GraphPad Curve Fitting Guide

GraphPad Software Inc.
www.graphpad.com

This is one of three companion guides to GraphPad Prism 8.
All are available as web pages on graphpad.com.

This guide was published on April 1, 2020

Table of Contents

1 Welcome to Prism 8 Curve Fitting Guide

Curve fitting is one of the most commonly used statistical techniques in research. This guide will help you learn the basics of curve fitting along with how to effectively perform curve fitting within Prism

Learn More about Curve Fitting

Read about mathematical models and how models are fit to data in the Principles of Regression 12 section of this guide!

Regression in Prism

Check out our Regression with Prism 8 115 section of this guide to learn how to start fitting models to your data using Prism!

More Guides!

You're currently browsing the Prism Curve Fitting Guide. We also have our User Guide, focused on getting to know Prism, and our Statistics Guide! Both of these guides do more than just explain how to use Prism. They also explain important concepts about data analysis!

- Switch to the Prism 8 User Guide

- Switch to the Prism 8 Statistics Guide

2 How to cite these pages

When citing one of these pages, reference the name of the page, the date you accessed it and the URL. All the pages in the statistics and curve fitting guides were written by GraphPad's founder, Harvey Motulsky.

Use this example as a guide to cite a single page:

> H. J. Motulsky, "Confidence intervals of parameters", GraphPad Curve Fitting Guide. Accessed 5 March 2016. http://www.graphpad.com/guides/prism/7/curve-fitting/index.htm?reg_standard_errors_and_confidence.htm

Note that the URL for each page in these guides is shown at the bottom of the page. Don't use the URL you see in the browser toolbar, as that may be generic and not apply just to the page you are looking at.

Use this example as a guide to citing the entire curve fitting guide:

> H. J. Motulsky, GraphPad Curve Fitting Guide. Accessed 5 March 2016. http://www.graphpad.com/guides/prism/7/curve-fitting/index.htm

3 PRINCIPLES OF REGRESSION

Many scientists fit curves more often than the use any other statistical technique. Yet few statistical texts really explain the principles of curve fitting. This Guide provides a concise introduction to fitting curves, especially nonlinear regression.

The first step is to be clear on what your goal is:

- If your goal is to fit a model to your data in order to obtain best-fit values of the parameters, and want to learn the principles first, then read this principles section before trying to fit curves.

- If you already understand the principles of nonlinear regression, and want to see how to fit curves with Prism, jump right to the tutorials 293.

- If your goal is to simply fit a smooth curve in order to interpolate values from the curve, there is no need to learn much theory. Jump right to an explanation of interpolation with Prism. 258

- If your goal is to create a spline (a curve that goes through every data point) or a lowess curve (shows the general trend with a curve that can be quite jagged), you can jump right to the instructions for that analysis 289.

3.1 Understanding mathematical models

A mathematical model is an equation that describes a physical, chemical or biological state or process. The goal of nonlinear regression is to fit a model to your data.

- What exactly is a mathematical model? 13

- Learn from three example models 14.

- It sure would be nice if models could be chosen by Prism. Why is this not possible? 16
- When trying to understand a model, here are some tips for figuring out what it means 17.
- In addition to fitting a model to data, Prism can plot a function 759, or a family of functions. You choose a function and values for the parameters, and Prism will plot predicted Y values for a range of X values.

3.1.1 What is a model?

The whole point of nonlinear regression is to fit a model to your data. So that raises the question: What is a model?

A mathematical model is a description of a physical, chemical or biological state or process. Using a model can help you think about chemical and physiological processes or mechanisms, so you can design better experiments and comprehend the results. When you fit a model to your data, you obtain best-fit values that you can interpret in the context of the model.

> *A mathematical model is neither a hypothesis nor a theory. Unlike scientific hypotheses, a model is not verifiable directly by an experiment. For all models are both true and false.... The validation of a model is not that it is "true" but that it generates good testable hypotheses relevant to important problems.*
>
> R. Levins, Am. Scientist 54:421-31, 1966

Your goal in using a model is not necessarily to describe your system perfectly. A perfect model may have too many parameters to be useful. Rather, your goal is to find as simple a model as possible that comes close to describing your system. You want a model to be simple enough so you can fit the model to data, but complicated enough to fit your data well and give you parameters that help you understand the system, reach valid scientific conclusions, and design new experiments.

3.1.2 Three example models

To give you a sense of how mathematical models work, below is a brief description of three commonly used models.

Optical density as a function of concentration

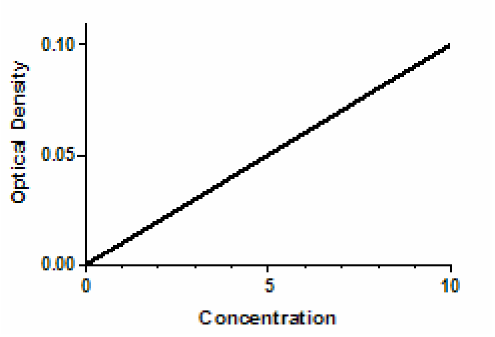
Background

Colorimetric chemical assays are based on a simple principle. Add appropriate reactants to your samples to initiate a chemical reaction whose product is colored. When you terminate the reaction, the concentration of colored product is proportional to the initial concentration of the substance you want to assay.

Model

Since optical density is proportional to the concentration of colored substances, the optical density will also be proportional to the concentration of the substance you are assaying.

$$\text{Optical Density} = Y = k \cdot [\text{substance}] = K \cdot X$$



Reality check

Mathematically, the equation works for any value of X. However, the results only make sense with certain values.

- Negative X values are meaningless, as concentrations cannot be negative.

- The model may fail at high concentrations of substance where the reaction is no longer limited by the concentration of substance.

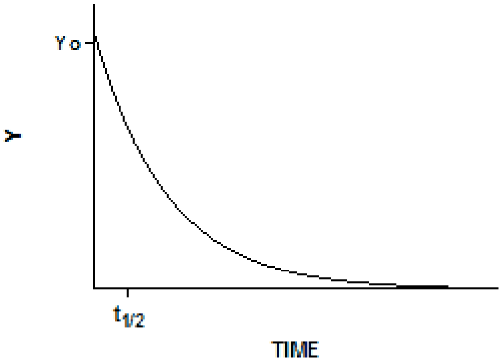
- The model may also fail at high concentrations if the solution becomes so dark (the optical density is so high) that little light reaches the detector. At that point, the noise of the instrument may exceed the signal.

It is not unusual for a model to work only for a certain range of values. You just have to be aware of the limitations, and not try to use the model outside of its useful range.

Exponential decay

Exponential equations whenever the rate at which something happens is proportional to the amount which is left. Examples include ligands dissociating from receptors, decay of radioactive isotopes, and metabolism of drugs. Expressed as a differential equation:

$$\frac{\Delta Y}{\Delta X} = \frac{dY}{dX} = -k \cdot Y$$



Converting the differential equation into a model that defines Y at various times requires some calculus. There is only one function whose derivative is proportional to Y, the exponential function. Integrate both sides of the equation to obtain a new exponential equation that defines Y as a function of X (time), the rate constant k, and the value of Y at time zero, Y_0.

$$Y = Y_0 \cdot e^{-k \cdot X} = Y_0 \cdot \exp(-k \cdot X)$$

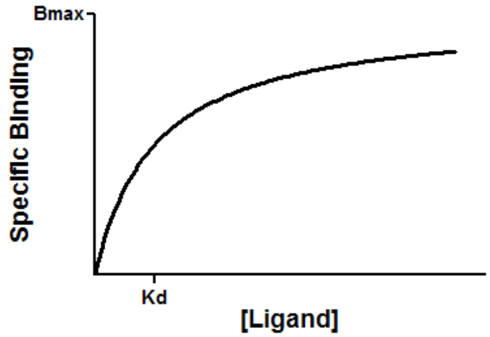
Equilibrium binding

When a ligand interacts with a receptor, or when a substrate interacts with an enzyme, the binding follows the law of mass action.

$$R + L \underset{k_{off}}{\overset{k_{on}}{\rightleftharpoons}} RL$$

You measure the amount of binding, which is the concentration of the RL complex, so plot that on the Y axis. You vary the amount of added ligand, which we can assume is identical to the concentration of free ligand, L, so that forms the X axis. Some simple (but tedious) algebra leads to this equation:

$$\text{Specific binding} = Y = \frac{B_{max} \cdot X}{K_d + X}$$



3.1.3 The problem with choosing models automatically

The goal of nonlinear regression is to fit a model to your data. The program finds the best-fit values of the parameters in the model (perhaps rate constants, affinities, receptor number, etc.) which you can interpret scientifically.

Choosing a model is a scientific decision. You should base your choice on your understanding of chemistry or physiology (or genetics, etc.). The choice should not be based solely on the shape of the graph.

Some programs (not available from GraphPad Software) automatically fit data to thousands of equations and then present you with the equation(s) that fit the data best. Using such a program is appealing because it frees you from the need to choose an equation. The problem is that the program has no understanding of the scientific context of your experiment. The equations that fit the data best are unlikely to correspond to scientifically meaningful models. You will not be able to interpret the best-fit values of the parameters, so the results are unlikely to be useful.

Letting a program choose a model for you can be useful if your goal is to simply create a smooth curve for simulations or interpolations. In these situations, you don't care about the value of the parameters or the meaning of the model. You only care that the curve fit the data well and does not wiggle too much. Avoid this approach when the goal of curve fitting is to fit the data to a model based on chemical, physical, or biological principles. Don't use a computer program as a way to avoid understanding your experimental system, or to avoid making scientific decisions.

3.1.4 Advice: How to understand a model

Encountering an equation causes the brains of many scientists to freeze. If you are one of these scientists who has trouble thinking about equations, here are some tips to help you understand what an equation means. As an example, let's use the Michaelis-Menten equation that describes enzyme activity as a function of substrate concentration:

Y=Vmax*X/(Km +X)

Tip 1. Make sure you know the meaning and units of X and Y

For this example, Y is enzyme activity which can be expressed in various units, depending on the enzyme. X is the substrate concentration in Molar or micromolar or some other unit of concentration.

Tip 2. Figure out the units of the parameters

In the example equation, the parameter Km is added to X. It only makes sense to add things that are expressed in the same units, so Km must be expressed in the same concentration units as X. This means that the units cancel in the term X/(Km +X), so Vmax must be expressed in the same units of enzyme activity as Y.

Tip 3: Figure out the value of Y at extreme values of X

Since X is concentration, it cannot be negative. But it can be zero. Substitute X=0 into the equation, and you will see that Y is also zero.

Let's also figure out what happens as X gets very large. As X gets large compared to Km, the denominator (X+Km) has a value very similar to X. So the ratio X/(X+Km) approaches 1.0, and Y approaches Vmax. So the graph of the model must level off at Y=Vmax as X gets very large.

Tip 4. Figure out the value of Y at special values of X

Since Km is expressed in the same units as X, you can ask what happens if X equals Km? In that case, the ratio X/(Km + X) equals 0.5, so Y equals half of Vmax. This means the Km is the concentration of substrate that leads to a velocity equal to half the maximum velocity Vmax.

Tip 5. Graph the model with various parameter values

Graphing a family of curves with various values for the parameters can help you visualize what the parameters mean. To do this with Prism, use the analysis "Create a family of theoretical curves 759".

3.2 Fitting models

Enter topic text here.

3.2.1 The goal of regression

The general idea of regression

All models define the outcome (Y) as a function of one or more parameters and an independent variable (X) [or several independent variables].

The goal of is to adjust the values of the model's parameters to find the line or curve that comes closest to your data. For example, with linear regression, the goal is to find the best-fit values of the slope and intercept that makes the line come close to the data. With nonlinear regression of a normalized dose-response curve, the goal is to adjust the values of the EC50 (the concentration that provokes a response halfway between the minimum and maximum responses) and the slope of the curve.

More precisely, the goal of regression is to find the values of the parameters that are most likely to be correct. To do this requires making an assumption about the scatter of data around the curve.

The goals of regression

Scientists use regression with one of three distinct goals:

- To fit a model to your data in order to obtain best-fit values of the parameters, or to compare the fits of alternative models. If this is your goal, you must pick a model (or two alternative models) carefully, and pay attention all the results. The whole point is to obtain best-fit values for the parameters, so you need to understand what those parameters mean scientifically.

- To fit a smooth curve in order to interpolate values from the curve, or perhaps to draw a graph with a smooth curve. If this is your goal, you can assess it purely by looking at the graph of data and curve. There is no need to learn much theory.

- To make predictions.

3.2.2 The distinction between linear and nonlinear

Linear is not defined as most scientists would guess. A model is said to be *linear* when the Y variable is linear with each parameter. To understand this, you need to think about the math. Hold all but one parameter constant and also hold X constant. Now vary the remaining parameter and watch how Y changes. If the change in Y is linear with the change in the parameter you vary, and this is true for all the parameters in the model, then that model is said to be linear.

By this definition, a polynomial model is linear. Let's take the example of a third order polynomial model:

$Y= A + BX + CX^2 + DX^3$

Hold A, B, D and X constant, and look at how Y will vary as you vary C. The equation now is $Y= [A + BX + DX^3] + C[X^2]$, with the two terms in brackets constant. That graph would be a straight line. A, B and D are also linear with Y (holding everything else constant) so the model is linear. But if you graph Y vs. X for a polynomial model, you'll almost always see a curve, not a line (it depends on what values you assign to A-D. So linear describes the model, not the graph of X vs. Y.

If the model is not linear, then it is *nonlinear*.

Why does it matter if a model is linear? Like linear regression, it is possible to fit polynomial models without fussing with initial values and without the possibility of a false minimum. For this reason, some programs (i.e. Excel) can perform polynomial regression, but not nonlinear regression. And some programs have separate modules for fitting data with polynomial models (linear) and nonlinear models. Prism fits polynomial models using the same analysis it uses to fit nonlinear models. Polynomial equations are available within Prism's nonlinear regression analysis.

From the point of view of a scientist using Prism, the distinction between linear and nonlinear models is not very important. Choose the model that makes sense for your data. The only issue is that with nonlinear models, it is essential to provide initial estimated values for each parameter. In some cases, these choices can be critical to getting useful results. If you choose a built-in model, Prism chooses initial values for you, and these are almost always good enough to get the job done.

While the terms linear and nonlinear have standard definitions in statistics, the term *curvilinear* does not have a standard meaning. It generally is used to describe a curve that is smooth (no discontinuities) but the underlying mathematical model could be either linear or nonlinear.

3.2.3 Different kinds of regression

All regression methods fit a model to data to find the values of the parameters that makes the model fit the best.

Simple vs. multiple regression

Simple regression fits models with one independent (X) variable. Multiple regression fits models with more than one independent variable. Prism offers multiple linear regression 178. In a restricted sense, it can do multiple nonlinear regression with two independent variables. See this example of enzyme inhibition data 622 to see how this works.

Linear vs. nonlinear regression

See the definition of linear and nonlinear. 19

Polynomial regression

Polynomial models are mathematically linear, but in Prism you use the nonlinear regression analysis to fit a polynomial model.

Poisson regression

Linear and nonlinear regression are usually run with the assumption that the residuals (vertical distance of the points from the best-fit line or curve) are sampled from Gaussian distributions. If the outcome is counts (number of objects or events), it is usually better to assume the residuals are sampled from a Poisson distribution. Prism 8 introduced Poisson regression to Prism as options in both multiple and nonlinear regression.

Logistic regression

A logistic regression model is used when the outcome, the dependent (Y) variable, has only two possible values. Did the person get the disease or not? Did the student graduate or not? There can be one or several independent variables. These independent variables can be a variable like age or blood pressure, or have discrete values to encode which treatment each subject received.

Prism 8.3 is the first version than can do logistic regression.

Proportional hazards regression

A proportional hazards model is used when the outcome is whether or not a one-time event (often death) occurred. One of the independent variables is time, and other independent variables can be used to account for treatment or other variables.

GraphPad Prism does not perform proportional hazards regression.

3.3 Principles of simple linear regression

Linear regression fits a straight line through your data to find the best-fit value of the slope and intercept.

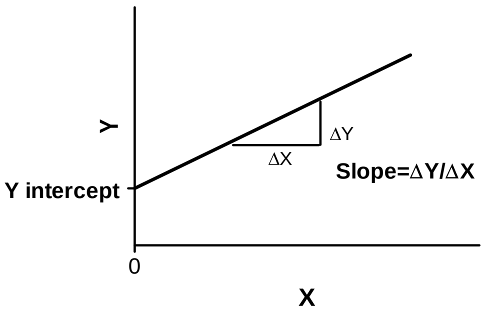
- What is the goal of linear regression? 22
- How does it work? 23
- Linear regression and correlation are often confused. How are they distinct? 25
- Linear regression is just a special case of nonlinear regression. How do they differ? 26
- Linear regression is sometimes used on transformed data to analyze Scatchard, Lineweaver-Burke and similar plots. Why is this is not a good way to analyze data? 28

3.3.1 The goal of linear regression

What is linear regression?

Linear regression fits this model to your data:

Y=intercept+slope×X



The slope quantifies the steepness of the line. It equals the change in Y for each unit change in X. It is expressed in the units of the Y axis divided by the units of the X axis. If the slope is positive, Y increases as X increases. If the slope is negative, Y decreases as X increases.

The Y intercept is the Y value of the line when X equals zero. It defines the elevation of the line.

Correlation and linear regression are not the same. Review the differences. 25

Simple vs. multiple linear regression

Simple linear regression is shown above. There is only a single X variable. In contrast, multiple linear regression defines Y as a function that includes several X variables. More generally, there are other types of relationships in which multiple X variables can be used to describe a single Y variable. These methods are collectively referred to as multiple regression (multiple linear regression is a type of multiple regression), and you can read more about the principles of multiple regression here 41 .

Linear vs. logistic regression

In simple linear regression, the dependent (Y) variable is continuous, meaning it can take on any range of values. In some cases, your Y variable may not be continuous. For example, if your Y variable can only be one of two values (for example, yes or no, heads or tails, male or female mice, etc.), then it's said to be a binary categorical variable. In this case, linear regression is not appropriate. Instead, you might consider using logistic regression, which models the probability of observing a given outcome (sometimes called a "success"). Like linear regression, you can have one or multiple X variables with logistic regression. Read more about simple logistic regression (with only one X variable) 31 and multiple logistic regression (with multiple X variables) 41 for more information.

3.3.2 How linear regression works

How linear regression works. Minimizing sum-of-squares.

The goal of linear regression is to adjust the values of slope and intercept to find the line that best predicts Y from X. More precisely, the goal of regression is to minimize the sum of the squares of the vertical distances of the points from the line. Why minimize the sum of the squares of the distances? Why not simply minimize the sum of the actual distances?

If the random scatter follows a Gaussian distribution, it is far more likely to have two medium size deviations (say 5 units each) than to have one small deviation (1 unit) and one large (9 units). A procedure that minimized the sum of the absolute value of the distances would have no preference over a line that was 5 units away from two points and one that was 1 unit away from one point and 9 units from another. The sum of the distances (more precisely, the sum of the absolute value of the distances) is 10 units in each case. A procedure that minimizes the sum of the squares of the distances prefers to be 5 units away from two points (sum-of-squares = 50) rather than 1 unit away from one point and 9 units away from another (sum-of-squares = 82). If the scatter is Gaussian (or nearly so), the line determined by minimizing the sum-of-squares is most likely to be correct.

The calculations are shown in every statistics book, and are entirely standard.

The term "regression"

The term "regression", like many statistical terms, is used in statistics quite differently than it is used in other contexts. The method was first used to examine the relationship between the heights of fathers and sons. The two were related, of course, but the slope is less than 1.0. A tall father tended to have sons shorter than himself; a short father tended to have sons taller than himself. The height of sons regressed to the mean. The term "regression" is now used for many sorts of curve fitting.

The details

The details of linear regression are explained in numerous text books, so will not be detailed here. Everything Prism does is standard. Here is one good source for the basics including confidence and prediction bands.

3.3.3 Comparing linear regression to correlation

Linear regression is distinct from correlation.

What is the goal?

Linear regression finds the best line that predicts Y from X.

Correlation quantifies the degree to which two variables are related. Correlation does not fit a line through the data points. You simply are computing a correlation coefficient (r) that tells you how much one variable tends to change when the other one does. When r is 0.0, there is no relationship. When r is positive, there is a trend that one variable goes up as the other one goes up. When r is negative, there is a trend that one variable goes up as the other one goes down.

What kind of data?

Linear regression is usually used when X is a variable you manipulate (time, concentration, etc.)

Correlation is almost always used when you measure both variables. It rarely is appropriate when one variable is something you experimentally manipulate.

Does it matter which variable is X and which is Y?

The decision of which variable you call "X" and which you call "Y" matters in regression, as you'll get a different best-fit line if you swap the two. The line that best predicts Y from X is not the same as the line that predicts X from Y (however both those lines have the same value for R^2).

With correlation, you don't have to think about cause and effect. It doesn't matter which of the two variables you call "X" and which you call "Y". You'll get the same correlation coefficient if you swap the two.

Assumptions

With linear regression, the X values can be measured or can be a variable controlled by the experimenter. The X values are not assumed to be sampled from a Gaussian distribution. The distances of the points from the best-fit line is assumed to follow a Gaussian distribution, with the SD of the scatter not related to the X or Y values.

The correlation coefficient itself is simply a way to describe how two variables vary together, so it can be computed and interpreted for any two variables. Further inferences, however, require an additional assumption -- that both X and Y are measured (are interval or ratio variables), and both are sampled from Gaussian distributions. This is called a bivariate Gaussian distribution. If those assumptions are true, then you can interpret the confidence interval of r and the P value testing the null hypothesis that there really is no correlation between the two variables (and any correlation you observed is a consequence of random sampling).

Relationship between results

Linear regression quantifies goodness of fit with r^2, sometimes shown in uppercase as R^2. If you put the same data into correlation (which is rarely appropriate; see above), the square of r from correlation will equal r^2 from regression.

Correlation computes the value of the Pearson correlation coefficient, r. Its value ranges from -1 to +1.

3.3.4 Comparing linear regression to nonlinear regression

The goal of linear and nonlinear regression

A line is described by a simple equation that calculates Y from X, slope and intercept. The purpose of **linear regression** is to find values for the slope and intercept that define the line that comes closest to the data.

Nonlinear regression is more general than linear regression and can fit any model (equation) to your data. It finds the values of those parameters that generate the curve that comes closest to the data.

How linear and nonlinear regression work

Both linear and nonlinear regression find the values of the parameters (slope and intercept for linear regression) that make the line or curve come as close as possible to the data. More precisely, the goal is to minimize the sum of the squares of the vertical distances of the points from the line or curve.

Linear regression accomplishes this goal using math that can be completely explained with simple algebra (shown in many statistics books). Put the data in, and the answers come out. There is no chance for ambiguity. You could even do the calculations by hand, if you wanted to.

Nonlinear regression uses a computationally intensive, iterative approach 97 that can only be explained using calculus and matrix algebra. The method requires initial estimated values for each parameter.

Linear regression is a special case of nonlinear regression

Nonlinear regression programs can fit any model, including a linear one. Linear regression is just a special case of nonlinear regression.

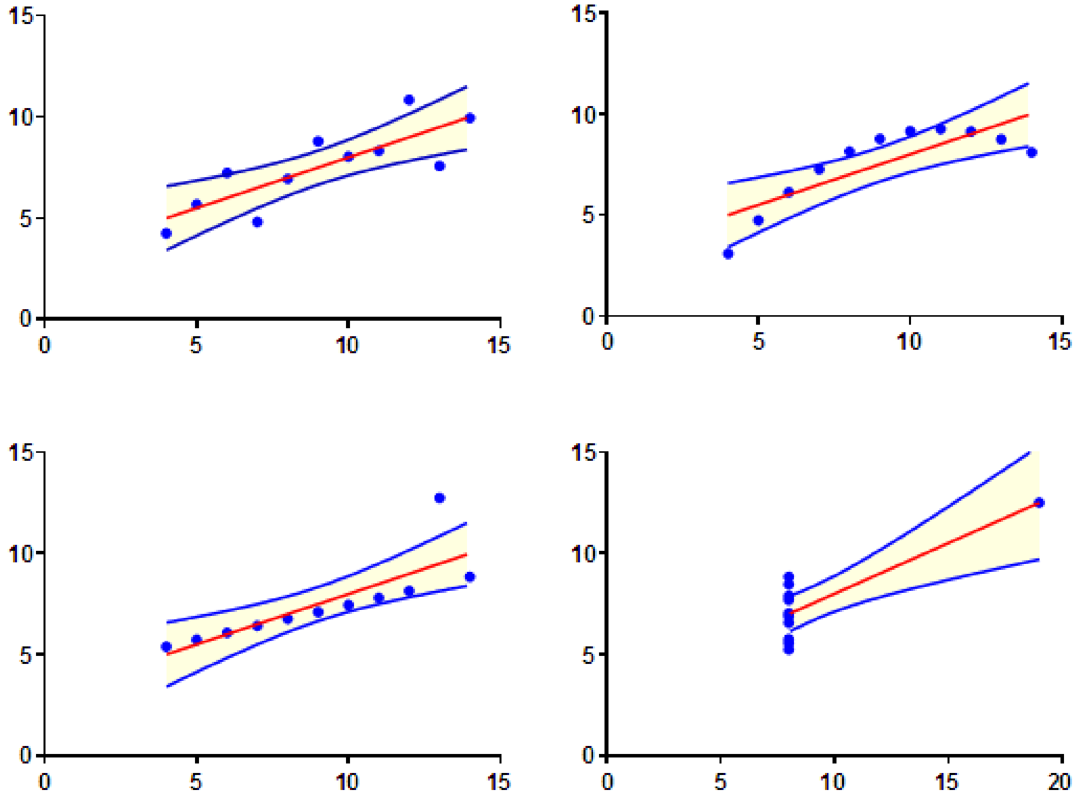
Even if your goal is to fit a straight line through your data, there are many situations 123 where it makes sense to choose nonlinear regression rather than linear regression.

Using nonlinear regression to analyze data is only slightly more difficult than using linear regression. Your choice of linear or nonlinear regression should be based on the model you are fitting. Do not use linear regression just to avoid using nonlinear regression. Avoid transformations 28 such as Scatchard or Lineweaver-Burke transforms whose only goal is to linearize your data.

3.3.5 Advice: Look at the graph

The four graphs below were created by Anscombe (1) to demonstrate the importance of looking at your data.The four data sets, the Anscome's quartet, all have 11 points. You can see at a glance that the data are very different. Surprisingly, the analyses of these four data sets are identical in many ways. When analyzed by linear regression, all these values are identical for all four graphs:

- The mean values of X and Y
- The slopes and intercepts
- r^2
- The standard errors and confidence intervals of the slope and intercept
- The standard deviation of the residuals (sy.x)
- The F ratio and P value when testing the null hypothesis that the slope is really 0.0 (horizontal).
- The covariance 393 between slope and intercept (computed by Prism's nonlinear regression analysis)



Differences seen in linear regression are:

- The confidence bands, shown in yellow below, are identical for three of the four graphs, but different for the fourth (which has different X values)
- The results of a runs test for linearity
- Normality tests of residuals

1. F.J. Anscombe, Graphs in Statistical Analysis, American Statistician, 27 (February 1973), 17-21.

3.3.6 Advice: Avoid Scatchard, Lineweaver-Burke and similar transforms

Before analyzing your data with linear regression, stop and ask yourself whether it might make more sense to fit your data with nonlinear regression. If you have transformed nonlinear data to create a linear relationship, you will almost certainly be better off fitting your original data using nonlinear regression.

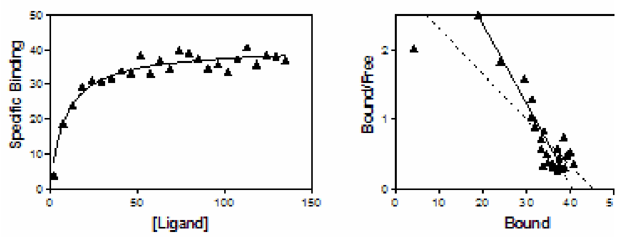
Before nonlinear regression was readily available, the best way to analyze nonlinear data was to transform the data to create a linear graph, and then analyze the transformed data with linear regression. Examples include Lineweaver-Burke plots of enzyme kinetic data, Scatchard plots of binding data, and logarithmic plots of kinetic data.

These methods are **outdated**, and should not be used to analyze data

The problem with these methods is that the transformation distorts the experimental error. Linear regression assumes that the scatter of points around the line follows a Gaussian distribution and that the standard deviation is the same at every value of X. These assumptions are rarely true after transforming data. Furthermore, some transformations alter the relationship between X and Y. For example, in a Scatchard plot the value of X (bound) is used to calculate Y (bound/free), and this violates the assumption of linear regression that all uncertainty is in Y while X is known precisely. It doesn't make sense to minimize the sum of squares of the vertical distances of points from the line, if the same experimental error appears in both X and Y directions.

Since the assumptions of linear regression are violated, the values derived from the slope and intercept of the regression line are not the most accurate determinations of the variables in the model. Considering all the time and effort you put into collecting data, you want to use the best possible technique for analyzing your data. Nonlinear regression produces the most accurate results.

The figure below shows the problem of transforming data. The left panel shows data that follows a rectangular hyperbola (binding isotherm). The right panel is a Scatchard plot of the same data. The solid curve on the left was determined by nonlinear regression. The solid line on the right shows how that same curve would look after a Scatchard transformation. The dotted line shows the linear regression fit of the transformed data. Scatchard plots can be used to determine the receptor number (B_{max}, determined as the X-intercept of the linear regression line) and dissociation constant (K_d, determined as the negative reciprocal of the slope). Since the Scatchard transformation amplified and distorted the scatter, the linear regression fit does not yield the most accurate values for B_{max} and K_d.



Don't use linear regression just to avoid using nonlinear regression. Fitting curves with nonlinear regression is not difficult.

Although it is usually inappropriate to analyze transformed data, it is often helpful to display data after a linear transformation. Many people find it easier to visually interpret transformed data. This makes sense because the human eye and brain evolved to detect edges (lines) — not to detect rectangular hyperbolas or exponential decay curves. Even if you analyze your data with nonlinear regression, it may make sense to display the results of a linear transformation.

3.4 Principles of simple logistic regression

The pages linked below provide an overview of many of the concepts involved in performing simple logistic regression. You can also read about how to perform simple logistic regression in Prism 143, or follow a guided walkthrough 146 for this analysis in Prism.

3.4.1 The goal of simple logistic regression

Logistic regression is used when the outcome variable (Y variable, dependent variable, response variable, etc.) can only take on two possible outcomes, and its goal is to model the probability of observing a success. In this sense, "success" simply refers to one of those two possible outcomes, and should be based on your experimental design. As with many terms in statistics, "success" in this case has a slightly different meaning than the one we're commonly used to using. For example, while studying the incidence of a rare disease in a population, what you may be interested in is the probability that an individual would get this disease. In this case, you would consider getting the disease a "success," if only for the sake of constructing the model.

Looking at another example, let's say you were given a dataset containing the length of time that students studied for a test in addition to whether or not those students passed the test. You would probably expect that the longer a student studies for the test, the more probable it is that the student will pass. Here, the "success" would be that the student passes. However, the Y variable for logistic regression could be just about anything, so long as it can only take on one of two possible values: yes/no, pass/fail, alive/dead, etc. Another way to say this is that the outcome variable must be "binary". Usually, these outcomes are encoded as a "1" (indicating a "success") or a "0" (indicating a "failure"). Note that in our example, if you were given the grades for each student (as a percent), you may have considered performing linear or nonlinear regression. However, because our outcome is binary, logistic regression is the appropriate choice.

In a sense, simple logistic regression can be thought of as an extension to simple linear regression to handle cases with binary outcomes: both simple linear regression and simple logistic regression build models with which you can predict an outcome value (Y) by knowing a single input value (X). Because of this, there are two very important things to remember when thinking about the similarities and differences of linear and logistic regression:

1. Linear regression works when the outcome is continuous, logistic regression works when the outcome is binary. Trying to use linear regression on a binary outcome variable simply won't work (well).

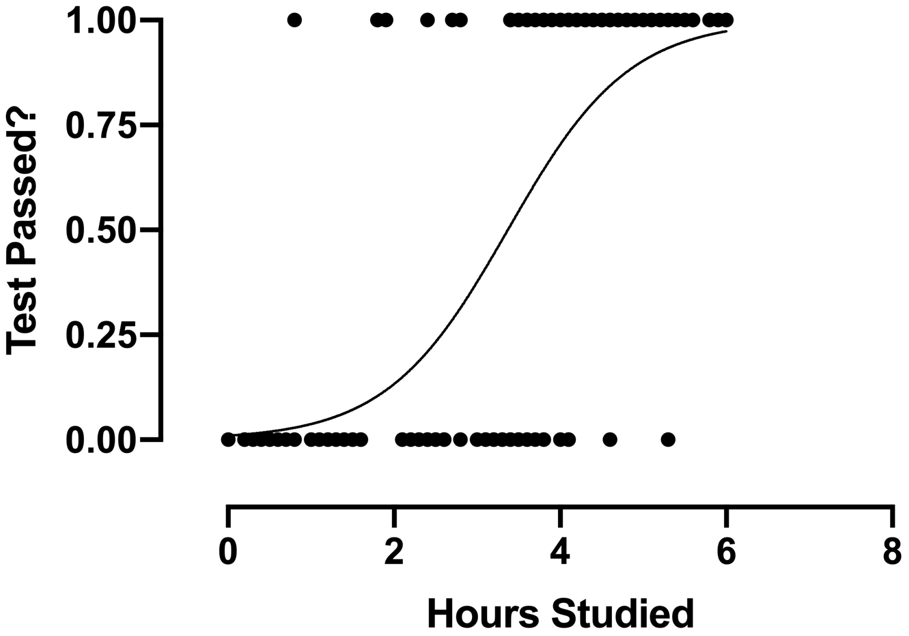
2. Logistic regression generates a model that allows you to predict the *probability of success*, given a certain X value. Data that you put into

the model will only include actual outcomes (at a given X value, a success was observed or it wasn't).

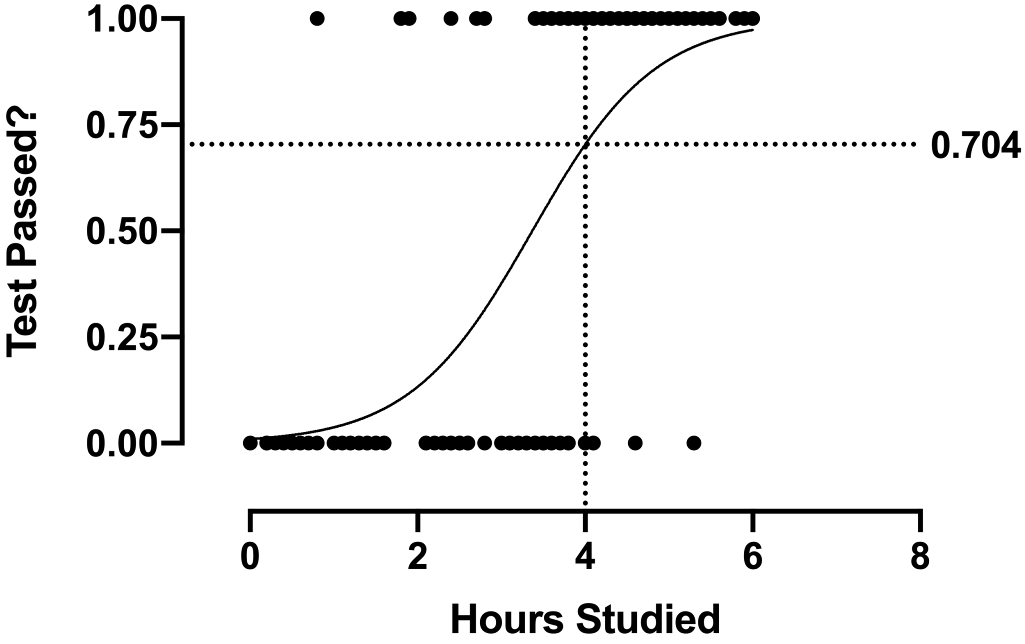
These two topics are discussed in more detail in the following sections

3.4.1.1 How simple logistic regression differs from simple linear regression

Linear regression works by fitting a model that you can use to determine the actual value of Y, given a value of X. This model provides information on the relationship between these two variables and answers the question: how much does the value of Y change with a change in X? In other words, using a linear regression model that defines the data well, you could predict the value of an outcome reasonably well just by knowing the value of the predictor. In contrast, logistic regression models the probability of observing a success, given the value of the predictor. Take the data shown below as an example:

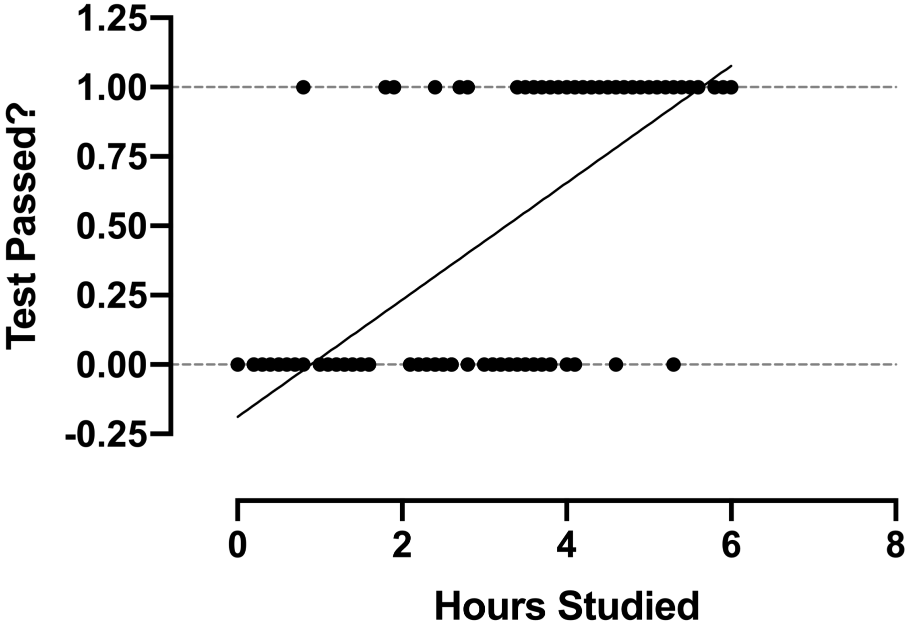


In this plot, all of our data points take either the value 0 (fail) or 1 (pass). The logistic fit is the S-curve that models the probability of success as a function of hours of study. In this example, instructors will be glad to observe that few students who studied 4 hours failed the exam. Indeed, for a student who studied 4 hours, the model predicts the probability of passing to be around 70%.



The S-curve is a byproduct of the way the logistic function estimates the probability. Note that probabilities are bound between 0 and 1, which makes sense: you can’t have a “negative probability” of an event happening, and a probability greater than 100% also doesn’t make any sense. As such, the upper and lower bounds of the S-curve are also limited by these values. But what this means is that - unlike with linear regression - the values we get from the model don’t give us direct estimates for the values we expect to observe. At X = 4, the value of the model is 0.704. However, for any observation that we make at X = 4, the outcome will ONLY be 0 or 1; the observed value would never be 0.704. The model simply tells us that we can expect ~70% of our outcomes to be 1 at X = 4. This is a critical point to understand for logistic regression.

If we were to compare the logistic regression model and the linear regression model on the same data, we would see quickly why the simple linear regression model simply doesn’t work for this kind of data.



Our data are still 0s and 1s, but, unlike the logistic model, the linear model is not predicting the probability of a success. Instead, it predicts values that can be less than 0 and larger than 1. For example, this model predicts that students who studied less than about 0.9 hours have a negative estimated value of passing the test. In some cases, linear models can be used on binary independent variables to do simple classification. However, these approaches cannot provide interpretability of coefficients, tests of significance, and confidence intervals. For these results (when your outcome is binary), use logistic regression.

3.4.1.2 Classification with logistic regression

As discussed in the previous section, the goal of logistic regression is to model the probability of a given outcome occurring. However, rather than predicting probabilities, researchers sometimes want the output of their model to indicate if either a success or a failure is expected for a given X value. This is called classification. The simplest way to perform classification is to set what is known as a cutoff value. This value is a number between 0 and 1 that serves as the division for what to call a “success” and what to call a “failure”. For example, setting the classification cutoff to 0.5 is common (and default for simple logistic regression in Prism), and means that if the model predicts a probability of success greater than or equal to 0.5, then that prediction is classified as a "success" (Y=1), while if the model predicts a probability less than 0.5, it will classify it as a "failure" (Y=0).

There are a LOT of metrics that researchers use from this sort of classification including concepts like the sensitivity and specificity of a model, the true positive rate (TPR) and false positive rate (FPR) of classification, the positive and negative predictive power of the model, and much more. Read more about classification 233.

3.4.2 Understanding Log Odds and Interpreting Coefficient Estimates

Logistic regression fits a model to the given data that assumes a linear relationship of the predictor variables with the log odds of the outcome variable. This section of the guide discusses what log odds are and how this concept is used in logistic regression.

3.4.2.1 What are Log Odds and why does logistic regression use them?

The model for simple logistic regression is written logit[P(Y=1)] = β0 + β1 * X + error.

On the right-hand side, this matches the model for simple linear regression (remember the simple linear regression model is Y = intercept + slope*X). The left-hand side includes a "logit" function (long o, soft g) that adjusts for the fact that Y is a variable that can only take on values of 0 and 1. Briefly, the logit is the log of the odds that Y=1, and "P(Y=1)" is the probability that Y is equal to 1. Note that "P" in this case is an abbreviation for probability and has nothing to do with P values.

To understand what "log odds" are, it's important to know what is meant by odds. The odds equals the probability that Y=1 divided by the probability that Y=0. For example, if the probability that Y =1 is 0.8 (or that there's an 80% probability of Y=1), then the probability that Y=0 is 1-0.8 or 0.2 (remember, Y can only be 0 or 1, so the probability that Y=0 is 1-[probability that Y=1]). Using these numbers, we can calculate the odds as the ratio of these two numbers:

Odds = P(Y=1)/P(Y=0) = 0.8/0.2 = 4

In this case, the odds is 4. You will often hear people refer to this as 4:1 odds, which you would read as "four to one odds." Now that we know how odds are related to probability, we can take the final step to calculate the log odds. This simply involves using the calculated value for odds and taking the natural logarithm (Ln) of that value:

Log odds = Ln(Odds) = Ln(P(Y=1)/P(Y=0)) = Ln(P(Y=1)/[1-P(Y=1)])

All of the forms of log odds listed above are equivalent, and while this math can sound quite confusing, the reason that we go through all of this work is that we want to model the probability that Y=1 (or Y=0).

More to the point, we want to use a linear model (the right hand side of the simple logistic regression equation) to model this probability. Recall that a probability ranges between 0 and 1. The right-hand side of the simple logistic regression model, like the simple linear regression model, can generate (in theory) any value from negative infinity to positive infinity. The logit function is used to serve as a link between these two ranges.

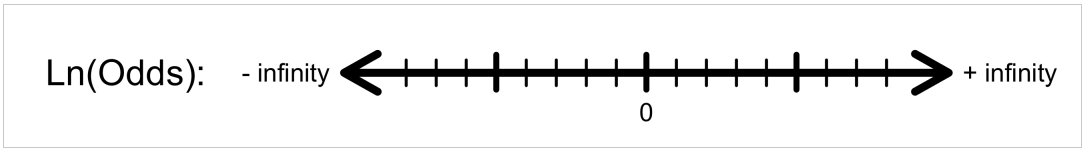
Start with probability: these values can only go from 0 to 1:

Probability:
0
1

First, we take the odds, which transforms this scale from 0 to 1 to a scale from 0 to +infinity (calculate the odds for any probability between 0 and 1 and see for yourself!):

Odds:
0
+ infinity

Next, we take the natural logarithm of the odds to get the log odds which transforms the scale again to one that goes from negative infinity to positive infinity:



So you can think of the logit function as using math to connect the values generated by the right-hand side of the model (can be any value) to the bound values of probability (must be between 0 and 1).

3.4.2.1.1 "Log" or "Ln"?

Logistic Regression uses the natural logarithm

Remember, when talking about log odds with logistic regression, we always mean the natural logarithm of the odds (Ln[Odds]). Natural log is often abbreviated as "log" or "ln," which can cause some confusion. In some contexts (not in logistic regression), "log" can be used as an abbreviation for base 10 logarithms. However, if used in the context of logistic regression, "log" means the natural logarithm!

Why is the natural log used instead of log base 10? Or log base 2? The short answer is tradition; that's just the way it's been done and so that's how everyone does it.

However, there are some interesting properties of the natural logarithm (and its inverse - the exponential function) that have contributed to its use over potential alternatives. For example, take the exponential function:

$\exp(x) = e^x$

The derivative of this function is… itself! Additionally, the derivative of Ln(x) = 1/x. These properties - among some other convenient attributes when dealing with growth rates, interest rates, decay rates, etc. - have made the natural logarithm the log of choice.

3.4.2.2 Interpreting the coefficients of logistic regression

Now that we know how logistic regression uses log odds to relate probabilities to the coefficients, we can think about what these coefficients are actually telling us. For simple logistic regression (like simple linear regression), there are two coefficients: an "intercept" (β0) and a "slope" (β1). Although you'll often see these coefficients referred to as intercept and slope, it’s important to remember that they don’t provide a graphical relationship between X and P(Y=1) in the way that their counterparts do for X and Y in simple linear regression. But what do they tell us?

- β0: the log odds when the X variable is 0
- β1: how much the log odds change with an increase (or decrease) in X by 1.0

Let’s consider a practical example. Let’s say our simple logistic regression model was Ln(odds) = -5.5 + 1.2*X. Here, β0 = -5.5 and β1 = 1.2. This means that when X = 0, the log odds equals -5.5. This also tells us that for every 1 unit increase in X, the log odds increases by 1.2 (a 2 unit increase in X results in an increase to the log odds of 2.4, etc.).

Thinking about log odds can be confusing, though. So using the math described above, we can re-write the simple logistic regression model to tell us about the odds (or even about probability).

Odds = $e^{\beta0+\beta1*X}$

Using some rules for exponents, we can obtain:

Odds = $(e^{\beta0})*(e^{\beta1*X})$

When X equals 0, the second term equals 1.0. Therefore, $e^{\beta0}$ is the Odds when X is zero. In our example above, when X is zero, Odds is $e^{-5.5}$, or about 0.009. Additionally, you can see that a 1 unit increase in X results in multiplying the odds by $e^{\beta1}$. Thus, if X is 1, the odds are $(e^{-5.5})*(e^{1.2})$ = 0.033. These values ($e^{\beta0}$ and $e^{\beta1}$) are called “odds ratios” and are reported by Prism for simple logistic regression. Note that for the sake of clarity, Prism simply reports the odds ratios as “β0” and “β1”, but numerically, these are actually $e^{\beta0}$ and $e^{\beta1}$, respectively.

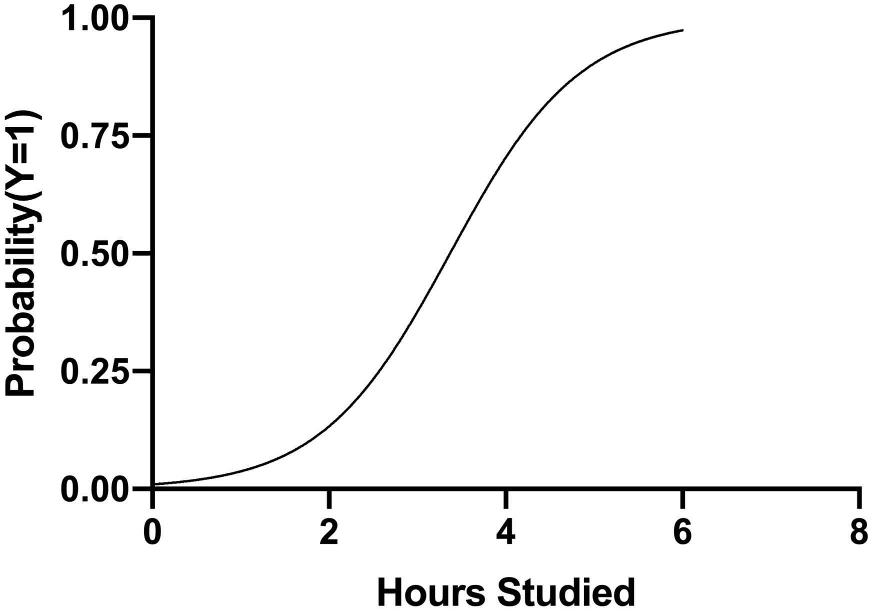
The form of the equation relating these coefficients to the probability that Y=1 is shown below. However, the interpretation of these coefficients in this equation is even more challenging than for odds, and so is not included here.

P(Y=1) = $(e^{\beta0+\beta1*X})/(1+e^{\beta0+\beta1*X})$

One final point to consider regarding all of these various transformations. While generally the graphical representation of simple logistic regression is the S-shaped logistic curve of probability vs. X, it's possible using the math shown above to plot the log odds vs. X. If you were to do this, what you would see is that a graph of the log odds vs. X generates a straight line with - you guessed it - an intercept equal to β0 and a slope equal to β1. This is demonstrated graphically below:

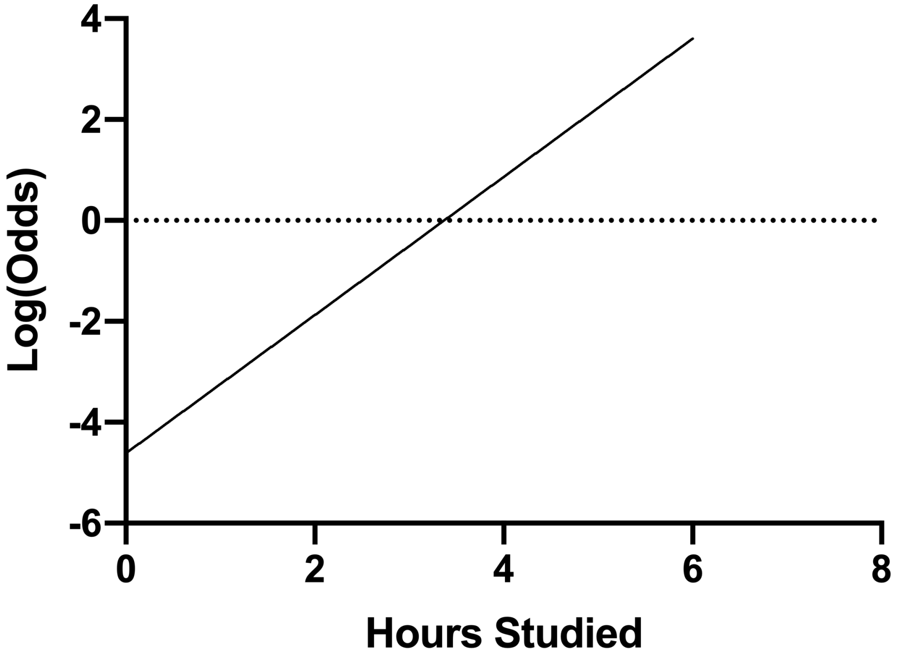
[Probability Y = 1] vs. X

β0 = -4.614, β1 = 1.370



Log(Odds) vs. X

β0 = -4.614, β1 = 1.370



3.4.3 How simple logistic regression works

Remember that with linear regression, the prediction equation minimizes the squared residual values (meaning it picks the line through the data points that has the smallest sum of the squared vertical distance between the line and all of the points). With logistic regression, all of the observed values are coded as 0 (failure) or 1 (success). Similar to simple linear regression, simple logistic regression also has a single independent (X) variable. The goal of logistic regression is to predict the probability of observing a 0 or 1, and simply fitting a straight line to the data by minimizing the sum of the squared distance from the points to this line would result in a nonsensical model (discussed in the previous section on "How simple logistic regression differs from simple linear regression 32").

So then how is the model fit? The model maximizes what statisticians call the likelihood. In very general terms, this process of maximizing the likelihood can be thought of as choosing the coefficient estimates for the model that would be most likely to generate the observed data. Something to keep in mind, therefore, is that logistic regression models will generally perform better at fitting (or classifying) the input data than they will at correctly predicting outcomes of new data.

Another key point to remember about logistic regression is that, unlike with linear regression, maximizing the likelihood for logistic regression requires an iterative solver, meaning that it picks some starting values and makes educated steps "up the mountain" towards the largest likelihood value in order to identify the maximum. This also means that – unlike linear regression, but similar to non-linear regression – there are times when the simple logistic regression model cannot be determined for a given set of data. In "statistician-speak", we would say that the maximum likelihood cannot be determined, or simply that the model did not converge.

There are a few common reasons why a simple logistic regression model may fail to converge, and these are covered on a separate page discussing each case 247.

3.5 Principles of multiple regression

3.5.1 The goal of multiple regression

Multiple regression means there are one or more independent (X) variables in the model with a single dependent (Y) variable. In order to model various types of dependent (Y) variables, Prism offers three types of multiple regression:

- Multiple linear regression 177 (used when Y is continuous)

- Poisson regression 177 (used when Y is a count; 0, 1, 2, ...)

- Logistic regression 198 (used when Y is binary; such as yes/no, success/failure, presence/absence, etc.)

All of these methods are members of the family of generalized linear models (GLMs). GLMs are a unifying theoretical framework that are quite flexible for modeling a variety of datasets. They all behave similarly, so once you've learned one, many of the ideas carry over to other regression types.

Multiple regression is useful in several contexts:

- To assess the impact of one variable after accounting for others. Does a treatment work after accounting for age differences between the patients who received the treatment and those who received a placebo? Does an environmental exposure increase the risk of a disease after taking into account other differences between people who were and were not exposed to that risk factor?

- To create an equation for making predictions. Given the data we know now, what is the chance that this particular man with chest pain is having a myocardial infarction (heart attack)?

- To understand how changes in several variables contribute to explaining an outcome of interest. For example, how do the concentrations of high-density lipoproteins (HDL, good cholesterol), low-density lipoproteins (LDL, bad cholesterol), triglycerides, C-reactive protein, and homocysteine predict the risk of heart disease? One goal might be to generate an equation that can predict the risk for an individual. Another goal is to understand the contributions of each risk factor in order to aid public health efforts and help prioritize research projects.

3.5.2 Lingo

Variables

A regression model predicts one variable Y from one or more other variables X. The Y variable is called the *dependent variable*, the *response variable* or the *outcome variable*. The X variables are called *independent variables*, *explanatory variables* or *predictor variables*.

Each X variable can be a value that the experimenter manipulated, a treatment that the experimenter selected or assigned, or a value that the experimenter measures.

Each independent variable can be: continuous (e.g., age, blood pressure, weight) or binary (e.g. gender, defining zero as male and one as female. These codes, of course, are arbitrary. When there are only two possible values for a variable, it is called a *dummy variable*.

Consult advanced texts if you need to use a categorical variable with three or more categories (e.g., four medical school classes, three different countries, or four different stages of cancer). Regression with a categorical variable with more than two categories is not straightforward, and it is easy to do it incorrectly.

Parameters

The multiple regression model defines the dependent variable as a function of the independent variables and a set of parameters, also called *regression coefficients*. Regression methods find the values of each parameter that make the model predictions come as close as possible to the data. This approach is analogous to linear regression, which determines the values of the slope and intercept (the two parameters or regression coefficients of the model) to make the model predict Y from X as closely as possible.

Simple regression versus multiple regression

Simple regression refers to models with a single X variable. *Multiple regression*, also called *multivariable regression*, refers to models with two or more X variables.

Univariate versus multivariate regression

Although they are beyond the scope of this guide, methods do exist that can simultaneously analyze several outcomes (Y variables) at once. These are called *multivariate* methods, and they include factor analysis, cluster analysis, principal components analysis, and multiple ANOVA (MANOVA). These methods contrast with *univariate* methods, which deal with only a single Y variable.

Note that the terms *multivariate* and *univariate* are used inconsistently. Sometimes multivariate is used to refer to *multivariable* methods for which there is one outcome and several independent variables (i.e., multiple and logistic regression). And sometimes *univariate* is used to refer to simple regression with only one independent variable.

Linear vs. nonlinear multiple regression

Prism only performs linear multiple regression. This means that each parameter is linear with Y. If you made a graph of how Y changes as you change any parameter (while holding all the X values and all the other parameters constant), the graph would be a straight line.

It is certainly possible to write models with one Y variable and multiple X values related to Y via a nonlinear function. But Prism does not (yet) perform multiple nonlinear regression. Let us know, with details, if this would be helpful to you.

3.5.3 How multiple regression works

The objective of multiple regression is to fit the chosen model to the entered data in order to determine values for the model parameters. The values determined for the parameters are those that will make the predictions of the model come as close as possible to the actual (entered) data. Prism offers three forms of multiple regression.

The first of these three forms is multiple linear regression. Like simple linear regression, multiple linear regression finds the values of the parameters (regression coefficients) in the model that minimize the sum of the squares of the discrepancies between the actual and predicted Y values. In other words, multiple linear regression is a least-squares method. The advantage of this method is that the parameter estimates can be determined using fairly simple calculations. Unlike Poisson, logistic and nonlinear regression, multiple linear regression does not require an iterative approach so does not require initial estimated values for the parameters.

The other two forms of multiple regression that Prism offers are Poisson regression and logistic regression. In both of these cases, parameter estimates are determined by maximizing the likelihood function. Unlike with multiple linear regression, this cannot be achieved simply by using the least-squares method. Instead, an iterative approach is utilized instead to determine the parameter estimates that would be most likely to randomly create the observed data given the model. But unlike nonlinear regression, you don't need to specify initial values (or review the suggested values) and consider the possibility that the reported fit is actually a false minimum. That is a potential problem with nonlinear regression, but not with multiple Poisson or logistic regression.

3.5.4 Why no stepwise regression?

What is stepwise regression?

Many multiple regression programs can choose variables automatically. You give the program data on lots of variables, and it decides which ones to actually use. The appeal of automatic variable selection is clear. You just put all the data into the program, and it makes all the decisions for you. Why *stepwise*? Because the automatic procedure fits several models in steps, adding (or removing) variables from the model to find the "best" one.

Prism does not offer automatic variable selection.

The problem with stepwise?

The problem is multiple comparisons. How many models does a multiple regression program compare when given data with k independent variables and instructed to use the all-subsets method to compare the fit of every possible model? Each variable can be included or excluded from the final model, so the program will compare 2k models. For example, if the investigator starts with 20 variables, then automatic variable selection compares 2^{20} models (more than a million), even before considering interactions.

When you read a paper presenting results of multiple regression, you may not even know the number of variables with which the investigator started. Peter Flom (1) explains why this ignorance makes it impossible to interpret the results of multiple regression with stepwise variable selection:

> If you toss a coin ten times and get ten heads, then you are pretty sure that something weird is going on. You can quantify exactly how unlikely such an event is, given that the probability of heads on any one toss is 0.5. If you have 10 people each toss a coin ten times, and one of them gets 10 heads, you are less suspicious, but you can still quantify the likelihood. But if you have a bunch of friends (you don’t count them) toss coins some number of times (they don’t tell you how many) and someone gets 10 heads in a row, you don’t even know how suspicious to be. That’s stepwise.

The consequences of automatic variable selection are pervasive and serious (1,2):

- The final model fits too well. R^2 is too high.

- The best-fit parameter values are too far from zero. This makes sense. Since variables with low absolute values have been eliminated, the remaining variables tend to have absolute values that are higher than they should be.

- The confidence intervals are too narrow, so you think you know the parameter values with more precision than is warranted.

- When you test whether the parameters are statistically significant, the P values are too small and cannot be interpreted.

References

1. Flom, P. L., & Cassell, D. L. (2007). Stopping stepwise: Why stepwise and similar selection methods are bad, and what you should use. NorthEast SAS Users Group.

2. Harrell, F. (2015). Regression Modeling Strategies: With Applications to Linear Models, Logistic and Ordinal Regression, and Survival Analysis. 2nd edition. Springer. ISBN: 978-3319194240

3.6 Principles of nonlinear regression

Enter topic text here.

3.6.1 Getting started with nonlinear regression

Nonlinear regression finds the values of parameters in a nonlinear model that bring the curve defined by that model close to your data points.

- Distinguishing nonlinear regression from other kinds of regression 20

- The goal of nonlinear regression 48

- The six steps of nonlinear regression 50
- Preparing data for nonlinear regression 52
- Don't fit a model to smoothed data 53

3.6.1.1 Distinguishing nonlinear regression from other kinds of regressions

Before choosing nonlinear regression, make sure you don't really need another kind of regression. Also read about how nonlinear regression differs from linear regression 26.

Polynomial regression

A polynomial model has this form: $Y= A + BX + CX^2 + DX^3$ ....

Like linear regression, it is possible to fit polynomial models without fussing with initial values. For this reason, some programs (i.e. Excel) can perform polynomial regression, but not nonlinear regression. And some programs have separate modules for fitting data with polynomial and nonlinear regression. Prism fits polynomial models using the same analysis it uses to fit nonlinear models. Polynomial equations are available within Prism's nonlinear regression analysis.

Multiple regression

A multiple regression model has more than one independent (X) variable. Like linear and nonlinear regression, the dependent (Y) variable is a measurement.

Logistic regression

A logistic regression model is used when the outcome, the dependent (Y) variable, has only two possible values. Did the person get the disease or not? Did the student graduate or not? There can be one or several independent variables. These independent variables can be a variable like age or blood pressure, or have discrete values to encode which treatment each subject received.

GraphPad Prism does not perform logistic regression.

Proportional hazards regression

A proportional hazards model is used when the outcome is whether or not a one-time event (often death) occurred. One of the independent variables is time, and other independent variables can be used to account for treatment or other variables.

GraphPad Prism does not perform proportional hazards regression.

3.6.1.2 The goal of nonlinear regression

Nonlinear regression is used for two purposes

Scientists use nonlinear regression with one of two distinct goals:

- To fit a model to your data in order to obtain best-fit values of the parameters, or to compare the fits of alternative models. If this is your goal, you must pick a model (or two alternative models) carefully, and pay attention all the results.

- To simply fit a smooth curve in order to interpolate values from the curve, or perhaps to draw a graph with a smooth curve. If this is your goal, you can assess it purely by looking at the graph of data and curve. There is no need to learn much theory. Jump right to an explanation of interpolation with Prism. 258

The general idea of regression

Linear regression fits a straight-line model to your data. Nonlinear regression extends this idea to fit any model to your data. Distinguish nonlinear regression from linear regression 26, and from other types of regression 20.

The goal of linear and nonlinear regression is to adjust the values of the model's parameters to find the line or curve that comes closest to your data. So with linear regression, the goal is to find the best-fit values of the slope and intercept that makes the line come close to the data. With nonlinear regression of a normalized dose-response curve, the goal is to adjust the values of the EC50 (the concentration that provokes a response halfway between the minimum and maximum responses) and the slope of the curve.

More precisely, the goal of regression is to find the values of the parameters that are most likely to be correct. To do this requires making an assumption about the scatter of data around the curve.

Least-squares regression

The most common assumption is that data points are randomly scattered around an ideal curve (or line) with the scatter following a Gaussian distribution. If you accept this assumption, then the goal of regression is to adjust the model's parameters to find the curve that minimizes the sum of the squares of the vertical distances of the points from the curve.

Why minimize the sum of the squares of the distances? Why not simply minimize the sum of the actual distances?

If the random scatter follows a Gaussian distribution, it is far more likely to have two medium size deviations (say 5 units each) than to have one small deviation (1 unit) and one large (9 units). A procedure that minimized the sum of the absolute value of the distances would have no preference over a curve that was 5 units away from two points and one that was 1 unit away from one point and 9 units from another. The sum of the distances (more precisely, the sum of the absolute value of the distances) is 10 units in each case. A procedure that minimizes the sum of the squares of the distances prefers to be 5 units away from two points (sum-of-squares = 25) rather than 1 unit away from one point and 9 units away from another (sum-of-squares = 82). If the scatter is Gaussian (or nearly so), the curve determined by minimizing the sum-of-squares is most likely to be correct.

3.6.1.3 Lingo

Variables

A nonlinear regression model predicts one variable Y from another variables X. The Y variable is called the *dependent variable*, the *response variable* or the *outcome variable*. The X variable is called *independent variables*, *explanatory variables* or *predictor variables*.

The X variable can be a value that the experimenter manipulated or assigned, or a value that the experimenter measures.

Parameters

The nonlinear regression model defines the dependent variable as a function of the independent variable and a set of parameters, also called *regression coefficients*. Regression methods find the values of each parameter that make the model predictions come as close as possible to the data. This approach is analogous to linear regression, which determines the values of the slope and intercept (the two parameters or regression coefficients of the model) to make the model predict Y from X as closely as possible.

3.6.1.4 The six steps of nonlinear regression

Before learning how to fit a model using GraphPad Prism (or any program), it is worth first reviewing the necessary steps.

Step 1. Choose a model

Nonlinear regression fits a model to your data. You must, therefore, choose a model or enter a new model. This is a scientific decision that must be made by someone who understands the scientific goals of the experiment. Why a computer program should not pick a model for you 16.

Step 2. Decide whether to constrain any parameters to constant values

When performing nonlinear regression, you don't have to fit each parameter in the equation. Instead, you may fix one or more of the parameters to constant values. It is often helpful to define constants when you have only a few data points. For example, you might fix the bottom plateau of a sigmoid curve or exponential decay to zero.

Don't expect computer programs to have common sense. That is your job! Think about how you did the experiment and prepared the data, and decide whether some of the parameters should be fixed. For example, if a background signal has already been subtracted, it makes sense to fix the bottom plateau of a dose-response curve or an exponential decay curve to zero.

Step 3. Choose (or review) initial values

Nonlinear regression is an iterative procedure. The program must start with estimated values for each parameter. It then adjusts these initial values to improve the fit. GraphPad Prism supplies initial values automatically if you choose a built-in equation. If you enter your own equation, you will also need to provide initial values, or rules to generate initial values from the range of the data.

You'll find it easy to estimate initial values if you have looked at a graph of the data, understand the model, and understand the meaning of all the parameters in the equation. Remember that you just need an estimate. It doesn't have to be very accurate. If you are having problems estimating initial values, set aside your data and simulate curves using the model. Change the variables one at a time, and see how they influence the shape of the curve. Once you have a better feel for how the parameters influence the curve, you might find it easier to estimate initial values.

When fitting a simple model to clean data, it won't matter much if the initial values are fairly far from the correct values. You'll get the same best-fit curve no matter what initial values you use, unless the initial values are extremely far from correct. Initial values matter more when your data have a lot of scatter or your model has many variables.

Step 4. If you are fitting two or more data sets at once, decide whether to share any parameters

If you enter data into two or more data set columns, Prism will fit them all in one analysis. But each fit will be independent of the others unless you specify that one or more parameters are shared. When you share parameters, the analysis is called a global nonlinear regression 70.

Step 5. Decide on a weighting scheme

Nonlinear regression programs generally weight each point equally. But there are many ways to differentially weight the points 329.

Step 6. Choose which results you want reported

Nonlinear regression always reports the best-fit values of the parameters. Beyond that, Prism (and most programs) gives you many choices for which results you want reported.

3.6.1.5 Preparing data for nonlinear regression

You must create an XY data table in Prism, for use with nonlinear regression.

Follow these guidelines to enter (or preprocess) data for nonlinear regression:

- Avoid linearizing transforms 28 such as Scatchard and Lineweaver-Burke plots. Such plots are useful for displaying data but are obsolete for data analysis.

- Transforming X values can be convenient, and will not change the results of regression (so long as the model is adjusted accordingly). Use Prism's Transform analysis to do this.

- Don't smooth your data 53. You will get invalid nonlinear regression results. Fit the raw data.

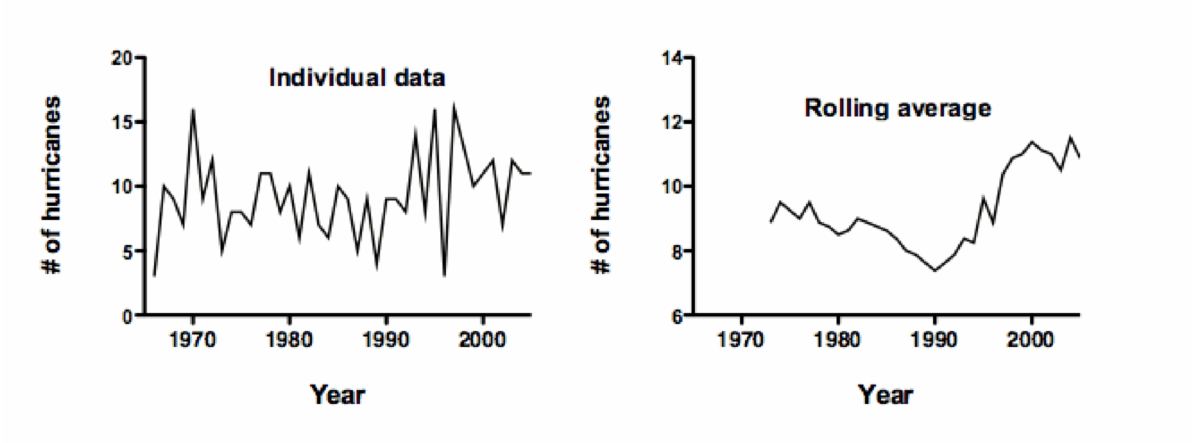
- Transforming Y values to change units or to subtract a baseline can be convenient, and will not substantially affect nonlinear regression. Use Prism's Transform analysis to do this.

- Avoid nonlinear Y transforms (reciprocals, logs) unless you have a very good reason. Such a transform can be useful it equalizes the variances (so the scatter at all points along the curve is about the same), but should not be done just to linearize the data.

- Enter raw replicates when possible, and not just mean and SD or SEM. Prism will fit the same curve either way, but there are two advantages of entering the raw data. First, you can plot every individual replicate so you see the actual data. Second, entering the individual replicates lets you choose robust nonlinear regression or automatic outlier elimination

- If you have entered replicates, first graph the individual replicates rather than mean and error bar. You may want to later plot mean and error bar, but look at a graph of the raw data first.

3.6.1.6 Don't fit a model to smoothed data



The figure shows the number of hurricanes over time. The left panel shows the number of hurricanes in each year, which jumps around a lot. To make it easier to spot trends, the right panel shows a rolling average. The value plotted for each year is the average of the number of hurricanes for that year plus the prior eight years. This smoothing lets you see a clear trend.

But there is a problem. These are not real data. Instead, the values plotted in the left panel were chosen randomly (from a Poisson distribution, with a mean of 10). There is no pattern. Each value was randomly generated without regard to the previous (or later) values.

Creating the running average creates the impression of trends by ensuring that any large random swing to a high or low value is amplified, while the year-to-year variability is muted.

You should not fit a model to the rolling average data with linear or nonlinear regression, or compute a correlation coefficient. Any such results would would be invalid and misleading. The problem is that regression assumes that each value is independent of the others, but the rolling average are not at all independent of each other. Rather each value is included as part of the neighboring values.

This example is adapted from Briggs (2008).

3.6.1.7 Reparameterizing an equation can help

Even though nonlinear regression, as its name implies, is designed to fit nonlinear models, some of the inferences actually assume that some aspects of the model are close to linear, so that the uncertainty about each parameter's value is symmetrical.

Reparameterizing an equation can make the uncertainty more symmetrical, making the SE easier to interpret and making the symmetrical asymptotic CI more helpful. Prism can compute an asymmetrical CI, and when you choose this approach it doesn't matter so much how you parametrize the equation.

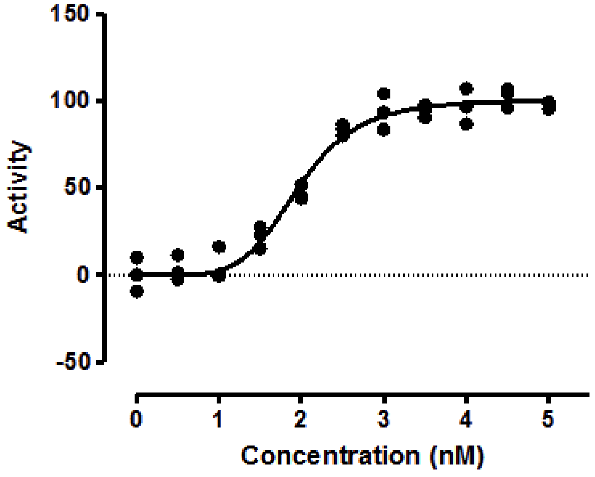
What is reparameterizing?

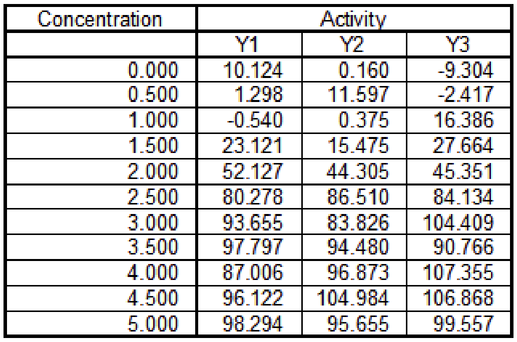
There are two forms of the model used to fit the sigmoidal enzyme kinetics data to a standard model:

```
Y=Vmax*X^h/(Khalf^h + X^h)
Y=Vmax*X^h/(Kprime + X^h)
```

The two equations are equivalent. They both fit Vmax (the maximum activity extrapolated to very high concentrations of substrate) and h (Hill slope, describing the steepness of the curve). But one model fits Khalf (the concentration needed to obtain a velocity half of maximal) and the other fits Kprime (a more abstract measure of substrate action).

Which model is best? The two are equivalent, with Kprime equal to Khalf^h , so they will generate exactly the same curve.



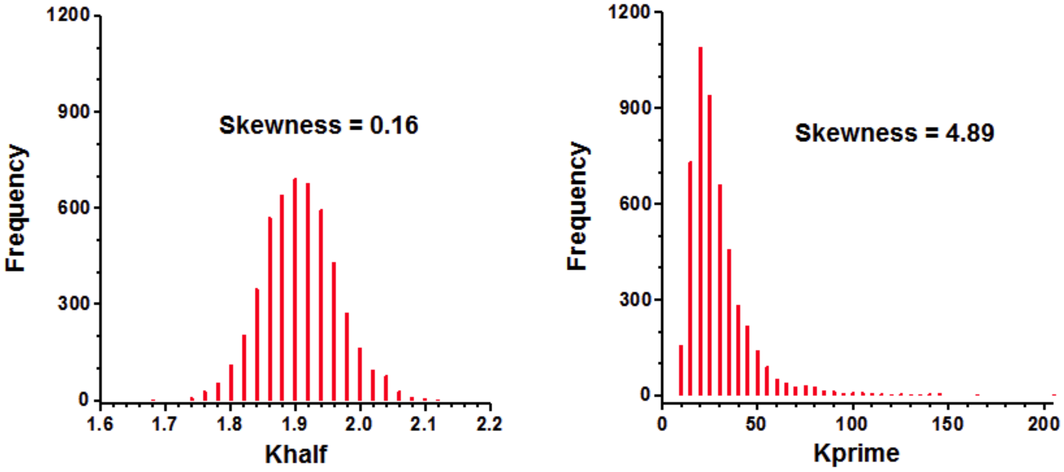


Concentration	Activity		
	Y1	Y2	Y3
0.000	10.124	0.160	-9.304
0.500	1.298	11.597	-2.417
1.000	-0.540	0.375	16.386
1.500	23.121	15.475	27.664
2.000	52.127	44.305	45.351
2.500	80.278	86.510	84.134
3.000	93.655	83.826	104.409
3.500	97.797	94.480	90.766
4.000	87.006	96.873	107.355
4.500	96.122	104.984	106.868
5.000	98.294	95.655	99.557

Since the sum-of-squares will be identical, as will the number of degrees of freedom, any comparison of this model with another will give identical results regardless of which form of this model you pick.

Distribution of parameters are not always symmetrical

Simulations can determine how symmetrical a parameter's uncertainty is. I simulated sigmoidal enzyme kinetics using Vmax=100, h=5, Kprime=25 (so Khalf=5), and Gaussian scatter with a SD equal to 7.5. The X values matched those in the figure above, with triplicate Y values at each X. Prism can repeat such simulations easily. I repeated the simulations 5000 times, fit each curve to both forms of the models, and tabulated the best-fit values of Kprime and Khalf, and computed the skewness of each.



Clearly the distribution of Khalf is quite symmetrical, and looks Gaussian. The skewness is close to zero, as expected for a symmetrical distribution. In contrast, the distribution of Kprime is quite skewed. Note that a few of the simulations had best-fit values of Kprime greater than 100. The skewness value (4.89) confirms what is obvious by inspection -- the distribution is far from symmetrical.

Quantifying asymmetry with Hougaard's skewness

The results above were obtained by running numerous simulations. There is an easier way to figure out how symmetrical a parameter is. Prism can compute Hougaards skewness 399 of each parameter, computed from the equation, the number of data points, the spacing of the X values, and the values of the parameters. For the simulated data set, Hougaard's skewness is 0.09 for Khalf and 1.83 for Kprime. A rule of thumb is to expect problems from asymmetry when the absolute value of the Hougaard's skewness is greater than 0.25, and big problems when the value is greater than 1.0. These values, which can be computed from one data set with no simulations, tells you that the symmetrical confidence intervals will be more accurate when you fit Khalf than when you fit Kprime.

Please note that while Prism 6 and 7 calculated Hougaard's skewness correctly for unweighted fits, they computed it incorrectly if you chose unequal weighting. This is fixed in Prism 8.

Consequence of asymmetrical parameters

Ideally, a confidence interval is easy to interpret. A 95% CI has a 95% chance of including the true population value of the parameter, and a 5% chance of missing it.

When analyzing real data, we never know the value of the true parameter, so never know if the interval includes it or not. But when you simulate data, you know the true values of the parameters, so can quantify the coverage of the confidence intervals. I set up the same simulations mentioned above, fit each data set to both equations, and tabulated whether each confidence interval included the true parameter value or not. This table shows the fraction of 5,000 simulations where the asymptotic symmetrical confidence interval did not include the true parameter value (25 for Kprime, and 1.9037 for Khalf).

	"95% CI"	**"99% CI"**
Ideal	5.0%	1.0%
Kprime	8.8%	4.8%
Khalf	5.1%	1.0%

These results show that Khalf is well behaved, as expected given its symmetry (see above). The 95% confidence interval is expected to miss the true value in 5.0% of the simulations. In fact, it happened 5.1% of the time. Similarly, the 99% CI is expected to miss the true value in 1.0% of the simulations, which is exactly what happened. In contrast, Kprime is less well behaved. The intervals computed to be 95% confidence intervals were not wide enough so missed the true value in 8.8% of the simulations. The 99% intervals were similarly not wide enough so missed the true value in 4.8% of the simulations. Thus the confidence intervals computed to be 99% intervals, actually turned out to be 95% intervals.

These simulations show the advantage of choosing the equation that fits Khalf, rather than the one that fits Kprime. Khalf has a symmetrical distribution so the confidence intervals computed from these fits can be interpreted at face value. In contrast, Kprime has an asymmetrical distribution and its confidence intervals cannot be interpreted at face value.

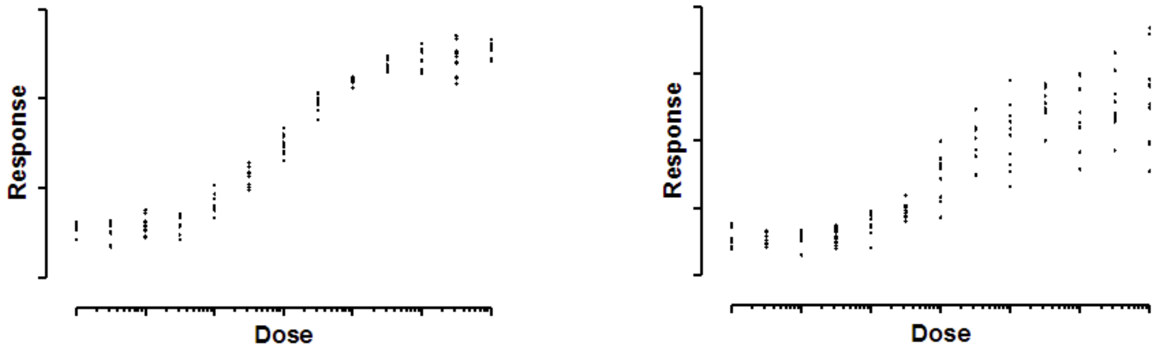
Parameterization doesn't matter if you ask Prism to present the aymmetrical profile likelihood confidence intervals

If you choose asymmetrical profile likelihood confidence intervals 343, then it doesn't matter which form of the equation you choose. The coverage will be the same in both cases, and will be very close to 95% or 99%. With this choice, you can choose the form of the equation that matches text books and papers, or that fits the way you think. If you prefer to think graphically, choose the Khalf. If you think mechanistically, choose Kprime.

3.6.2 Weighted nonlinear regression

.

3.6.2.1 The need for unequal weighting in nonlinear regression



Both graphs above show dose-response curves, with response measured in ten replicate values at each response. In the left graph, the standard deviation of those replicates is consistent. It is about the same all the way along the curve. In the right graph, the standard deviation of the replicates is related to the value of Y. As the curve goes up, variation among replicates increases.

These data are simulated. In both cases, the scatter among replicates is sampled from a Gaussian distribution. In the graph on the left, the SD of that Gaussian distribution is the same for all doses. On the right, the SD is a constant fraction of the mean Y value. When a response is twice as high as another response, the standard deviation among replicates is also twice as large. In other words, the coefficient of variation (CV) is constant.

What happens if you fit a model to the data on the right without taking into account the fact that the scatter increases as Y increases? Consider two doses that give different responses that differ by a factor of two. The average distance of the replicates from the true curve will be twice as large for the higher response. Since regression minimizes the sum of the squares of those distances those points will be expected to contribute four times as much to the sum-of-squares as the points with the smaller average Y value. In other words, the set of replicates whose average Y value is twice that of another set will be given four times as much weight. This means, essentially, that the curve fitting procedure will work harder to bring the curve near these points, and relatively ignore the points with lower Y values. You'd need to have four times as many replicates in the lower set to equalize the contribution to the sum-of-squares.

The goal of weighting is for points anywhere on the curve to contribute equally to the sum-of-squares. Of course random factors will give some points more scatter than others. But the goal of weighting is to make those differences be entirely random, not related to the value of Y. The term weight is a bit misleading, since the goal is to remove that extra weight from the points with high Y values. The goal is really to unweight.

Prism offers six choices on the Method tab of nonlinear regression, and lets you test for appropriate weighting 401.

3.6.2.2 Math theory of weighting

Relative weighting (weighting by 1/Y2)

In many experimental situations you expect the average distance (or rather the average absolute value of the distance) of the points from the curve to be higher when Y is higher. Another way to say this is that you expect the standard deviation of the residuals (distance of points from the curve) to get larger when Y is higher -- for the standard deviation to be proportional to the value of Y. Yet another way to say this is that you expect the coefficient of variation to be constant, but for the SD to not be constant.

If the SD of residuals is related to the value of Y, then the points with larger Y values will tend to be further from the curve (have larger residuals). These points will therefore contribute more to the sum-of-squares, and thus dominate the calculations.

The solution is not to minimize the sum-of-squares, but rather to minimize the weighted sum-of-squares. In other words, minimize the relative distances of the Y values of the data (Ydata) from the Y values of the curve (Ycurve). When you choose relative weighting, nonlinear regression minimizes this quantity:

$$\sum\left(\frac{Y_{data}-Y_{curve}}{Y_{curve}}\right)^2=\sum\frac{\left(Y_{data}-Y_{curve}\right)^2}{Y_{curve}{}^2}$$

The left side is easiest to understand. For each point, compute how far it is (in the Y direction) from the curve, divide that value by the Y value of the curve, and square that ratio. Add up that squared ratio for all points. The right side is equivalent. It squares the numerator and denominator separately, and then computes the ratio of those two squared values. This is the way most mathematical statisticians think about weighting, so relative weighting is also called weighting by Y^2.

Poisson weighting

Weighting by 1/Y is a compromise between minimizing the actual distance squared and minimizing the relative distance squared. One situation where 1/Y weighting is appropriate is when the Y values follow a Poisson distribution. This would be the case when Y values are radioactive counts and most of the scatter is due to counting error. With the Poisson distribution, the standard error of a value equals the square root of that value. Therefore you divide the distance between the data and the curve by the square root of the value, and then square that result. The equation below shows the quantity that Prism minimizes, and shows why it is called weighting by 1/Y (but Prism actually weights by the absolute value of Y.)

$$\sum\left(\frac{Y_{data}-Y_{curve}}{\sqrt{Y_{curve}}}\right)^2=\sum\frac{\left(Y_{data}-Y_{curve}\right)^2}{\left|Y_{curve}\right|}$$

General weighting

The first three equations below shows how absolute, Poisson and relative weighting are related. Note that taking anything to the zero power results in 1.0, so the denominator of equation on the left always equals 1.0.

Absolute	Poisson	Relative	General
$\sum\frac{\left(Y_{data}-Y_{curve}\right)^2}{Y_{curve}{}^{0}}$	$\sum\frac{\left(Y_{data}-Y_{curve}\right)^2}{Y_{curve}{}^{1}}$	$\sum\frac{\left(Y_{data}-Y_{curve}\right)^2}{Y_{curve}{}^{2}}$	$\sum\frac{\left(Y_{data}-Y_{curve}\right)^2}{Y_{curve}{}^{K}}$
K=0	**K=1**	**K=2**	**K= ?? (you enter)**

The equation on the right shows general weighting. You enter K so can customize the weighting scheme to fit your data. Generally this choice is used with values of K between 1.0 and 2.0. Reference 1 below uses this approach.

If you want to experimentally determine the best value of K, you can do so:

1. Collect data with lots (over a dozen; maybe several dozen) replicates at many points along the curve.

2. Plot the data the usual way to make sure the data seem correct.

3. Create a second graph that ignores the X values (time or concentration...). Instead, in this new graph, X is the logarithm of mean of the replicate Y values for each point on the first graph, and Y on this new graph is the logarithm of variance (square of the standard deviation) among the replicate values. You can use either natural logarithms or logarithms base 10. It doesn't matter, so long as both logarithms uses the same base.

4. Fit a straight line to this graph using linear regression. Since the assumption of a Gaussian variation around this line is dubious, use nonlinear regression and choose a robust fit.

5. The slope of this regression line is K. If K is close to 0.0, then the SD does not vary with Y so no weighting is needed. If K is close to 2.0, then the standard deviation is proportional to Y, and relative weighting is appropriate. If K is close to 1.0, choose Poisson weighting. If K has some other value, use general weighting and enter this value of K as a constant.

Note that if Ycurve is negative, Prism actually takes the the absolute value of Ycurve to the K power.

Rarely used weighting choices in Prism

The choices to weight by 1/X or $1/X^2$ weight the points at the left part of the graph more than points to the right. This is common in some fields when fitting straight lines to bioassay data.

Prism also offers the choice to weight by the reciprocal of the standard deviation squared. This option minimizes the following:

$$\sum\left(\frac{Y_{data}-Y_{curve}}{SD}\right)^2=\sum\frac{\left(Y_{data}-Y_{curve}\right)^2}{SD^2}$$

This method is most useful when you have formatted the data table for entry of SD values, but then entered values that you had calculated elsewhere based on understanding how the scatter (or errors) arise in your experimental system. The "SD" values you enter should be computed weighting factors, not the actual SD of the data.

If you enter the actual SD into the SD subcolumn, or enter replicate values so Prism computes the SD, then Prism will use these actual SD values as weighting factors. This is way less useful than it sounds. With small to moderate sample sizes, the SD will jump around a lot by chance, and it is not appropriate to use these random SD values for weighting. Weighting should be by predicted SD not actual SD (which is subject to random factors).

Of course, weighting by SD is impossible if any of the entered SD values are zero. It also is impossible if Prism is computing the SD from replicates and there is only one replicate, or if all replicates are identical (so their SD equals zero).

Prism doesn't use weighting on the first iteration

There is one situation where weighting by the predicted Y values of the curve causes problems - when the initial values are really bad. The initial curve can be quite far from the points, and in some cases can have Y=0 for some X values, which makes weighting by 1/Y or 1/Y2 impossible. To improve the fitting, even if the initial values generate a curve far from the data, Prism uses no weights for the first iteration. This first iteration brings the curve closer to the points. From then on, Prism uses the weighting function you specify. Essentially it uses the results of one iteration of unweighted fitting as the initial values for the weighted fit.

Reference

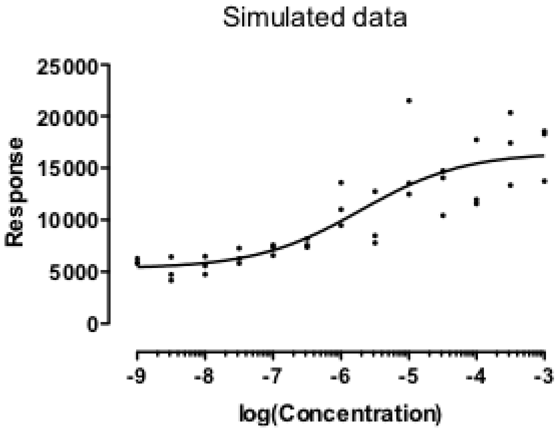
1. LM Lavasseur, H Faessel, HK SLocum, and WR Greco, Implications for Clinical Pharmacodynamic Studies of the Statistical Characterization, of an In Vitro Antiproliferation Assay, J. Pharmacokinetics and pharmacodynamics, 26: 717-733, 1998.

3.6.2.3 Don't use weighted regression with normalized data

Before fitting a log(dose) vs. response curve, it is common to first normalize the data 458. This page explains why one should not use weighted regression in this situation.

Weighted fits to nonnormalized data -- works great

The graph below shows simulated data that show a common situation. The scatter among replicates is greater when the Y values are greater. In fact, these data were simulated so the SD among replicates is proportional to the Y value.



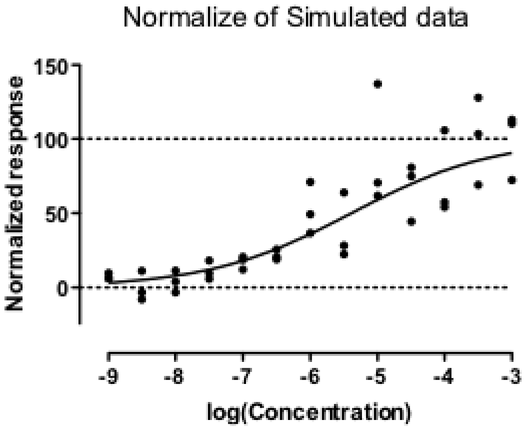
These data fit nicely using relative weighting. This minimizes the sum of the squares of the relative distance between the points and the curve. In other words it minimizes:

$$\sum\left(\frac{Y_{data}-Y_{curve}}{Y_{curve}}\right)^2=\sum\frac{\left(Y_{data}-Y_{curve}\right)^2}{Y_{curve}{}^2}$$

I ran 10,000 simulations, and found that in every case the fit worked well and gave a reasonable answer (with the EC50 within the range of the data). No surprise so far. The data were fitting assuming a model that exactly matched the method used to simulate the data, and those fits worked well.

Weighted fits to normalized data -- ugh

People often like to normalize dose-response data so the Y values range from 0% to 100%. What happens if these normalized data are fit with weighted nonlinear regression?



I answered that question with simulations. Of 1,000 simulated data sets, 223 could not be fit at all. Moreover, 60 simulated data sets gave nonsense results, with the EC50 outside of the range of the data. The remaining 72% of the simulations seemed ok, but the confidence intervals were very wide in some.

What's going on?

With the data that were not normalized, the Y value of the best-fit curve at X = -4 is 3.05 times larger than the Y value of the curve at X=-8. Since the weighting is proportional the the Y value of the curve squared, the points near the top of the curve (at X=-4) get 3.05^2 , or 9.28, times as much weight as points near the bottom of the curve (at X=-8).

The situation is very different with the normalized data. With the first simulated data set, the Y value of the best-fit curve at X = -4 is 17.77 times larger than the Y value of the curve at X=-8. Since the weighting is proportional the the Y value of the curve squared, the points near the top of the curve (at X=-4) get 17.77^2 , or 315.8, times as much weight as points near the bottom of the curve (at X=-8). Since the points at the top of the curve gets so much more weight that those at the bottom, those at the bottom are essentially ignored, making the entire curve fit not work very well.

Another problem. Some of the Y values near the bottom of the normalized data sets are negative. Weighting factors really don't make any sense at all when some values are negative and some are positive.

In summary: After normalizing the SD among replicates is no longer proportional to the Y values, so relative weighting is not appropriate.

Bottom line

The whole idea of weighted nonlinear regression is to match the weighting scheme used by the regression to the variation in the actual data. If you normalize the data, none of the usual weighting schemes will work well.

If you really want to show your data on a normalized axis running from 0% to 100%, you can do so. First fit the model to the actual data using an appropriate weighting scheme. Then normalize both the data and the curve.

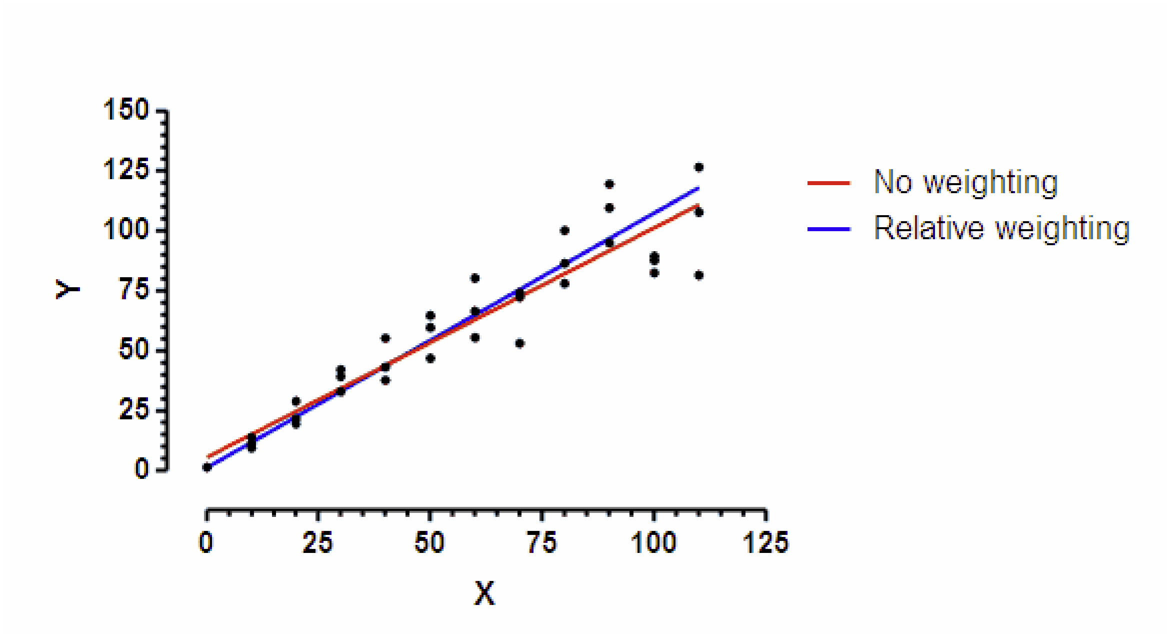
Details and link to Prism file

3.6.2.4 What are the consequences of choosing the wrong weighting method

What happens if you leave the choice on the weighting tab set to its default -- no weighting -- when the scatter is actually related to the Y value? Of course, the answer is "it depends"? It depends on how large the scatter is, and how large a range the Y values span. The rest of this article looks at one example, and simulates many data sets to see what happens when the fitting is done inappropriately.

Simulations

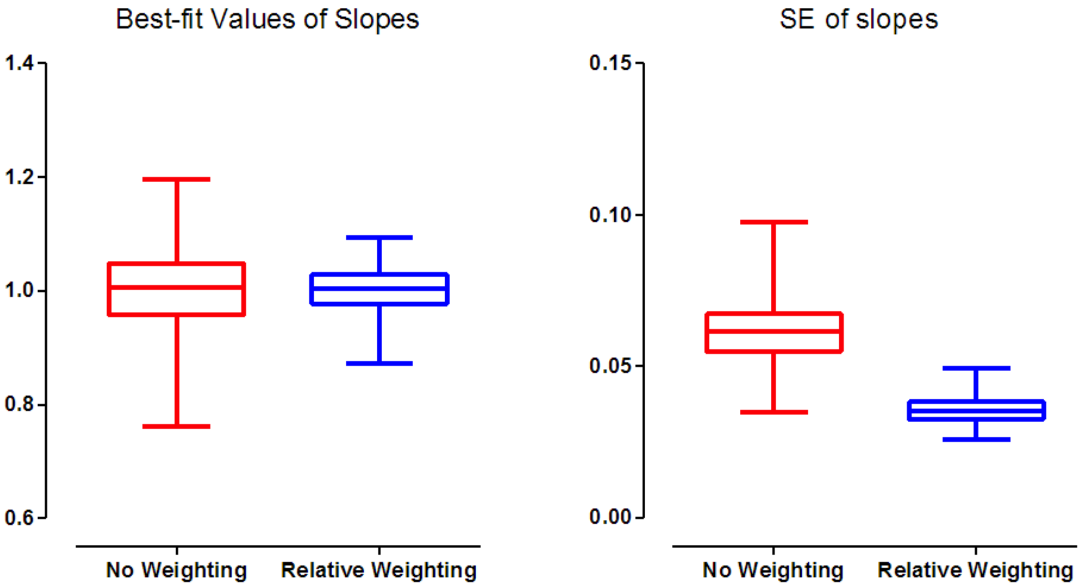
I picked a very simple model -- a straight line. I simulated the data so the SD of scatter at any point along the line is proportional to the Y value of that point. The graph below shows a simulated data set. You can clearly see that the scatter among replicates increases as the line goes up.



The line was fit to the data by "nonlinear" regression. Prism does not offer differential weighting as part of its linear regression analysis, but "nonlinear" regression can fit a straight line with many options not available in the linear regression analysis 123.

The red line used the default choice -- no weighting; minimize sum of squares. The blue line used relative weighting. This choice is appropriate when you expect the SD of replicate residuals to be proportional to Y. The two lines are not identical.

I simulated 5000 such data sets using the Monte Carlo analysis of Prism. Each of the 5000 simulated data sets was fit with both unweighted and weighted (relative weighting) regression. I recorded both the best-fit value of the slope and its standard error (SE) for both analyses of each of the 5000 data sets.



The scatter among replicates was proportional to Y, so relative weighting is appropriate. The results of these simulations are shown in blue. The results in red show the results when the regression used equal weighting (the default). The boxes extend from the 25th to the 75th percentile, with a line at the median (50th percentile). The whiskers extend down to the smallest value and up to the largest.

The left graph above shows the distribution of the best-fit values of the slopes. Chossing the wrong weighting scheme would not systematically result in a best-fit value of slope that is too high (too steep) or too low (too shallow). Indeed, the median best-fit values for slope are almost identical for the no-weighting and relative-weighting fits. But note that the results with no weighting has a much broader spread of slopes. Since these are simulated data, we know the true slope (1.0) and can quantify how far the best-fit slope is from that idea for each simulated data set. The median error when the data were fit without differential weighting (red) was 0.04610, compared to 0.02689 for the simulations with relative weighting (blue).In this example, choosing the wrong weighting scheme increased the median error by 71%.

The graph on the right above shows the distribution of the SE of the slopes. The SE of the slope is much higher on average and less consistent (wider spread) when the fits were done with equal weighting for all points. The median SE for the unweighted fits was 73% larger than the median SE for the weighted fits. Since the width of the confidence interval is proportional to the SE, this means the median confidence interval was 73% wider when the fits were done without weighting.

Since these are simulated data, we know the true population slope (it is 1.0). So we can ask for each simulation whether the reported 95% confidence interval included the true value. For the relative weighting simulations, the 95% confidence intervals included the true value in 95.3% of the simulated data sets. When those same data sets were analyzed without weighting, only 92.6% of the "95%" confidence intervals included the true value.

Summary

In summary, when we chose the wrong weighting scheme in this example:

- The best-fit value of the slope was less precise.

- The SE of the slope was larger, and thus the confidence intervals were wider.

- Even though the 95% confidence intervals were wider, they weren't wide enough. The "95%" confidence interval only included the true value in fewer than 95% of the simulations.

This is just one example. With other examples, the choice of weighting method matters less. But with still other examples, it may matter more. It is worth trying to choose the appropriate weighting scheme.

Download the Prism file for this example.

3.6.2.5 Poisson regression

.When to use Poisson regression

Use Poisson regression when the Y values are the actual number of objects or events counted. Be sure the values are not normalized in any way but represent the number of objects or events actually counted. Let's say that the Y values are radioactivity, you counted each sample for 10 minutes, but the instrument reported counts per minute. It would be a mistake to analyze the counts per minute data with Poisson regression, since those are not the actual number of radioactive decay events that were counted. You'd need to multiply all those values by 10 (the number of minutes each sample was counted) to have the actual number of events counted.

Since the Y values are counts, they can't be negative or fractional.

What choices are unavailable with Poisson regression

These options are not available with Poisson nonlinear regression

- No standard errors of parameters.
- Confidence intervals of parameters are always computed using an algorithm that usually produces asymmetrical confidence intervals. There us no option for a simpler method that reports symmetrical confidence intervals.
- No confidence or prediction bands.
- No normality tests of residuals (since the residuals are not expected to be Gaussian).
- Different ways to quantify goodness-of-fit 387.

Poisson regression works differently

Ordinary regression (assuming residuals are Gaussian) works by minimizing the sum-of-squares of the residuals. This works because minimizing the sum-of-squares is the same as maximizing likelihood when the residuals are sampled from a Gaussian distribution.

Poisson regression works differently, directly maximizing the likelihood.

3.6.3 The many uses of global nonlinear regression

A global model defines a family of curves, rather than just a single curve, with some parameters shared between data sets. For each shared parameter, Prism fits one (global) best-fit value that applies to all the data sets. For each non-shared parameter, Prism fits a separate (local) best-fit value for each data set.

- What is global nonlinear regression? 70

- Using global regression to fit incomplete datasets 72
- Fitting models where the parameters are defined by multiple data sets 74
- Advice: Don't use global regression if datasets use different units 76

3.6.3.1 What is global nonlinear regression?

A global model defines a family of curves, rather than just a single curve. Some parameters are shared so a single parameter value applies to all the curves, while other parameters apply to each data individually. For each shared parameter, fit one (global) best-fit value that applies to all the data sets. For each non-shared parameter, fit a separate (local) best-fit value for each data set.

Nonlinear regression finds parameters of a model that make the curve come as close as possible to the data. This is done by minimizing the sum of the squares of the vertical distances between the data points and curve. Global nonlinear regression extends this idea to fitting several data sets at once and minimizes the sum (of all data sets) of sum (of all data points) of squares.

Prism makes it very easy to perform global nonlinear regression. Enter your data on one data table, click analyze, choose nonlinear regression and choose a model. On the Constrain tab 336 of the Nonlinear regression dialog, choose which parameter(s) to share among data sets.

3.6.3.2 How global nonlinear regression works

Nonlinear regression works iteratively. With each iteration, it alters the values of the parameters to lower the sum-of-squares.

With global fitting, Prism does not fit each data set individually and then average them together. Rather, it fits all the data at once to the global model. With each iteration, it minimizes the sum (over data sets) of sum (over data points) of squares. There is nothing special about any particular parameter in any particular equation. It doesn't know about EC50s or plateaus or rate constants. It just uses the nonlinear regression algorithm to step by step improve the sum of sum of squares.

Here are the details:

Let's assume that we have four data sets and equation with three parameters. One way to represent this is with the equation y=f(X, A, B, C), or in other words, your model "y" is a function defined by X and three variables; A, B, and C. In this case, let's assume that A is shared among your four data sets, while B and C are calculated individually for each.

The above is the user-friendly description of experimental design (or problem being solved by fitting engine). From the math point of view, Prism actually fits the following function to all the data (from all data sets).

```
F(X, A, B1, B2, B3, B4, C1, C2, C3, C4) =
{ IF (X from Data set 1) THEN f(X, A, B1, C1) { IF (X from Data set 2)
THEN f(X, A, B2, C2) { IF (X from Data set 3)
THEN f(X, A, B3, C3) { IF (X from Data set 4) THEN f(X, A, B4, C4)
```

Note that parameter A (which is shared) looks like "common" one while parameters B and C (not shared) are split on "individual" parameters B1, B2, B3, B4 and C1, C2, C3, C4 respectively.

Here is an alternative way to write this function:

```
F(X, A, B1, B2, B3, C1, C2, C3) = f(X, A, B1*DummyB_1 + B2* DummyB_2 + B3* Dum
+ B4* DummyB_4, C1* DummyC_1 + C2* DummyC_2 + C3* DummyC_3 + C4* DummyC_4)
```

In this version of the equation, DummyB_i = DummyC_i = 1, if X belongs to i-th data set DummyB_i = DummyC_i = 0, if X doesn't belong to i-th data set

Both forms of the equation (which are equivalent) will compute a Y value for each curve for every X value in the data, using the current values of the parameters. Prism can then compute the difference between the actual Y value (for each data set for each point) and the predicted Y value. Sum the squares of all those points to get the sum-of-squares which nonlinear regression minimizes. So the nonlinear regression works as usual. It just has a fancier model that accounts for which parameters have one value for all data sets and which have one value for all data sets (shared).

3.6.3.3 The uses of global nonlinear regression

Global nonlinear regression is useful in (at least) three situations:

- Test whether a parameter value differs significantly between data sets. Compare the sum-of-squares (to assess goodness-of-fit) when the parameter is shared, with the sum of the sum-of-squares when the parameter is fit individually to each dataset. In Prism, set up this kind of comparison in the Compare tab 336.

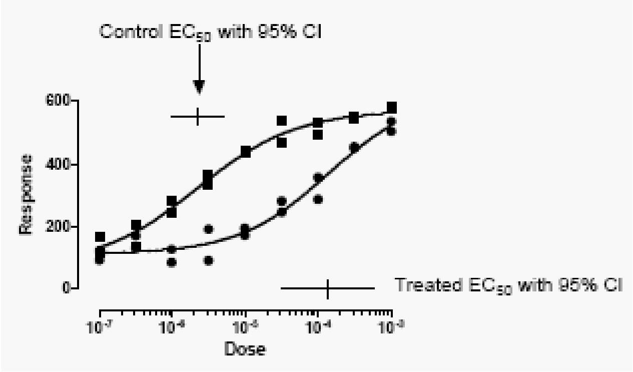
- Fit families of data where each dataset is incomplete, but the entire family of datasets defines the parameters. For example, one data set may do a great job of defining the bottom plateau of a dose-response curve, while another data set defines the top. Fit the two data sets separately, and the results may be ambiguous (very wide confidence intervals). Globally fit both curves, and the results might be very tight. See an example 72.

- Fit models where the parameter(s) you care about cannot be determined from any one dataset, but only from the relationship between several data sets 74. Another example is fitting enzyme inhibition data 622.

The first two uses of global fitting do not require writing special models. The third use requires that you write a model for this purpose.

3.6.3.4 Using global regression to fit incomplete datasets

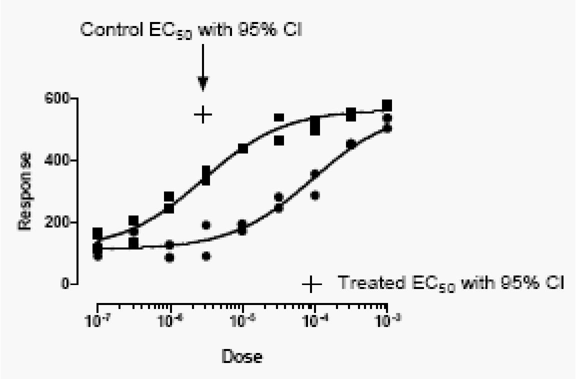
The graph below shows two dose-response curves. The goal of the experiment is to determine the two EC_{50} values. The EC_{50} is the concentration (dose) that gives a response half-way between the minimum and maximum responses. Each curve in the graph below was fit individually to one of the data sets. The horizontal lines show the 95% confidence interval of the EC_{50}.



While the curves nicely fit the data points, the confidence intervals are quite wide. We really haven't determined the EC_{50} with sufficient precision to make useful conclusions. The problem is that the control data (squares) don’t really define the bottom plateau of the curve, and the treated data (circles) don’t really define the top plateau of the curve. Since the data don’t define the minimum and maximum responses very well, the data also don’t define very clearly the point half-way between the minimum and maximum responses. Accordingly, the confidence intervals for each EC_{50} extend over more than an order of magnitude. The whole point of the experiment was to determine the two EC_{50} values, but there is an unacceptable amount of uncertainty in the value of the best-fit values of the EC_{50}.

The problem is solved by sharing parameters. For this example, share the parameters that define the top and bottom plateaus and the slope. But don’t share the EC50 value, since the EC50 values for control and treated data are clearly distinct.

Here are the results.



The graph of the curves looks only slightly different. But now the program finds the best-fit parameters with great confidence. The 95% confidence intervals for the EC_{50} values span about a factor of two (compared to a factor of ten or more when no parameters were shared).

The control data define the top of the curve pretty well, but not the bottom. The treated data define the bottom of the curve pretty well, but not the top. By fitting both data sets at once, sharing some parameters, both EC50 values were determined with reasonable certainty.

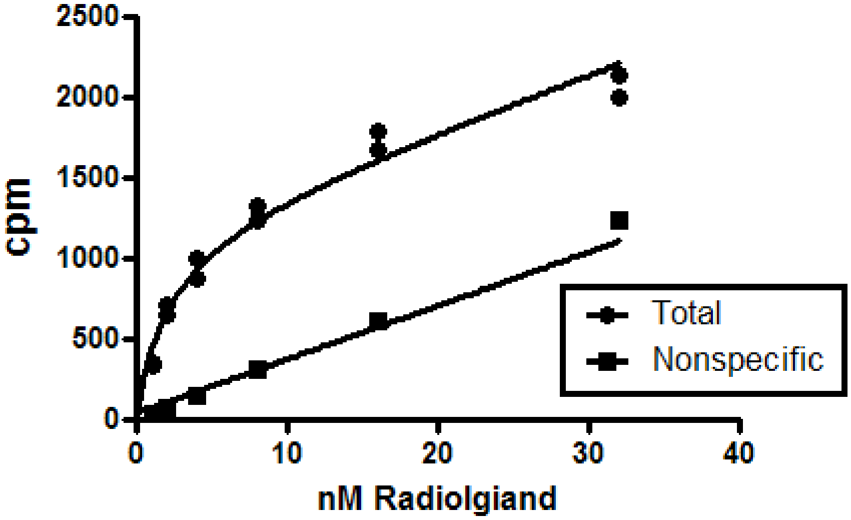
3.6.3.5 Fitting models where the parameters are defined by multiple data sets

Global fitting is most useful when the parameters you care most about are not defined by any one data set, but rather by the relationship between two data sets.

Sample data

Choose the XY sample data set: Binding --Saturation binding to total and nonspecific

Fit the data using nonlinear regression, open the "Binding -- Saturation" list of equations, and choose "One site -- total and nonspecific". You'll see the fit below.



Explanation of the equation and global fitting

This experiment measured equilibrium binding of radioligand at various concentrations of radioligand to find the B_{max} and K_d of the radioligand. Since the ligand binds to nonspecific sites as well as the receptor of interest, the experiment measured both total binding and nonspecific binding (binding of radioligand in the presence of an excess of an unlabeled receptor blocker).

These kind of data are often analyzed by first subtracting the nonspecific binding from the total binding. The resulting specific binding is then fit to a model that describes equilibrium binding to one receptor site.

Global fitting simultaneously fits both the total binding and the nonspecific binding. There is no need to first subtract the two data sets. The only trick is to write a model that fits different equations to each data set. Prism's built in equation is set up as follows:

```
specific=Bmax*X/(X+Kd)
nonspecific=NS*X + Background
<A>Y=specific+nonspecific
<B>Y=nonspecific
```

The first line defines specific saturable binding.

The second line defines nonspecific binding to be a constant fraction of added radioligand (X) plus a background (which is often zero).

The third line is preceded by <A>, so it only applies to the first data set (column A, total binding). It defines the Y values in that dataset to equal the sum of total and nonspecific binding.

The fourth line is preceded by <B> so only applies to the second data set, and defines those Y values to equal nonspecific binding.

The equation is defined with the constraint that the parameters NS and background are shared between the two data sets. That way, Prism finds one best-fit value for NS and background, based on fitting both data sets. Since Bmax and Kd are only used in fitting the first dataset, it wouldn't be meaningful to share these parameters.

The parameters you care about (Bmax and Kd) cannot be determined precisely by fitting just one dataset. But fitting a model that defines both data sets (and their relationship) while sharing the parameter NS between the datasets, lets Prism get the most information possible from the data.

3.6.3.6 Advice: Don't use global regression if datasets use different units

Global fitting works by minimizing the sum (for all data sets) of the sum (for all data points) of squares of distances of the data points from the curve. This approach only makes sense when all the data are expressed in the same units.

If different data sets are expressed in different units, be very cautious about using global fitting. The problem is that your decision about which units to use can change the results.

For example, imagine what happens if you change one data set from expressing weight in grams to expressing weight in milligrams. All the values are now increased by a factor of one thousand, and the sum-of-squares for that data set is increased by a factor of one million (one thousand squared). Compared to other data sets, expressed in different units, this data set now has a much greater impact on the fit. If you really need to do global fit to data sets using different units, consider first normalizing the data so they are comparable.

3.6.4 Comparing fits of nonlinear models

When fitting a nonlinear model to data, your main objective is often to discriminate between different models, or to ask whether an experimental intervention changed a parameter.

- Comparing models can answer four distinct questions[77]:

  ✓ For each data set, which of two equations (models) fits best?

  ✓Do the best-fit values of selected unshared parameters differ between data sets?

  ✓For each dataset, does the best-fit value of a parameter differ from a theoretical value?

✓Does one curve adequately fit all the data

- Prism lets you choose from two approaches to comparing models 78: the extra sum-of-squares F tests and the AICc approach based on information theory.

- How the F test works to compare models 82

- How the AICc computations work 84

- The idea of comparing models extends way beyond nonlinear regression. In fact, much of statistics can be viewed as comparing models 86

3.6.4.1 Questions that can be answered by comparing models

When fitting data with regression, in many cases your main objective may be to compare the fits of different models, or to ask if an experimental intervention changed a parameter.

On the Compare tab 334 of the Nonlinear regression dialog, Prism offers four choices:

For each data set, which of two equations (models) fits best?

Compare the fit of two models, taking into account differences in the number of parameters to be fit. Most often, you will want to compare two related equations. Comparing the fits of two unrelated equations is rarely helpful.

Example: Compare a one-phase exponential decay with a two-phase exponential decay.

Do the best-fit values of selected unshared parameters differ between data sets?

Compare the fit when the selected parameter(s) are shared among all datasets with the fit when those parameter(s) are fit individually to each dataset.

If you pick one parameter, you are asking whether the best-fit value of that one parameter differs among datasets.

If you pick all the parameters, you are asking whether a single curve adequately fits all the data points, or if you get a better fit with individual curves for each dataset.

Example: Fit a family of dose-response curves and compare the fit when the slope factor (Hill slope) is shared with the fit when each curve is fit individually. This is a way to test whether the curves are parallel.

For each dataset, does the best-fit value of a parameter differ from a theoretical value?

You may have theoretical reasons to believe that a parameter will have a certain value (often 0.0, 100, or 1.0). Compare the fit when the parameter is constrained to that value with the unconstrained fit.

Example: Test if a Hill Slope differs from 1.0 (a standard value).

Does one curve adequately fit all data sets?

This choice compares the fits of separate curves to each data set with the fit of a single curve fit to all the data sets. It asks whether there is evidence that the treatments did anything to shift the curves.

This choice is identical to choosing "Do the best-fit values of selected unshared parameters differ between data sets?" and then selecting all the parameters.

3.6.4.2 Approaches to comparing models

Approach to comparing models

Which model is 'best'? At first, the answer seems simple. The goal of nonlinear regression is to minimize the sum-of-squares, so it seems as though the model with the smaller sum-of-squares is best. If the two alternative models both have the same numbers of parameters, then indeed that is the best approach.

But that approach is too simple when the models have different numbers of parameters, which is usually the case. A model with more parameters can have more inflection points, so of course comes closer to the points. It can bend and twist more to get near the data points. Accordingly, a two-phase model almost always fits better than a one-phase model, and a three-phase model fits even better. So any method to compare a simple model with a more complicated model has to balance the decrease in sum-of-squares with the increase in the number of parameters.

Three statistical approaches to comparing models

Prism offers three approaches to comparing models with different numbers of parameters. These are not the only methods that have been developed to solve this problem, but are the most commonly used methods.

Extra sum-of-squares F test

The Extra sum-of-squares F test 82 is based on traditional statistical hypothesis testing. It is used only for least-squares regression (not Poisson regression).

The null hypothesis is that the simpler model (the one with fewer parameters) is correct. The improvement of the more complicated model is quantified as the difference in sum-of-squares. You expect some improvement just by chance, and the amount you expect by chance is determined by the number of data points and the number of parameters in each model. The F test compares the difference in sum-of-squares with the difference you would expect by chance. The result is expressed as the F ratio, from which a P value is calculated.

The P value answers this question:

> If the simpler model was correct, in what fraction of experiments (the size of yours) will the difference in sum-of-squares be as large as you observed, or even larger?

If the P value is small, conclude that the simple model (the null hypothesis) is wrong, and accept the more complicated model. Usually the threshold P value is set at its traditional value of 0.05. If the P value is less than 0.05, then you reject the simpler (null) model and conclude that the more complicated model fits significantly better.

Likelihood ratio test

When you choose Poisson nonlinear regression, Prism does not offer the F test (above) but instead the likelihood ratio test.

The likelihood ratio answers this question: How much more likely would the data have been observed one model were true compared to if the other model were true. A P value is computed from the likelihood ratio and the the difference in df between the two models.

The P value answers this question:

> If the simpler model was correct, in what fraction of experiments (the size of yours) will the likelihood ratio be as large as you observed, or even larger?

If the P value is small, conclude that the simple model (the null hypothesis) is wrong, and accept the more complicated model. Usually the threshold P value is set at its traditional value of 0.05. If the P value is less than 0.05, then you reject the simpler (null) model and conclude that the more complicated model fits significantly better.

The extra sum-of-squares F test is equivalent to the likelihood ratio test when you choose least-squares regression.

Information theory approach Akaike's criterion (AIC)

This alternative approach is based on information theory, and does not use the traditional “hypothesis testing” statistical paradigm. Therefore it does not generate a P value, does not reach conclusions about “statistical significance”, and does not “reject” any model.

The method determines how well the data supports each model, taking into account both the goodness-of-fit (sum-of-squares) and the number of parameters in the model. The results are expressed as the probability that each model is correct, with the probabilities summing to 100%. If one model is much more likely to be correct than the other (say, 1% vs. 99%), you will want to choose it. If the difference in likelihood is not very big (say, 40% vs. 60%), you will know that either model might be correct, so will want to collect more data. How the calculations work 84.

Which approach to choose?

In most cases, the models you want to compare will be 'nested'. This means that one model is a simpler case of the other. For example, a one-phase exponential model is a simpler case of a two-phase exponential model. A three parameter dose-response curve with a standard Hill slope of 1.0 is a special case of a four parameter dose-response curve that finds the best-fit value of the Hill slope as well.

If the two models are nested, you may use either the F test (or likelihood ratio, if Poisson regression) or the AIC approach. The choice is usually a matter of personal preference and tradition. Basic scientists in pharmacology and physiology tend to use the F test. Scientists in fields like ecology and population biology tend to use AIC approach.

If the models are not nested, then the F test and the likelihood ratio test are not valid, so you should choose the information theory approach. Note that Prism does not test whether the models are nested.

How do these methods work to compare data sets?

The Compare tab 334 of Prism lets you ask "Do the best-fit values of selected unshared parameters differ between data sets?" or "Does one curve adequately fit all data sets?". Applying the F test or Akaike's method to answering these questions is straightforward. Prism compares the sum-of-squares of two fits.

- In one fit, the model is separately fit to each data set, and the goodness-of-fit is quantified with a sum-of-squares. The sum of these sum-of-square values quantifies the goodness of fit of the family of curves fit to all the data sets.
- The other fit is a global fit to all the data sets at once, sharing specified parameters. If you ask Prism whether one curve adequately fits all data sets, then it shares all the parameters.

These two fits are nested (the second is a simpler case of the first, with fewer parameters to fit) so the sums-of-squares (actually the sum of sum of squares for the first fits) can be compared using either the F test 82 or Akaike's method 84.

Don't compare models using R^2 or the adjusted R^2

R^2 is a measure of how well a model fits your data, so it would seem to make sense to choose among competing models by picking the one that has the lowest R^2 or adjusted R^2. In fact, it works really poorly (1). Don't use this method!

1. Spiess, A.-N. & Neumeyer, N. An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach. BMC Pharmacol 10, 6–6 (2010).

Interpreting comparison of models 410

3.6.4.3 How the F test works to compare models

The extra sum-of-squares F test compares the fits of two nested models fit with least-square regression. *Nested* means one model (the simpler one, model 1 below) is a special case of the other model (the more complicated one; model 2 below).

If the simpler model is correct, the relative increase in the sum of squares (going from more complicated to simpler model) is expected to equal the relative increase in degrees of freedom. In other words, if the simpler model is correct you expect that:

$$(SS1 - SS2) / SS2 \approx (DF1 - DF2) / DF2$$

SS1 is the sum-of-squares for the simpler model (which will be higher) and SS2 is the sum-of-squares of the more complicated model. Recall that the number of degrees of freedom equals the number of data points minus the number of parameters. The more complicated model has more parameters so has fewer degrees of freedom.

If the more complicated model is correct, then you expect the relative increase in sum-of-squares (going from complicated to simple model) to be greater than the relative increase in degrees of freedom:

$$(SS1 - SS2) / SS2 > (DF1 - DF2) / DF2$$

The F ratio quantifies the relationship between the relative increase in sum-of-squares and the relative increase in degrees of freedom.

$$F = \frac{(SS1 - SS2)/SS2}{(DF1 - DF2)/DF2}$$

That equation is more commonly shown in an equivalent form:

$$F = \frac{(SS1 - SS2)/(DF1 - DF2)}{SS2/DF2}$$

F ratios are always associated with a certain number of degrees of freedom for the numerator and a certain number of degrees of freedom for the denominator. This F ratio has DF1-DF2 degrees of freedom for the numerator, and DF2 degrees of freedom for the denominator.

If the simpler model is correct you expect to get an F ratio near 1.0. If the ratio is much greater than 1.0, there are two possibilities:

- The more complicated model is correct.
- The simpler model is correct, but random scatter led the more complicated model to fit better. The P value tells you how rare this coincidence would be.

The P value answers this question:

> If model 1 is really correct, what is the chance that you would randomly obtain data that fits model 2 so much better?

If the P value is low, conclude that model 2 is statistically significantly better than model 1. Otherwise, conclude that there is no compelling evidence supporting model 2, so accept the simpler model (model 1).

3.6.4.4 How the Likelihod ratio test works to compare models

The likelihood ratio test compares the fits of two nested models fit by Poisson regression. *Nested* means one model (the simpler one, model 1 below) is a special case of the other model (the more complicated one; model 2 below).

The X^2 ratio quantifies the relative goodness of fit of the two models:

$$X^2=2\ln(LR)$$

If the simpler model is correct you expect to get an LR ratio near 0, so Q will be near 2. If the ratio is much greater than 2.0, there are two possibilities:

- The more complicated model is correct.
- The simpler model is correct, but random scatter led the more complicated model to fit better. The P value tells you how rare this coincidence would be.

The P value is computed from X^2 using the chi-square distribution. The degrees of freedom equals the difference between the df of the two models.

The P value answers this question:

> If model 1 is really correct, what is the chance that you would randomly obtain data that fits model 2 so much better?

3.6.4.5 How the AICc computations work

While the theoretical basis of Akaike's method is difficult to follow, it is easy to do the computations and make sense of the results.

The fit of any model to a data set can be summarized by an information criterion developed by Akaike, called the AIC. If you accept the usual assumptions of nonlinear regression (that the scatter of points around the curve follows a Gaussian distribution), the AIC is defined by a simple equation from the sum-of-squares and number of degrees of freedom of the two models. It is not possible to make sense of this AIC value itself, because its units depend on which units you use for your data.

To compare models, it is only the difference between the two AIC values that matter. When you take the difference, the units cancel out and the result is unitless.

For least squares regression, the equation is:

$$\Delta \mathrm{AIC} = \mathrm{N} \times \ln\left(\frac{\mathrm{SS2}}{\mathrm{SS1}}\right) + 2\Delta \mathrm{DF}$$

For Poisson regression, the equation is:

$$\Delta AIC = 2\ln(LR) + 2\Delta df$$

The equation for AICc for logistic regression 230 is nearly identical to the equation for Poisson regression (using the number of parameters in place of the degrees of freedom in the equation).

The equation now makes intuitive sense. Like the F test, it balances the change in goodness-of-fit as assessed by sum-of-squares (or likelihood ratio for Poisson regression) with the change in the number of degrees of freedom (due to differences in the number of parameters to be fit). Since model 1 is the simpler model, it will almost always fit worse, so SS1 will be greater than SS2. Since the logarithm of a fraction is always negative, the first term will be negative. Model 1 has fewer parameters and so has more degrees of freedom, making the last term positive. If the net result is negative, that means that the difference in sum-of-squares is more than expected based on the difference in number of parameters, so you conclude that the more complicated model is more likely.

Prism reports the difference between the two AICc values as the AICc of the simpler model minus the AICc of the more complicated model. When the more complicated (more parameters) model has the lower AICc and so is preferred, Prism reports the difference of AICc as a positive number. When the simpler model has the lower AICc and so is preferred, Prism reports the difference of AICc as a negative number.

The equation above helps you get a sense of how AIC works – balancing change in goodness-of-fit vs. the difference in number of parameters. But you don’t have to use that equation. Just look at the individual AIC values, and choose the model with the smallest AIC value. That model is most likely to be correct.

Prism actually doesn't report the AIC, but rather the AICc. That value includes a correction for low sample size. The equation is a bit more complicated, and is more accurate with small sample size. With larger sample sizes, the AIC and AICc are almost the same.

Note that these calculations are based on information theory, and do not use the traditional “hypothesis testing” statistical paradigm. Therefore there is no P value, no conclusion about “statistical significance”, and no “rejection” of a model.

From the difference in AICc values, Prism calculates and reports the probability that each model is correct, with the probabilities summing to

100%. If one model is much more likely to be correct than the other (say, 1% vs. 99%), you will want to choose it. If the difference in likelihood is not very big (say, 40% vs. 60%), you will know that either model might be correct, so will want to collect more data. These probabilities are computed with this equation, where Δ is the difference between the AICc values.

$$probability = \frac{e^{0.5\Delta}}{1+e^{0.5\Delta}}$$

Note that this method simply compares the fits of the two models you chose. It is possible that a third model, one you didn't choose, fits far better than either model you chose.

3.6.4.6 Much of statistics can be viewed as comparing models

Finding the mean by fitting a model

You already know how to find the mean of a bunch of numbers: Add them up, and divide the total by the sample size. That is a convenient shortcut method, but you can also do it via a much harder route!

You can also find the mean by fitting a model. One way to think about this is to fit a linear regression model, but with the slope constrained to equal 0 so only the intercept is fit. When you fit such a model, the best-fit value of the Y intercept is the mean of your data. The mean is simply the best-fit parameter value from a simple model fit to your data.

Comparing two means by fitting a model

The usual way to compare two means is to perform an unpaired t test.

You can also compare to means by comparing two linear regression models. In both models, constrain the slope to equal zero. With a slope of zero, you are fitting a horizontal line.models.

- The first model constrains (shares) the intercepts to be the same for both groups. You are fitting one horizontal line through all the values. The intercept of this line is the grand mean of all the values.

- The second model doesn't constrain the intercepts so finds individual intercepts for each group. In other words, it fits one horizontal line to the data from one group, and another line to data from the other group. Each intercept is the mean of the respective group of data.

The P value from this comparison will be the same as the P value from an unpaired t test. The P value answers this question:

> If the first model is really correct, what is the chance that just by coincidence the data will fit the other model as much better as observed?

Interpreting P values via comparing models

In almost all cases, you can understand a P value as the answer to a question in comparing models. In a clinical study analyzed by logistic regression, the key question is whether the treatment made a difference. Fit one logistic regression model where treatment is ignored, and another model where the treatment is included as one of the parameters in the model. If the P value is small, you can conclude that the treatment mattered. Does it work equally in men and women? Fit one model where gender is one of the parameters and compare to a model that omits gender. If the P value is small, you can conclude that the results differed in men and women.

3.6.5 Outlier elimination and robust nonlinear regression

Nonlinear regression, like linear regression, assumes that the scatter of data around the ideal curve follows a Gaussian or normal distribution. Outliers can violate this assumption and invalidate nonlinear regression results. To deal with outliers, Prism offers robust regression and automatic outlier removal.

- What are the advantages of removing outliers. But isn't it cheating? 88
- Outlier removal has to be used carefully. It must be avoided 89 in many situations.
- Once you've identified an outlier, beware or removing it from graphs and analyses. An outlier is not always a 'bad' point. 91

- Prism uses an outlier removal method created by GraphPad Software --the ROUT method. Learn how it works. 92

- When identifying outliers, the step is to fit a curve using a robust method designed so outliers won't affect the curve much. Learn about robust nonlinear regression. 94

3.6.5.1 When to use automatic outlier removal

The problem with outliers

Nonlinear regression, like linear regression, assumes that the scatter of data around the ideal curve follows a Gaussian or normal distribution. This assumption leads to the familiar goal of regression: to minimize the sum of the squares of the vertical or Y-value distances between the points and the curve. However, experimental mistakes can lead to erroneous values – outliers. Even a single outlier can dominate the sum-of-the-squares calculation, and lead to misleading results.

Is it 'cheating' to remove outliers?

Some people feel that removing outliers is ‘cheating’. It can be viewed that way when outliers are removed in an *ad hoc* manner, especially when you remove only outliers that get in the way of obtaining results you like. But leaving outliers in the data you analyze is also ‘cheating’, as it can lead to invalid results.

Here is a Bayesian way to think about systematic approaches to removing outliers. When a value is flagged as an outlier, there are two possibilities.

- A coincidence occurred, the kind of coincidence that happens in few percent of experiments even if the entire scatter is Gaussian (depending on how aggressively you define an outlier).

- A ‘bad’ point got included in your data.

Which possibility is more likely?

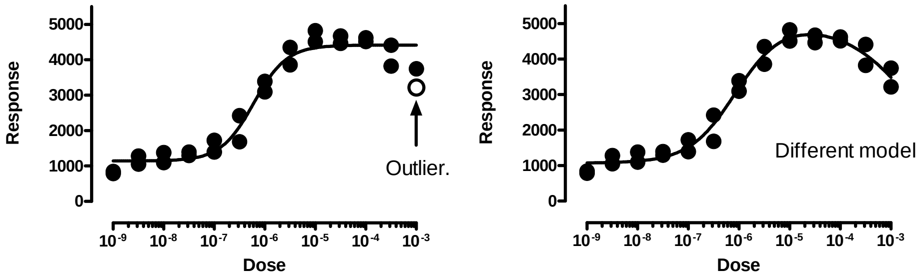
It depends on your experimental system.

If your experimental system generates a 'bad' point in a few percent of experiments, then it makes sense to eliminate the point as an outlier. It is more likely to be a 'bad' point than a 'good' point that just happened to be far from the curve.

If your system is very pure and controlled, so 'bad' points occur very rarely, then it is more likely that the point is far from the curve due to chance (and not mistake) and you should leave it in. Alternatively in that case, you could set Q to a lower value in order to only detect outliers that are much further away.

3.6.5.2 When to avoid automatic outlier removal

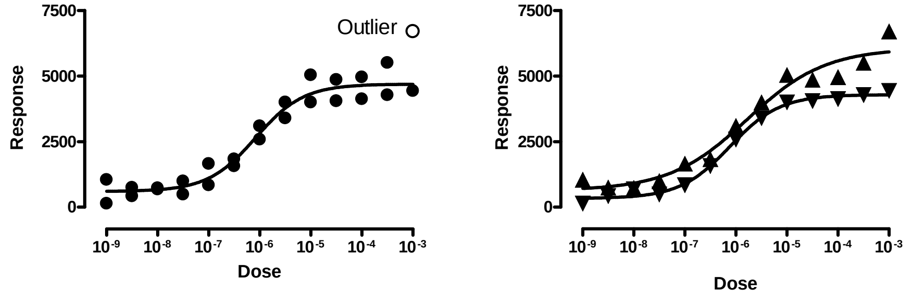
Outlier elimination is misleading when you are fitting the wrong model



The left panel above shows the data fit to a dose response curve 473. In this figure, one of the points is a significant outlier. But this interpretation assumes that you've chosen the correct model. The right panel shows the data fit to an alternative bell-shaped dose-response model 534, where high doses evoke a smaller response than does a moderate dose. The data fit this model very well, with no outliers detected (or even suspected).

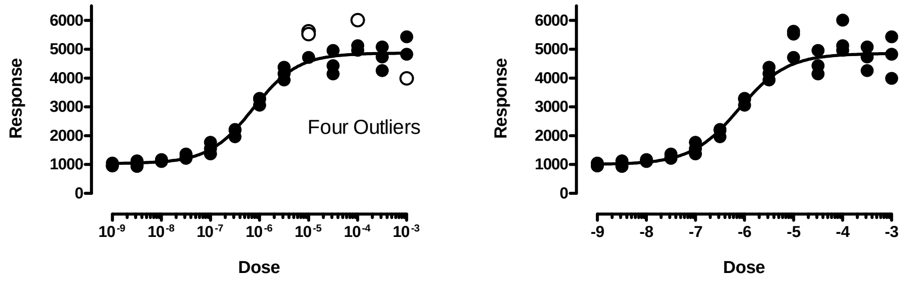
This example points out that outlier elimination is only appropriate when you are sure that you are fitting the correct model.

Outlier elimination is misleading when data points are not independent



The left panel above show data fit to a dose-response model with one point (in the upper right) detected as an outlier. The right panel shows that the data really come from two different experiments. Both the lower and upper plateaus of the second experiment (shown with upward pointing triangles) are higher than those in the first experiment (downward pointing triangles). Because these are two different experiments, the assumption of independence was violated in the analysis in the left panel. When we fit each experimental run separately, no outliers are detected.

Outlier elimination is misleading when you chose incorrect weighting factors



The left panel above shows data fit to a dose-response model. Four outliers were identified (two are almost superimposed). But note that the values with larger responses (Y values) also, on average, are further from the curve. This makes least-squares regression inappropriate. To account for the fact that the SD of the residuals is proportional to the height of the curve, we need to use weighted regression 57. The right panel shows the same data fit to the same dose-response model, but minimizing sum of the squares of the distance of the point from the curve divided by the height of the curve, using relative weighting. Now no outliers are identified. Using the wrong weighting method created false outliers.

3.6.5.3 Outliers aren't always 'bad' points

Definition of an 'outlier'

The term 'outlier' is defined fairly vaguely, but refers to a value that is far from the others. In Prism's nonlinear regression, an outlier is a point that is far from the best-fit curve defined by robust regression.

Of course, there is some possibility that an outlier really comes from the same Gaussian population as the others, and just happens to be very high or low. You can set the value of Q to control how aggressively Prism defines outliers.

Outliers are not always due to mistakes

Nonlinear regression is usually used with experimental data, where X is a variable like time or concentration or some other variable you manipulate in the experiment. Since all the scatter is due to experimental error, it can make sense to eliminate any extreme outlier since it is almost certainly the result of an experimental mistake.

In other situations, each data point can represent a different individual. In this case, an outlier may not be due to experimental mistakes, but rather be the result of biological variation, or differences in some other variable that is not included in your model. Here, the presence of the outlier may be the most interesting finding in the study. While the ROUT outlier method might prove useful to flag an outlier in this situation, it would be a big mistake to automatically exclude such outliers without further thought (or experimentation).

In quality control analyses, an outlier can tell you about a process that is out of control. You wouldn't want to delete outliers, without first figuring out why the value is far from the others. The outlier might be telling you something important.

3.6.5.4 The ROUT method of identifying outliers

How the ROUT method of removing outliers works

Prism offers a unique approach to identifying and removing outliers, detailed in reference 1.Because this method combines **Ro**bust regression and **Out**lier removal, we call it the **ROUT** method.

The ROUT method of regression follows these steps.

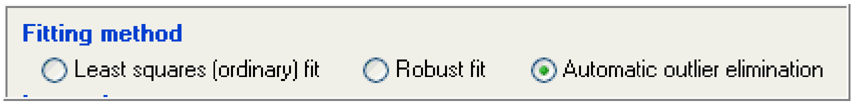
1. Our robust nonlinear regression 94 method is used to fit a curve that is not influenced by outliers.

2. The residuals of the robust fit are analyzed to identify any outliers. This step uses a new outlier test adapted from the False Discovery Rate approach of testing for multiple comparisons.

3. Remove the outliers, and perform ordinary least-squares regression on the remaining data.

How to use the ROUT method with Prism

Although the ROUT method requires three steps (listed above), Prism does all this automatically. All you have to do is check an option on the Fit tab of nonlinear regression dialog:



Fitting method

○ Least squares (ordinary) fit ○ Robust fit ● Automatic outlier elimination

Prism then identifies the outliers, eliminates them, and fits the remaining points. The outliers are shown in a separate table, and the number of outliers is tabulated on the main results table.

The ROUT coefficient Q

The value of Q determines how aggressively the ROUT method defines outliers. The mathematical details are explained in reference 1. This value is set in the Method tab 327 of the Nonlinear regression dialog.

If you set Q to a higher value, the threshold for defining outliers is less strict. This means that Prism will have more power to detect outliers, but also will falsely detect 'outliers' more often.

If you set Q to a lower value, the threshold for defining outliers is stricter. This means that Prism will have a less power to detect real outliers, but also have a smaller chance of falsely defining a point to be an outlier.

Unless you have a strong reason to choose otherwise, we recommend sticking with the default value of 1%. Our simulations have shown that if all the scatter is Gaussian, Prism will falsely find one or more outliers in about 2-3% of experiments. This does not mean that a few percent of all values are declared to be outliers, but rather that one or more outliers will be detected in a few percent of experiments. If there really are outliers present in the data, Prism will detect them with a False Discovery Rate less than 1%.

Unequal weighting, robust regression and outlier removal

As we explain in reference 1, unequal weighting is not useful with robust regression. The problem is that outliers can get way too much weight. Robust regression, essentially, applies differential weights so it doesn't make sense to also add another weigting scheme.

If you choose both unequal weighting and robust fitting, therefore, Prism does the fitting assuming equal weights. However it does use your weighting choice when creating a table of residuals.

If you choose both unequal weighting and automatic outlier removal, Prism first fits using robust regression (ignoring your weighting choice). It does use the weighting factors when identifying the outliers, as explained in reference 1. It then performs weighted nonlinear regression on the outlier-depleted data.

Reference

1. Motulsky HM and Brown RE, Detecting outliers when fitting data with nonlinear regression – a new method based on robust nonlinear regression and the false discovery rate, BMC Bioinformatics 2006, 7:123.

3.6.5.5 Robust nonlinear regression

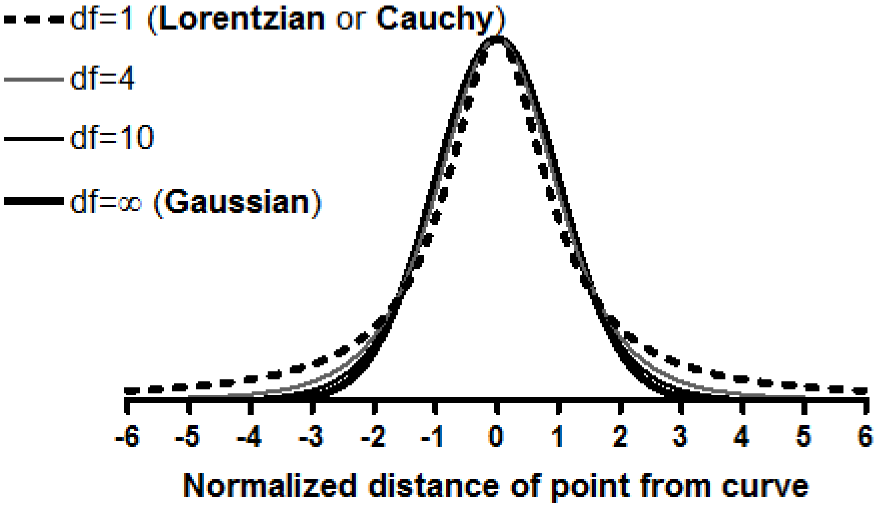
The need for robust regression

Nonlinear regression, like linear regression, assumes that the scatter of data around the ideal curve follows a Gaussian or normal distribution. This assumption leads to the familiar goal of regression: to minimize the sum of the squares of the vertical or Y-value distances between the points and the curve. This standard method for performing nonlinear (or linear regression) is called **least-squares**.

Experimental mistakes can lead to erroneous values whose values are way too high or too low– outliers. Even a single outlier can dominate the sum-of-the-squares calculation, and lead to misleading results. One way to cope with this problem is to perform a **robust fit** using a method that is not very sensitive to violations of the Gaussian assumption. Another approach is to use automatic outlier elimination 88 to identify and remove the outliers, and then run least-squares regression. Prism offers both choices. The outlier identification method actually first does a robust fit, so it has a baseline from which to decide when a point is too far from that baseline so becomes defined as an outlier. Then after removing those outliers, it it performs a standard least-squares fit on the remaining points.

How robust regression works

Based on a suggestion in Numerical Recipes (1), we based our robust fitting method on the assumption that variation around the curve follows a Lorentzian distribution, rather than a Gaussian distribution. Both distributions are part of a family of t distributions:



The widest distribution in that figure, the t distribution for df=1, is also known as the *Lorentzian* distribution or *Cauchy* distribution. The Lorentzian distribution has wide tails, so outliers are fairly common and therefore have little impact on the fit.

We adapted the Marquardt 97 nonlinear regression algorithm to accommodate the assumption of a Lorentzian (rather than Gaussian) distribution of residuals, and explain the details in reference 2.

When does it make sense to choose robust nonlinear regression?

The main use of robust regression in Prism is as a 'baseline' from which to remove outliers 88. You may want to experiment with robust regression in order to better understand the outlier-removal method (which begins with robust regression). You may also find it useful if your only goal is to interpolate from a standard curve, and that standard curve has one or more outliers.

We recommend that you avoid robust regression (at least as implemented by Prism) for most uses, because it has these drawbacks:

- Robust regression cannot calculate standard errors or confidence intervals for the parameters.
- Robust regression cannot plot confidence or prediction bands.

- Robust regression cannot compute the degree to which parameters are intertwined. It cannot compute the covariance matrix or the dependency values.

- Robust regression cannot compare the fits of two models or two data sets.

The inability to compare models does not matter in most contexts. But the inability to provide diagnostic information (SE, CI, Covariance matrix, Dependency, Confidence and prediction bands) seriously limits the usefulness of robust regression for most research.

References

1. Press WH, Teukolsky SA, Vettering WT, Flannery BP: *Numerical Recipes in C. the Art of Scientific Computing.* New York, NY: Cambridge University Press; 1988.
2. Motulsky HM and Brown RE, Detecting outliers when fitting data with nonlinear regression – a new method based on robust nonlinear regression and the false discovery rate, BMC Bioinformatics 2006, 7:123. Download as pdf.

3.6.6 How nonlinear regression works

If you want to really understand how nonlinear regression works, you will need to learn matrix algebra and study advanced math books. But you can use nonlinear regression effectively so long as you understand the basic principles of regression.

- Why minimize the sum-of-squares? 97
- How nonlinear regression works 97
- The need for unequal weighting in nonlinear regression 57
- How standard errors and confidence intervals are computed 103
- How confidence and prediction bands are computed 108

- Replicates 109
- How dependency is calculated 111

3.6.6.1 Why minimize the sum-of-squares?

The goal of nonlinear regression is to adjust the values of the model's parameters to find the curve that best predicts Y from X. More precisely, the goal of regression is to minimize the sum of the squares of the vertical distances of the points from the curve.

Why minimize the sum of the squares of the distances? Why not simply minimize the sum of the actual distances?

If the random scatter follows a Gaussian distribution, it is far more likely to have two medium size deviations (say 5 units each) than to have one small deviation (1 unit) and one large (9 units). A procedure that minimized the sum of the absolute value of the distances would have no preference over a curve that was 5 units away from two points and one that was 1 unit away from one point and 9 units from another. The sum of the distances (more precisely, the sum of the absolute value of the distances) is 10 units in each case. A procedure that minimizes the sum of the squares of the distances prefers to be 5 units away from two points (sum-of-squares = 50) rather than 1 unit away from one point and 9 units away from another (sum-of-squares = 82). If the scatter is Gaussian (or nearly so), the curve determined by minimizing the sum-of-squares is most likely to be correct.

3.6.6.2 How nonlinear regression works

The basic idea of nonlinear regression

You won't be able to understand the mathematical details of nonlinear regression unless you first master matrix algebra. But the basic idea is pretty easy to understand. Every nonlinear regression method follows these steps:

1. Start with initial estimated values for each parameter in the equation.

2. Generate the curve defined by the initial values. Calculate the sum-of-squares -- the sum of the squares of the vertical distances of the points from the curve. (Or compute the weighted sum-of-squares if you are including weighting factors.)

3. Adjust the parameters to make the curve come closer to the data points -- to reduce the sum-of-squares. There are several algorithms for adjusting the parameters, as explained below.

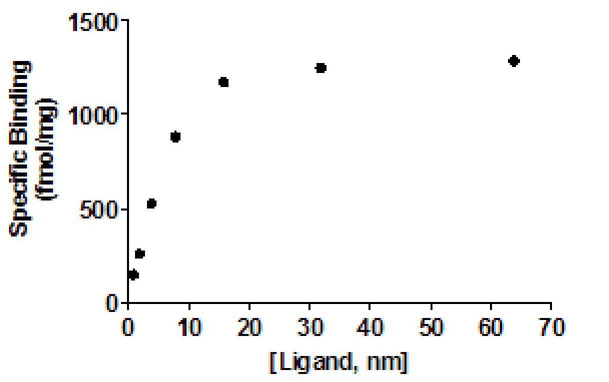
4. Adjust the parameters again so that the curve comes even closer to the points. Repeat.

5. Stop the calculations when the adjustments make virtually no difference in the sum-of-squares.

6. Report the best-fit results. The precise values you obtain will depend in part on the initial values chosen in step 1 and the stopping criteria of step 5. This means that repeat analyses of the same data will not always give exactly the same results.

The Marquardt method

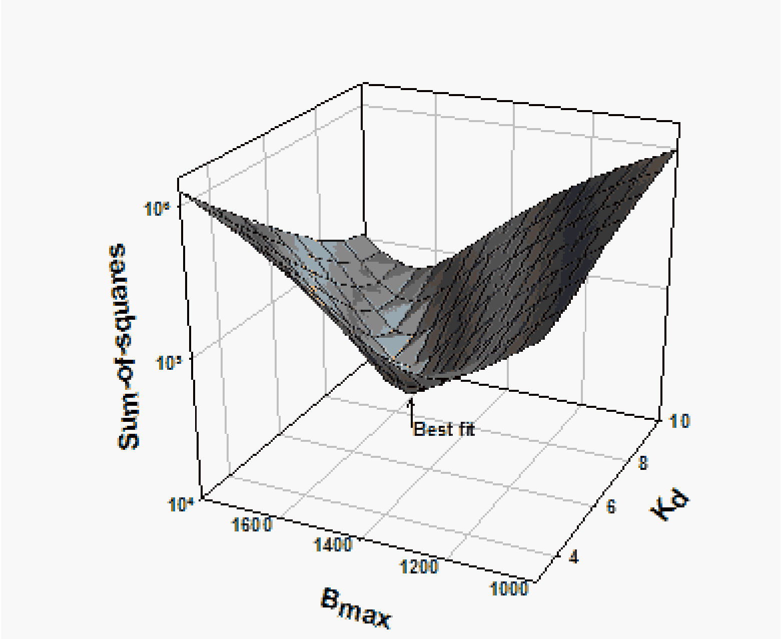
Step 3 is the only difficult one. Prism (and most other nonlinear regression programs) uses the method of Marquardt and Levenberg, which blends two other methods, the method of linear descent and the method of Gauss-Newton. The best way to understand these methods is to follow an example. Here are some data to be fit to a typical binding curve (rectangular hyperbola).



You want to fit a binding curve to determine Bmax and Kd using the equation

$$Y = \frac{B_{max} \cdot X}{K_d + X}$$

How can you find the values of Bmax and Kd that fit the data best? You can generate an infinite number of curves by varying Bmax and Kd. For each of the generated curves, you can compute the sum-of-squares to assess how well that curve fits the data. The following graph illustrates the situation.



The X- and Y-axes correspond to two parameters to be fit by nonlinear regression (Bmax and Kd in this example). The Z-axis is the sum-of-squares. Each point on the surface corresponds to one possible curve. The goal of nonlinear regression is to find the values of Bmax and Kd that make the sum-of-squares as small as possible (to find the bottom of the valley).

The method of linear descent follows a very simple strategy. Starting from the initial values, try increasing each parameter a small amount. If the sum-of-squares goes down, continue. If the sum-of-squares goes up, go back and decrease the value of the parameter instead. You've taken a step down the surface. Repeat many times. Each step will usually reduce the sum-of-squares. If the sum-of-squares goes up instead, the step must have been so large that you went past the bottom and back up the other side. If this happens, go back and take a smaller step. After repeating these steps many times, you will reach the bottom.

The Gauss-Newton method is a bit harder to understand. As with the method of linear descent, start by computing how much the sum-of-squares changes when you make a small change in the value of each parameter. This tells you the slope of the sum-of-squares surface at the point defined by the initial values. If the equation really is linear, this is enough information to determine the shape of the entire sum-of-squares surface, and thus calculate the best-fit values of Bmax and Kd in one step. With a linear equation, knowing the slope at one point tells you everything you need to know about the surface, and you can find the minimum in one step. With nonlinear equations, the Gauss-Newton method won't find the best-fit values in one step, but that step usually improves the fit. After repeating many iterations, you reach the bottom.

This method of linear descent tends to work well for early iterations, but works slowly when it gets close to the best-fit values (and the surface is nearly flat). In contrast, the Gauss-Newton method tends to work badly in early iterations, but works very well in later iterations. The two methods are blended in the method of Marquardt (also called the Levenberg-Marquardt method). It uses the method of linear descent in early iterations and then gradually switches to the Gauss-Newton approach.

Prism, like most programs, uses the Marquardt method for performing nonlinear regression. The method is pretty standard. The only variations are what value to use for lambda (which determines step size) and how to change lambda with successive iterations. We follow the recommendations of Numerical Recipes. Lambda is initialized to 0.001. It is decreased by a factor of 10 after a successful iteration and increased by a factor of 10 after an unsuccessful iteration.

References

Chapter 15 of *Numerical Recipes in C*, Second Edition, WH Press, et. Al. , Cambridge Press, 1992

Chapter 10 of *Primer of Applied Regression and Analysis of Variance* by SA Glantz and BK Slinker, McGraw-Hill, 1990.

3.6.6.3 Nonlinear regression with unequal weights

How Prism implements weighting

Prism 5 and later weight by the Y value of the curve. Prism 4 and earlier weighted by the Y value of the data. The distinction is subtle and rarely matters much, but our simulations show that the results are sometimes more accurate when weights are based on the value of the curve rather than the data.

The situation is a bit tricky. The goal is to adjust the values of the parameters to minimize the weighted sum-of-squares. But the values of the weights depend on the values of those parameters. Here is how Prism resolves this issue:

1. Perform the first iteration without any weights. If the initial parameter values are not very accurate, weighted regression can cause problems on the first iteration. This step was introduced in Prism 5.02 and 5.0b. Without it, some fits with bad initial values simply won't converge.

2. Compute Ycurve at all values of X based on the curve generated by this first iteration of nonlinear regression, and use those values to compute weights for each points.

3. Keeping those weights constant, use nonlinear regression to minimize the weighted sum-of-squares.

4. Recompute Ycurve based on the results of that curve fit and generate a new set of weights.

5. Keeping those new weights constant, use nonlinear regression again to minimize the weighted sum-of-squares.

6. Repeat steps 4 and 5 until the weighted sum-of-squares no longer changes.

If you enter data as mean, n and SD or SEM, Prism obtains the same results as if you had entered raw data, even with unequal weighting. This document explains how. The only exception is robust regression, which only "sees" the means and ignores the n and SD or SEM values you entered.

Weighting with robust regression or automatic outlier removal

As we explain in reference 1, it doesn't make sense to perform robust regression using unequal weights. The problem is that outliers can get too much weight.

If you choose both unequal weighting and robust fitting, therefore, Prism does the fitting assuming equal weights. However it uses your weighting choice when creating a table of residuals, and when counting the number of outliers (a choice you can make in the preferences tab).

If you choose both unequal weighting and automatic outlier removal, Prism first fits using robust regression (ignoring your weighting choice), and then uses the weighting factors in identifying the outliers, as explained in reference 1.

Reference

1. Motulsky HM and Brown RE, Detecting outliers when fitting data with nonlinear regression – a new method based on robust nonlinear regression and the false discovery rate, BMC Bioinformatics 2006, 7:123. Download as pdf.

3.6.6.4 Regression on data entered as mean, n and SD or SEM

If your data has replicate Y values for each X value, there are two ways to enter the data into Prism. You can enter the raw replicates. Or you can enter mean, n and SD or SEM. How do regression results differ?

- With ordinary nonlinear regression, Prism gives exactly the same results as if you had entered raw data, even with weighted regression. This document explains the mathematical details about how Prism does it.

- If you choose robust nonlinear regression, Prism fits only the means and ignores the n and SD or SEM values you entered.

- If you choose outlier detection, this uses only the information from the means (ignoring n and SD or SEM). But once any outliers are identified and removed (if you chose that), the fitting proceeds as usual and does account for n and SD or SEM.

- If you want Prism to only fit the means with nonlinear regression, this is a choice on the Weights tab. It is

- It is better to enter the raw data so you can plot the raw data to look for outliers or problems. You can then change the graph to plot mean and SD to avoid making the graph too busy.

- If you want to see the SD (or SEM) at each X value, Prism has an analysis for that (Row Means...).

3.6.6.5 How standard errors are computed

Standard error

Prism uses a standard method to compute the standard error and confidence interval for each parameter fit with nonlinear regression.

Each parameter's standard error is computed using this equation:

SE(Pi) = sqrt[(SS/DF) * Cov(i,i)]

where:

Pi : i-th adjustable(non-constant) parameter
SS : sum of squared residuals
DF : degrees of freedom (the number of data points minus number of parameters fit by regression)
Cov(i,i) : i-th diagonal element of covariance matrix
sqrt() : square root

3.6.6.6 How symmetrical asymptotic CIs are calculated

Asymptotic confidence intervals are always centered on the best-fit value of the parameter, and extend the same distance above and below that value.

The 95% confidence intervals are computed by this equation:

From [BestFit- t*SE] TO [BestFit+ t*SE]

where BestFit is the best fit value for the parameter, SE is its standard error 103, and t is the value from the t distribution for the desired level of confidence (95% is standard) and the number of degrees of freedom (which equals the number of data points minus the number of parameters fit by regression). With 95% confidence and many degrees of freedom (more than a few dozen), this multiplier is very close to 1.96. Note that the value of t is not computed from your data, but is a constant that depends on the confidence level you choose, the number of data points, and the number of parameters.

3.6.6.7 How profile likelihood asymmetrical confidence intervals are computed

The basic idea of profile likelihood asymmetrical confidence intervals

Before Prism 7, Prism only reported asymptotic confidence intervals that are always symmetrical around the best-fit values.

For some parameters in some models, an asymmetrical interval does a much better job of expressing precision. Prism (starting with version 7) offers this with *profile likelihood confidence intervals,* as a choice in the Confidence tab of the nonlinear regression dialog 343. The disadvantages are that they are unfamiliar to many and that they take longer to compute (but with a fast computer, you may not even notice unless you have huge data sets and choose a user-defined equation).

The idea is pretty simple. The extra sum-of-squares test 82 compares two models.

- The more complicated model is the model you chose. The entire page of results is for this model.

- The simpler model fixes one parameter to a constant value. The idea is to fix that parameter to various values until you find the confidence limit (as explained below).

Here is a very simplified algorithm that sort of explains the idea behind the method. Define the sum-of-squares to be SS and the degrees of freedom to be DF.

1. Set a variable Delta to the SE of the parameter you are finding the CI for (then repeat for the other parameters).

2. Hold the parameter fixed to its best-fit value *minus* delta, and run the fit again letting all the other parameter values vary. Record the new SS and DF of this fit.

3. Compare the original best-fit with this fit that forces the parameter to be decreased by delta using the extra sum-of-squares F test 82. The second fit holds one parameter to a constant value so fits one fewer parameter as so has one more degree of freedom. Compute the P value.

    a. If the P value is less than 0.05, delta is too large. Make it smaller and go back to step 2.

    b. If the P value is greater than 0.05, delta is too small. Make it larger and go back to step 2.

    c. If the P value is very close to 0.05, then the lower confidence limit equals the original best fit value minus the current value of delta.

4. Holding your parameter fixed to its best-fit value *plus* delta, run the fit again letting all the other parameter values vary. Record the SS and DF of this fit.

5. Compare the original best-fit with this fit that forces the parameter to be *increased* by delta using the extra sum-of-squares F test. Compute the P value.

    a. If the P value is less than 0.05, delta is too large. Make it smaller and go back to step 5.

    b. If the P value is greater than 0.05, delta is too small. Make it larger and go back to step 5.

    c. If the P value is very close to 0.05, then the upper confidence limit equals the original best fit value plus the current value of delta.

6. Repeat for each parameter.

This creates a 100*(1-ɑ)% confidence interval for a parameter (95% interval for the common situation where ɑ is set to 0.05). If you were to test the null hypothesis that the true parameter value is equal to its best-fit value, that null hypothesis would not be rejected for any value of the parameter with the confidence interval.

More formally: Define θbf the best fit value of the parameter and θhyp a hypothetical different value for the parameter. The null hypothesis that θbf = θhyp will not be rejected at the ɑ level of significance for any value of θhyp within the confidence interval but will be rejected for any value of θhyp outside the confidence interval.

How Prism computes profile likelihood confidence intervals

Prism actually uses the steps detailed by Venzon and Moolgavkar(1) for each parameter. This method creates a likelihood profile for each parameter. For various possible values of the parameter, the algorithm fits the curve (optimizing the other parameters) and determines the likelihood that the data would have come from this model. The confidence interval is the range of parameter values where the likelihood is not too much lower than its maximum value. Of course, "too low" is defined rigorously.

The maximum likelihood is at the best-fit value of the parameter. When these profiles are graphed in texts, it is usually the negative logarithm of the likelihood that is plotted. The maximum likelihood is the same as the minimum -log(likelihood) so in these graphs the best fit value is the X value where Y is at its lowest value.

If you assume all residuals follow a Gaussian distribution, maximizing the likelihood is the same as minimizing the sum-of-squares.

Notes

- The final delta value for computing the upper confidence limit may not be equal to (or even be close to) the final value of delta for computing the lower limit. That's why the confidence interval may be asymmetrical around the best-fit value.

- The P value goal of 0.05 above is used only when you want 95% confidence intervals. If you want 99% confidence intervals. use 0.01, etc.

- The method in reference 1 (that Prism uses) is way more clever than described above so takes fewer computations.

- The confidence intervals computed this way are for just that one parameter. The ideas is that each confidence interval has a 95% chance of including the true parameter value. The 95% does not apply to the set of intervals. It is not correct to say that we expect there to be a 95% chance that *all* the confidence intervals include the respective true parameter values.

- When computing the extra sum of squares F test above, note that the two models differ by one degree of freedom. This is because we are fixing one parameter and letting Prism fit the others. Some publications (2) assume that you are fixing all the parameters, not just one. So the two models entered into the F test differ by K degrees of freedom, where K is the number of parameters fit. These intervals are wider, and I think the intent is that the 95% confidence level apply simultaneously to all the intervals, rather than to just one. Prism does not use this approach. With prism, the two models being compared always differ by one degree of freedom.

- The method we use is also described by Watts(3). Prism matches the results he presents in Table IV for data in Table III.

- In some cases, the method is unable to find one of the confidence limits and reports "???" instead of a value.

Reference

1. Venzon DJ, Moolgavkar SH. A method for computing profile-likelihood-based confidence intervals. Applied Statistics. 1988;37(1):87.

2. Kemmer, G., & Keller, S. (2010). Nonlinear least-squares data fitting in Excel spreadsheets. Nature Protocols, 5(2), 267–281. http://doi.org/10.1038/nprot.2009.182

3. Watts, D.G. (2010) Parameter estimates from nonlinear models, Chapter 2 of Essential *Numerical Computer Methods* by M Johnson, Academic Press 2010.

3.6.6.8 How confidence and prediction bands are computed

The difference between confidence and prediction bands

Note the difference between confidence and prediction bands:

- The 95% confidence bands enclose the area that you can be 95% sure contains the true curve. If you have many data points, the confidence bands will be near the line or curve, and most of your data will lie outside the confidence bands.

- The 95% prediction bands enclose the area that you expect to enclose 95% of future data points. They are wider than confidence bands -- much wider with large data sets.

Also distinguish between the 95% confidence interval of the parameters (a range of values), and the 95% confidence bands around the curve.

How confidence and prediction bands are computed

The calculation of the confidence and prediction bands are fairly standard, and can only be expressed with matrices. A brief explanation follows. More details can be found here.

First, define G|x, which is the gradient of the parameters at a particular value of X and using all the best-fit values of the parameters. The result is a vector, with one element per parameter. For each parameter, it is defined as dY/dP, where Y is the Y value of the curve given the particular value of X and all the best-fit parameter values, and P is one of the parameters.)

G'|x is that gradient vector transposed, so it is a column rather than a row of values.

Cov is the covariance matrix (inverted Hessian from last iteration). It is a square matrix with the number of rows and columns equal to the number of parameters. Each item in the matrix is the covariance between two parameters. Note that this is the actual covariance matrix, which is distinct from the normalized covariance matrix (where each value is between -1 and 1) that Prism can report.

Now compute c = G|x * Cov * G'|x. The result is a single number for any value of X.

The confidence and prediction bands are centered on the best fit curve, and extend above and below the curve an equal amount.

The confidence bands extend above and below the curve by:

= sqrt(c)*sqrt(SS/DF)*CriticalT(Confidence%, DF)

The prediction bands extend a further distance above and below the curve, equal to:

= sqrt(c+1)*sqrt(SS/DF)*CriticalT(Confidence%, DF)

In both these equations, the value of c (defined above) depends on the value of X, so the confidence and prediction bands are not a constant distance from the curve. The value of SS is the sum-of-squares for the fit, and DF is the number of degrees of freedom (number of data points minus number of parameters). CriticalT is a constant from the t distribution based on the amount of confidence you want and the number of degrees of freedom. For 95% limits, and a fairly large df, this value is close to 1.96. If DF is small, this value is higher.

3.6.6.9 Replicates

If you entered data with replicates (in side-by-side subcolumns), Prism gives you two choice for how to deal with the replicates.

- Consider each replicate as an individual point.

- Fit the means of each set of replicates.

The rest of this page explains how to decide. When in doubt, choose to fit individual replicates. The other choice is only rarely useful.

Independent replicates

In most experiments, it is fair to consider each replicate to be an independent data point. Each particular replicate is subject to random factors, which may increase or decrease its value. Each random factor affects individual replicates, and no random factor affects the replicates as a group. In any kind of biochemical experiment, where each value comes from a test tube or plate well, the replicates are almost certain to be independent.

When your replicates are independent, Prism will treat each replicate as a separate point. If there are four replicates at one X value and two at another, the four replicates will automatically get twice the weight, since the program considers them to be four separate data points.

If you ask Prism to fit the mean values, rather than individual replicates, you won't get valid standard errors and confidence intervals. If you have different number of replicates at different X values, you will lose the extra weights that the points with more replicates deserve, so will get incorrect best-fit values.

Replicates that are not independent

In some experimental situations, the replicates are not independent. Random factors can affect all the replicates at once. Two examples:

- You performed a binding experiment with a single tube at each concentration, but measured the radioactivity in each tube three times. Those three values are not independent. Any experimental error while conducting the experiment would affect all the replicates.

- You performed a dose-response experiment, using a different animal at each dose with triplicate measurements. The three measurements are not independent. If one animal happens to respond more than the others, that will affect all the replicates. The replicates are not independent.

Treating each replicate as a separate data point would not be appropriate in these situations. Most of the random variation is between tubes (first example) or animals (second example). Collecting multiple replicates does not give you much additional information. Certainly, each replicate does not give independent information about the values of the parameters. Here is one way to look at this. Imagine that you have performed a dose-response experiment with a separate animal for each dose. You measure one animal in duplicate (for one dose) and another animal (another dose) ten times. It would be a mistake to enter those as individual values, because that would give five times more weight to the second dose compared to the first. The random factors tend to affect the animal, not the measurement, so measuring an animal ten times does not give you five times more information about the true value than measuring it two times.

Since each tube (first example, above) or animal (second example) is the experimental unit, you should enter each tube or animal once. If you measured several replicates, average these and enter the average. Don’t enter individual values. Don’t weight the means by sample size. Doing so would inflate the number of degrees of freedom inappropriately, and give you SE that are too small and CI that are too narrow. Doing so, when you have unequal number of replicates would give artificial, and undeserved, weight to the tubes or animals with more replicates, so would affect the best-fit curve and you would get less than optimal best fit parameter values.

If you are confused by the choice of fitting to individual replicates vs. the mean, choose to fit individual replicates (which is the default

3.6.6.10 How dependency is calculated

What is dependency?

When the model has two or more parameters, as is almost always the case, the parameters can be intertwined.

What does it mean for parameters to be intertwined? After fitting a model, change the value of one parameter but leave the others alone. The curve moves away from the points. Now, try to bring the curve back so it is close to the points by changing the other parameter(s). If you can bring the curve closer to the points, the parameters are intertwined. If you can bring the curve back to its original position, then the parameters are redundant.

Prism can quantify the relationships between parameters by reporting the correlation matrix or reporting dependency.

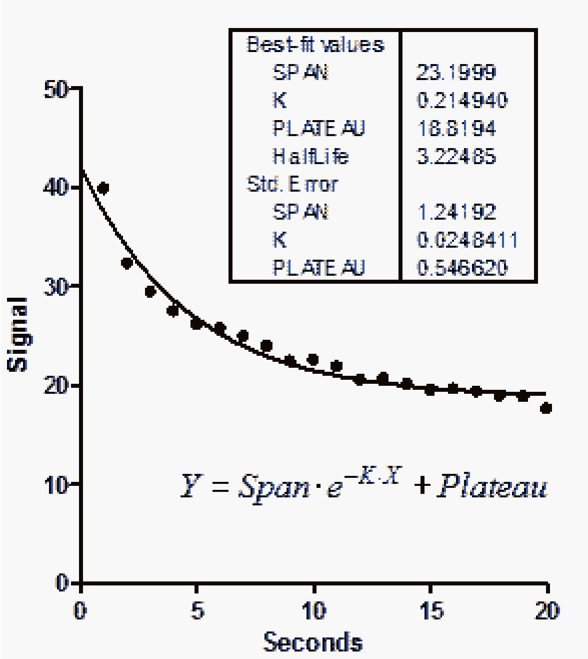
Interpreting dependency

You can interpret dependency 393 without knowing much about how it is calculated. Read on if you are interesting in knowing how the value is computed.

Example of dependency calculations

This example is an exponential decay (taken from pages 128-130 of the MLAB Applications Manual, www.civilized.com.

Time	Signal
1.0	39.814
2.0	32.269
3.0	29.431
4.0	27.481
5.0	26.086
6.0	25.757
7.0	24.932
8.0	23.928
9.0	22.415
10.0	22.548
11.0	21.900
12.0	20.527
13.0	20.695
14.0	20.105
15.0	19.516
16.0	19.640
17.0	19.346
18.0	18.927
19.0	18.857
20.0	17.652



Best-fit values	
SPAN	23.1999
K	0.214940
PLATEAU	18.8194
HalfLife	3.22485
Std. Error	
SPAN	1.24192
K	0.0248411
PLATEAU	0.546620

We will focus on the rate constant, K. The best fit value is 0.2149 sec-1, which corresponds to a half-life of 3.225 seconds. Its SE is 0.0248 sec-1, which corresponds to a 95% confidence interval of 0.1625 to 0.2674 sec^{-1}.

It is clear that the three parameters are not entirely independent. If you forced *K* to have a higher value (faster decay), the curve would get further from the points. But you could compensate a bit by starting the curve at a higher value and ending at a lower one (increase *Span* and decrease *Plateau*). The SE values of the parameters depend on one another.

Fix *Span* and *Plateau* to their best fit values, and ask Prism to fit only the rate constant K. This will not change the best fit value, of course, since we fixed *Span* and *Plateau* to their best-fit values. But the SE of K is lower now, equal to 0.008605. This makes sense. Changing the value of K has a bigger impact on goodness-of-fit (sum-of-squares) when you fix the *Span* and *Plateau* than it does when you allow the values of *Span* and *Plateau* to also change to compensate for the change in *K*.

The lower value of the SE of K when you fix the other parameters tells you that the uncertainty in K is dependent on the other parameters. We want to quantify this by computing the *dependency*.

Before we can compare the two SE values, we have to correct for a minor problem. When computing the SE, the program divides by the square root of the number of degrees of freedom (df). For each fit, df equals the number of data points minus the number of parameters fit by the regression. For the full fit, df therefore equals 20 (number of data points) minus 3 (number of parameters) or 17. When we held the values of Plateau and Span constant, there was only one parameter, so df=19. Because the df are not equal, the two SE values are not quite comparable. The SE when other parameters were fixed is artificially low. This is easy to fix. Multiply the SE reported when two of the parameters were constrained by the square root of 19/17. This corrected SE equals 0.00910.

Now we can compute the dependency. It equals 1.0 minus the square of the ratio of the two (corrected) SE values. So the dependency for this example equals 1.0-(0.0091/0.0248)2, or 0.866. Essentially, this means that 86.6% of the variance in K is due to is interaction with other parameters.

Each parameter has a distinct dependency (unless there are only two parameters). The dependency of Span is 0.613 and the dependency of Plateau is 0.813.

The origin of the idea of dependency

There appears to be no paper to cite regarding the first use of dependency. The idea of dependency apparently was developed by Dick Shrager at the NIH, and then enhanced by Gary Knott. MLAB was the first software to compute dependency, and it is explained well in the MLAB manual. GraphPad Prism simply implements the method as it is explained there. (I learned this history in an email from Gary Knott in 2007). Here is an early paper that discusses the basic ideas of dependency, but it is defined differently (ranging from 1 to infinity, rather than 0 to 1, and the math is not fully explained.

3.6.6.11 How Prism identifies unstable parameters

Starting with Prism 8.2 as a Prism Labs experimental feature, Prism offers the option of identifying unstable parameters instead of ambiguous fits. You can choose this option on the Confidence tab 343 of the nonlinear regression dialog. Here is a brief explanation of how this feature works:

1. After completing the fit, Prism prepares the Hessian matrix. This is a square matrix with one row (and one column) for each parameter fit by the regression. The values in the matrix are the product of the partial derivative of the goodness-of-fit as you change the parameter that defines the row times the corresponding partial derivative for the parameter that defines the column. Goodness of fit is usually the sum-of-squares (or weighted sum-of-squares), but is defined differently for Poisson regression. This matrix defines how sensitive goodness of fit is to changes in the parameters.

2. Prism computes the "condition" of the matrix. This basically quantifies how error-prone it will be to invert the matrix (which is needed as part of curve fitting). The condition is a number. If that number is low, there are no unstable parameters. If the number is high, the matrix is "ill conditioned", which means a small change in the Hessian matrix would result in a large change in the best-fit parameters.

3. Prism concludes there is one or more unstable parameter when the condition is greater than 10/sqrt(epsilon), where epsilon is the 64 bit machine epsilon, 2.22e-16, an upper bound on the relative error due to rounding in floating point arithmetic. So if the matrix condition is greater than about 670 million, Prism concludes that one (or more) of the parameters must be unstable. Of course, this is a somewhat arbitrary cutoff. The threshold of 10 divided by the square root of epsilon is not a typo; we find that using 10 (not 1) in the numerator provides a more useful threshold.

4. Prism identifies the parameter that causes the biggest increase in the Condition of the matrix and removes the corresponding row and column from the Hessian matrix and marks that parameter as "unstable". Then it repeats steps 2 and 3 to see if the Hessian matrix is still unstable, in which case another parameter needs to be removed as unstable. Repeat until the condition of the matrix is low enough.

5. In the results table, Prism reports the unstable parameter(s), along with their standard error(s) and confidence interval(s), as "Unstable".

3.6.6.12 Who developed nonlinear regression?

According to wikipedia (Sept. 2011), the algorithm was first published by Kenneth Levenberg, while working at the Frankford Army Arsenal. It was rediscovered by Donald Marquardt who worked as a statistician at DuPont and independently by Girard, Wynn and Morrison.

K Levenberg (1944). A Method for the Solution of Certain Non-Linear Problems in Least Squares. The Quarterly of Applied Mathematics 2: 164–168.

DW Marquardt, An Algorithm for Least-Squares Estimation of Nonlinear Parameters, J. Society for Industrial and Applied Mathematics, 11:431-441, 1963.

4 REGRESSION WITH PRISM 8

Prism offers several analyses that can fit lines and curves.

- Simple linear regression 117
- Simple logistic regression 142
- Deming linear regression 172
- Multiple regression with Prism 176, including
    - Multiple linear regression 177
    - Poisson regression 177
    - Multiple Logistic regression 198
- Nonlinear regression 294 (which also does polynomial regression)
- Spline and Lowess 289

- Interpolating from a standard curve 258 (linear or nonlinear)

4.1 Simple linear and simple logistic regression with Prism

Prism offers two forms of simple regression: simple linear regression and simple logistic regression. Although these analyses are related, we discuss them separately. To learn more about the similarities and differences of simple linear regression and simple logistic regression, read more about them in the “Principles of Regression” portion of this guide (simple linear regression, simple logistic regression).

Simple linear regression fits a straight line through your data to find the best-fit value of the slope and intercept.

Simple logistic regression estimates the probability of obtaining a “positive” outcome (when there are only two possible outcomes, such as “positive/negative”, “success/failure”, or “alive/dead”, etc.).

4.1.1 How to: Simple linear regression

4.1.1.1 Finding the best-fit slope and intercept

1. Create a data table

From the Welcome or New Table dialog, choose to create XY data table.

If you are just getting started, choose the sample data: Linear regression -- Compare slopes.

If you are entering your own data, choose the subcolumn format. Choose replicate values if you have replicates to enter. Prism can plot error bars automatically. You can also choose to enter data where the mean and SD (or SEM) have already been calculated. In this case, if you want to take into account variations in the SD from point to point, use nonlinear regression to fit the line 123.

2. Enter data

If you chose sample data, you'll see these values:

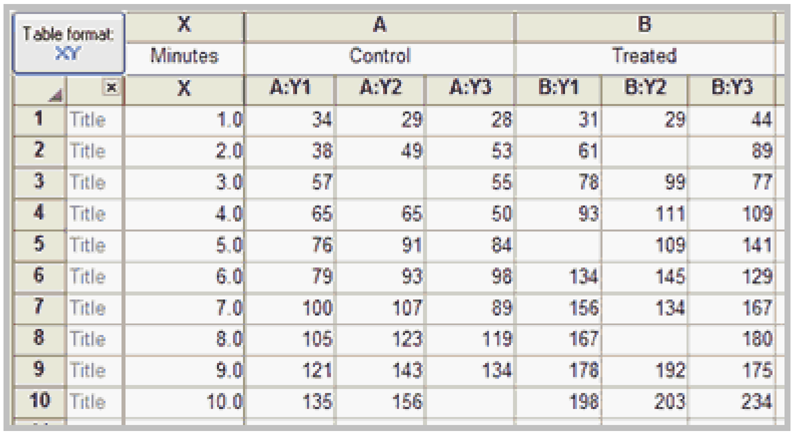
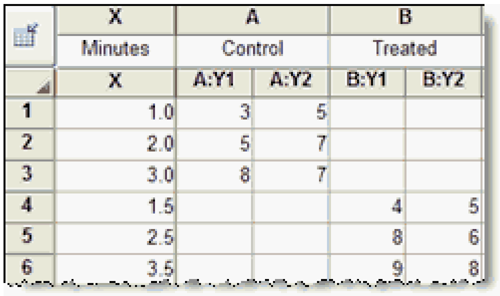


Table format: XY		X	A			B		
		Minutes	Control			Treated		
		X	A:Y1	A:Y2	A:Y3	B:Y1	B:Y2	B:Y3
1	Title	1.0	34	29	28	31	29	44
2	Title	2.0	38	49	53	61		89
3	Title	3.0	57		55	78	99	77
4	Title	4.0	65	65	50	93	111	109
5	Title	5.0	76	91	84		109	141
6	Title	6.0	79	93	98	134	145	129
7	Title	7.0	100	107	89	156	134	167
8	Title	8.0	105	123	119	167		180
9	Title	9.0	121	143	134	178	192	175
10	Title	10.0	135	156		198	203	234

If you enter Y values for several data sets (column A, B and C), Prism will report regression results for X vs. YA, for X vs. YB, and for X vs. YC. It can also test whether the slopes (and intercepts) differ significantly.

If the different data sets don't share the same X values, use different rows for different data sets like this:



	X	A		B	
	Minutes	Control		Treated	
	X	A:Y1	A:Y2	B:Y1	B:Y2
1	1.0	3	5		
2	2.0	5	7		
3	3.0	8	7		
4	1.5			4	5
5	2.5			8	6
6	3.5			9	8

3. Analysis choices

Click Analyze, and then choose linear regression from the list of XY analyses.

Force the line to go through a specified point (such as the origin)?

If you choose regression, you may force the line to go through a particular point such as the origin. In this case, Prism will determine only the best-fit slope, as the intercept will be fixed. Use this option when scientific theory tells you that the line must go through a particular point (usually the origin, X=0, Y=0) and you only want to know the slope. This situation arises rarely.

Use common sense when making your decision. For example, consider a protein assay. You measure optical density (Y) for several known concentrations of protein in order to create a standard curve. You then want to interpolate unknown protein concentrations from that standard curve. When performing the assay, you adjusted the spectrophotometer so that it reads zero with zero protein. Therefore you might be tempted to force the regression line through the origin. But this constraint may result in a line that doesn't fit the data very well. Since you really care that the line fits the standards very well near the unknowns, you will probably get a better fit by not constraining the line.

If in doubt, you should let Prism find the best-fit line without any constraints.

Fit linear regression to individual replicates or to means?

If you collected replicate Y values at every value of X, there are two ways to calculate linear regression. You can treat each replicate as a separate point, or you can average the replicate Y values, to determine the mean Y value at each X, and do the linear regression calculations using the means.

You should consider each replicate a separate point when the sources of experimental error are the same for each data point. If one value happens to be a bit high, there is no reason to expect the other replicates to be high as well. The errors are independent.

Average the replicates and treat the mean as a single value when the replicates are not independent. For example, the replicates would not be independent if they represent triplicate measurements from the same animal, with a different animal used at each value of X (dose). If one animal happens to respond more than the others, that will affect all the replicates. The replicates are not independent.

Test departure from linearity with runs test

See Runs test 140

Test whether slope and intercept are significantly different

If you have entered data for two or more datasets, Prism can test whether the slopes differ significantly 139.

Confidence and prediction bands

Learn about confidence and prediction bands 129.

4.1.1.2 Interpolating from a linear standard curve

1. Create a data table

From the Welcome or New Table dialog, choose to create an XY data table.

If you are just getting started, choose the sample data: Linear regression -- Interpolate from standard curve

If you are entering your own data, choose the subcolumn format. Choose replicate values if you have replicates to enter. Prism can plot error bars automatically. You can also choose to enter data where the mean and SD (or SEM) have already been calculated. In this case, if you want to take into account variations in the SD from point to point, use nonlinear regression to fit the line 123.

2. Enter data

Enter the unknowns below the standards on the same table. Enter Y values with no X values in those rows (example below), or X values with no Y values in those rows. Optionally enter row titles to label those unknowns.

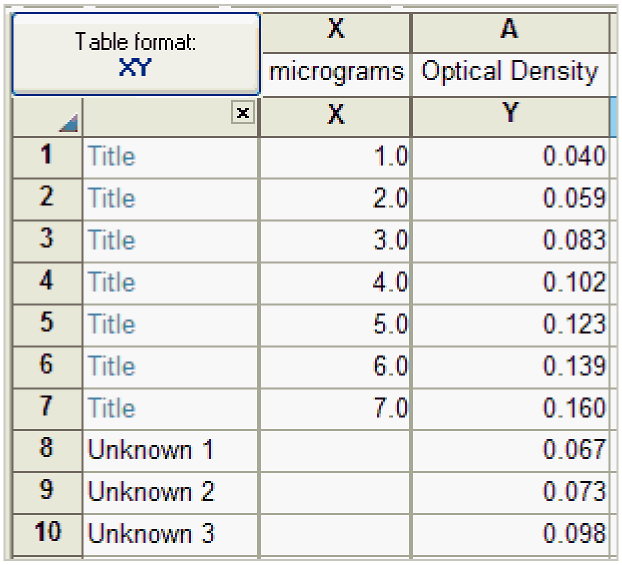
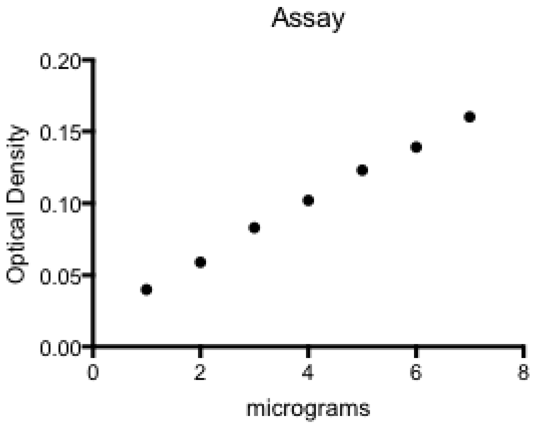


Table format: XY		X	A
		micrograms	Optical Density
		X	Y
1	Title	1.0	0.040
2	Title	2.0	0.059
3	Title	3.0	0.083
4	Title	4.0	0.102
5	Title	5.0	0.123
6	Title	6.0	0.139
7	Title	7.0	0.160
8	Unknown 1		0.067
9	Unknown 2		0.073
10	Unknown 3		0.098

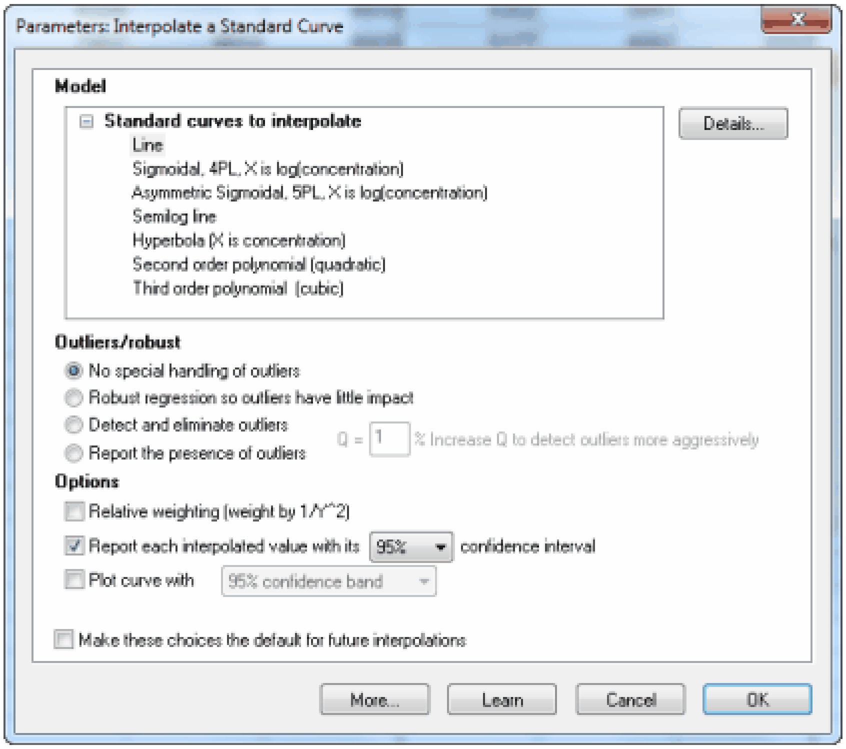
3. View the graph



4. Analysis choices

Click the Analyze button and from the list of XY analyses choose: Interpolate a Standard Curve.

Alternatively, you can click the “Interpolate a standard curve” button right on top of the Analyze button.



Parameters: Interpolate a Standard Curve

Model

⊟ **Standard curves to interpolate**
- Line
- Sigmoidal, 4PL, X is log(concentration)
- Asymmetric Sigmoidal, 5PL, X is log(concentration)
- Semilog line
- Hyperbola (X is concentration)
- Second order polynomial (quadratic)
- Third order polynomial (cubic)

Outliers/robust

- ● No special handling of outliers
- ○ Robust regression so outliers have little impact
- ○ Detect and eliminate outliers
- ○ Report the presence of outliers

Q = 1 % Increase Q to detect outliers more aggressively

Options

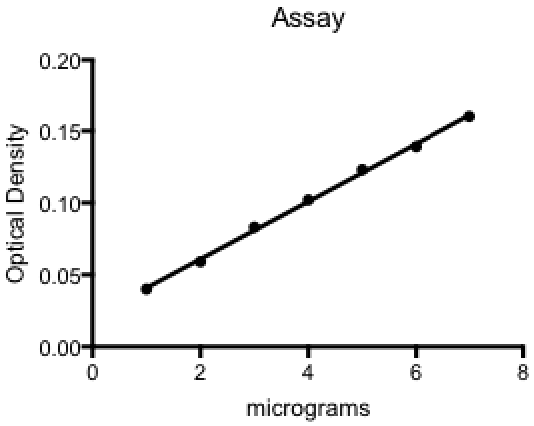
- □ Relative weighting (weight by 1/Y^2)
- ☑ Report each interpolated value with its 95% confidence interval
- □ Plot curve with 95% confidence band

□ Make these choices the default for future interpolations

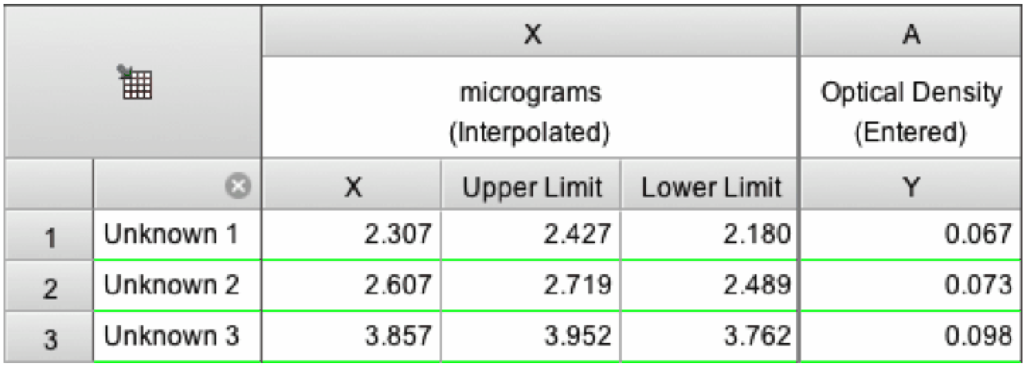
Choose Line in the list of Standard curves to interpolate. Unless you have a good reason, you can leave the other choices on that dialog set to their default values.

5. Inspect the results

View the graph with the superimposed line.



The results will appear in several pages, including one with the interpolated values, which will be in the same units as your original data.



		X			A
		micrograms (Interpolated)			Optical Density (Entered)
		X	Upper Limit	Lower Limit	Y
1	Unknown 1	2.307	2.427	2.180	0.067
2	Unknown 2	2.607	2.719	2.489	0.073
3	Unknown 3	3.857	3.952	3.762	0.098

The second page of results tabulates the best-fit values of the parameters and much more. For this example, we aren't too interested in these results.

4.1.1.3 Advice: When to fit a line with nonlinear regression

Linear regression is a special case of nonlinear regression

Simple linear regression is just a simpler, special, case of nonlinear regression. The calculations are a bit easier (but that only matters to programmers). You can use Prism's nonlinear regression analysis to fit a straight-line model, and the results will be identical to simple linear regression.

How to convert a linear regression analysis to nonlinear regression

To convert simple linear regression to nonlinear regression, bring up the Parameters dialog for linear regression, and click "More Choices" at the bottom.

Nonlinear regression offers more options

Conceptually, simple linear regression is just a special case of nonlinear regression. But Prism offers many more options when using nonlinear regression. Therefore, it often makes sense to use Prism's nonlinear regression analysis to fit a straight line. In particular, the nonlinear regression analysis lets you:

- Fit to both a linear and nonlinear model, and compare the two models 334.

- Apply differential weighting 57.
- Automatically exclude outliers 94.
- Use a robust 94 fitting method.
- Graph the residuals 361 in four ways.
- Perform a normality test 375 on the residuals.
- Inspect the correlation matrix or dependencies 393.
- Compare the scatter of points from the line with the scatter among replicates with a replicates test 390.
- Report the best-fit values with 90% confidence limits (or any others). Prism's linear regression analysis only reports 95% CI. Nonlinear regression lets you choose the confidence level you want.
- Report the results of interpolation 260 from the line/curve along with 95% confidence intervals of the predicted values. Prism's linear regression analysis does not include those confidence intervals.
- With linear regression, the SE of the slope is always reported with the slope as a plus minus value. With nonlinear regression, the SE values are a separate block of results that can be copy and pasted elsewhere.
- Use global nonlinear regression 70 to fit one line to several data sets. Or share the intercept or slope among several data sets, while fitting the other parameter individually to each data set.
- Run a Monte Carlo analysis.
- When you enter data with multiple replicates at each X value, Prism's nonlinear regression can perform the replicates test 390 to ask whether the data deviate systematically from the straight line model. Prism does not offer the replicates test with linear regression.
- Test whether the slope (or intercept) significantly differs from some proposed value. For example, test whether the slope differs from a

hypothetical value of 1.0, or whether an intercept differs significantly from 0.0.

- Find the crossing point 667 of two lines.

Lines that must be fit with nonlinear regression

There are two situations where you might first think that linear regression is the best analysis, but in fact nonlinear regression is necessary:

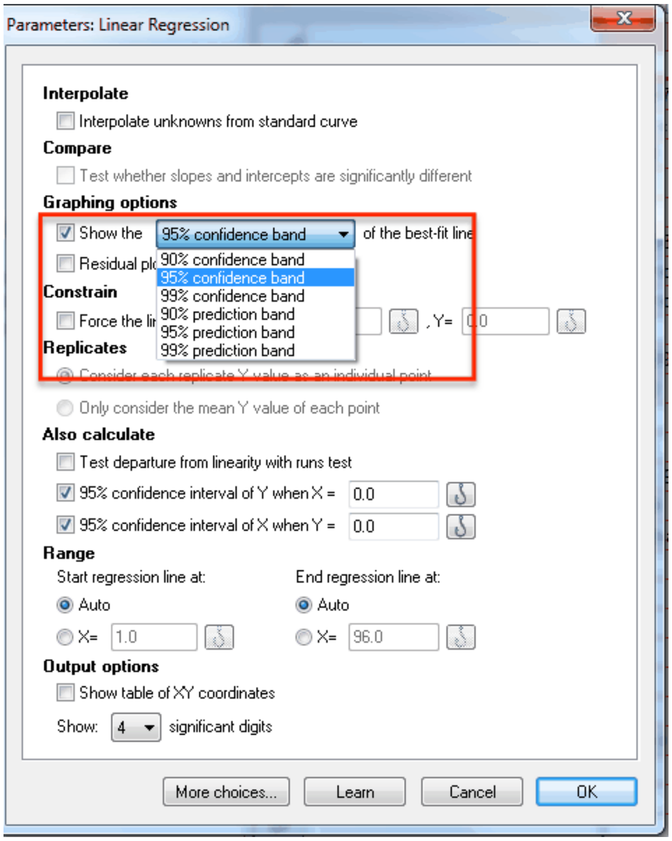
- If your Y axis uses a logarithmic or probability scale, then a straight line on the graph is created by a nonlinear model. In this case, although the line on the graph is straight, the model is not actually linear. You need to fit the 'line' 652 with nonlinear regression.

- If you want to fit two lines to different segments of the data, this cannot be done with Prism's simple linear regression analysis. However, Prism's nonlinear regression can fit segmental linear regression 656.

4.1.1.4 Confidence and prediction bands (linear regression)

Plotting confidence or prediction bands

If you check the option box on the top of the Simple linear regression parameters dialog, Prism will calculate and graph either the 95% confidence band or 95% prediction band of the regression line.



Parameters: Linear Regression

Interpolate

☐ Interpolate unknowns from standard curve

Compare

☐ Test whether slopes and intercepts are significantly different

Graphing options

☑ Show the 95% confidence band of the best-fit line

☐ Residual pl

- 90% confidence band
- 95% confidence band
- 99% confidence band
- 90% prediction band
- 95% prediction band
- 99% prediction band

Constrain

☐ Force the li , Y= 0.0

Replicates

● Consider each replicate Y value as an individual point

○ Only consider the mean Y value of each point

Also calculate

☐ Test departure from linearity with runs test

☑ 95% confidence interval of Y when X = 0.0

☑ 95% confidence interval of X when Y = 0.0

Range

Start regression line at: ● Auto ○ X= 1.0

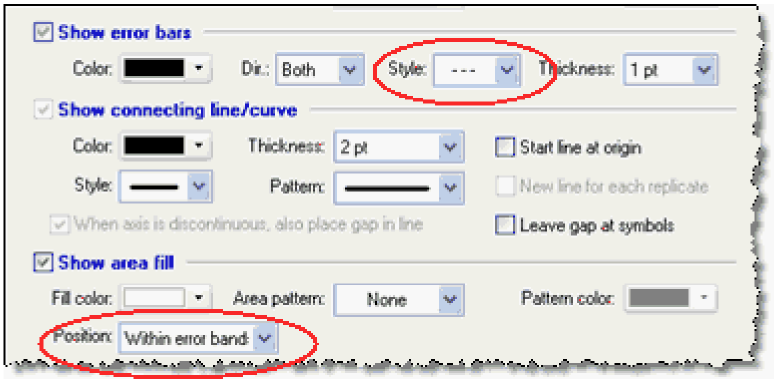
End regression line at: ● Auto ○ X= 96.0

Output options

☐ Show table of XY coordinates

Show: 4 significant digits

To adjust the appearance of the confidence or prediction bands, go to the Format Graph dialog, select the dataset that represents the regression line, and adjust the error bars and area fill settings. You can also choose to fill the area enclosed by the confidence or prediction bands.



☑ Show error bars

Color: ■ Dir.: Both Style: --- Thickness: 1 pt

☑ Show connecting line/curve

Color: ■ Thickness: 2 pt ☐ Start line at origin

Style: ── Pattern: ── ☐ New line for each replicate

☑ When axis is discontinuous, also place gap in line ☐ Leave gap at symbols

☑ Show area fill

Fill color: □ Area pattern: None Pattern color: ■

Position: Within error band

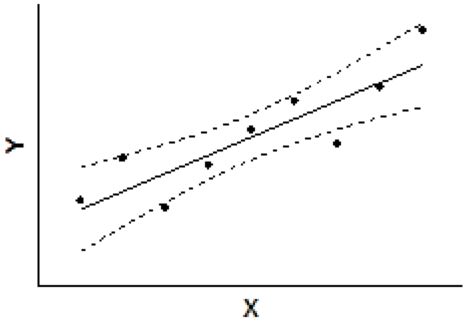
Plotting both confidence and prediction bands on the same graph

Prism lets you choose either a confidence band or a prediction band as part of the linear regression dialog. But not both. To plot both on one graph, you need to analyze your data twice, choosing a confidence band the first time and a prediction band the second time.

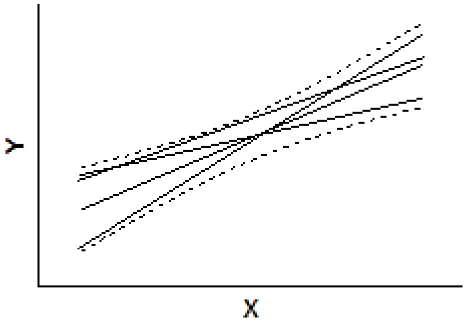
The regression lines (and bands) are data sets that you can add to any graph by drag-n-drop or using the Change menu from the graph. So go to the graph with the first regression line, and drag the second regression analysis onto that graph. Or double click on the graph, and use the middle tab of Format Graph to add the other regression data set.

Meaning of confidence bands

Two confidence bands surrounding the best-fit line define the confidence interval of the best-fit line.



The dashed confidence bands are curved. This does not mean that the confidence band includes the possibility of curves as well as straight lines. Rather, the curved lines are the boundaries of all possible straight lines. The figure below shows four possible linear regression lines (solid) that lie within the confidence band (dashed).

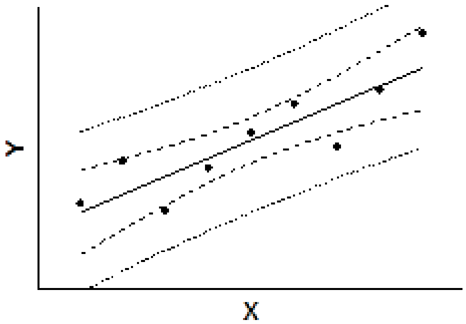


Given the assumptions of linear regression, you can be 95% confident that the two curved confidence bands enclose the true best-fit linear regression line, leaving a 5% chance that the true line is outside those boundaries.

Many data points will be outside the 95% confidence bands. The confidence bands are 95% sure to contain the best-fit regression line. This is not the same as saying it will contain 95% of the data points.

Meaning of the prediction bands

Prism can also plot the 95% prediction bands. The prediction bands are further from the best-fit line than the confidence bands, a lot further if you have many data points. The 95% prediction band is the area in which you expect 95% of all data points to fall. In contrast, the 95% confidence band is the area that has a 95% chance of containing the true regression line. This graph shows both prediction and confidence intervals (the curves defining the prediction intervals are further from the regression line).



When to plot confidence and prediction bands

The confidence bands sort of combine the confidence intervals of the slope and intercept in a visual way. Use confidence bands to learn how precisely your data define the best-fit line.

Prediction bands are wider, to also include the scatter of the data. Use prediction bands when your main goal is show the variation in your data.

4.1.1.5 Graphing tips: Simple linear regression

Graphing the regression line

When Prism performs simple linear regression, it automatically superimposes the line on the graph.

If you need to create additional graphs, or change which line is plotted on which graph, keep in mind that the line generated by linear regression is seen by Prism as a data set. You can add lines to a graph or remove lines from a graph on the 'Data sets on graph' tab of the Format Graph dialog.

Residuals

If you check an option on the simple linear regression dialog, Prism will create a results table with residuals, which are the vertical distances of each point from the regression line. The X values in the residual table are identical to the X values you entered. The Y values are the residuals. A residual with a positive value means that the point is above the line; a residual with a negative value means the point is below the line.

When Prism creates the table of residuals, it also automatically makes a new graph containing the residuals and nothing else. You can treat the residuals table like any other table, and do additional analyses or make additional graphs.

If the assumptions of simple linear regression have been met, the residuals will be randomly scattered above and below the line at Y=0. The scatter should not vary with X. You also should not see large clusters of adjacent points that are all above or all below the Y=0 line.

See an example 361 of residuals from nonlinear regression.

4.1.1.6 Difference between linear regression and correlation

Correlation and linear regression are not the same.

What is the goal?

Correlation quantifies the degree to which two variables are related. Correlation does not fit a line through the data points. You simply are computing a correlation coefficient (r) that tells you how much one variable tends to change when the other one does. When r is 0.0, there is no relationship. When r is positive, there is a trend that one variable goes up as the other one goes up. When r is negative, there is a trend that one variable goes up as the other one goes down.

Linear regression finds the best line that predicts Y from X.

What kind of data?

Correlation is almost always used when you measure both variables. It rarely is appropriate when one variable is something you experimentally manipulate.

Linear regression is usually used when X is a variably you manipulate (time, concentration, etc.)

Does it matter which variable is X and which is Y?

With correlation, you don't have to think about cause and effect. It doesn't matter which of the two variables you call "X" and which you call "Y". You'll get the same correlation coefficient if you swap the two.

The decision of which variable you call "X" and which you call "Y" matters in regression, as you'll get a different best-fit line if you swap the two. The line that best predicts Y from X is not the same as the line that predicts X from Y (however both those lines have the same value for R^2).

Assumptions

The correlation coefficient itself is simply a way to describe how two variables vary together, so it can be computed and interpreted for any two variables. Further inferences, however, require an additional assumption -- that both X and Y are measured (are interval or ratio variables), and both are sampled from Gaussian distributions. This is called a bivariate Gaussian distribution. If those assumptions are true, then you can interpret the confidence interval of r and the P value testing the null hypothesis that there really is no correlation between the two variables (and any correlation you observed is a consequence of random sampling).

With linear regression, the X values can be measured or can be a variable controlled by the experimenter. The X values are not assumed to be sampled from a Gaussian distribution. The distances of the points from the best-fit line is assumed to follow a Gaussian distribution, with the SD of the scatter not related to the X or Y values.

Relationship between results

Correlation computes the value of the Pearson correlation coefficient, r. Its value ranges from -1 to +1.

Linear regression quantifies goodness of fit with r^2, sometimes shown in uppercase as R^2. If you put the same data into correlation (which is rarely appropriate; see above), the square of r from correlation will equal r^2 from regression.

4.1.1.7 How to fit one line to two data sets

If you entered data for more than one data set on one table and then choose simple linear or nonlinear regression, Prism usually fits one line or curve through each data set. Follow these steps to fit one line or curve through all the data.

1. Use the nonlinear regression analysis to fit the data, even if you are fitting a straight line. Don't use simple linear regression.

2. In the first tab of nonlinear regression dialog, pick an equation. If you want to use linear regression, open the "lines" folder of equations and choose "straight line". Fitting this model with nonlinear regression is the same as simple linear regression, except that Prism offers many more choices in its nonlinear regression dialog.

3. Go to the Constrain tab (of the nonlinear regression dialog), and share all the parameters. If you are fitting a line, share both the slope and also the intercept. "Sharing" means that Prism fits one best-fit value for all the data sets, rather than one for each data set. If you share all the parameters, then Prism fits and draws only one curve or line to all the data.

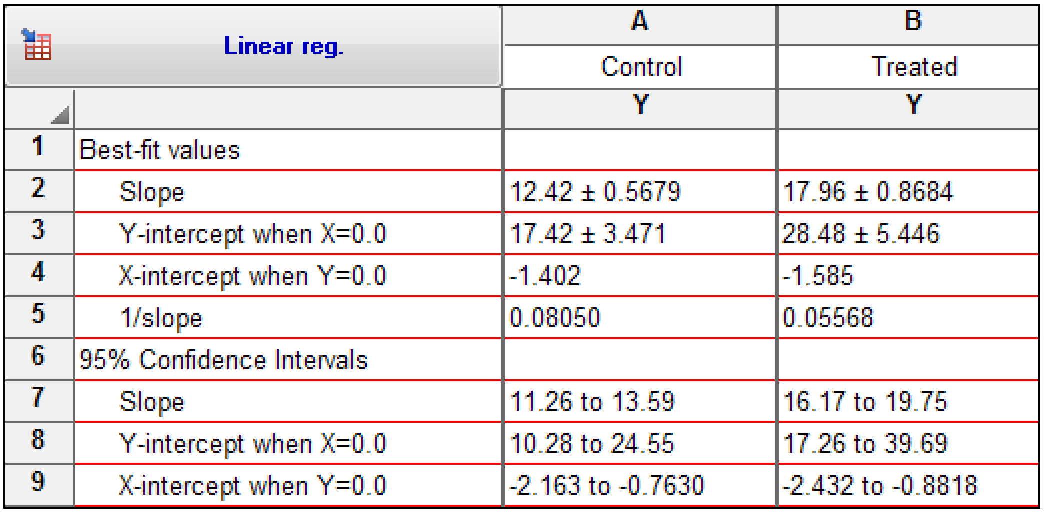
Caution: One of the assumptions of linear and nonlinear regression is that each data point contributes independent information. This means that it is totally random whether each point is above or below the curve, and by how far. When you fit two data sets to one line or curve, it is likely that you will be violating this assumption. Be careful when interpreting the results.

4.1.2 Results of simple linear regression

4.1.2.1 Slope and intercept

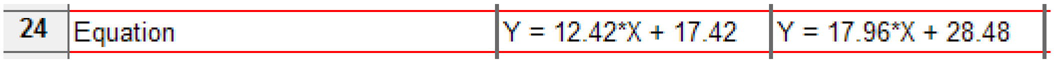
How Prism reports the slope and intercept

Prism first reports the best-fit values of the slope and intercept, along with their standard errors. It also reports the X intercept and the reciprocal of the slope. Below those values, it reports the 95% confidence interval of the slope and both intercepts.



	Linear reg.	A	B
		Control	Treated
		Y	Y
1	Best-fit values		
2	Slope	12.42 ± 0.5679	17.96 ± 0.8684
3	Y-intercept when X=0.0	17.42 ± 3.471	28.48 ± 5.446
4	X-intercept when Y=0.0	-1.402	-1.585
5	1/slope	0.08050	0.05568
6	95% Confidence Intervals		
7	Slope	11.26 to 13.59	16.17 to 19.75
8	Y-intercept when X=0.0	10.28 to 24.55	17.26 to 39.69
9	X-intercept when Y=0.0	-2.163 to -0.7630	-2.432 to -0.8818

At the bottom of the results page, the slope and intercept are reported again in the form of the equation that defines the best-fit line. You can copy this equation and paste onto a graph, or into a manuscript.

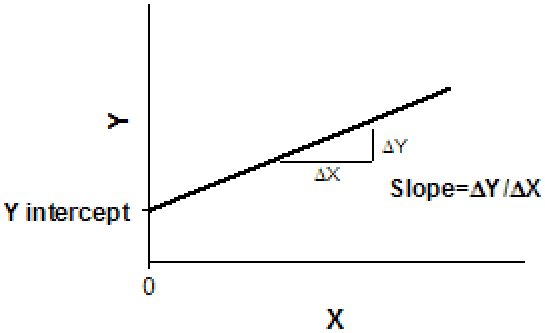


24	Equation	Y = 12.42*X + 17.42	Y = 17.96*X + 28.48

Interpreting the slope and intercept

The slope quantifies the steepness of the line. It equals the change in Y for each unit change in X. It is expressed in the units of the Y-axis divided by the units of the X-axis. If the slope is positive, Y increases as X increases. If the slope is negative, Y decreases as X increases.

The Y intercept is the Y value of the line when X equals zero. It defines the elevation of the line.



Interpreting the standard errors and confidence intervals

The standard error values of the slope and intercept can be hard to interpret, but their main purpose is to compute the 95% confidence intervals.

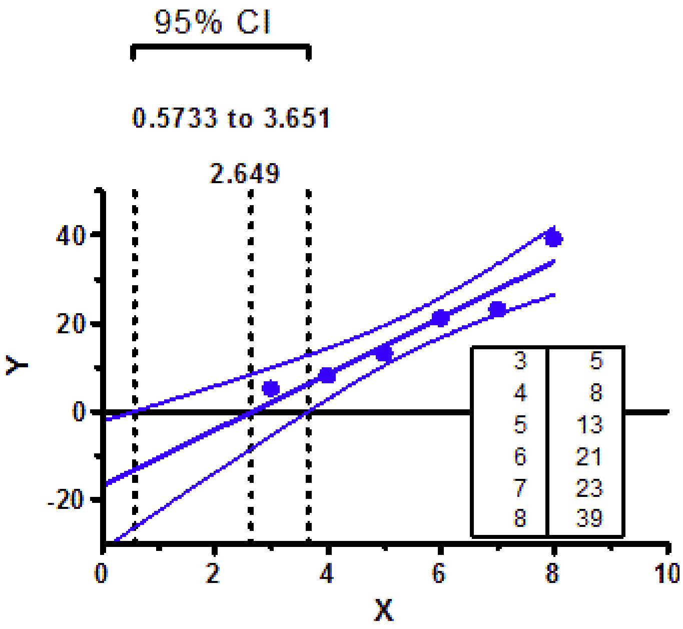
If you accept the assumptions of linear regression 141, there is a 95% chance that the 95% confidence interval of the slope contains the true value of the slope, and that the 95% confidence interval for the intercept contains the true value of the intercept. The width of the confidence intervals is determined by the number of data points, their distances from the line, and the spacing of the X values.

Prism can also plot either the confidence or prediction bands 352.

The X intercept and its confidence interval

GraphPad Prism reports the 95% confidence interval of the X intercept if you check an option on the Linear regression parameters dialog.

The 95% confidence interval for the X-intercept is not symmetrical around the X-intercept. It goes further in one direction than the other, as illustrated in the graph below.



Follow the Y=0 baseline from left to right. The region between the 95% confidence bands for the best fit line (blue curves) is the 95% CI of the X intercept. You can see that this confidence interval (between the two outermost dotted lines) is not symmetrical around the X intercept (the middle dotted line).

This asymmetry will be very noticeable if you only have a few points with lots of scatter, and will be almost unnoticeable with lots of points with little scatter.

Because the uncertainty is not symmetrical, it rarely makes sense to report a standard error of the X-intercept. It is much better to report both ends of the 95% confidence interval, which Prism reports. If you really want to compute a single standard error for the X intercept, you can do so by choosing nonlinear regression, and fitting this user-defined equation to the data:

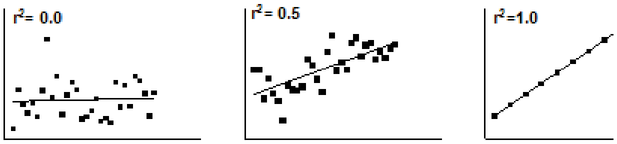
```
Y = slope*(X-Xintercept)
```

Prism will report the best-fit value of the X intercept along with a SE and 95% confidence interval. Since this confidence interval will be computed from the SE value it will be symmetrical around the X intercept, and so won't be as accurate as the asymmetrical interval reported by linear regression.

4.1.2.2 r2, a measure of goodness-of-fit of simple linear regression

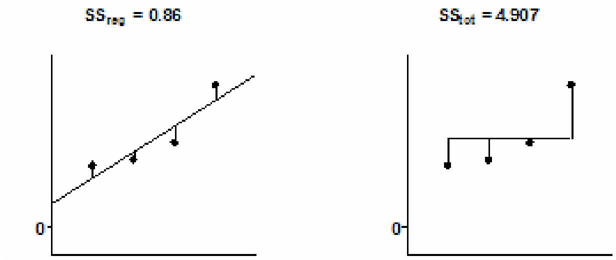
The meaning of r^2

The value r^2 is a fraction between 0.0 and 1.0, and has no units. An r^2 value of 0.0 means that knowing X does not help you predict Y. There is no linear relationship between X and Y, and the best-fit line is a horizontal line going through the mean of all Y values. When r^2 equals 1.0, all points lie exactly on a straight line with no scatter. Knowing X lets you predict Y perfectly.



How r^2 is computed

This figure demonstrates how Prism computes r^2.



$$r^2 = 1 - \frac{SS_{reg}}{SS_{tot}} = 1 - \frac{0.86}{4.91} = 0.84$$

The left panel shows the best-fit linear regression line This lines minimizes the sum-of-squares of the vertical distances of the points from the line. Those vertical distances are also shown on the left panel of the figure. In this example, the sum of squares of those distances (SSreg) equals 0.86. Its units are the units of the Y-axis squared. To use this value as a measure of goodness-of-fit, you must compare it to something.

The right half of the figure shows the null hypothesis -- a horizontal line through the mean of all the Y values. Goodness-of-fit of this model (SStot) is also calculated as the sum of squares of the vertical distances of the points from the line, 4.907 in this example. The ratio of the two sum-of-squares values compares the regression model with the null hypothesis model. The equation to compute r^2 is shown in the figure. In this example r^2 is 0.8428. The regression model fits the data much better than the null hypothesis, so SSreg is much smaller than SStot, and r^2 is near 1.0. If the regression model were not much better than the null hypothesis, r^2 would be near zero.

You can think of r^2 as the fraction of the total variance of Y that is "explained" by variation in X. The value of r^2 (unlike the regression line itself) would be the same if X and Y were swapped. So r^2 is also the fraction of the variance in X that is "explained" by variation in Y. In other words, r^2 is the fraction of the variation that is shared between X and Y.

In this example, 84% of the total variance in Y is "explained" by the linear regression model. The variance (SS) of the data from the linear regression model equals only 16% of the total variance of the Y values (SStot).

How is r^2 useful?

Many people find r^2 useful in this way: When you run a series of experiments, you want to make sure that today's experiment is consistent with other runs of the experiment. For example, if you always get r^2 between 0.90 and 0.95 but today you got r^2=0.75, then you should be suspicious and look carefully to see if something went wrong with the methods or reagents used in that particular experiment. And if a new employee brings you results showing r^2 of 0.99 using that same system, you should look carefully at how many"outliers" were removed, and whether some data were made up.

Why Prism doesn't report r^2 in constrained linear regression

Prism does not report r^2 when you force the line through the origin (or any other point), because the calculations would be ambiguous. There are two ways to compute r^2 when the regression line is constrained. As you saw in the previous section, r^2 is computed by comparing the sum-of-squares from the regression line with the sum-of-squares from a model defined by the null hypothesis. With constrained regression, there are two possible null hypotheses. One is a horizontal line through the mean of all Y values. But this line doesn't follow the constraint -- it does not go through the origin. The other null hypothesis would be a horizontal line through the origin, far from most of the data.

Because r^2 is ambiguous in constrained linear regression, Prism doesn't report it. If you really want to know a value for r^2, use nonlinear regression to fit your data to the equation Y=slope*X. Prism will report r^2 defined the first way (comparing regression sum-of-squares to the sum-of-squares from a horizontal line at the mean Y value).

Upper or lower case?

With linear regression, it is conventional to use the abbreviation r^2. With nonlinear regression, the convention is to use R^2. There appears to be no reason for this distinction.

4.1.2.3 Standard deviation of the residuals

Prism quantifies goodness of fit by reporting the standard deviation of the residuals, written Sy.x by Prism (but sometimes called Se). Remember that the residual is the vertical distance (in Y units) of the point from the fit line or curve. If you have n data points, after the regression, you have n residuals.

Sy.x is expressed in the same units as Y and all can be interpreted as the typical deviation of the points from the line or curve. It is computed by the equation below, where n is the number of points and K is the number of parameters fit by regression (usually 2, but 1 if you force the curve through a point). The value n-K is the number of degrees of freedom of the regression.

$$Sy.x = \sqrt{\frac{\sum(residual^2)}{n - K}}$$

4.1.2.4 Is the slope significantly different than zero?

Interpreting the P value when Prism fits both slope and intercept

Prism reports the P value testing the null hypothesis that the overall slope is zero. The P value answers this question:

> If there were no linear relationship between X and Y overall, what is the probability that randomly selected points would result in a regression line as far from horizontal (or further) than you observed?

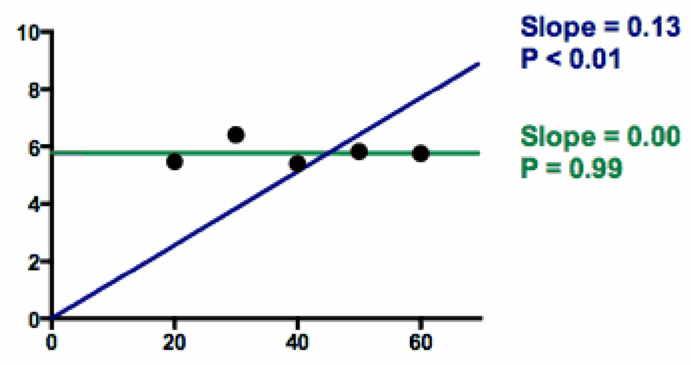
Equivalently:

> If there were no linear relationship between X and Y overall, what is the probability that randomly selected points would result in an R^2 value as high (or further) as you observed?

The P value is calculated from an F test, and Prism also reports the value of F and its degrees of freedom. You would get exactly the same P value from the t ratio computed by dividing the slope by its standard error.

Interpreting the P value when Prism fits only the slope

Beware of the P value from linear regression when you force the line to go through the origin or some other point. It is easy to misinterpret. The graph below demonstrates the problem:



The data points almost form a horizontal line.

When fit with linear regression the usual way (fit both slope and intercept; green line), the best fit value of the slope is 0.00. The P value answers the question: If the true slope is zero, what is the chance that the slope will be further from zero than the observed slope due only to random sampling. Since the observed slope is zero, there is almost a 100% chance of obtaining a slope that is further than zero than observed! So the P value is greater than 0.99, as high as a P value can be. Some people are confused and think the P value should be small purely because the points for a pattern. Not so. The P value, from conventional linear regression fitting both slope and intercept, will be small only when the points form a linear pattern that is not horizontal.

The results are very different when you fit linear regression with the constraint that the line has to go through the origin (blue line). To make the line go through the origin and also go near the points, the best-fit line has a slope is far from zero. Since the line is far from horizontal, the P value is tiny. Given the constraint that the line must go through the origin (X=0, Y=0; lower-left of graph), the data are quite convincing that best fit line is far from horizontal, so it makes sense that the P value is tiny.

Constraining a line to go through the origin (or some other point) can be very useful in some situations. Usually this option is used to fit calibration curves used for interpolation, in which case the P value is not useful. If you force the line through the origin, be very wary when interpreting the P value. It is rarely useful, and easy to misinterpret.

Download the Prism file.

4.1.2.5 Comparing slopes and intercepts

Prism compares slopes of two or more regression lines if you check the option: "Test whether the slopes and intercepts are significantly different".

Comparing slopes

Prism compares slopes first. It calculates a P value (two-tailed) testing the null hypothesis that the slopes are all identical (the lines are parallel). The P value answers this question:

> If the slopes really were identical, what is the chance that randomly selected data points would have slopes as different (or more different) than you observed.

If the P value is less than 0.05

If the P value is low, Prism concludes that the lines are significantly different. In that case, there is no point in comparing the intercepts. The intersection point of two lines is:

$$X = \frac{\text{Intercept}_1 - \text{Intercept}_2}{\text{Slope}_2 - \text{Slope}_1}$$

$$Y = \text{Intercept}_1 + \text{Slope}_1 \cdot X = \text{Intercept}_2 + \text{Slope}_2 \cdot X$$

If the P value for comparing slopes is greater than 0.05

If the P value is high, Prism concludes that the slopes are not significantly different and calculates a single slope for all the lines. Essentially, it shares 70 the Slope parameter between the two data sets.

Comparing intercepts

If the slopes are significantly different, there is no point comparing intercepts. If the slopes are indistinguishable, the lines could be parallel with distinct intercepts. Or the lines could be identical. with the same slopes and intercepts.

Prism calculates a second P value testing the null hypothesis that the lines are identical. If this P value is low, conclude that the lines are not identical (they are distinct but parallel). If this second P value is high, there is no compelling evidence that the lines are different. It does this by sharing the slopes (so they are forced to be the same) and then it compares elevations. With equal slopes, comparing elevations tests if the lines are identical. It doesn't matter if you compare the elevation at X=0 or X=any other value.

Relationship to ANCOVA and global regression

This method is equivalent to an Analysis of Covariance (ANCOVA), although ANCOVA can be extended to more complicated situations. It also is equivalent to using Prism's nonlinear regression analysis with a straight-line model, and using an F test to compare a global model where slope is shared among the data sets with a model where each dataset gets its own slope.

Reference

Chapter 18 of J Zar, *Biostatistical Analysis*, 2nd edition, Prentice-Hall, 1984.

4.1.2.6 Runs test following linear regression

The runs test determines whether your data differ significantly from a straight line. Prism can only calculate the runs test if you entered the X values in order.

A run is a series of consecutive points that are either all above or all below the regression line. In other words, a run is a consecutive series of points whose residuals are either all positive or all negative.

If the data points are randomly distributed above and below the regression line, it is possible to calculate the expected number of runs. If there are Na points above the curve and Nb points below the curve, the number of runs you expect to see equals [(2NaNb)/(Na+Nb)]+1. If you observe fewer runs than expected, it may be a coincidence of random sampling or it may mean that your data deviate systematically from a straight line. The P value from the runs test answers this question:

> If the data really follow a straight line, and you performed many experiments like this one, what fraction of the time would you obtain as few (or fewer) runs as observed in this experiment?

If the runs test reports a low P value, conclude that the data do not really follow a straight line, and consider using nonlinear regression to fit a curve.

The P values are always one-tail, asking about the probability of observing as few runs (or fewer) than observed. If you observe more runs than expected, the P value will be higher than 0.50.

.

4.1.2.7 Analysis checklist: Simple linear regression

✓ Can the relationship between X and Y be graphed as a straight line?

In many experiments the relationship between X and Y is curved, making linear regression inappropriate. It rarely helps to transform the data to force the relationship to be linear. Better, use nonlinear curve fitting.

✓ Is the scatter of data around the line Gaussian (at least approximately)?

Linear regression analysis assumes that the scatter of data around the best-fit line is Gaussian. In other words, it assumes that the residuals (the vertical distances of the points from the best-fit line) are sampled from a Gaussian (normal) distribution.

✓ Is the variability the same everywhere?

Linear regression assumes that scatter of points around the best-fit line has the same standard deviation all along the curve. The assumption is violated if the points with high or low X values tend to be further from the best-fit line. The assumption that the standard deviation is the same everywhere is termed homoscedasticity. (If the scatter goes up as Y goes up, you need to perform a weighted regression. Prism can't do this via the linear regression analysis. Instead, use nonlinear regression but choose to fit to a straight-line model.

✓ Do you know the X values precisely?

The linear regression model assumes that X values are exactly correct, and that experimental error or biological variability only affects the Y values. This is rarely the case, but it is sufficient to assume that any imprecision in measuring X is very small compared to the variability in Y.

✓ Are the data points independent?

Whether one point is above or below the line is a matter of chance, and does not influence whether another point is above or below the line.

✓ Are the X and Y values intertwined?

If the value of X is used to calculate Y (or the value of Y is used to calculate X) then linear regression calculations are invalid. One example is a Scatchard plot, where the Y value (bound/free) is calculated from the X value. Another example would be a graph of midterm exam scores (X) vs. total course grades(Y). Since the midterm exam score is a component of the total course grade, linear regression is not valid for these data.

4.1.3 How to: Simple logistic regression

Logistic regression was added with Prism 8.3.0. This section of the guide will provide you with information on how to perform simple logistic regression within Prism.

Fitting a simple logistic regression model 143

Example: Simple logistic regression 146

4.1.3.1 Fitting a simple logistic regression model

1. Create a data table

From the Welcome or New Table dialog, choose to create an XY data table. Be sure to select the option “Enter and plot a single Y value for each point.” Simple logistic regression in Prism currently does not allow for replicates in subcolumns. To enter replicates, simply add each replicate on its own row with its associated X value and observed outcome.

If you would like to see how Prism works on a sample data set, choose the sample data: Simple logistic regression. Otherwise, if you want to enter your own data, be sure that the Y values are binary and encoded as strictly 0s and 1s. Prism will not perform simple logistic regression (or multiple logistic regression) if your outcome variable is not encoded as 0s and 1s (read more about binary outcome variables 31).

Tip: An easy way to keep track of which outcome is coded as a 1 and which is coded as a 0 is to use a yes/no question as the title of your Y column. Using this method, you could enter a “1” to answer the question “yes” and a “0” to answer the question “no.”

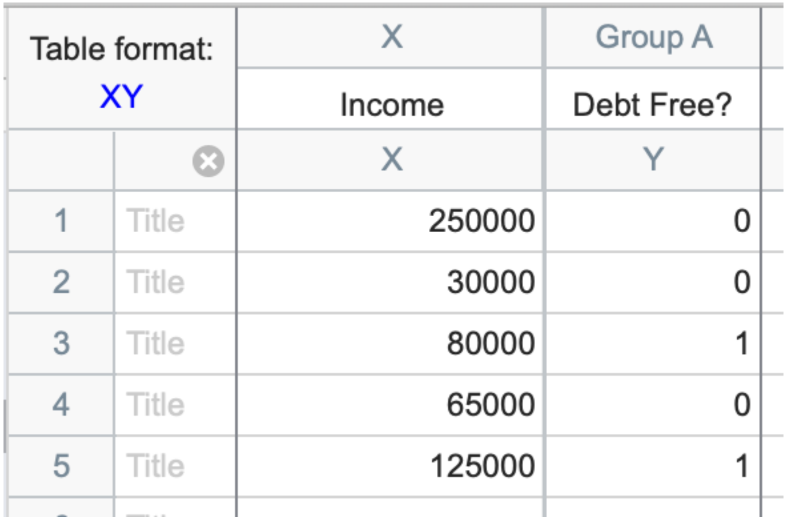
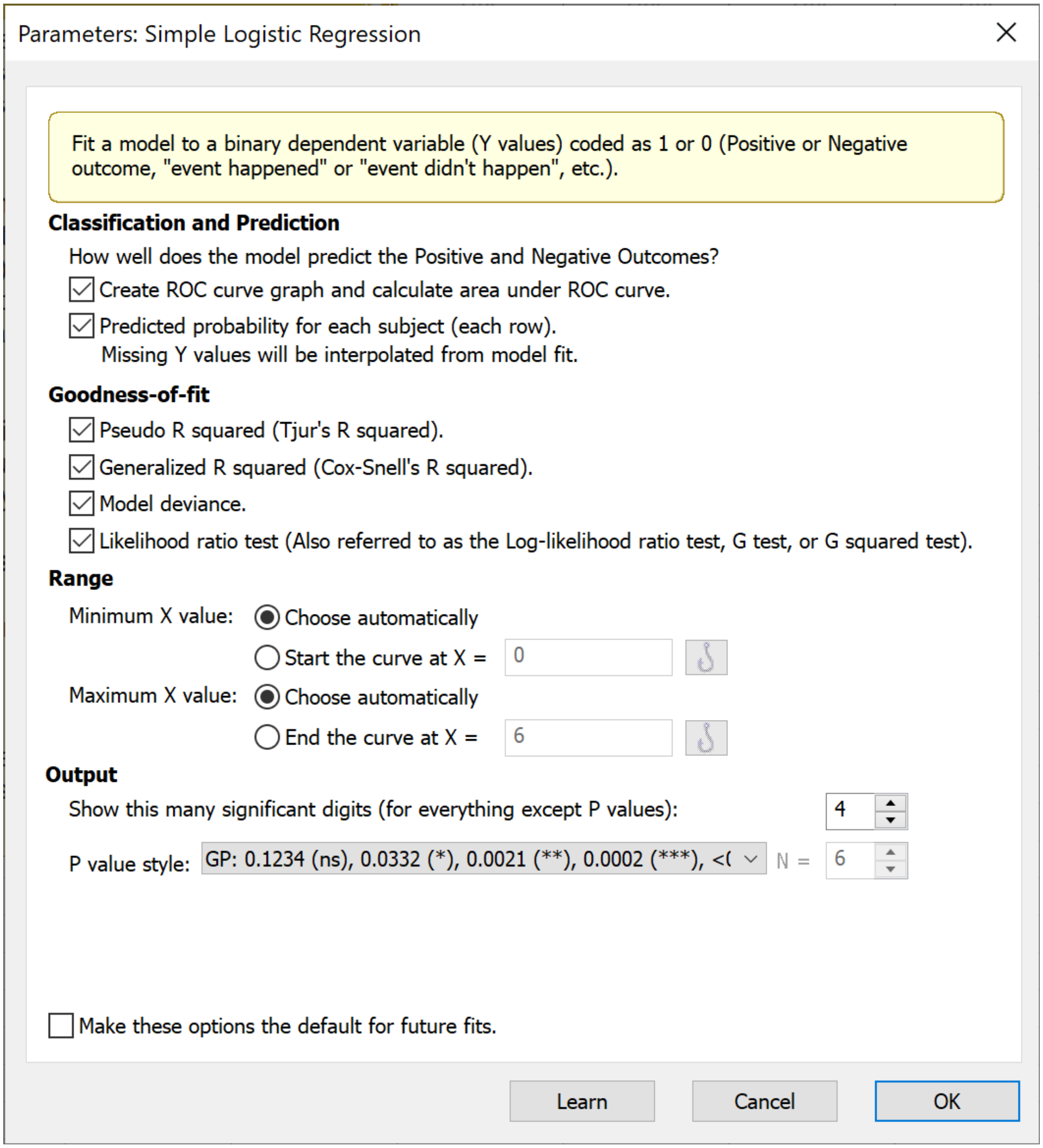


Table format: XY		X	Group A
		Income	Debt Free?
	⊗	X	Y
1	Title	250000	0
2	Title	30000	0
3	Title	80000	1
4	Title	65000	0
5	Title	125000	1

2. Analysis choices

To run simple logistic regression, click the Analyze button in the toolbar and choose simple logistic regression from the list of XY analyses. The parameters dialog for simple logistic regression offers several customization choices. For even more analysis options, consider copying your data to a multiple variables data table and using the multiple logistic regression analysis 198.



Parameters: Simple Logistic Regression

Fit a model to a binary dependent variable (Y values) coded as 1 or 0 (Positive or Negative outcome, "event happened" or "event didn't happen", etc.).

Classification and Prediction

How well does the model predict the Positive and Negative Outcomes?

☑ Create ROC curve graph and calculate area under ROC curve.

☑ Predicted probability for each subject (each row).
Missing Y values will be interpolated from model fit.

Goodness-of-fit

☑ Pseudo R squared (Tjur's R squared).

☑ Generalized R squared (Cox-Snell's R squared).

☑ Model deviance.

☑ Likelihood ratio test (Also referred to as the Log-likelihood ratio test, G test, or G squared test).

Range

Minimum X value: ● Choose automatically
○ Start the curve at X = 0

Maximum X value: ● Choose automatically
○ End the curve at X = 6

Output

Show this many significant digits (for everything except P values): 4

P value style: GP: 0.1234 (ns), 0.0332 (*), 0.0021 (**), 0.0002 (***), <(N = 6

□ Make these options the default for future fits.

a. Classification and interpolation

The options in this section provide information on how well the model does at predicting the positive and negative outcomes (entered as 1s and 0s in the data, respectively). You can request that Prism generate an ROC curve and report the area under the ROC curve (AUC) in the table of results. The AUC provides information on how well the logistic regression model classifies the observed data at various possible cutoffs. Check out this page 239 for more on understanding ROC curves.

Selecting the “Classification of each subject” will output a new green table titled “Row Classification” that first copies the column of observed X values and adds a column containing the predicted probabilities for each of those values from the simple logistic regression model. Note that you can use this option to predict new observations at specific values of X. To do this, simply add a row at the bottom of the original data table with the desired X value (or values) and leave the corresponding Y values blank (note that this process is very similar to the way interpolation is performed for simple linear regression, but is done automatically without selecting additional options).

b. Goodness-of-fit

In addition to classification performance, Prism offers four ways to evaluate model performance.

The standard metric for evaluating the fit of a linear model is R squared. However, because the simple logistic regression model is not fit using the same techniques as simple linear regression, this metric is not appropriate for logistic regression. For simple logistic regression, Prism offers two alternatives to R squared. Although these have the term R squared in their name, they do not have the same interpretation as R squared from simple linear regression (i.e. they do not represent the proportion of variance explained by the model).

Instead, both Tjur’s Pseudo R squared and Cox-Snell’s Generalized R squared are values that range between 0 and 1, with larger values representing better predictive performance of the model. More details on these pseudo R squared metrics (and others offered through multiple logistic regression from the multiple variables data table) can be found here 236.

Model deviance produces a value sometimes referred to as G squared which is used to calculate the test statistic for the likelihood ratio test. If you are comparing different X predictors for your model, you can choose the predictor that results in the smallest model deviance. You can also use the Pseudo and Generalized R squared values in this way.

The likelihood ratio test (LRT) compares the logistic model that includes the given X predictor to a model without that X predictor (i.e. an “intercept-only” model). Like other hypothesis tests, the LRT uses a null hypothesis, and generates a P value to test that null hypothesis. In this case, the null hypothesis is that the intercept-only model (the model without the X variable) fits the data better than the model with the X variable. If your X variable contributes to model prediction performance, you would expect that the P value for this test would be small.

c. Range

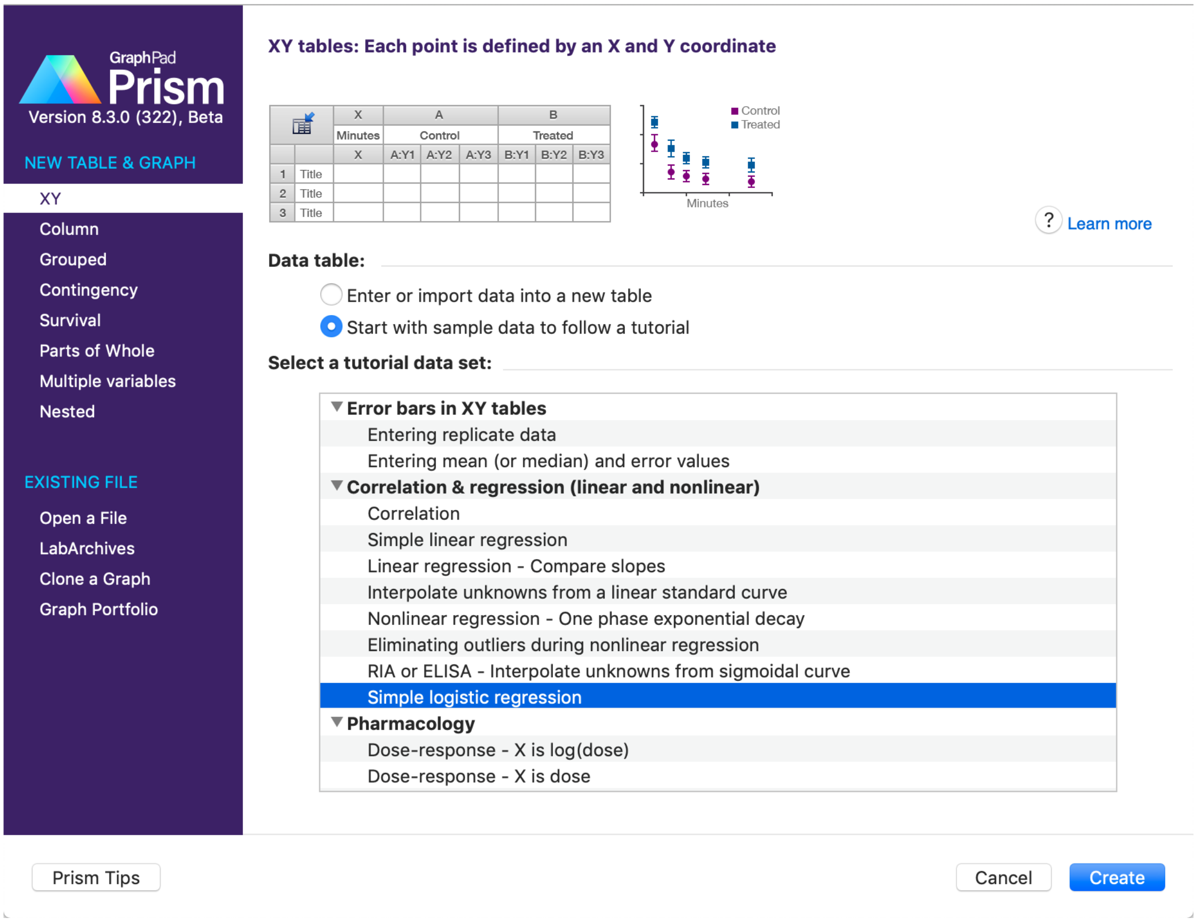
A logistic plot is automatically generated when you perform simple logistic regression. You can allow Prism to choose the default minimum and maximum X values for plotting this curve based on your data, or you may choose to increase or decrease these limits as you choose.

4.1.3.2 Example: Simple logistic regression

This guide will walk you through the process of performing simple logistic regression with Prism. Logistic regression was added with Prism 8.3.0

The data

To begin, we'll want to create a new XY data table from the Welcome dialog



XY tables: Each point is defined by an X and Y coordinate

Data table:

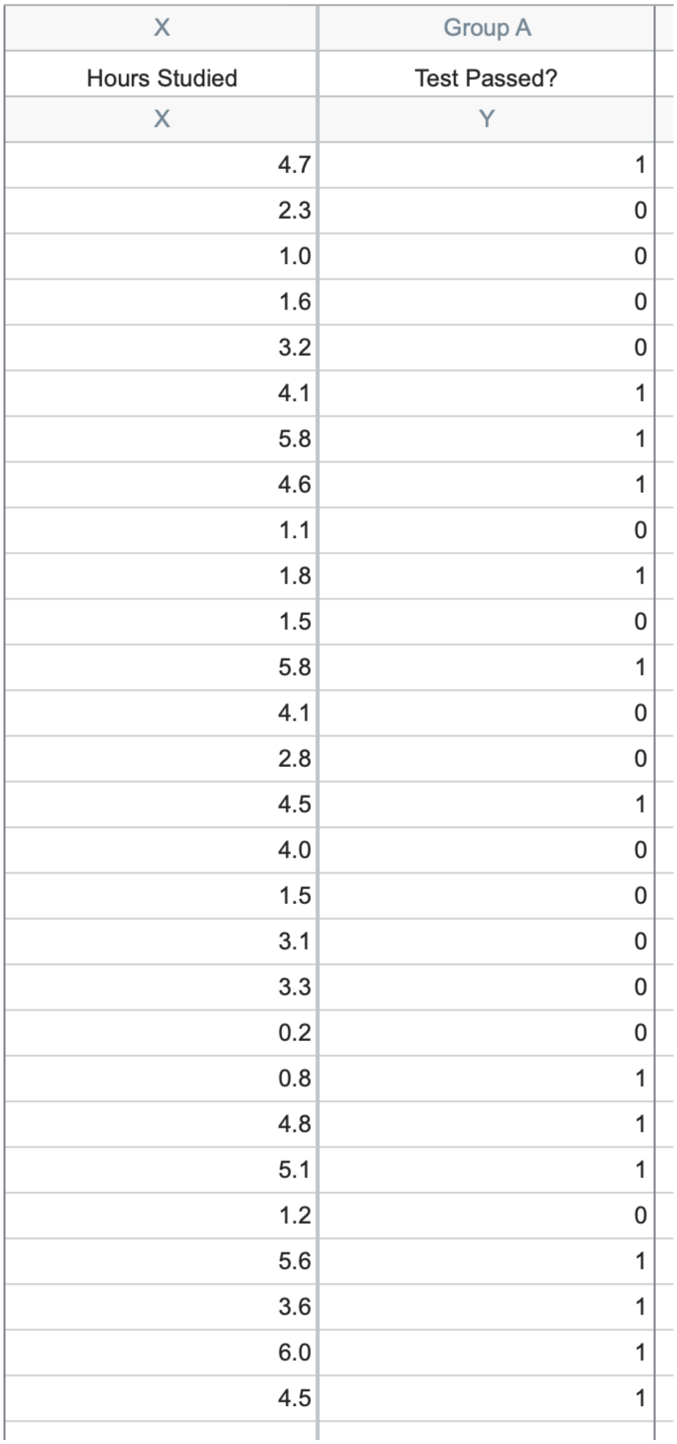
○ Enter or import data into a new table

● Start with sample data to follow a tutorial

Select a tutorial data set:

- ▼ Error bars in XY tables
  - Entering replicate data
  - Entering mean (or median) and error values
- ▼ Correlation & regression (linear and nonlinear)
  - Correlation
  - Simple linear regression
  - Linear regression - Compare slopes
  - Interpolate unknowns from a linear standard curve
  - Nonlinear regression - One phase exponential decay
  - Eliminating outliers during nonlinear regression
  - RIA or ELISA - Interpolate unknowns from sigmoidal curve
  - Simple logistic regression
- ▼ Pharmacology
  - Dose-response - X is log(dose)
  - Dose-response - X is dose

For the purposes of this walkthrough, we will be using the Simple logistic regression sample data found in the "Correlation & regression" section of the sample files. To use this data, click on "Simple logistic regression" in the list, and then click "Create". You will then be shown a set of data with two columns: "Hours Studied" in the X column and "Test Passed?" in the Y column.

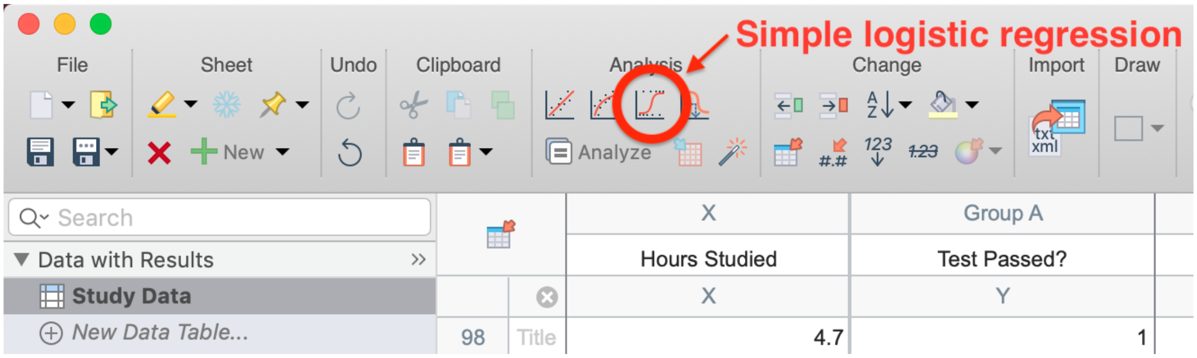


X	Group A
Hours Studied	Test Passed?
X	Y
4.7	1
2.3	0
1.0	0
1.6	0
3.2	0
4.1	1
5.8	1
4.6	1
1.1	0
1.8	1
1.5	0
5.8	1
4.1	0
2.8	0
4.5	1
4.0	0
1.5	0
3.1	0
3.3	0
0.2	0
0.8	1
4.8	1
5.1	1
1.2	0
5.6	1
3.6	1
6.0	1
4.5	1

This data represents a collection of 125 students and the amount of time that they spent preparing for a test along with the outcome of the test: did the student pass (entered as a 1 in the data table) or did the student fail (entered as a 0 in the data table)?

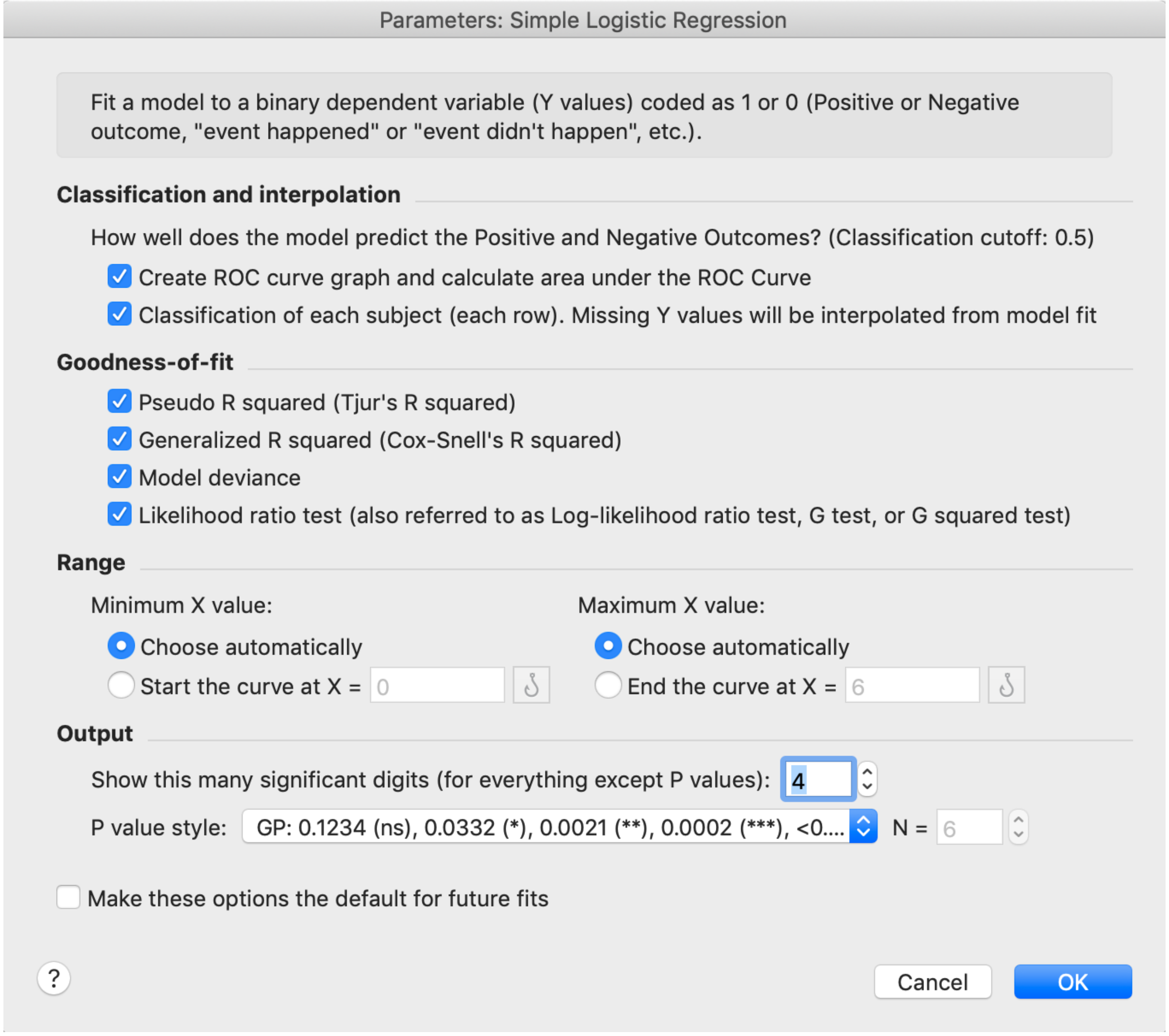
Initiating the analysis

To perform simple logistic regression on this dataset, click on the simple logistic regression button in the toolbar (shown below). Alternatively, you can click on the "Analyze" button in the toolbar, then select "Simple logistic regression" from the list of available XY analyses.



The analysis dialog

After clicking the simple logistic regression button, the parameters dialog for this analysis will appear. For the purposes of this walkthrough, we won't need to change any of the default options. The results for some of these options are discussed below, but additional information can for these options can be found here 143.



Parameters: Simple Logistic Regression

Fit a model to a binary dependent variable (Y values) coded as 1 or 0 (Positive or Negative outcome, "event happened" or "event didn't happen", etc.).

Classification and interpolation

How well does the model predict the Positive and Negative Outcomes? (Classification cutoff: 0.5)

☑ Create ROC curve graph and calculate area under the ROC Curve

☑ Classification of each subject (each row). Missing Y values will be interpolated from model fit

Goodness-of-fit

☑ Pseudo R squared (Tjur's R squared)

☑ Generalized R squared (Cox-Snell's R squared)

☑ Model deviance

☑ Likelihood ratio test (also referred to as Log-likelihood ratio test, G test, or G squared test)

Range

Minimum X value:

● Choose automatically

○ Start the curve at X = 0

Maximum X value:

● Choose automatically

○ End the curve at X = 6

Output

Show this many significant digits (for everything except P values): 4

P value style: GP: 0.1234 (ns), 0.0332 (*), 0.0021 (**), 0.0002 (***), <0.... N = 6

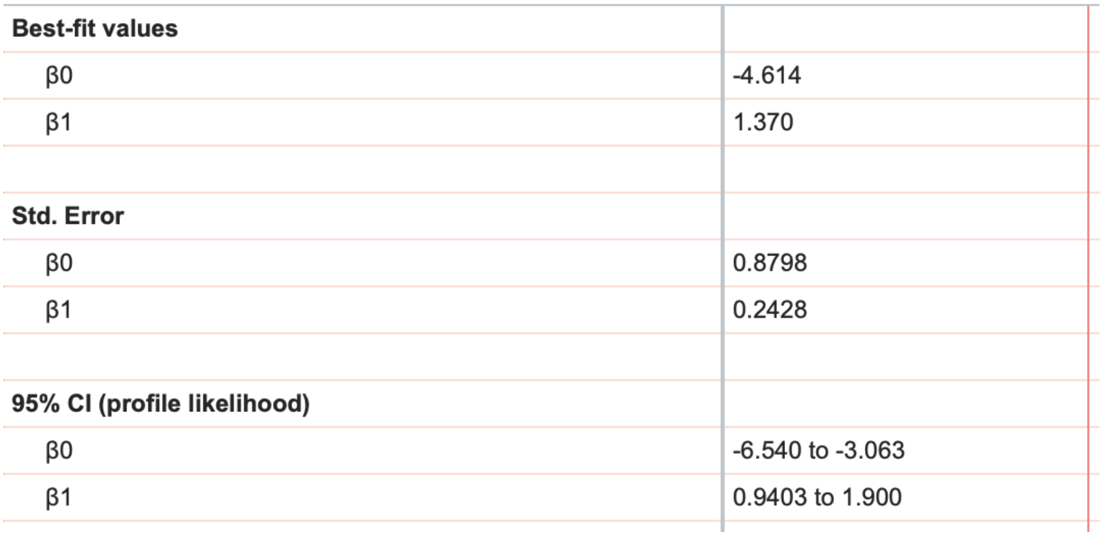
□ Make these options the default for future fits

Once you click "OK", you'll be taken to the main results sheet which will be discussed in the next section.

Results of simple logistic regression

Parameter estimates

The first thing that you'll see on the results sheet are the best fit value estimates along with standard errors and 95% confidence intervals for both β0 and β1.

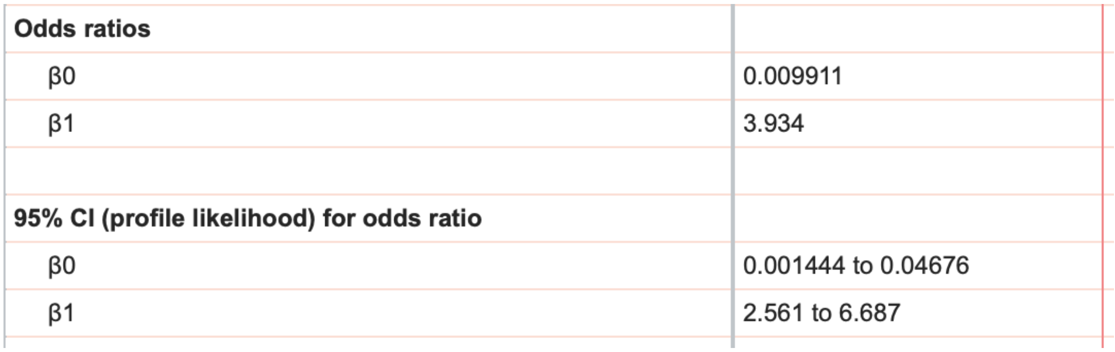


Best-fit values	
β0	-4.614
β1	1.370
Std. Error	
β0	0.8798
β1	0.2428
95% CI (profile likelihood)	
β0	-6.540 to -3.063
β1	0.9403 to 1.900

These parameters are sometimes referred to as the "intercept" and "slope", respectively, based on their relationship to the "log odds"[162].

Odds ratios

Because it is hard to directly interpret β0 and β1, it's common to look instead to the odds ratios and their 95% confidence intervals (reported farther down on the results sheet).



Odds ratios	
β0	0.009911
β1	3.934
95% CI (profile likelihood) for odds ratio	
β0	0.001444 to 0.04676
β1	2.561 to 6.687

A more detailed explanation of odds ratios can be found here[165], but what the odds ratio for β1 tells us is that increasing X by 1 multiplies the odds of success by the value for β1. Take the values in these results, for example. Recall that X represents the number of hours studied. Thus, an odds ratio of 3.934 tells us that for every additional hour studied, the odds of passing the test are multiplied by almost 4!

In case you haven't read about the relationship of probability and odds, here's a quick summary:

Odds = Probability of success/Probability of failure

Since the probability of failure is just 1 - Probability of success, we can write this as:

Odds = (Probability of success)/(1 - Probability of success)

For example, let's say there's a 75% probability of success, the odds would then be calculated as:

Odds = 0.75/(1 - 0.75) = 0.75/0.25 = 3

Commonly, we would say that "the odds are 3:1" (read "three to one").

X at 50%

Another key value that Prism reports for simple logistic regression is the value of X when the probability of success is predicted to be 50% (or 0.5). Interestingly, using our equation for odds given above, we can see that when probability is 50%, the odds are equal to 1 (also known as "even odds"). In our case, the value of X at 50% is 3.37, meaning that for those students that studied 3.37 hours, the odds of passing the test were 1:1 (a 50% probability of passing.. not great!).

If we combine this result with the odds ratio, we can quickly determine the odds and probability of passing if the student studies one hour more. Remember, the odds ratio that Prism reports tells us by how much the odds are multiplied when X increases by 1. We know that the odds are 1 when X is 3.37, and the odds ratio is 3.934. Thus, increasing X by 1, from 3.37 to 4.37 gives us a new odds of 1*3.934, or just 3.934. That is the predicted odds of passing for students that studied 4.37 hours (just one hour extra).

It is easy to convert this odds to a probability:

Odds = 3.934 = (Probability of success)/(1 - Probability of success)

3.934*(1 - Probability of success) = Probability of success

3.934 - 3.934*(Probability of success) = Probability of success

3.934 = Probability of success + 3.934*(Probability of success)

3.934 = (Probability of success)*(1 + 3.934)

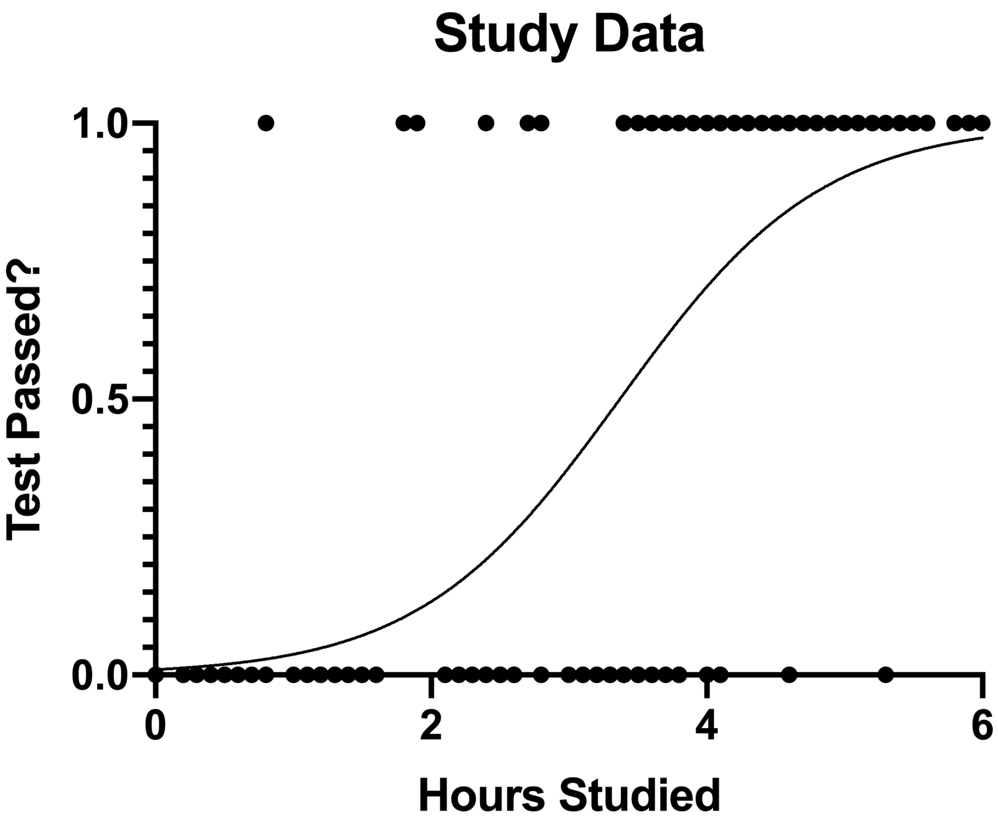
3.934 = (Probability of success)*4.934

Probability of success = 3.934/4.934

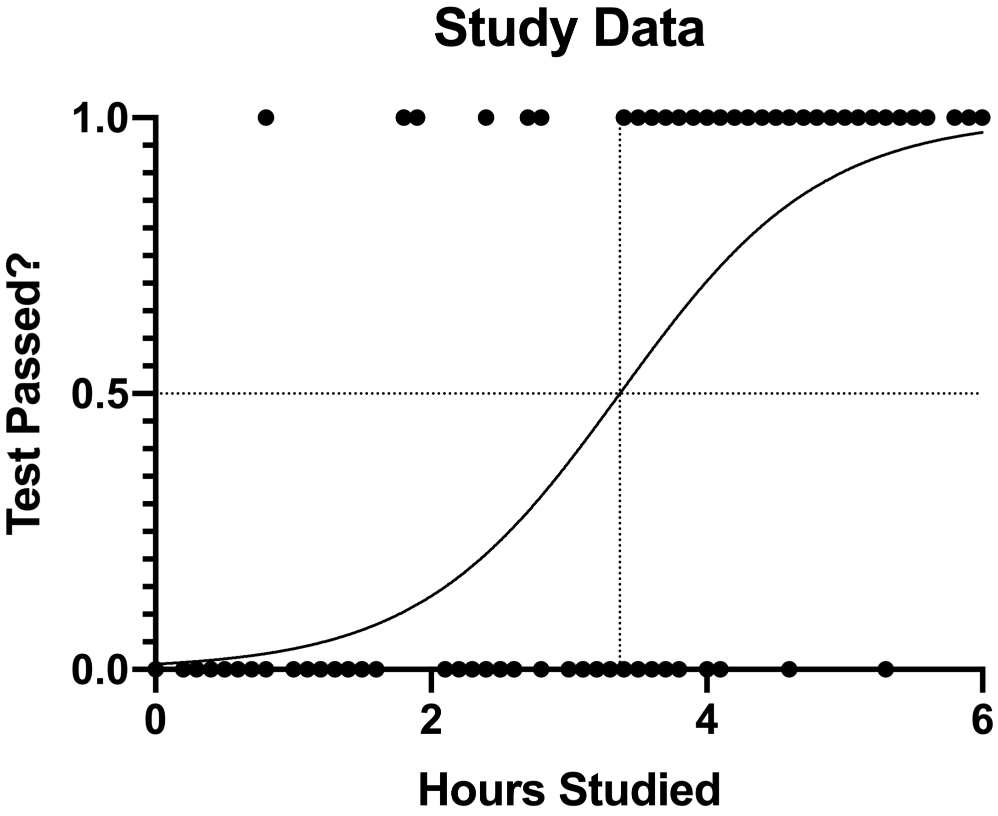
Probability of success = 0.797 or 79.7%

The logistic regression curve

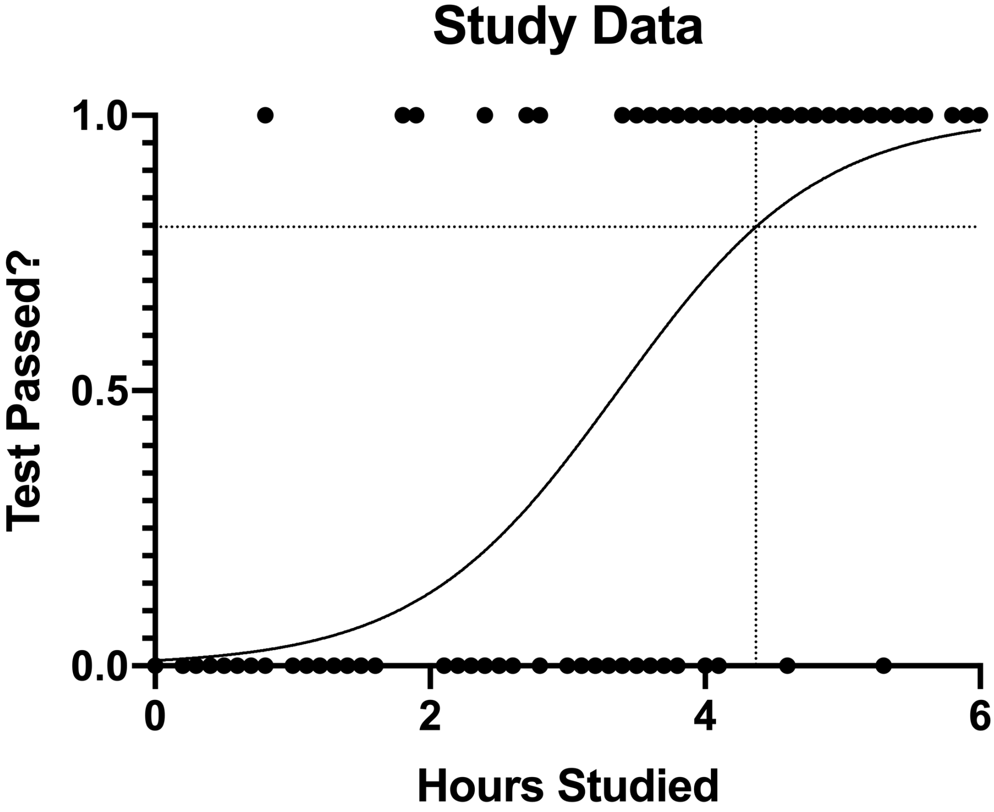
If we break away from the results sheet for just a moment, we can take a look at the curve that logistic regression plotted for our data. This graph (shown below), confirms some of the observations that we made in the previous sections:



The curve on this graph plots the predicted probability of passing the test (Y) as a function of the number of hours studied (X). As we discussed, we can quickly see that for a study time of 3.37 hours, the predicted probability of passing the test is 50%:



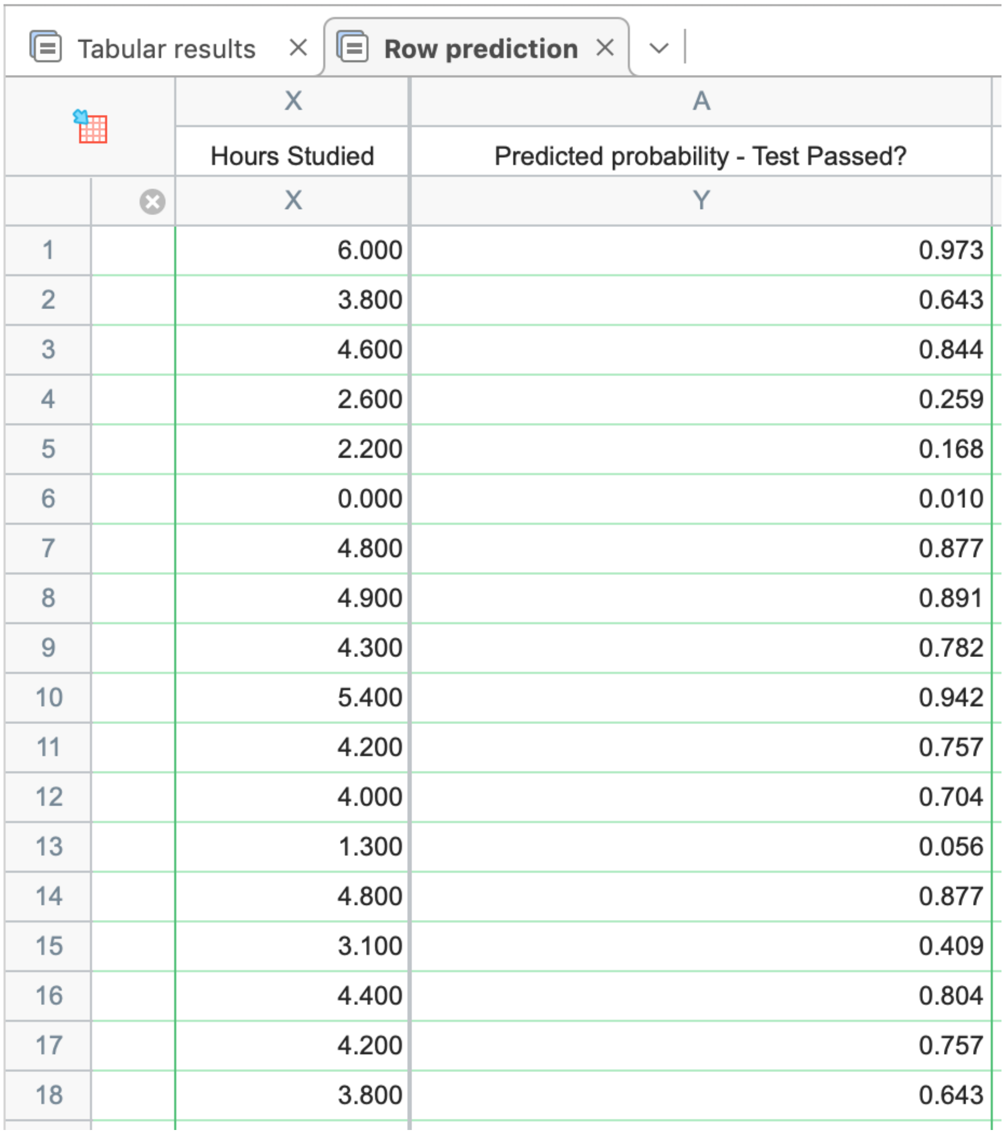
We can also confirm our claims about odds ratios from this graph. We can see that by increasing the amount of time studied by one hour (to 4.37 hours total), the predicted probability of passing the test increases to ~80%:



In fact, you can use this curve to determine the predicted probability of passing the test for any given amount of time studied. The next section discusses how to determine the predicted probability for any entered X value

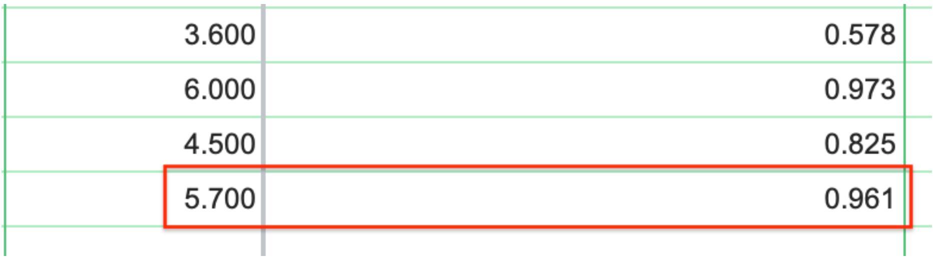
Row prediction

If we switch back to the main results sheet for simple logistic regression, you'll see at the top a sheet tab titled "Row prediction". When clicking on this tab, Prism will provide a complete list of predicted probabilities for all entered X values:



		X	A
		Hours Studied	Predicted probability - Test Passed?
		X	Y
1		6.000	0.973
2		3.800	0.643
3		4.600	0.844
4		2.600	0.259
5		2.200	0.168
6		0.000	0.010
7		4.800	0.877
8		4.900	0.891
9		4.300	0.782
10		5.400	0.942
11		4.200	0.757
12		4.000	0.704
13		1.300	0.056
14		4.800	0.877
15		3.100	0.409
16		4.400	0.804
17		4.200	0.757
18		3.800	0.643

This table provides predicted probabilities for all X values in the data table. This includes both X values from the data that were fit as well as lone X values that were entered without a Y value. Say, for example, you wanted to know the probability of passing the test given this model and a study time of 5.7 hours. Of the 125 students in this data set, none studied 5.7 hours, but going back to the original data table, you can enter 5.7 at the bottom of the X column (without an associated Y value), and then return to the Row Classification table:

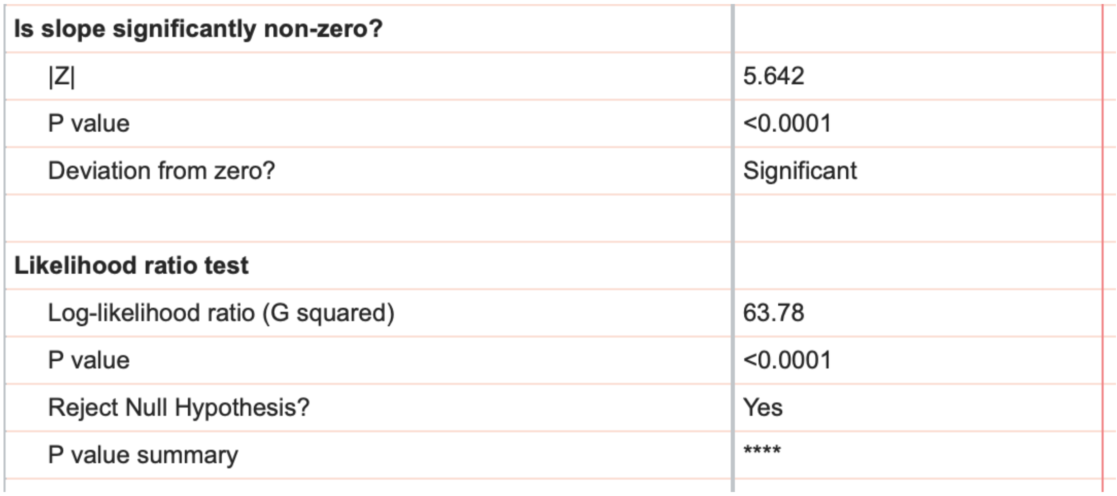


3.600	0.578
6.000	0.973
4.500	0.825
5.700	0.961

This result tells us that - based on the observations of the 125 students - if a new student were to study for this same test for 5.7 hours, they would have a 96.1% probability of passing!

Hypothesis tests

If we click back to the "Tabular results" tab of the results sheet, we can continue investigating the other results reported by simple logistic regression. The next section of the results provides two different ways to test how well the model fits the data 165. While these tests are quite similar, it's important to understand how each of them work and the hypothesis that each are testing to interpret the results.

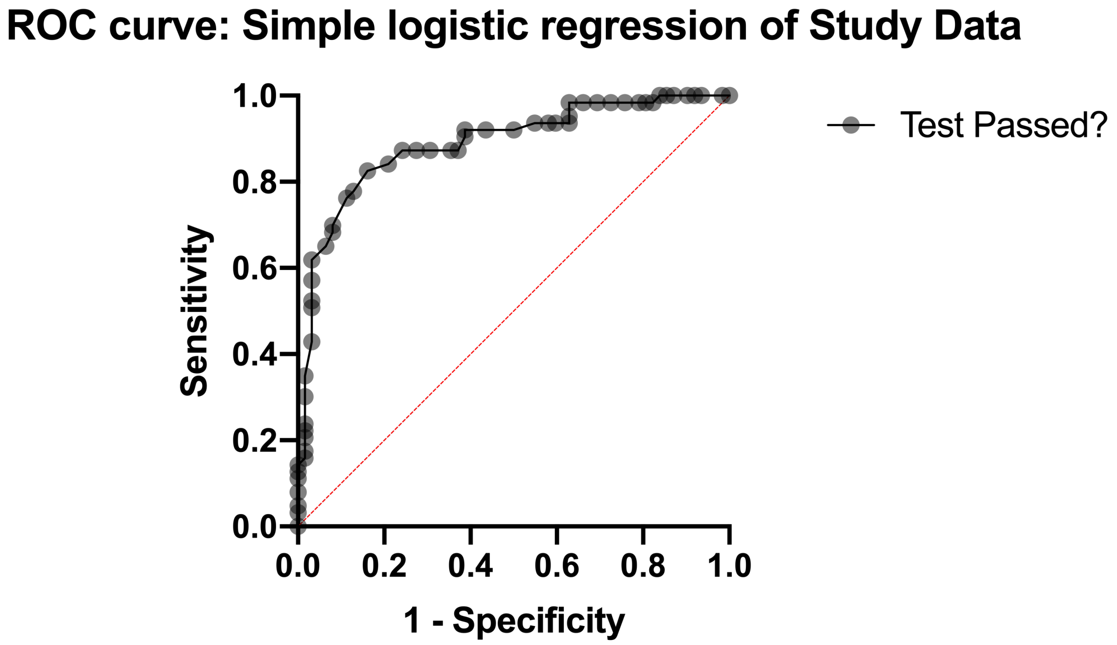


Is slope significantly non-zero?	
|Z|	5.642
P value	<0.0001
Deviation from zero?	Significant
Likelihood ratio test	
Log-likelihood ratio (G squared)	63.78
P value	<0.0001
Reject Null Hypothesis?	Yes
P value summary	****

Based on the results given for our data, we can conclude that the effect of studying (given by the coefficient β1) is definitely non-zero; in other words, the amount of time studied had a definite effect on the probability of passing the test.

The ROC curve and the area under the ROC curve

The next section of the results deals exclusively with something called an ROC curve. The ROC curve for this analysis is provided in the Graphs section of the Navigator, and looks like this:

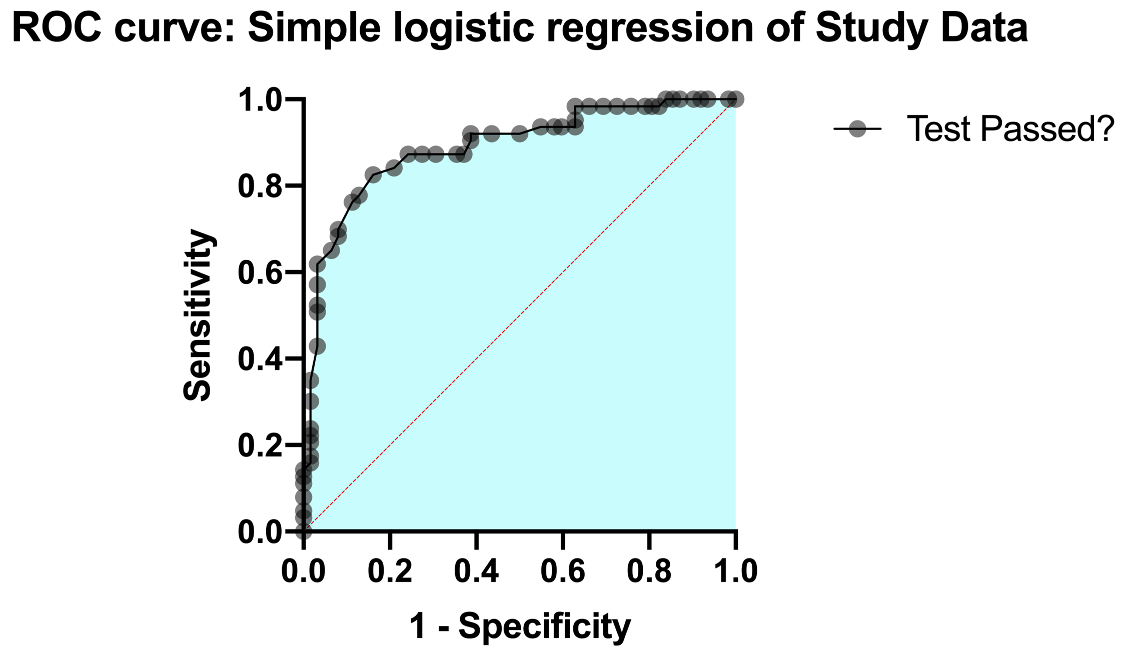


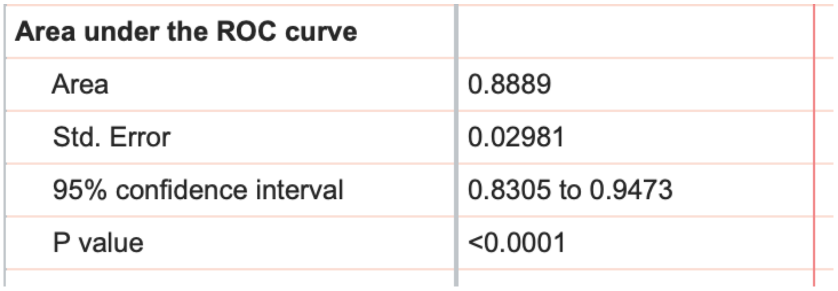
Understanding ROC curves takes a bit of experience, but ultimately what these graphs are showing you is the relationship between the model's ability to correctly classify successes correctly and its ability to classify failures correctly. The way the model classifies observations is by setting a cutoff value. Any predicted probability greater than this cutoff is classified as a 1, while any predicted probability less than this cutoff is classified as a 0. If you set a very low cutoff, you would almost certainly classify all of your observed successes correctly. The proportion of observed successes correctly classified is referred to as the Sensitivity, and is plotted on the Y axis of the ROC curve (a Y value of 1 represents perfect classification of successes, while a Y value of 0 represents complete mis-classification of successes).

However, with a very low cutoff, you would also likely *incorrectly* classify many failures as successes as well. Specificity is the proportion of correctly classified failures, and "1-Specificity" is plotted on the X axis (so that an X value of 0 represents perfect classification of failures, and an X value of 1 represents complete mis-classification of failures).

You can imagine that as the cutoff is varied (from 0 to 1), there will be a trade-off of the observed successes and failures that are correctly (and incorrectly) classified. That trade-off is what the ROC curve shows: as sensitivity increases, specificity must decrease (i.e. 1-specificity must increase). Each point along this ROC curve represents a different cutoff value with corresponding sensitivity and specificity values.

The area under the ROC curve (AUC) is a measure of how well the fit model correctly classifies successes/failures. This value will always be between 0 and 1, with a larger area representing a model with better classification potential. In our case, the AUC for the the ROC curve (depicted below) is 0.8889, and is listed in the results table along with the standard error of the AUC, the 95% confidence interval, and P value (null hypothesis: the AUC is 0.5). Read more about ROC curves for logistic regression 239 for even more information and some of the math involved.





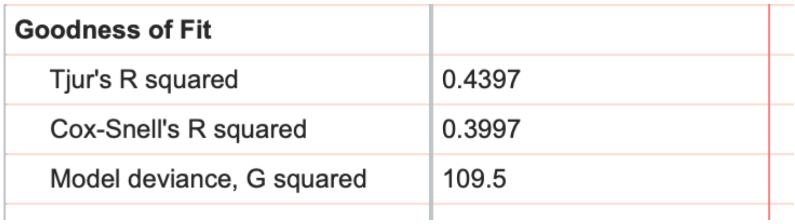
Area under the ROC curve	
Area	0.8889
Std. Error	0.02981
95% confidence interval	0.8305 to 0.9473
P value	<0.0001

Goodness of fit and additional model details

In the final section of calculated results, Prism provides some additional metrics that attempt to summarize how well the model fits the given data. The first two of these are Tjur's R squared and Cox-Snell's R squared, and while "R squared" may be in their name, these metrics simply cannot be interpreted in the same way that R squared for linear and nonlinear regression is interpreted. Instead, these values are known as pseudo R squared 166 values, provide different kinds of information about the model fit. For these metrics, the calculated value will be between 0 and 1, with higher values indicating a better fit of the model to the data.

Of the two pseudo R squared values provided, Tjur's R squared is much easier to calculate and interpret: find the average predicted probability of success for the observed successes and the average predicted probability for the observed failures. Then calculate the absolute value of the difference between these two values. That's Tjur's R squared!

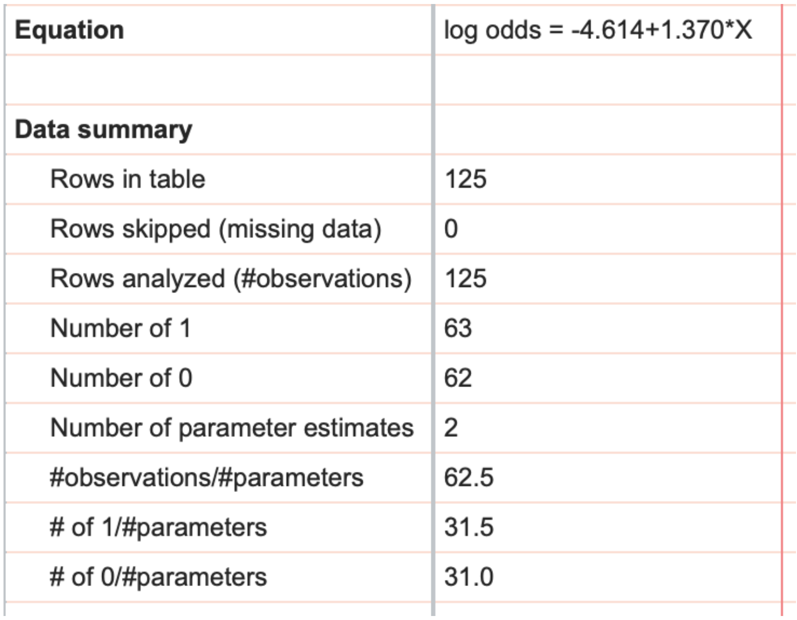
The last metric Prism reports is the model deviance. This value requires by far one of the hardest calculations 230 of the metrics that simple logistic regression reports, and so it won't be explained here. However, this metric provides a numeric estimate for "how likely" it is that the model (with the parameters given earlier in the results) would have generated the observed data. Sounds confusing, but the key here is that if you're comparing multiple models to describe the same data, a smaller value for model deviance represents a better model fit (model deviance cannot be a negative value, with a deviance of zero indicating a perfect fit of the model to the data).



Goodness of Fit	
Tjur's R squared	0.4397
Cox-Snell's R squared	0.3997
Model deviance, G squared	109.5

Equation and Data Summary

The final pieces of information that Prism provides from simple logistic regression include the model equation (given in terms of log odds 35), and a data summary that includes the number of rows in the data table, the number of rows that were skipped, and the difference of these two values providing the number of observations in the analysis. Additionally in the data summary, the total number of 1s and the total number of 0s is given. Finally, three ratios are provided: number of observations to number of parameters, number of 1s to number of parameters, and number of 0s to number of parameters (we recommend that these last two ratios should be at least 10 for logistic regression).



Equation	log odds = -4.614+1.370*X
Data summary	
Rows in table	125
Rows skipped (missing data)	0
Rows analyzed (#observations)	125
Number of 1	63
Number of 0	62
Number of parameter estimates	2
#observations/#parameters	62.5
# of 1/#parameters	31.5
# of 0/#parameters	31.0

4.1.4 Results of simple logistic regression

Logistic regression was added with Prism 8.3.0. This section of the guide will provide you with information on how to interpret simple logistic regression results generated by Prism.

Interpreting the coefficient estimates 162

X at 50% 164

Odds Ratios 165

Relating coefficients to probability 165

Hypothesis tests (P values) for β1 165

Area under the ROC curve 166

Goodness-of-fit metrics 166

Analysis checklist: Simple logistic regression 168

Error messages from simple logistic regression 169

4.1.4.1 Interpreting the coefficient estimates

Prism provides best-fit values for β0 and β1 based off of maximum likelihood estimation. Remember that the form of the logistic model can be expressed in two equivalent forms:

Log(Odds) = β0 + β1*X, or

Logit(P(Y=1)) = β0 + β1*X

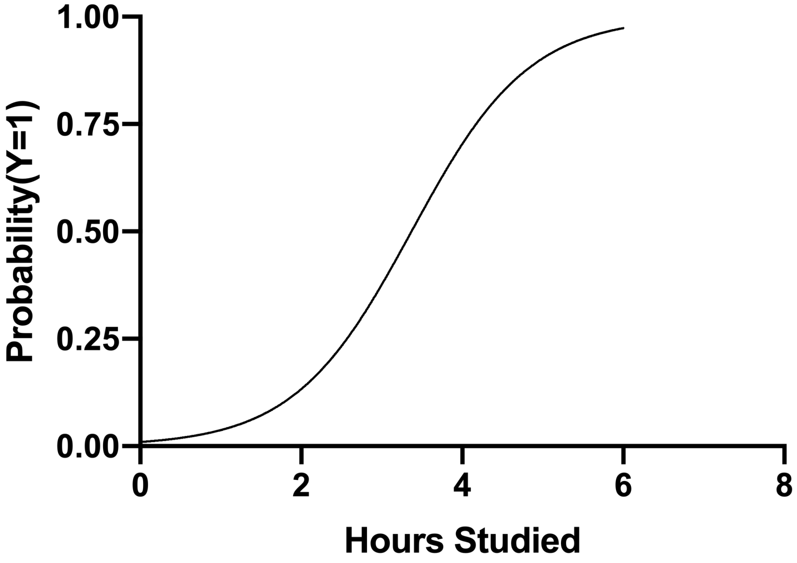
Because of the logit transformation, interpretation of the model coefficients is more difficult than with linear regression. β0 and β1 are, respectively, the slope and intercept of the model for the log odds of Y = 1. That's a mouthful, and is difficult for most people to interpret. It is correct (although not particularly helpful to say), "For a 1 unit increase in X, the log odds increases by β1." Instead, people usually interpret the coefficients using odds ratios as described in a later section 165. You can learn more about the relationship of probability, odds, and log odds here 37.

Prism reports both standard errors and (profile likelihood) confidence intervals for β0 and β1 estimates.

Often, β0 and β1 will be referred to as the "intercept" and "slope" of the simple logistic regression, respectively. Note that these terms only relate to the graph of the regression when Log(Odds) is plotted on the Y axis (see below). Typically, however, probability of Y=1 is plotted on the Y axis, generating the characteristic S-shaped logistic curve:

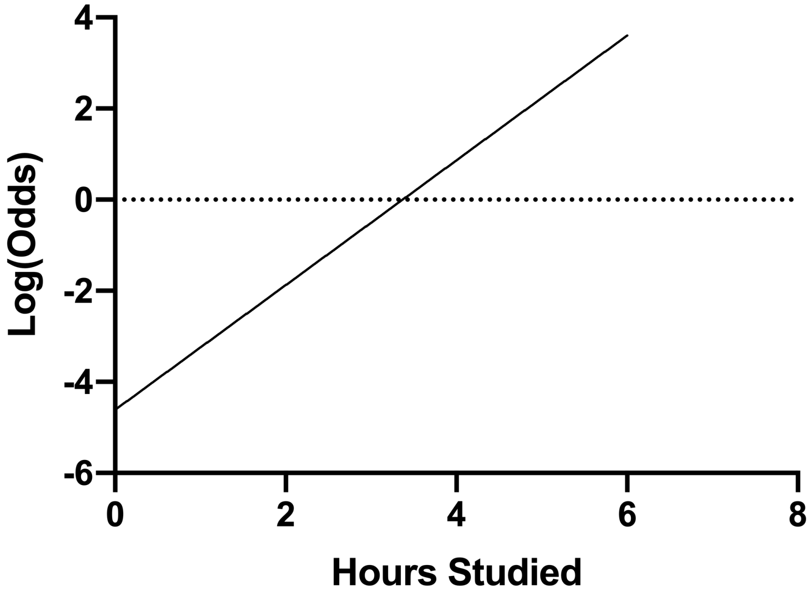
Probability Y=1 vs. X

β0 = -4.614, β1 = 1.370



Log Odds vs. X

β0 = -4.614, β1 = 1.370



4.1.4.2 X at 50%

This number that Prism reports represents the value of the X variable when the Y value of the logistic curve is 0.5. Another way to say this is that this is the value of X where the model predicts the probability of observing a success to be 0.5 (or 50%). Mathematically, because:

$P(Y=1) = 0.5$,

$P(Y=0) = 1 - P(Y=1) = 1 - 0.5 = 0.5$

Thus,

$\text{Odds} = P(Y=1)/P(Y=0) = 0.5/0.5 = 1$, and

$\text{Ln}(\text{Odds}) = \text{Ln}(1) = 0$

Using the formula for the simple logistic model,

$\text{Ln}(\text{Odds}) = 0 = \beta 0 + \beta 1 * X$, we can solve for X:

$\beta 0 + \beta 1 * X = 0$

$\beta 1 * X = -\beta 0$

$X = -\beta 0/\beta 1$

Thus, for a given model, the value that Prism reports here is equal to -β0/β1. Note that this value has some additional interpretations in special cases. For example, if X represents the amount (or concentration) of some drug or other administered treatment, and Y=1 indicates the death of a subject, then X at P(Y=1) = 0.5 is the LD50 (or lethal dose, 50%). This value represents the amount/concentration of substance required to kill 50% of the test population. This is a very common value to calculate in toxicology.

In other dose-response type relationships, this value may also be referred to as the EC50 or IC50, depending on the specifics of the experiment.

4.1.4.3 Odds Ratios

It's generally easier to think and speak in terms of the odds as opposed to the log odds (discussed previously 162). We can convert the simple logistic regression model from one that predicts the log odds to one that predicts the odds by exponentiating both sides:

Odds = $e^{\beta 0+\beta 1*X}$ = $(e^{\beta 0})$ * $(e^{\beta 1*X})$

This provides an equivalent, but slightly different interpretation for the coefficients. If, for example, $(e^{\beta 1})$ = 2, then we could say, "For a one unit increase in X, the odds increase by a multiplicative factor of 2." This is an important distinction as effects caused by increases in X multiply the odds but add to the log odds.

Note that in Prism's results output, odds ratios are reported for β0 and β1. These values are mathematically equivalent to $e^{\beta 0}$ and $e^{\beta 1}$.

4.1.4.4 Relating coefficients to probability

The effect of changes in the β0 and β1 coefficients calculated by simple logistic regression can't easily be translated to changes in probability. Why not?

The reason β0 and β1 can't easily be applied directly to an increase or decrease in probability is because the change in probability depends on the specific value of X. When the probability is near 0.5, a one unit increase in X will have a much larger effect than, for example, when the probability is near 0 or 1. If you're interested in this, you can read more about the relationships of log odds, odds, and probability 37.

4.1.4.5 Hypothesis tests (P values) for β1

As part of the results output for simple logistic regression, Prism will test if the slope of the simple logistic model (β1) is significantly different from 0, which is equivalent to whether the odds ratio is 1.0. Prism tests this null hypothesis in two ways.

- Wald test. This result is shown under the heading "Is slope significantly non-zero".

- Likelihood ratio test. This tests whether an intercept-only model (a model with only β0) fits the data better than the model containing both β0 and β1.

We include the Wald test so you can compare with other program, but recommend that you rely on the results of the likelihood test. The Wald test is an approximation of the likelihood test, and approximation that is quite good with large sample size.

4.1.4.6 Area under the ROC curve

One option Prism provides in the Classification and Interpolation section of options for simple logistic regression is to generate an ROC curve and to calculate the area under this curve (AUC). Learn more about interpreting AUC. 239 The results for the area under the ROC curve that Prism reports include:

- Area

- Standard error of the area

- 95% confidence interval of the area

- P value (testing the null hypothesis that AUC = 0.5)

4.1.4.7 Goodness-of-fit metrics

Prism offers a number of goodness-of-fit metrics that can be reported for simple logistic regression. Three of them (Tjur’s R squared, Cox-Snell’s R squared, and Model deviance) are reported in the Goodness of Fit section of the results for simple logistic regression, and are briefly discussed below. A fourth option offered in the goodness-of-fit section of the simple logistic regression dialog is the likelihood ratio test which is described here 165, and reported in its own section of the simple logistic regression results sheet.

Tjur’s R squared

Tjur’s R squared is one of a number of metrics developed for logistic regression collectively referred to as “Pseudo R squared” values. If you’re familiar with linear regression, you’ve probably encountered the concept of R squared in the past. It is critical to understand that these pseudo R squared values for logistic regression are NOT the same as R squared for linear regression (read more about R squared for linear regression).

Tjur’s R squared is one of the easier pseudo R squared values for logistic regression to understand and interpret: find the average predicted probability for all rows where Y was entered as a 0 and the average predicted probability for all rows where Y was entered as a 1, and then find the absolute value of the difference between these two values. In other words:

Tjur’s R squared = |Average Predicted value for 0s – Average Predicted value for 1s|

For good models, you would expect that the average for 0s would be close to zero and the average for 1s would be close to one. Thus, like R squared for linear regression, this will be a value between 0 and 1, and values that are closer to 1 indicate a better model fit to the data.

Cox-Snell’s R squared

Cox-Snell’s R squared is sometimes referred to as “generalized R squared”, but is just another pseudo R squared intended to provide an idea of how well a model fits the given data. The calculation of Cox-Snell’s R squared is more complicated than Tjur’s R squared, as is its interpretation. However, this is a common metric among other statistics packages, and Prism offers it for the sake of comparison to results calculated elsewhere. You can read more about how Cox-Snell’s R squared is calculated here 236.

Model Deviance

Model deviance is a metric that can be used to assess how well a given model fits to the entered data. Deviance is calculated based on another metric known as likelihood (or log likelihood). Calculation and interpretation of these values can be somewhat complicated, but for those interested, methods to calculate model deviance, likelihood, and log likelihood for logistic regression 230 are provided in the multiple logistic regression portion of this guide.

4.1.4.8 Analysis checklist: Simple logistic regression

To check that simple logistic regression is an appropriate analysis for your these data, ask yourself these questions:

✓ Is the outcome (Y) variable binary (dichotomous)?

The independent (Y) variable may only take on two values and in Prism, these must be coded as a 0 and a 1.

✓ Are the observations (rows) independent?

One of the fundamental assumptions of logistic regression is that each row of data is a unique, independent observation. An example of independent observations is a study on 100 randomly selected people where a 1 indicates a positive outcome and a 0 a negative outcome, and each person is recorded on a single row. If each person was measured more than once (say at various time points in the study), then the observations are not independent and logistic regression isn't appropriate. Independence would also be questionable if some of the participants were in the same family as almost any outcome is likely to be more similar between two individuals from one family than from two unrelated individuals.

✓ Does the model fit and predict the data well?

All models are wrong, but some are useful…

Prism offers a variety of metrics to evaluate how well the simple logistic model fits to the entered data. However, you should keep in mind that fitting models to data and interpretation of model fits is - to some extent - subjective. Some possibilities to consider when evaluating a given model include:

- Does the model classify data well? In other words, does the model correctly predict the observed 0s and 1s? You can evaluate this in Prism a number of different ways such as with Tjur’s R squared, an ROC plot (with area under the ROC curve), and the row classification table.

- Does the logistic model out-perform an intercept-only model? Prism tests this concept in two related, but slightly different ways: using the Wald test to examine if β1 is significantly non-zero, and using the likelihood ratio test to directly compare the given model with an intercept-only model.

✓ Do you have sufficient data to trust your results?

As with all stats modeling, the more data (generally) the better. At the bottom of the tabular results sheet of the analysis results, Prism reports how many observations were included in the model (Rows analyzed). For simple logistic regression, a general rule of thumb is to have at least ten observations with an outcome of zero and ten observations with an outcome of 1.

✓ Are you underfitting?

In the case of simple logistic regression, it’s possible that your predictor (X) variable is only one of multiple variables that affect if an outcome is a success or not. If the model prediction performance isn’t as good as desired, perhaps you’re missing some key variable(s) that you either didn’t measure or chose not to model. If you simply chose not to model them, you should definitely investigate their impact using multiple logistic regression - which is a natural extension of simple logistic regression. Read more about multiple logistic regression here 198. On the other hand, if the key variable is one that you didn’t measure, you’re out of luck. Go back and repeat the experiment with a focus on collecting more information. THEN come back and perform multiple logistic regression!

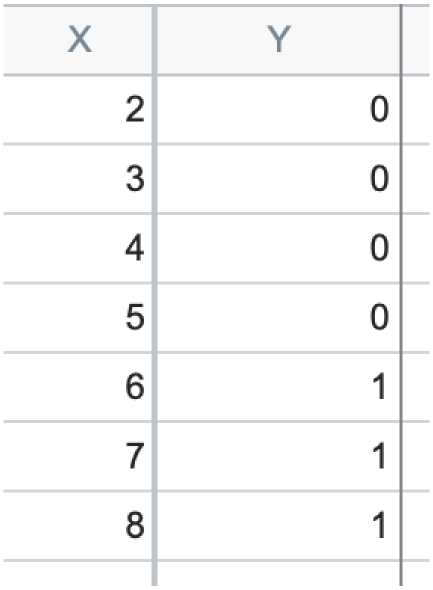
4.1.4.9 Error messages from simple logistic regression

Similar to simple linear regression, simple logistic regression attempts to find best-fit values for a set of parameters. Unlike simple linear regression, however, simple logistic regression finds best-fit values through an iterative procedure that starts with some initial values for these parameters, and works toward the best-fit values one step at a time. Because of this, for some data sets, it’s impossible for this iterative algorithm to calculate these best-fit values. When this happens, Prism provides an error message indicating that there may be an issue with “perfect separation” in the data, or that there is only a single X value.

The mathematical concepts involved in the process to determine best-fit values (and the errors that can occur during this process) are the same for both simple and multiple logistic regression. Below, very basic explanations for why this error message might show up in simple logistic regression are provided. More detailed explanations for the model fitting process and the errors that can occur are provided on the corresponding page for multiple logistic regression 247.

Perfect separation

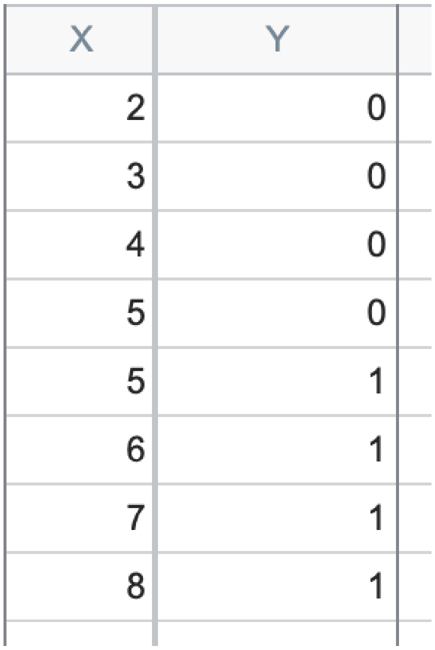
Separation is a concept that applies to data sets when the value of the X variable is a perfect predictor for the value of the Y variable. In other words, for every observation in the data set, when X is less than a certain value, Y will take on one outcome, while when X is greater than that value, Y will take on the opposite outcome. The following data represent an example of perfect separation:



X	Y
2	0
3	0
4	0
5	0
6	1
7	1
8	1

In the data set above, all Y values are 0 when X is less than or equal to 5, while all Y values are 1 when X is greater than 5. It is impossible to fit the S shaped logistic curve as the data gives no clue as what outcome to expect when X is 5.1 or 5.5 or any value between 5 and 6.

A very closely related issue occurs when data aren't "perfectly" separated, such as in this data set:



X	Y
2	0
3	0
4	0
5	0
5	1
6	1
7	1
8	1

Here, you can see that for X values less than 5, Y is 0, and for X values greater than 5, Y is 1. But at X = 5, we have observations of both Y = 0 and Y = 1. This is a special case of separation called "quasi-perfect separation".

There is only a single X value

The goal of simple logistic regression is to predict the probability of a "success" (Y = 1) given the value of the predictor (X variable). However, if all of your observations have the same value for the predictor, it's impossible to fit a model that predicts the probability of "success" as a function of that predictor.

Mathematically, this issue is caused by linear dependence between β0 and β1. This is a problem that can also arise in multiple logistic regression when the model has multiple predictor variables.

4.1.5 Deming regression

4.1.5.1 Key concepts: Deming regression

Standard linear regression 22 assumes that you know the X values perfectly, and all the uncertainty is in Y. It minimizes the sum of squares of the vertical distance of the points from the line.

If both X and Y variables are subject to error, fit linear regression using a method known as **Deming**, or Model II, regression.

If your goal is to compare two analysis methods, consider using a Bland-Altman plot instead.

4.1.5.2 How to: Deming regression

1. Create a data table

From the Welcome or New Table dialog, choose to create XY data table.

If you are just getting started, choose the sample data: Linear regression -- Compare slopes.

If you are entering your own data, choose a to enter a single Y value for each point (no replicates, no error values).

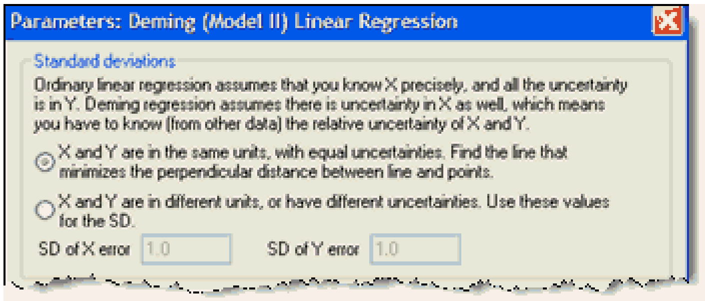
2. Enter data

If you enter Y values for several data sets (column A, B and C), Prism will report regression results for X vs. YA, for X vs. YB, and for X vs. YC.

3. Analysis choices

Click Analyze and choose Deming regression from the list of XY analyses.

Most of the dialog is self-explanatory. The choices on top are the hardest to understand, but also are the most important as the calculations depend on knowing relative magnitude of the errors in X and Y.



Parameters: Deming (Model II) Linear Regression

Standard deviations

Ordinary linear regression assumes that you know X precisely, and all the uncertainty is in Y. Deming regression assumes there is uncertainty in X as well, which means you have to know (from other data) the relative uncertainty of X and Y.

● X and Y are in the same units, with equal uncertainties. Find the line that minimizes the perpendicular distance between line and points.

○ X and Y are in different units, or have different uncertainties. Use these values for the SD.

SD of X error 1.0 SD of Y error 1.0

X and Y have equal uncertainty

If you are comparing two analytical methods, X and Y, assessed in the same units, you will probably find it reasonable to assume that both share the same standard deviation. In this case, check the first option. There is no need to estimate that standard deviation. It is enough to declare that X and Y have equal standard deviations. In this case, Deming regression minimizes the sum of the squares of the perpendicular distances of the points from the line. This is also called *orthogonal regression*.

X and Y have different uncertainties

If you are not willing to assume that X and Y have the same amount of uncertainty, then you need to enter the SD of each. Enter the SD of X values in the same units as X values are entered, and the SD of Y values in the same units as Y values are entered. Don't enter the SD of all X (or Y) values you enter. Enter the average SD (determined separately) of repeated measurements of X and Y.

How do you know what values to enter? To assess the uncertainty (error) of a method, collect duplicate measurements from a number of samples using that method. Calculate the standard deviation of the error using the equation below(1), where each d_i is the difference between two measurements of the same sample (or subject), and N is the number of measurements you made (N equals twice the number of samples, since each sample is measured twice).

$$SD_{error} = \sqrt{\frac{\sum d_i^2}{N}}$$

Compute the SD of the error for each method or variable (X and Y), enter the two SD_{error} values into the Deming regression analysis dialog, and Prism will fit the line for you. If the X variable has a much smaller SD than the Y value, the results will be almost identical to standard linear regression.

If you try to compare Prism’s results with those of another program or book, you may encounter the variable λ (lambda), which quantifies the inequality between X and Y errors.

$$\lambda=\left(\frac{SD_{X\ error}}{SD_{Y\ error}}\right)^2$$

Prism requires you to enter individual SD values, but uses these values only to calculate λ, which is then used in the Deming regression calculations. If you know λ, but not the individual SD values, enter the square root of λ as the SD of the X values, and enter 1.0 as the SD of the Y error. The calculations will be correct, since Prism uses those two values only to compute λ.

Reference

1. PW Strike, Statistical Methods in Laboratory Medicine, ISBN:0750613459. Equation 8.15.

4.1.5.3 Q&A: Deming Regression

Why doesn't Prism report R^2 with Deming Regression results?

When Prism performs Deming regression, it reports the slope and intercepts with confidence intervals, and reports a P value testing the null hypothesis that the slope is really zero. However, Prism does not report any measure of goodness-of-fit with Deming regression, and so does not report R^2. The reason is that we have been unable to find any paper or text that would explain how to compute or interpret such a value. In ordinary linear or nonlinear regression, R^2 is the fraction of the variation that is accounted for by the model. But with Deming regression, this definition doesn't really make sense, and it isn't obvious to us how to extend it.

Is Deming regression the same as orthogonal linear regression?

In Prism's Deming dialog, you specify whether X and Y are in the same units with equal uncertainties (variation). If you choose this option, Deming regression minimizes the sum of the square of the perpendicular distances of the points from the line. This is also called *orthogonal linear* *regression*. If you specify different SD values for X and Y, then Deming regression is not the same as orthogonal linear regression.

How was the equation to compute the SD from a set of duplicate measurements derived?

The previous page showed the equation for assessing the uncertainty (error) of a method from duplicate measurements from a number of samples using that method:

$$SD_{error}=\sqrt{\frac{\sum d_i^2}{N}}$$

It is easy to derive that equation. To compute the variance, you find the average square of the difference between each value and the mean. For pairs, the mean is half the difference. So take half that distance squared, but also compute it for the other value of the pair. For both values, therefore, the contribution is d_i squared. Add that up for all the measurements and divide by the number of measurements, and you have the variance. Take the square root to get the SD.

Why N, rather than N-1? I suspect it should be N-1, but that won't matter much. It will make the SD a bit smaller than it should be, but that will be true for both the X and Y SD, so the ratio will hardly be changed.

4.1.5.4 Analysis checklist: Deming regression

✓ Can the relationship between X and Y be graphed as a straight line?

In many experiments the relationship between X and Y is curved, making linear regression inappropriate. It rarely helps to transform the data to force the relationship to be linear. Better, use nonlinear curve fitting.

✓ Are the data points independent?

Whether one point is above or below the line is a matter of chance, and does not influence whether another point is above or below the line.

✓ Are the X and Y values intertwined?

If the value of X is used to calculate Y (or the value of Y is used to calculate X) then linear regression calculations are invalid. One example is a Scatchard plot, where the Y value (bound/free) is calculated in part from the X value (bound). Another example would be a graph of midterm exam scores (X) vs. total course grades(Y), since Y is in part computed from X.

✓ **Do you know the relative uncertainty of X and Y?**

Ordinary linear regression assumes that you know the X values perfectly, and all the uncertainty is in Y. Deming regression assumes there is uncertainty in both variables. You have to specify the relative uncertainties, either by specifying that X and Y are equally uncertain or by entering the SD of each. If these values are incorrect, the Deming results won't be useful.

4.2 Multiple regression with Prism

How to: Multiple regression 177

Results of multiple regression 187

How to: Multiple logistic regression 198

Results of multiple logistic regression 226

4.2.1 Getting started with multiple regression

As discussed in Principles of multiple regression 41 section, multiple linear regression, multiple logistic regression and Poisson regression are all related modeling techniques. In each of these three cases, you may have one or more independent (X) variables (also called predictor variables), and you’ll have a single dependent (Y) variable (also called a response variable or an outcome variable). Despite the similarity of the three methods, choosing which regression method to use is easy; it's based on the type of dependent (Y) variable you are modeling.

- When Y is continuous use multiple linear regression 177
- When Y is count data (0,1,2, ...) use Poisson regression 177. In Prism, this is an option within the multiple linear regression analysis.

- When Y is binary (yes/no, presence/absence, etc.) use multiple logistic regression 198

There's a natural hierarchy of complexity with these modeling types. If you're new to regression modeling, we recommend learning simple linear regression 117 first. Simple linear regression has some intuitive extensions: nonlinear regression and multiple linear regression are two of them. Multiple linear regression, in turn, has two extensions available in Prism: multiple logistic regression and Poisson regression. Finally, as described above, multiple logistic regression involves fitting a model to a set of multiple independent variables. However, if you only have one independent variable, this method can be simplified into simple logistic regression (similar to the relationship of simple linear regression and multiple linear regression). In this way, simple logistic regression can also be thought of as an extension of simple linear regression.

If this all sounds a bit overwhelming, that’s ok. You can read more about each of the multiple regression methods in the Principles of Regression (link) portion of this guide, or you can continue with any of the links below to learn more about how each method is performed within Prism.

4.2.2 How to: Multiple linear regression (and Poisson regression)

Enter data for multiple regresssion 178

Choosing a model for multiple regression 179

Comparing multiple regression models 182

Weighting in mulitple regression 183

Choosing diagnostics for mulitple regression 184

Plotting residuals from multiple regression 186

Why no stepwise regression? 45

4.2.2.1 Enter data for multiple regression

1. Create a data table

From the Welcome or New Table dialog, choose to create a multiple variable data table.

If you are just getting started, choose the sample data for multiple linear regression

2. Enter data

Each row represents a different individual, animal, experiment or something.

Each column represents a different observational unit, for example an individual, animal, or experimental replicate. All variables must be entered as numbers. Usually these will be continuous variables.

If you want to enter a categorical variable with two possible values you must encode the values (perhaps 1 for Male, 0 for Female) and can not enter category names directly.

If a categorical variable has three or more possible values, you'll have to do some extra work. The simplest approach is called dummy coding (also called indicator coding or reference coding), but there are other alternative codings such as effects coding. One good source to learn about these coding methods is Glantz and Slinker, cited below.

Note that there is no need to code interaction(s) manually. Prism will allow you to add interaction(s) automatically in the parameters dialog.

3. Run multiple regression

Click Analyze, choose multiple linear regression from the list of analyses for multiple variable tables. the multiple regression dialog has five tabs:

- Model 179. Choose which variable is the dependent variable and which other variables to include as independent variables Also choose any interactions or transforms you wish to include in the model.

- Compare 182. Choose a second model and specify how the fit of the two models should be compared.

- Weighting 183. Usually all data are weighted equally, but you can specify another weighting scheme.

- Diagnostics 184. Specify which results Prism should report.

- Residuals 186. Plot the residuals (the difference between actual and predicted Y values) in several different ways.

Reference

Glantz and Slinker, *Primer of Applied Regression and Analysis of Variance*, 3rd edition, Chapter "Using linear regression to do one-way analysis of variance with any number of treatments", page 391

4.2.2.2 Choosing a model for multiple regression

Prism currently offers three different multiple regression model frameworks: linear, Poisson, and logistic. This section describes options for linear and Poisson. For more information about how to perform multiple logistic regression, check out its section of the guide 198.

Multiple linear regression is used when Y is a continuous variable. Prism minimizes the sum-of-squares of the vertical distances between the data points and the curve. This method is often called a least squares method. This is the appropriate choice if you assume that the distribution of residuals (distances of the points from the predicted values) are Gaussian.

Poisson regression is used when every Y value is a count (0, 1, 2, ..) of objects or events. These must be the actual counts, not normalized in any way. If a machines says your sample had 98.5 radioactive decays per minute, but you asked the counter to count each sample for ten minutes, then it counted 985 radioactive decays. That is the value you should enter for Poisson regression. If the Y values are normalized counts, and are not actual counts, then you should not choose Poisson regression.

Choose dependent variable

One variable is the dependent, Y, variable and you must tell Prism which variable it is. The goal of multiple regression is to find the model that best predicts that variable.

Note that the Y variable should be continuous. If your outcome (Y) variable is binary (has only two possible values), you should use *logistic regression*[198] rather than multiple regression.

Define model

Prism requires you to specify exactly what model you want to fit. It cannot automatically choose a set of variables or interactions for you. Read why[45].

Intercept

The intercept is the value of Y when all the X values equal zero. You will almost always want to include the intercept. Only remove it from the model if you have a very strong reason, as this makes sense very rarely.

Main effects

Each main effect multiplies one parameter by a regression coefficient (parameter). You will almost always want to include all main effects in your model. If you uncheck one of the main effects, that X variable will essentially not be part of the analysis (unless that variable is part of an interaction or transform as explained below).

Two-way interactions

Each two-way interaction multiplies two parameters together, and multiplies that product by a regression coefficient (parameter). Two-way interactions are often, but not always, used in multiple regression. Why "interaction"? Because the model uses the product of two variables. Of course, two variables can interact in many ways, not just the way captured by multiplying the two variables together.

Three-way interactions

Each three-way interaction multiplies three parameters together, and multiplies that product by a regression coefficient (parameter). Three-way interactions are used less commonly than two-way interactions.

Transforms

Prism lets you use the square, the cube, or the square root of any parameter in the model. Let us know if you'd like Prism to offer other transforms when defining a multiple regression model.

Example model

In this example, column A is blood pressure in mmHg, column B is weight in kg, column C is gender coded as 0=male and 1=female. If you select column A to be the dependent (outcome variable) and include both columns B and C in the model, the resulting equation (model) is:

```
Blood pressure = Beta0 + Beta1*age +Beta2*weight +Beta3*gender + random scatte
```

Prism finds the values of the three coefficients (beta values). that that minimize the sum of the square of the differences between the Y values in your data and the Y values predicted by the equation.

The model is very simple, and it is surprising that it turns out to be so useful. For the blood pressure example, the model assumes:

- On average, blood pressure increases (or decreases) a certain amount (the best- fit value of Beta1) for every year of age. This amount is the same for men and women of all ages and all weights.

- On average, blood pressure increases (or decreases) a certain amount per pound (the best-fit value of Beta2). This amount is the same for men and women of all ages and all weights.

- On average, blood pressure differs by a certain amount between men and women (the best-fit value of Beta3). This amount is the same for people of all ages and weights.

The mathematical terms are that the model is linear and allows for no interaction. Linear means that holding other variables constant, the graph of blood pressure vs. age (or vs. weight) is a straight line. No interaction means that the slope of the blood pressure vs. age line is the same for all weights and for men and women.

If you checked the option to include the interaction between age and gender, the model would be:

```
Blood pressure = Beta0 + Beta1*age +Beta2*weight +Beta3*gender + Beta3*age*ger
```

4.2.2.3 Comparing multiple regression models

Prism lets you compare the fits of two alternative models.

Choose the second model

On the Compare tab of the multiple regression dialog, first choose the second model. In most cases, the second model will be nested within the first model. This means that the second model is simpler, maybe leaving out one independent variable or leaving out one or more interactions.

Choose a method to compare

Prism offers two approaches to comparing models with different numbers of parameters. These are not the only methods that have been developed to solve this problem, but are the most commonly used methods.

Extra sum-of-squares F test

The Extra sum-of-squares F test 82 is based on traditional statistical hypothesis testing.

The null hypothesis is that the simpler model (the one with fewer parameters) is correct. The improvement of the more complicated model is quantified as the difference in sum-of-squares. You expect some improvement just by chance, and the amount you expect by chance is determined by the number of data points and the number of parameters in each model. The F test compares the difference in sum-of-squares with the difference you would expect by chance. The result is expressed as the F ratio, from which a P value is calculated.

The P value answers this question:

> If the null hypothesis is really correct, in what fraction of experiments (the size of yours) will the difference in sum-of-squares be as large as you observed, or even larger?

If the P value is small, conclude that the simple model (the null hypothesis) is wrong, and accept the more complicated model. Usually the threshold P value is set at its traditional value of 0.05. If the P value is less than 0.05, then you reject the simpler (null) model and conclude that the more complicated model fits significantly better.

Information theory approach Akaike's criterion (AIC)

This alternative approach is based on information theory, and does not use the traditional “hypothesis testing” statistical paradigm. Therefore it does not generate a P value, does not reach conclusions about “statistical significance”, and does not “reject” any model.

The method determines how well the data supports each model, taking into account both the goodness-of-fit (sum-of-squares) and the number of parameters in the model. The results are expressed as the probability that each model is correct, with the probabilities summing to 100%. If one model is much more likely to be correct than the other (say, 1% vs. 99%), you will want to choose it. If the difference in likelihood is not very big (say, 40% vs. 60%), you will know that either model might be correct, so will want to collect more data. How the calculations work 84.

Which approach to choose?

In most cases, the models you want to compare will be 'nested'. This means that one model is a simpler case of the other. For example one model may include an interaction term, and the other model does not but is otherwise identical.

If the two models are nested, you may use either the F test or the AIC approach. The choice is usually a matter of personal preference and tradition. Basic scientists in pharmacology and physiology tend to use the F test. Scientists in fields like ecology and population biology tend to use AIC approach.

If the models are not nested, then the F test is not valid, so you should choose the information theory approach. Note that Prism does not test whether the models are nested.

4.2.2.4 Weighting in multiple regression

It is often useful to differentially weight the data points. Learn why. 57

Prism offers four choices on the Weight tab of multiple regression:

No weighting. Regression is most often done by minimizing the sum-of-squares of the vertical distances of the data from the line or curve. Points further from the curve contribute more to the sum-of-squares. Points close to the curve contribute little. This makes sense, when you expect experimental scatter to be the same, on average, in all parts of the curve.

Weight by 1/Y^2. In many experimental situations, you expect the average distance (or rather the average absolute value of the distance) of the points from the curve to be higher when Y is higher. The points with the larger scatter will have much larger sum-of-squares and thus dominate the calculations. If you expect the relative distance (residual divided by the height of the curve) to be consistent, then you should weight by 1/Y2.

Weight by 1/Y. This weighting is sometimes used when the scatter follows a Poisson distribution -- when Y represents the number of objects in a defined space or the number of events in a defined interval. Since Prism offers Poisson regression (a choice on the Model tab 179), there is little use for 1/Y weighting.

Weight by 1/Y^{K}. Also called "General weighting". Read more. 59

4.2.2.5 Choosing diagnostics for multiple regression

How precise are the best-fit values of the parameters?

The best fit value of each parameter is, of course, an estimate of the true value which you could only know if you had an infinite amount of data. The precision of each parameter can be expressed as a standard error and/or a confidence interval. Standard errors are more conventional. Confidence intervals are easier to interpret.

You can also ask Prism to test the statistical significance of each parameter. The null hypothesis is that the true population value of the parameter is 0.0. The P value answers the question: If that null hypothesis were true, what is the chance that analysis of a randomly selected data sample will result in a parameter as far, or further, from zero than reported from this analysis? Prism also reports whether the result is "statistically significant" defined as a P value less than 0.05.

Only choose to compute significance when you really wonder if the true parameter value isn't zero (so that variable has no impact on the model). This will not be the case for many parameters.

Are the parameters intertwined or redundant?

Prism can report a covariance matrix to show you how each variable is correlated with each other variable. It can also quantify multicollinearity 193 -- how well each variable can be predicted from the other variables.

How to quantify goodness of fit

If you chose least-squares regression

- Multiple R, coefficient of multiple correlation, is the correlation between the Y values and the predicted Y values. It is the square root of R2 and its value is always between 0 and 1.
- R^2 377 is the standard way to assess goodness-of-fit.
- The adjusted R 377 ^{2} 377 takes into account the number of parameters fit to the data, so has a lower value than R^2 (unless you fit only one parameter, in which case R^2 and adjusted R^2 are identical).
- The sum-of-squares (or weighted sum-of-squares) is the value that Prism minimizes when it fits the model. Reporting this value is useful only if you want to compare Prism's results to those of another program, or you want to do additional calculations by hand.
- Sy.x and RMSE 383 are alternative ways to quantify the standard deviation of the residuals. We recommend the Sy.x, which is also called Se. Sy.x divides the sum-of-squares by N-K, where N is number of rows analyzed and K is the number of parameters fit. RMSE uses N-1 in the denominator.
- The AICc 84 is useful only if you separately fit the same data to three or more models. You can then use the AICc to choose between them. But note that it only makes sense to compare AICc between fits when the only difference is the model. If the data or weighting are not identical between fits, then any comparison of AICc values would be completely meaningless.

If you chose Poisson regression

If you chose Poisson regression, Prism offers three ways to quantify the goodness of fit 387: the pseudo-R2, the dispersion index and the model deviance. The pseudo-R2 can be interpreted pretty much like an ordinary R2. The other two values will be of interest only to those who have studied Poisson regression in depth.

Are the residuals Gaussian ?

Least-squares regression assumes that the distribution of residuals follows a Gaussian distribution. Prism can test this assumption by running a normality test 375 on the residuals. Prism offers four normality tests. We recommend the D'Agostino-Pearson test.

Normality tests. Are the residuals Gaussian?

Least squares regression is based on the assumption that the data are sampled from a population where the model is correct with a Gaussian distribution of residuals (discrepancy between observed and predicted Y values). Prism can test the normality of residuals using four normality tests.

4.2.2.6 Plotting residuals from multiple regression

Prism can plot the residuals in four distinct ways:

	X axis	Y axis
Residual plot	Predicted Y value	Residual
Homoscedasticity plot	Predicted Y values	Absolute value of residual
QQ plot	Actual residual	Predicted residual if residuals are sampled from a Gaussian distribution
Residual vs. row number	Actual residual	Row number

The residual plot is used most often. For each row of data, Prism computes the predicted Y value from the regression equation and plots this on the X axis. The Y axis shows the residual. If the data follow the assumptions of multiple regression, you shouldn't see any clear trend. The size of the residuals should not be related to the predicted Y values.

The homoscedasticity plot is the same, except the Y axis shows the absolute value of the residuals. This means all the Y values are positive, showing the length of the residual. It is a little easier to look for a relationship between predicted Y and the size of the residuals when shown this way. It helps you assess whether the assumption of homoscedasticity is valid. This is the assumption that absolute value of the residual is not related to the Y values. If you see larger absolute residuals when Y is larger, then this assumption is dubious. You might want to consider unequal weighting 183 of the points.

The QQ plot graphs the actual residuals on the X axis vs. the predicted residuals on the Y axis. This prediction is based on the assumption that the residuals were sampled from a Gaussian distribution. If the assumption of normality is met, you'd expect the points on this graph to form a straight line, near the line of identity. Various programs plot QQ plots in various ways. Prism always plots the actual residual vs. predicted and doesn't show percentiles or quantiles.

Plotting residuals vs. row numbers only makes sense when the order of rows is meaningful. If you see a trend that the residuals get smaller (or larger) with increasing row number may help you make sense of your data.

If you choose unequal weighting, the residual plots all show the weighted residual, not the actual residual.

You can read more about the various kinds of residual plots in the discussion of residuals from nonlinear regression 361.

4.2.3 Results of multiple linear regression

Parameter values from multiple regression 188

P values from multiple regression 189

ANOVA table from multiple regression 190

Goodness of fit from multiple regression 190

Multicollinearity 193

Graphing the results of multiple regression 196

Analysis checklist: Multiple regression 196

4.2.3.1 Parameter values from multiple regression

Units of the parameters

The parameter Beta0 has the same units as the Y values (the outcome variable).

The other best-fit parameters have the units of the Y variable divided by the units of the corresponding X variable.

Consider this example model again.

```
  Blood pressure = Y = Beta0 + Beta1*age +Beta2*weight +Beta3*gender + random sc
```

Beta0 is expressed in units of the Y variable, which is mmHg. It is the predicted value of Y when all X variables equal zero. For this example (and many others) this is a bit silly, as it would be the average blood pressure of men (since gender is coded as zero) with age=0 and weight=0! Better just to think of it as a constant in the model.

If blood pressure is measured in mmHg and age is measured in years, the variable Beta1 will have units of mmHg/year. It is the amount by which blood pressure increases, on average, for every year increase in age, after correcting for differences in gender and weight.

If weight is measured in kg, then Beta2 has units of mmHg/kg. It is the average amount by which blood pressure increases for every kg increase in weight, adjusting for differences in age and gender.

Gender is a dummy variable with no units, coded so that males are 0 and females are 1. Therefore, Beta3 has units of mmHg. It is the average difference in blood pressure between men and women, after taking into account for differences in age and weight.

Standard errors and confidence intervals

The only way you could really know the best-fit values of the parameters in the model would be to collect an infinite amount of data. Since you can't do this, the best-fit values reported by Prism are influenced, in part, by random variability in picking subjects. Prism reports this uncertainty as a 95% confidence interval for each parameter. These take into account the number of subjects in your study, as well as the scatter of your data from the predictions of the model. If the assumptions of the analysis are true, you can be 95% sure the true best-fit value of the parameter lies within that confidence interval.

Prism also presents the standard error of each parameter in the model. These are hard to interpret, but are used to compute the 95% confidence intervals for each coefficient. Prism shows them so that its results can be compared to those of other programs.

4.2.3.2 P values from multiple regression

Prism tests the statistical significance of each parameter.

The null hypothesis is that the true population value of the parameter is 0.0. What this means is that the corresponding variable simply has no impact on the outcome variable. The P value answers the question:

> If that null hypothesis were true, what is the chance that analysis of a randomly selected data sample will result in a parameter as far, or further, from zero than reported from this analysis?

For each parameter, Prism reports:

- The absolute value of t, which is computed as the parameter value divided by its standard error. With Poisson regression, Prism reports the absolute value of z instead.

- The P value which is computed from the t ratio and the number of degrees of freedom (equal to the number of rows of data minus the number of columns of data).

- A P value summary, as ns or one or more asterisks.

4.2.3.3 ANOVA table from multiple regression

Prism provides the ANOVA table so you can compare results with other programs.

The ANOVA table starts with the variation of the outcome (Y) variable, without considering the other variables or the regression. This is calculated as the sum of the square of the difference between each Y value and the mean of all the Y values. This is reported as the total sum of squares in the bottom row of the ANOVA table.

Then it computes the sum of the square of the discrepancy (difference) between the actual Y values and the Y values predicted by the regression model. This is the sum of squares of the residuals.

The difference between these two values is the sum of squares of the regression.

The ANOVA table computes a F ratio, which is used to compute a P value. This P value tests the null hypothesis that the regression model is entirely useless so the predictions of the regression model are no better than just predicting that each Y value equals the mean of all Y values. It is only worth looking at the other regression results when this P value is small.

4.2.3.4 Goodness of fit from multiple regression

Least squares regression

Degrees of freedom. The number of degrees of freedom equals the number of rows of data analyzed (Prism skips any rows with missing or excluded values) minus the number of parameters in the model.

Multiple R. The coefficient of multiple correlation is the correlation between the Y values and the predicted Y values. It is the square root of R^2 and its value is always between 0 and 1.

$R^{2.}$ The fraction of all variance in Y that is explained by the multiple regression model. If R^2 equals 1.0, then each Y value is predicted perfectly by the model, with no random variability. If R^2 equals 0.0, then the regression model does a terrible job of predicting Y values - you'll get equally accurate predictions by simply predicting that each Y value equals the mean of the Y values you measured. With real data, of course, you won't see those extreme R^2 values, but instead will see R2 values between 0.0 and 1.0. If you computed the r^2 from linear regression on the graph of actual vs. predicted Y values, that r^2 (from linear regression) would be the same as R^2 from multiple regression.

Adjusted R^2. Even if the data are all random, you expect R^2 to get larger as you add more variables to the equation. Just by chance the model will predict the data better if it has more components. The adjusted R^2 value corrects for this, by correcting for the number of X variables in the model. Notes:

- If you collect random data, you'd expect the adjusted R^2 value to be zero on average. If you collected many sets of random data, the adjusted R^2 value will be negative half the time, and positive half the time. How is it possible for the adjusted R^2 to be negative? If the adjusted R^2 were really the square of anything, then it would always be positive. But the adjusted R^2 is not the square of anything - it is just R^2 minus a correction.

- The adjusted R^2 is mostly useful for comparing the fits of models with different numbers of independent variables. You can't compare R^2, because you expect R^2 to be smaller in the fit with more variables just by chance.

Sum-of-squares. Multiple regression finds values for coefficients in the model that minimize the sum-of-squares of the differences between the predicted Y values and the actual Y values.

Sy.x and RMSE. These are alternative ways to quantify the standard deviation of the residuals. We recommend the Sy.x, which is also called Se. Learn how these are calculated. 383

AICc. This is a value that accounts for goodness of fit and the also the number of parameters in the model. If you fit the same data with the same weighting to two models the one with the lower AICc is more likely to be the correct model. This may not be the model that fits the data better. It is easy to fit the data better by adding lots of independent variables (or interactions) and so increasing the number of parameters fit by the model. The AICc gets smaller when the model fits the data better, but gets larger when you add parameters to the model. The value of the AICc depends on the units of the dependent variable, so cannot be interpreted in any useful way as a single value. Only the difference between two AICc values can be interpreted. Learn more about AICc (and how it is computed) in our explanation of how it is used in nonlinear regression 84.

Poisson regression

Prism can compute goodness-of-fit of Poission in four ways, selectable in the Diagnostics tab.

Pseudo R-Squared

It is not possible to compute R^2 with Poisson regression models. Instead, Prism reports the **pseudo R^2.** You can interpret it as you do a regular R^2. This is the simplest goodness-of-fit measure to understand, so we recommend it.

Pseudo $R^2$is computed from log-likelihoods of three models: *LLo*, the log-likelihood of horizontal-line model; *LLfit*, the log-likelihood of the model you chose; and *LLmax*, the maximum log-likelihood possible, which would occur when the actual responses exactly equal the predicted responses so the model exactly predicts every point and all the residuals equal 0.0. The equation that computes the pseudo R^2 is:

```
   R2 = (LLfit - LLo) / (LLmax - LLo)
```

Negative log likelihood

Least squares regression minimizes the sum-of-squares, which Prism reports. Poisson regression maximizes the negative log of the likelihood, which Prism can report.

Deviance or G2

The deviance is twice the difference between the maximum possible log-likelihood (see above) and the log-likelihood of the fitted model. The formula for the deviance is D=2(*LLmax* - *LLfit*). This is also called G^2.

Dispersion ratio

When data are sampled from the Poisson distribution, the variance equals the mean. Prism can report the variance-to-mean ratio (VMR), called the *dispersion ratio*. Prism reports the degree of overdispersion with a value phi. . If phi is much greater than 1.0, then the actual variance of points around the curve is greater than the mean, and the Poisson model may not be appropriate. This is called *overdispersion*.

Some programs offer extensions to Poisson regression to deal with overdispersion, but Prism does not (let us know if you need this).

AICc

The AICc 84 is useful only if you separately fit the same data to two or more models. You can then use the AICc to choose between them. But note that it only makes sense to compare AICc between fits when the only difference is the model you chose. If the data are not identical between fits, then any comparison of AICc values would be meaningless. It is also essential that the weighting, or regression method, be the same for all the fits. If you use Poisson regression for one fit, you need to use it for all.

4.2.3.5 Multicollinearity

What is multicollinearity?

It is important to understand the concept of multicollinearity, as it can interfere with proper interpretation of multiple regression results.

To understand multicollinearity, first consider an absurd example. Imagine that you are running multiple regression to predict blood pressure from age and weight. Now imagine that you've entered weight-in-pounds and weight-in-kilograms as two separate X variables. The two X variables measure exactly the same thing - the only difference is that the two variables have different units. The P value for the overall fit is likely to be low, telling you that blood pressure is linearly related to age and weight. Then you'd look at the individual P values. The P value for weight-in-pounds would be very high - after including the other variables in the equation, this one adds no new information. Since the equation has already taken into account the effect of weight-in-kilograms on blood pressure, adding the variable weight-in-pounds to the equation adds nothing. But the P value for weight-in-kilograms would also be high for the same reason. After you include weight-in-pounds to the model, the goodness-of-fit is not improved by including the variable weight-in-kilograms. When you see these results, you might mistakenly conclude that weight does not influence blood pressure at all since both weight variables have very high P values. The problem is that the P values only assess the incremental effect of each variable. In this example, neither variable has any incremental effect on the model. The two variables are collinear.

That example is a bit absurd, since the two variables are identical except for units. The blood pressure example -- model blood pressure as a function of age, weight and gender - is more typical. It is hard to separate the effects of age and weight, if the older subjects tend to weigh more than the younger subjects. It is hard to separate the effects of weight and gender if the men weigh more than the women. Since the X variables are intertwined, multicollinearity will make it difficult to interpret the multiple regression results.

Quantifying multicollinearity

Multicollinearity is an intrinsic problem of multiple regression, and it can frustrate your ability to make sense of the data. All Prism can do is warn you about the problem. It does this by asking how well each independent (X) variable can be predicted from the other X variables (ignoring the Y variable), expressing the results in two ways:

- R^2 with other X variables. The fraction of the variance in one X variable that can be predicted from the other X variables. The Y variable is not part of the calculation.

- Variance Inflation Factor (VIF). If the X variables contain no redundant information, you expect VIF to equal one. If the X variables are collinear (contain redundant information), then VIF will be greater than one. VIF is related to R^2 by this equation: $VIF=1/(1-R^2)$.

Some programs also compute Tolerance, but Prism does not. You can easily calculate it yourself for each variable as $1.0 - R^2$.

When multicollinearity is high

If R^2 and VIF are high for some X variables, then multicollinearity is a problem in your data. How high is high? Any threshold is arbitrary, but here is one rule of thumb. If any of the R^2 values are greater than 0.75 (so VIF is greater than 4.0), suspect that multicollinearity might be a problem. If any of the R^2 values are greater than 0.90 (so VIF is greater than 10) then conclude that multicollinearity is a serious problem.

If multicollinearity is a big problem, the results of multiple regression are unlikely to be helpful. Possible methods to resolve the issue:

1. Make sure you didn't include redundant information. Say your study includes both men and women so you have one independent variable "Woman" that is 1 for females and 0 for males, and another variable "Man" that is 0 for females and 1 for males. You've introduced collinearity because the two variables encode the same information. Only one variable is needed.

2. Combine variables. An example of correlated variables would be including both weight and height in a model, as people who are taller also tend to weigh more. One way around this issue would be to compute the body mass index (BMI) from height and weight, and only include that single variable in the mode, rather than including both height and weight.

3. In some cases, removing one or more variables from the model will reduce multicollinearity to an acceptable level.

4. In other cases, you may be able to reduce multicollinearity by collecting data over a wider range of experimental conditions.

This is a difficult problem, and you may need to seek statistical guidance elsewhere.

Notes

- Don't confuse these individual R^2 values for each X variable with the overall R^2. The individual R^2 values quantify how well each X variable can be predicted from the other X variables. The overall R^2 quantifies goodness-of-fit of the entire multiple regression model. Generally you want the overall R2 value to be high (good fit) while all the individual R^2 values to be low (little multicollinearity).

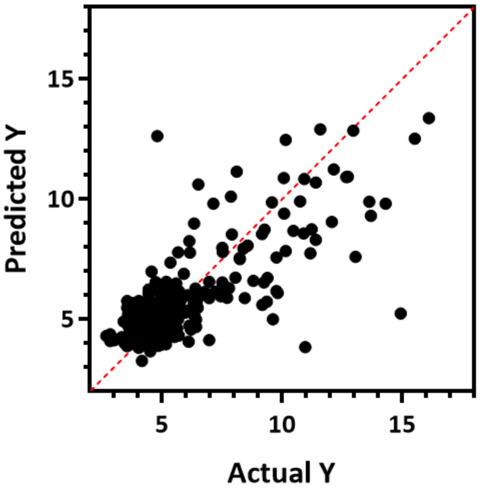
- Multicollinearity increases the width of the confidence interval (which is proportional to the square root of variance) by a factor equal to the square root of VIF. If a variable has a VIF of 9, the confidence interval of that coefficient is three times wider than it would be were it not for multicollinearity.

- When you only have two independent variables, the problem is called *collineariaty*. With three or more, the term *multicollinearity* is used.

4.2.3.6 Graphing the results of multiple regression

Prism automatically graphs the actual Y values (on the X axis) vs. the predicted Y values (from the regression; on the Y axis). If the model is useful, the data should cluster around the line of identity, which is shown. This graph gives you a visual sense for how well the multiple regression model explains your data. These kinds of graphs are not often shown, but they should be!

In this example, some of the predictions of the model are far from the actual data, suggesting that the model fit here is incomplete.



The r^2 from linear regression of this graph is the same as the R^2 from the multiple regression. (By tradition, a lower case r is used with linear regression and an upper case R with multiple regression).

Prism can also plot residuals 186 in up to four different ways.

4.2.3.7 Analysis checklist: Multiple regression

To check that multiple regression is an appropriate analysis for these data, ask yourself these questions.

✓ **Is the relationship between each X variable and Y linear?**

In many experiments, the relationship between X and Y is nonlinear, making multiple regression inappropriate. In some cases you may be able to transform one or more X variables to create a linear relationship. You may also be able to restrict your data to a limited range of X variables, where the relationship is close to linear. Some programs (but none currently from GraphPad Software) can perform nonlinear regression with multiple independent variables.

✓ Are the residuals (discrepancy between actual and predicted Y values) Gaussian (at least approximately)?

Multiple regression assumes that the distribution of residuals is random and Gaussian.

✓ Is the variability the same everywhere?

Multiple regression assumes that scatter of data from the predictions of the model has the same standard deviation for all values of X. The assumption is violated if the points with higher (or lower) X values also tend to be further from the best-fit line. The assumption that the standard deviation is the same everywhere is termed homoscedasticity. Prism offers unequal weighting, but then the assumption is that the weighted residuals are, on average, the same everywhere.

✓ Do you know the X values precisely?

The regression model assumes that all the X values are exactly correct, and that experimental error or biological variability only affects the Y values. This is rarely the case, but it is sufficient to assume that any imprecision in measuring X is very small compared to the variability in Y.

✓ Are the data points independent?

Whether one value is higher or lower than the regression model predicts should be random, and should not influence whether another point is above or below the line.

✓ Are you overfitting?

The goal of regression, as in all of statistics, is to analyze data from a sample and make valid inferences about the overall population. That goal cannot always be met using multiple regression techniques. It is too easy to reach conclusions that apply to the fit of the sample data but are not really true in the population. When the study is repeated, the conclusions will not be reproducible.This problem is called *overfitting*. It happens when you ask more questions than the data can answer— when you have too many independent variables in the model compared to the number of subjects. How many independent variables is too many? For multiple regression, a rule of thumb is to have at least 10–20 subjects (cases; rows in Prism) per independent variable (column in Prism). Fitting a model with five independent variables thus requires about 50 to 100 subjects or cases. That is a rule of thumb, not a strict criterion.

Are you *really* overfitting?

If you have fewer cases than variables, your analysis is almost certainly meaningless.

4.2.4 How to: Multiple logistic regression

Logistic regression was added with Prism 8.3.0. This section of the guide will provide you with information on how to perform multiple logistic regression with Prism.

4.2.4.1 Overview of multiple logistic regression

Multiple logistic regression is an extension of multiple linear regression. Logistic regression is used to model a dependent variable with binary responses such as yes/no or presence/absence. The goal of logistic regression is to perform predictions or inference on the probability of observing a 0 or a 1 given a set of X values. In the following, we write the probability of Y = 1 as P(Y=1).

Logistic regression fits a linear regression model to the log odds. The odds are defined mathematically as P(Y=1) / P(Y=0). People have often seen odds used in reference to betting. For example, 3 to 1 odds is another way of saying that the P(Y=1) is 0.75. The log odds are then just the natural log (Ln) of the odds.

Why not just fit a multiple linear regression (MLR) instead of a multiple logistic regression? There are two main reasons:

1. The standard statistical tests for MLR (goodness of fit, parameter estimates, standard errors, etc.) assume a Gaussian distribution for the residuals. This assumption is violated.

2. For interpretability, it’s preferable to talk about the probability of observing a “success” or a “failure.” This can be done with logistic regression, but not with linear regression.

So why do we model the log odds?

It's essentially a mathematical technicality that lets us use much of the framework developed for MLR models. While probability is constrained between 0 and 1, odds can be any positive number and the range for log odds, then, is any real number. Thus, we can use a familiar form of our model equation: log odds (Y) = β0 + β1*X1 + β2*X2 + ... We can then back transform estimates as needed to provide inference about odds or predictions for probabilities.

4.2.4.2 Entering data for multiple logistic regression

1. Create a data table

From the Welcome or New Table dialog, choose to create a multiple variables data table. If you are just getting started, you can choose to use the sample data for multiple logistic regression

2. Enter data

Each row represents a different observational unit, for example an individual, animal, or experimental replicate

Each column represents a different variable. All variables must be entered as numbers. Usually these will be continuous variables. Note that the exception is that for multiple logistic regression, the dependent (Y) variable must contain values of only 0 or 1 (read more about why).

Other variables (predictor variables) can also be categorical. If a categorical variable only has two possible values, we recommend that you encode the values using a 1 or 0, the same approach used when encoding the dependent variable. For example, if studying mice, you may want to include the mouse sex as a variable, and could enter "0" for male and "1" for female.

If a categorical variable has three or more possible values, you'll have to do some extra work. The simplest approach is called dummy coding (also called indicator coding or reference coding), but there are other alternative methods that could be used such as effects coding. One good source to learn about these coding methods is Glantz and Slinker, cited below.

Note that there is no need to code interactions manually. Prism will allow you to add interactions automatically in the parameters dialog

3. Run the multiple logistic regression

Click Analyze, and then choose multiple logistic regression from the list of analyses for multiple variables tables. The multiple logistic regression dialog has five tabs:

- Model 201. Choose which variable is the dependent (Y) variable. Add labels for the dependent variable outcomes 1 and 0. Specify which other variables to include as dependent variables, and choose any interactions or transforms you wish to include in the model.

- Compare 203. Choose a second model and specify how the fit of the two models should be compared.

- Options 205. Specify which results Prism should report (note, additional results options on Goodness-of-fit tab).

- Goodness-of-fit 206. Select from a range of various metrics that provide some insight into how well the model fits the entered data.

- Graphs 209. Select which visualizations you would like Prism to generate from the fit.

Reference

Glantz and Slinker, Primer of Applied Regression and Analysis of Variance, 3rd edition, Chapter “Using linear regression to do one-way analysis of variance with any number of treatments”, page 391

4.2.4.3 Choosing a model for multiple logistic regression

Multiple logistic regression is used when the dependent (Y) variable is dichotomous (yes/no, success/fail, etc.). Y values must all be coded as 0 or 1. Multiple logistic regression will provide a fit for the probability that Y equals 1 given particular values of the one or more X variables.

Choose dependent variable

Select the appropriate column from the drop down menu that contains your dependent (Y) variable. Note that all of the values in this column must be 0 or 1.

Label 0 and 1 outcomes (Optional)

To make the results easier to interpret, text labels (such as "presence" and "absence", “yes” and “no”, or “alive” and “dead”) may be added here. These text labels will be used in the results output of the model fit.

Define model

Other than the restriction on the Y values, data gets entered in the same way as it does for multiple linear regression (link to enter data for multiple regression). Prism makes it easy to choose which variables to include in the model including two- and three-way interactions and transformations. However, Prism cannot automatically choose a set of variables or interactions for you. Read why 45.

Intercept

For logistic regression, the intercept is the expected log odds when all of the X values are zero. If that doesn't provide you with much intuitive value, you're not alone. This value is much less interpretable than with linear regression due to the log odds transformation of the dependent variable used in logistic regression. Perhaps the easiest way to understand it is to think in terms of the logistic, S-curve that can be plotted from simple logistic regression 37. The intercept term can provide information on the value of that S-curve when X is zero. Using some simple algebra, it can be shown that the value of this S-curve at X=0 is given by:

Probability when X is 0 = $e^{\beta0}/(1+e^{\beta0})$

Another way to think of this is that the intercept term allows you to determine the probability of success in the absence of (or accounting for) your predictor variable(s).

One very important thing to keep in mind about the intercept of a logistic regression model is that **there’s almost never a reason to exclude it**. As discussed above, the value of the intercept ($\beta0$) of a logistic regression model tells you what the predicted log odds are when all of the predictor variables (X variables) are equal to zero. If we convert this information from a log odds scale to a probability scale, the intercept of the model provides the information for the probability of observing a “success” (Y=1) after accounting for all other variables. However, by excluding the intercept from the model, you’re making the assumption that the probability of observing a “success” when all X variables are zero is equal to 0.5 (or 50%). You can read more about the math involved here 253 if you’re interested. But the bottom line is that it’s quite uncommon that you would be able to assume (or that you’d want to assume) that the probability of a “success” in the absence of predictors is 0.5. Therefore, you’ll almost always want to include the intercept term in your model.

Main Effects

Use the checkboxes to select the desired X variables to include in the model. When fitting the model, Prism will determine a regression coefficient for each of the selected main effects (along with each of the interactions and transforms selected). By default, all of the main effects are selected. If you uncheck one of the main effects, that X variable will essentially not be part of the analysis (unless that variable is part of an interaction or transform as explained below).

Interactions

Prism makes it easy to include two- and three-way interactions of the independent variables in the model. A two-way interaction multiplies two variables together to create a new variable for which the model will determine a regression coefficient. Similarly three-way interactions multiply together three variables. Three-way interactions are used less commonly than two-way interactions.

Transforms

Prism lets you use the square, the cube, or the square root of any variable in the model. Let us know if you’d like Prism to offer other transforms when defining multiple regression models.

4.2.4.4 Comparing multiple logistic regression models

Comparing models works similarly to multiple linear regression 182.

Choose the second model

On the Compare tab of the multiple logistic regression dialog, first specify the main effects, interactions, and transforms for the second model. In many cases, the second model will be nested within the first model (i.e. it will use a subset of the effects, interactions, and transforms of the first). If this is the case, the second model will be a “simpler” model.

Choose a method to compare

For logistic regression, Prism offers two approaches for comparing models: Akaike's criterion (AICc) and Likelihood Ratio Tests (LRT).

Akaike’s Information Criterion (AICc)

AICc is an information theory approach using a corrected version of Akaike's criterion. This method determines how well the data supports each model, taking into account the model deviance for each model. The option to report model deviance for the selected model is also available on the Goodness-of-fit tab 206 of the multiple logistic regression dialog. The results are expressed as the probability that each model is correct, with the probabilities summing to 100%. Obviously, this method does not consider the possibility that a different model is correct. It just compares the two models it is asked to compare. If one model is much more likely to be correct than the other (say 1% vs. 99%), you will want to choose it. However, if the difference is not very big (say 40% vs. 60%), you wouldn’t know for sure which model was better, and you’d want to collect more data. Read more about how these calculations work. Importantly, AICc can be used to compare any two models on the same data set.

This test reports the difference of AICc values calculated for each of the selected models. More information on how AICc is calculated for logistic regression is given here 230.

Likelihood Ratio Test (LRT)

Similar to the AICc, the LRT also uses the model deviance to determine which model is preferred. However, unlike AICc, LRTs are only appropriate for testing when one model is a reduced version of the other. Another way to describe this scenario is to say that the two models are “nested”. Although this test is only valid when the models are nested, Prism will not check if the models are nested. Therefore, you must be careful when selecting to compare two models using this test.

The test statistic is calculated as the difference between the deviance of the simpler model (the model with fewer parameters) and the more complex model (the model with more parameters):

LRT statistic = Deviance(simpler model) - Deviance(more complex model)

Adding parameters to a model (very nearly) always decreases the model deviance. Thus, the test statistic is how much smaller the deviance is for the more complex model. The value of this statistic is used to calculate a P value. A small P value suggests rejecting the null hypothesis that the simpler model is correct. In Prism, you can specify how small the P value must be to reject this null hypothesis (with a default of 0.05).

4.2.4.5 Options for multiple logistic regression

How precise are the best-fit values of the parameters?

Many of the same ideas for testing best-fit values apply from multiple linear regression 184. Prism will optionally report standard errors, confidence intervals and P values for each secoefficient estimate. These values can be used to assess how stable the coefficient estimates are. Large standard errors, which subsequently mean large confidence intervals, imply that there is considerable uncertainty with the point estimates. The P values provide an assessment for whether the true value of the coefficient is equal to zero 230.

Are the variables intertwined or redundant?

Prism offers two ways to evaluate the linear dependence of predictors in multiple logistic regression. You may evaluate multicollinearity using variance inflation factors or evaluate pairwise correlation with the correlation matrix. See here for details 193.

Comparative model diagnostics

Selecting these options provides the raw output for the corrected Akaike Information Criterion (AICc), log-likelihood, or the model deviance.

Prism has easier ways to compare models in two special cases. First, if you wish to compare your model to a model with only an intercept, then the easiest way to do this in Prism is to run hypothesis tests in the Goodness-of-fit tab 206.

Second, if you wish to compare two different logistic regression models, you can use the Compare tab 203.

If neither of those options meet your need, and you just want the raw numbers, then select the desired box in this section.

4.2.4.6 Goodness of fit from multiple logistic regression

Goodness-of-fit metrics for multiple logistic regression can be grouped into three main categories. Explore each of these categories below for more information on the metrics found within each.

- Classification methods 206: area under the ROC curve, classification table, and row classification
- Pseudo R squared values 207: Tjur's R squared, McFadden's R squared, Cox-Snell's R squared, and Nagelkerke's R squared
- Hypothesis tests 208: Hosmer-Lemeshow test, Likelihood ratio test (LRT)

4.2.4.6.1 Classification methods for multiple logistic regression

A reasonable question to ask when evaluating a model might be, “How well does the model work for classifying the 0s and 1s observed in the data?”

Logistic regression computes the probability of receiving a "positive" result (encoded in the data table as a 1). To use logistic regression to predict if a new observation is “positive” or “negative”, specify a cutoff value that specifies the minimum probability that would be considered a “positive”. The standard cutoff is 0.5, which means that if the predicted probability is greater than 0.5, that observation is classified as a “positive” (or simply as a 1). Now that we understand what cutoff values tell us, let’s look at the three classification methods Prism offers.

Area under the ROC curve

Area under the ROC curve (AUC) provides an aggregate value of how well the model correctly classifies the 0s and 1s with all possible cutoff values. AUC values range between 0.5 and 1, where an area of 0.5 means that the model predicts which outcomes will be 1 or 0 no better than flipping a coin, and an area of 1 means that the model predicts perfectly. Provided are some examples of various extremes for ROC curves from logistic regression 239.

Classification table

The classification table reports a 2x2 table that displays the numbers of correctly classified values at the user-specified cutoff. This table has four entries that report the number of observed 0s (and 1s) that were correctly (and incorrectly) predicted. Additionally, the classification table will provide information on total number of observed 1s and 0s, total number of predicted 1s and 0s, the percent of correctly classified 1s and 0s, the percent of total correctly classified observations, and the positive and negative predictive power. Details 233.

Row classification

Row Classification generates an additional table containing two columns. The first column contains a copy of the values in the selected independent (Y) variable column as found in the data table. The second column contains predicted probabilities generated by the model corresponding to each entry (each row) in the first column.

4.2.4.6.2 Pseudo R squared values for multiple logistic regression

R squared 377 is a useful metric for multiple linear regression, but does not have the same meaning in logistic regression. Statisticians have come up with a variety of analogues of R squared for multiple logistic regression that they refer to collectively as “pseudo R squared”. These do not have the same interpretation, in that they are not simply the proportion of variance explained by the model.

In fact, the magnitude of any particular pseudo R squared value can't be used to compare across datasets. Instead, the primary use for these pseudo R squared values is for comparing multiple models fit to the same dataset. There is not a clear "best" pseudo R squared, and Prism offers four different ones from which to choose. More information on how these pseudo R squared values are calculated 236 is provided.

Tjur’s R squared

This is one of the simplest pseudo R squared values to explain. Find the predicted probability from the model for every entered value of the dependent variable. For each category of the dependent variable (0 and 1), find the average predicted probability. Then, find the absolute value of the difference between these two averages. That value is Tjur’s R squared.

McFadden’s R squared

This value uses the log-likelihood of the specified model and a corresponding “intercept-only” model (values that Prism can report 230) and determines their ratio. This ratio is then subtracted from 1 to determine the reported value. A small ratio (and thus a final value close to 1) indicates that the specified model is better than an intercept-only model.

Cox-Snell’s R squared (“generalized” R squared)

Cox-Snell’s R squared uses the likelihood (as opposed to the log-likelihood), so some additional mathematical manipulation would be required to calculate this value. Unlike other pseudo R squared values here, the maximum of Cox-Snell’s R squared is less than 1. However, this value is commonly reported by other software, and so is an option here.

Nagelkerke’s R squared

This pseudo R squared is very similar to Cox-Snell’s R squared. The primary difference is that it adjusts Cox-Snell’s R squared to have a maximum value of 1. This is done by dividing the Cox-Snell’s R squared by its maximum possible value.

4.2.4.6.3 Hypothesis tests for multiple logistic regression

Prism offers two standard Goodness-of-fit tests, Hosmer-Lemeshow and likelihood ratio test (LRT). You should be cautious when utilizing these tests, as they do not test the same null hypotheses,and the P values need to be interpreted very differently. Each test is briefly explained below, while some additional information is provided in the results interpretation section 237 of this guide.

Hosmer-Lemeshow test

This test uses the null hypothesis that the specified model is correct. As such, a small P value would suggest that the model is incomplete.

Perhaps you would need to add additional independent variables or maybe interactions. This test is commonly included in other software, so we provide as an option. However, there has been much criticism regarding this test, and it is not recommended. Prism does the test as it was initially described, dividing the data into 10 groups by predicted outcome. The value 10 is arbitrary, and some programs (not Prism) let you choose another value (which can change the P value a lot).

Likelihood Ratio Test (LRT)

This null hypothesis is that an intercept-only model is correct. In other words, the null hypothesis is that none of the independent variables provides any information to help predict the outcome. As such, a small P value would suggest that an intercept-only model is insufficient. As implied by the name, this test uses the ratio of the log-likelihood values calculated for the specified model and the intercept only model.

4.2.4.7 Graphs for multiple logistic regression

Prism offers four ways to graph results of logistic regression.

Name	X axis	Y axis
Actual vs Predicted	Actual (Entered) Y	Predicted probability
ROC Curve	1 - Specificity	Sensitivity
Logistic Plot	Actual (Entered) X	Actual (Entered) Y
Proportion Correct vs Cutoff	Cutoff value	Proportion classified correctly

Actual vs. Predicted

The Actual vs Predicted plot generates a violin plot with two groups: one group contains the observations entered in the data table with a dependent (Y) value of 0, and the other with a Y value of 1. For each of these groups, Prism plots the corresponding predicted values (from the model) as a violin plot.

ROC Curve

The ROC curve evaluates classification performance. As is typical with ROC curves, the values are plotted on axes of Sensitivity and 1-Specificity for every possible cutoff value, where (for any given cutoff value):

- Sensitivity is the number of values entered as 1 and correctly classified as 1, divided by the total number of values entered as 1

- Specificity is the number of values entered as 0 and correctly classified as 0, divided by the total number of values entered as 0

Logistic Plot

The Logistic plot option is only available when a single predictor (X) variable is included in the model. This generates the typical S-Curve (or part of an S-Curve) that represents the predicted probability of Y=1 at the given X value. If two models are being compared (via the Compare tab 203), this option will be available only if both models contain only one predictor (main effect) in addition to an intercept term.

Proportion Correct vs. Cutoff

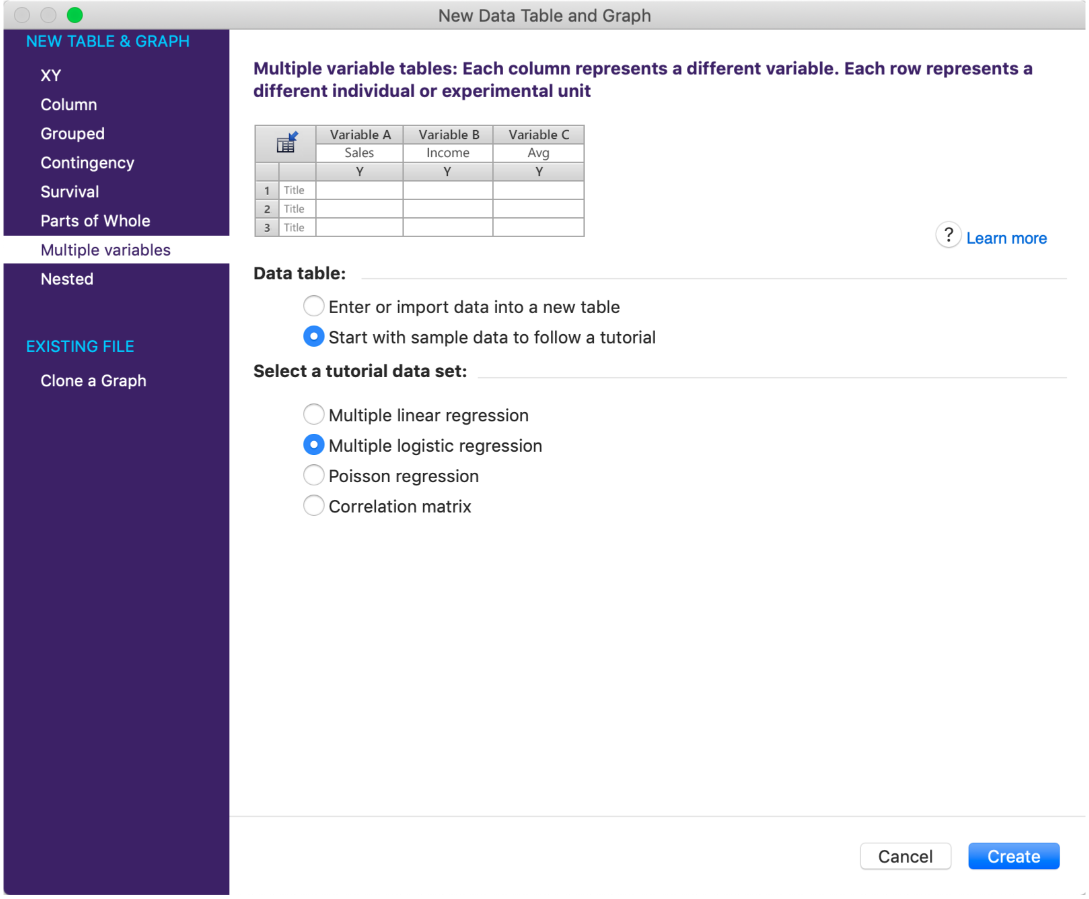
The Proportion Correct vs Cutoff graph is an alternative to looking at the ROC curve. Similar to the ROC curve, it investigates every possible cutoff value (the X axis for this graph), and plots the corresponding proportion of observations that were correctly classified for that cutoff.

4.2.4.8 Example: Multiple logistic regression

This guide will walk you through the process of performing multiple logistic regression with Prism. Logistic regression was added with Prism 8.3.0

The data

To begin, we'll want to create a new Multiple variables data table from the Welcome dialog



Multiple variable tables: Each column represents a different variable. Each row represents a different individual or experimental unit

		Variable A	Variable B	Variable C
		Sales	Income	Avg
		Y	Y	Y
1	Title			
2	Title			
3	Title			

Data table:

○ Enter or import data into a new table

● Start with sample data to follow a tutorial

Select a tutorial data set:

○ Multiple linear regression

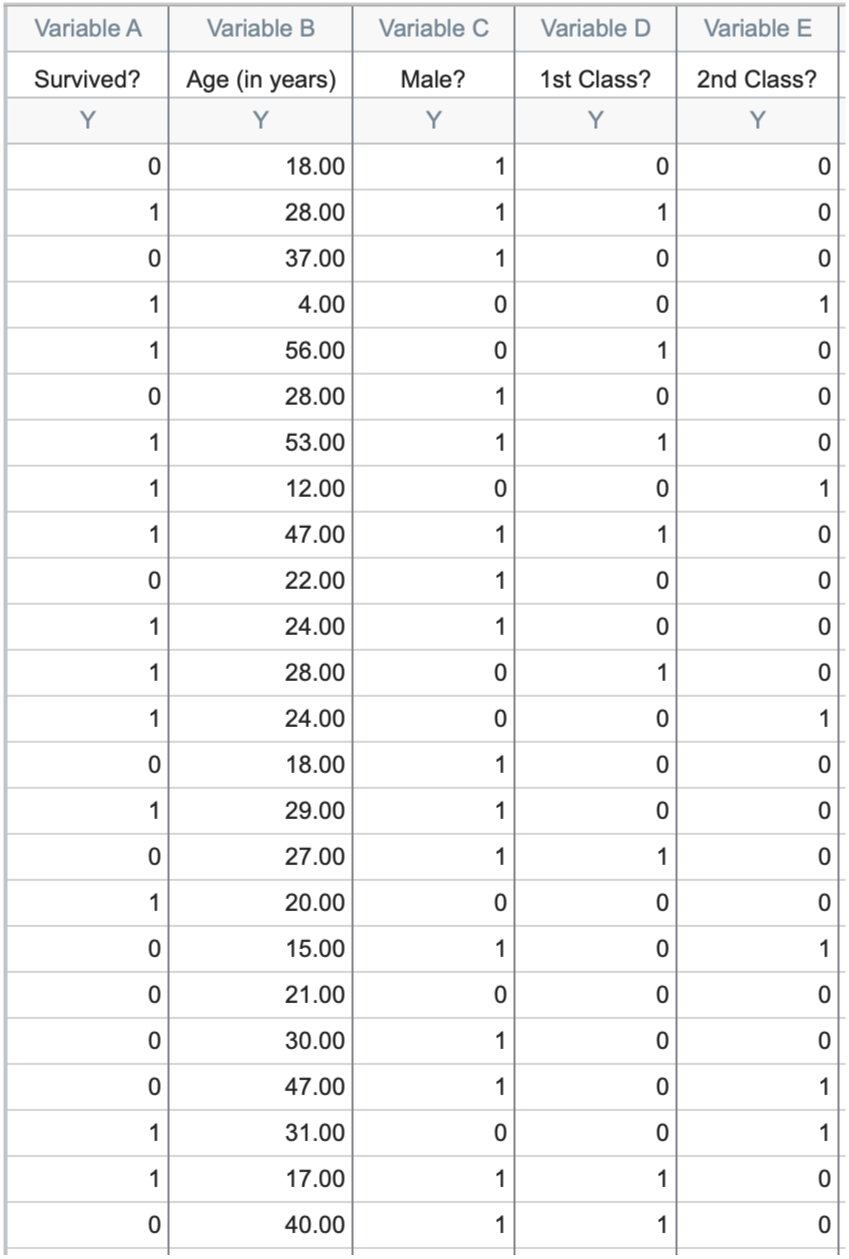
● Multiple logistic regression

○ Poisson regression

○ Correlation matrix

Choose the Multiple logistic regression sample data found in the list of tutorial data sets for the multiple variables data table.

The sample data has five total columns: "Survived?", "Age (in years)", "Male?", "1st Class?", and "2nd Class?".



Variable A	Variable B	Variable C	Variable D	Variable E
Survived?	Age (in years)	Male?	1st Class?	2nd Class?
Y	Y	Y	Y	Y
0	18.00	1	0	0
1	28.00	1	1	0
0	37.00	1	0	0
1	4.00	0	0	1
1	56.00	0	1	0
0	28.00	1	0	0
1	53.00	1	1	0
1	12.00	0	0	1
1	47.00	1	1	0
0	22.00	1	0	0
1	24.00	1	0	0
1	28.00	0	1	0
1	24.00	0	0	1
0	18.00	1	0	0
1	29.00	1	0	0
0	27.00	1	1	0
1	20.00	0	0	0
0	15.00	1	0	1
0	21.00	0	0	0
0	30.00	1	0	0
0	47.00	1	0	1
1	31.00	0	0	1
1	17.00	1	1	0
0	40.00	1	1	0

These data represent 1,314 passengers on board the RMS Titanic, with each row representing a different passenger (note that the crew is not included in this data set, and that the official number of passengers varies among available sources). The first column "Survived?" provides the fate for each passenger, with a 1 indicating that the passenger survived and a 0 indicating that the passenger died when the ship sank. The other four columns provide demographic information and the ticketed cabin for each passenger. Note that three of these four columns are also coded (with all values in these columns being either a 1 or 0). A brief explanation of how this coding works is given below. To get straight to the analysis, skip to "Initiating the analysis 213"

What about 3rd Class? (A quick intro to dummy coding)

As mentioned above, a number of the variables in this data set are coded as either a 1 or 0. In each case, the column titles have been entered as a question, and the presence of a 1 for a given observation means that the answer to the question is "yes", while the presence of a 0 for an observation means that the answer to the question is "no". For example, we can discern some information about the passenger in the first row in the table above. The 0 in the first column means that the passenger did not survive, they were 18 years old, and the 1 in the third column indicates that the passenger was male.

But what ticket class was this passenger? We see in the fourth column that the passenger wasn't 1st class, and we see in the fifth column that the passenger wasn't 2nd class (with 0s in both of these columns). Thus, we can conclude that this passenger must have been 3rd class (because he wasn't 1st or 2nd class, and those are the only possibilities).

When a variable is categorical (survived/died, male/female, 1st/2nd/3rd class, etc.), we can encode those responses as a set of 0s and 1s in a process called dummy coding (there are other coding techniques, but dummy coding is probably the easiest to understand). Using dummy coding, variables with two outcomes (survived/died, male/female, etc.) end up getting coded using one column. You don't need one column for "Male?" and one column for "Female?" because you can get all of the information from a single column. Similarly, variables with three outcomes (first class/second class/third class) end up getting coded using two columns. In this case, we don't need a column for "3rd Class?" because we can discern that information from the values in the other two columns. In fact, if you DID try to include a column for "3rd Class" in the logistic regression model, the analysis would fail because of linear dependence of the variables 251.

Initiating the analysis

Click on the multiple logistic regression button in the toolbar (shown below), or click on the "Analyze" button in the toolbar, and then select "Multiple logistic regression" from the list of available Multiple variable analyses.

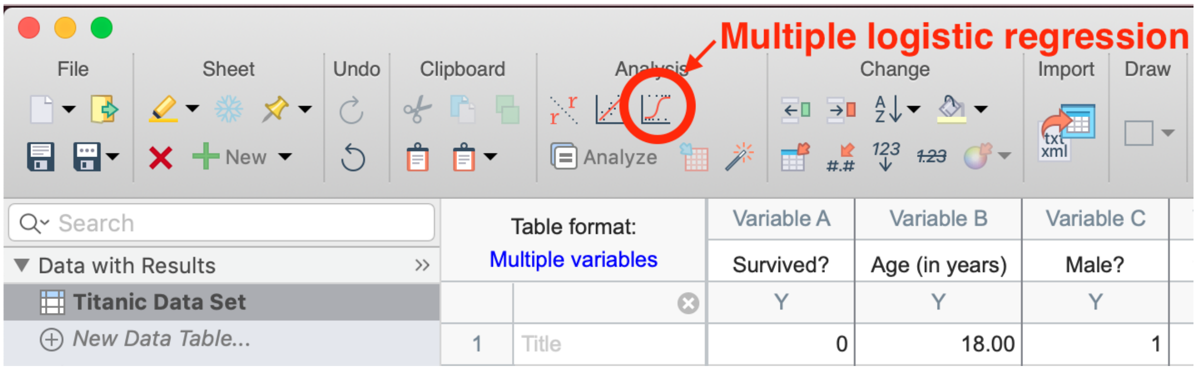
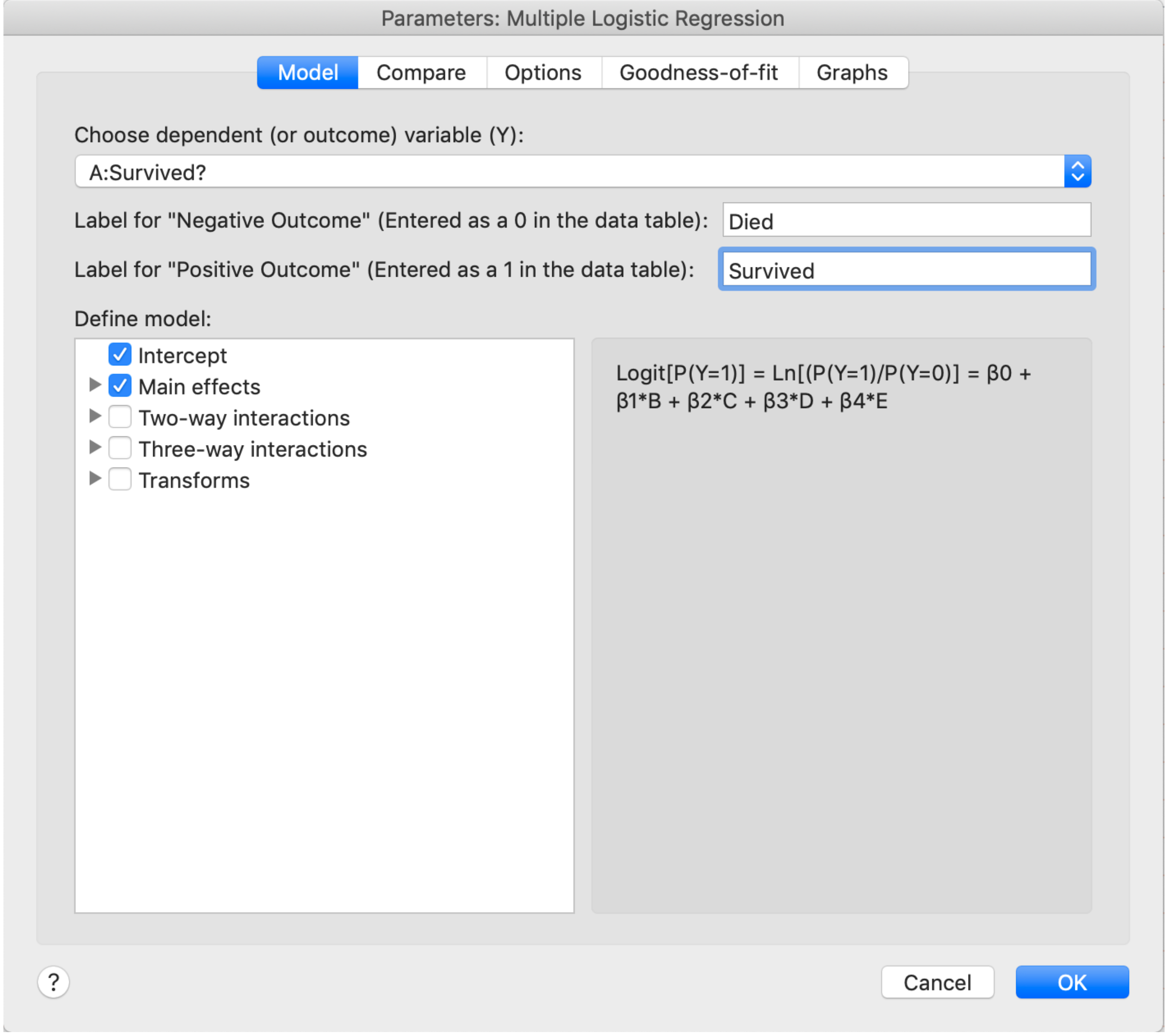


Table format: Multiple variables	Variable A	Variable B	Variable C
	Survived?	Age (in years)	Male?
	Y	Y	Y
1 Title	0	18.00	1

The analysis dialog

After clicking the multiple logistic regression button, the parameters dialog for this analysis will appear. For the purposes of this walkthrough, we'll simply accept most of the default options. The results for these default options are discussed below, but there are many more options available on each of the tabs of the multiple logistic regression parameters dialog 200. Something that we will change to make interpretation of the results a bit easier are the labels for the "Negative" and "Positive" outcomes (i.e. what it means for the entered response to be a 0 or a 1). In the space for the "Negative Outcome" label, enter "Died", and in the space for the "Positive Outcome" label, enter "Survived".



Parameters: Multiple Logistic Regression

Choose dependent (or outcome) variable (Y): A:Survived?

Label for "Negative Outcome" (Entered as a 0 in the data table): Died

Label for "Positive Outcome" (Entered as a 1 in the data table): Survived

Define model:

- ☑ Intercept
- ▶ ☑ Main effects
- ▶ □ Two-way interactions
- ▶ □ Three-way interactions
- ▶ □ Transforms

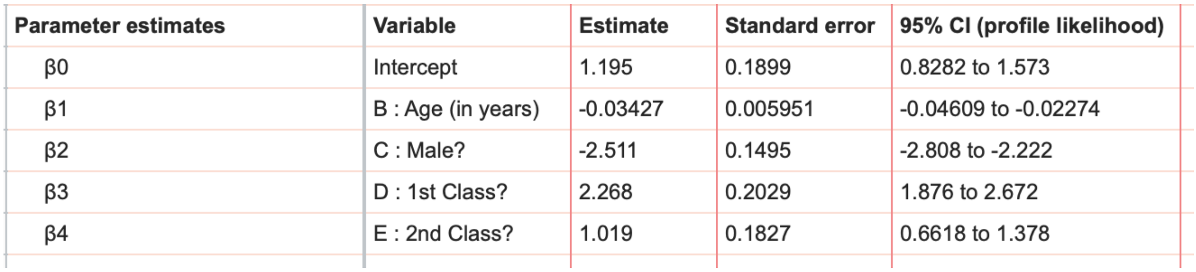
Logit[P(Y=1)] = Ln[(P(Y=1)/P(Y=0)] = β0 + β1*B + β2*C + β3*D + β4*E

Once you click "OK", you'll be taken to the main results sheet which will be discussed in the next section.

Results of logistic regression

Parameter estimates

The first thing that you'll see on the results sheet are the best fit value estimates along with standard errors and 95% confidence intervals for β0 along with one parameter estimate for each component (main effect or interaction) in the model. In this case, we have the parameter estimate for the intercept (β0) and four main effects.

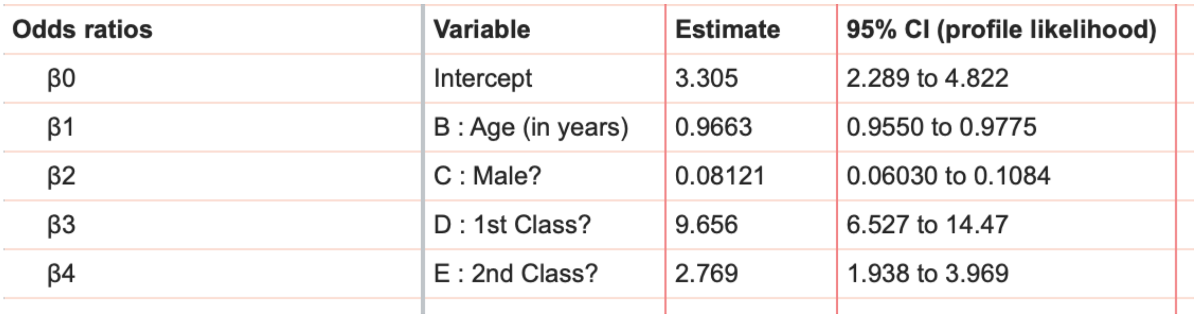


Parameter estimates	Variable	Estimate	Standard error	95% CI (profile likelihood)
β0	Intercept	1.195	0.1899	0.8282 to 1.573
β1	B : Age (in years)	-0.03427	0.005951	-0.04609 to -0.02274
β2	C : Male?	-2.511	0.1495	-2.808 to -2.222
β3	D : 1st Class?	2.268	0.2029	1.876 to 2.672
β4	E : 2nd Class?	1.019	0.1827	0.6618 to 1.378

Interpretation of these parameters is a little different than with linear regression due to the fact that logistic regression models the log odds. Make sure you understand the relationship between log odds, odds, and probability 35 before trying to make sense of these parameter estimates. In this example, we see that the estimate for "Male?" is -2.511, while the estimate for "1st class?" is 2.268. This means that if a passenger was male (all other variables held constant), their log odds of survival was *decreased* by 2.511, while if a passenger was in first class (all other variables held constant), their log odds of survival was *increased* by 2.268.

Odds ratios

As an alternative to seeing the results as log odds, Prism also reports the odds ratios and their 95% confidence intervals (reported farther down on the results sheet).



Odds ratios	Variable	Estimate	95% CI (profile likelihood)
β0	Intercept	3.305	2.289 to 4.822
β1	B : Age (in years)	0.9663	0.9550 to 0.9775
β2	C : Male?	0.08121	0.06030 to 0.1084
β3	D : 1st Class?	9.656	6.527 to 14.47
β4	E : 2nd Class?	2.769	1.938 to 3.969

A more detailed explanation of odds ratios can be found here 228, but what the odds ratios for each variable (β1, β2, β3, and β4) tells us is that increasing that variable by 1 multiplies the odds of success by the value of the odds ratio. The odds ratio for "Age (in years)" is reported as 0.9663. This means that for each year older a passenger was, their odds of survival was multiplied by 0.9663. Since this odds ratio is less than 1, this means that as a passengers age increases, their odds of survival actually decreased (all other factors held constant).

Another point to note is that, because we've used dummy coding for some of our variables, these odds ratios assume a certain "default" or "reference" status for these passengers. For example, we can set the variables "Male?", "1st Class?", and "2nd Class?" to equal zero to investigate the effects of age on the odds of survival. However, because we've used dummy coding, what we're actually looking at is the effect of age on the odds of survival for 3rd class female passengers. Let's say we determined the odds of survival for a 3rd class female passenger at 25 years of age. In this case, the values of "Male?", "1st Class?", and "2nd Class?" would all be 0, and the resulting Odds would be 1.402. Knowing this, we could then quickly determine the odds of survival for a 1st class female passenger at age 25 (Odds = 1.402*9.656 = 13.538). Thus, when using dummy coding, the odds ratio actually tells us how the odds change *relative to the reference case*.

Review of probability and odds

In case you haven't read about the relationship of probability and odds, here's a quick summary:

Odds = Probability of success/Probability of failure

Since the probability of failure is just 1 - Probability of success, we can write this as:

Odds = (Probability of success)/(1 - Probability of success)

For example, let's say there's a 75% probability of success, the odds would then be calculated as:

Odds = 0.75/(1 - 0.75) = 0.75/0.25 = 3

Commonly, we would say that "the odds are 3:1" (read "three to one").

P values

By default, P values for the parameter estimates aren't given in the results, and thus won't be discussed here. However, you can select to display these values using the Options tab of the Multiple Logistic Regression Parameters dialog 205. You'll also want to read about how to interpret these values 230 if you choose to have Prism report them.

Model diagnostics

The next section of the multiple logistic regression results provides a number of useful model diagnostics for determining how well the data fit the selected model. By default, the two values reported here include the degrees of freedom and corrected Akaike's Information Criterion (AICc) for both an "Intercept-only model" and the "Selected model". These two models are fairly easy to describe:

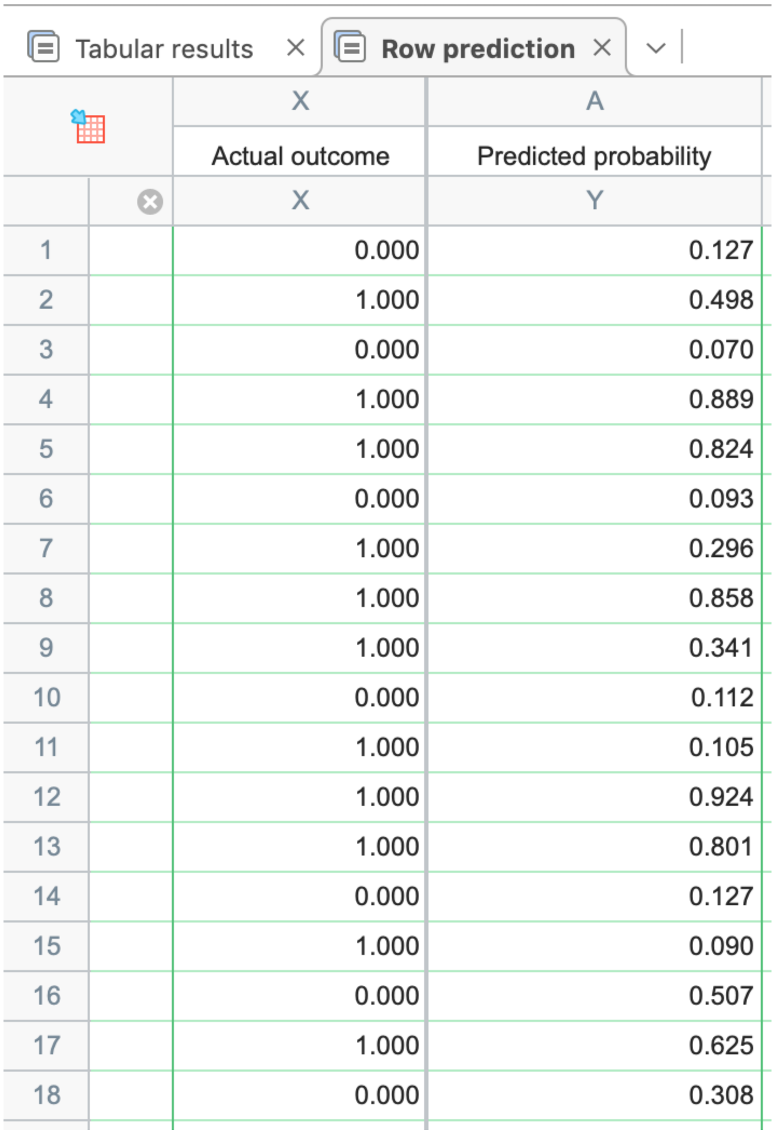
- Intercept-only model: a logistic regression model including only the intercept term β0. This model uses none of the independent variables when predicting the outcome. Therefore the same prediction is made for every case.

- Selected model: the logistic regression model that you chose to fit to the data

The results of this section actually provide you with a way to determine if your selected model does a better job at fitting the data than an intercept-only model. Another way to say this is that the variables in your selected model provide useful information about the observed data. The way you determine which model does a better job fitting the data is by using the AICc. The way the AICc is calculated 230 is a bit complex, but interpretation in this case is fairly simple: a smaller AICc indicates a better model fit. With values of 1746 for the intercept-only model and 1235 for our selected model, we can determine that our selected model does a better job describing the observed data.

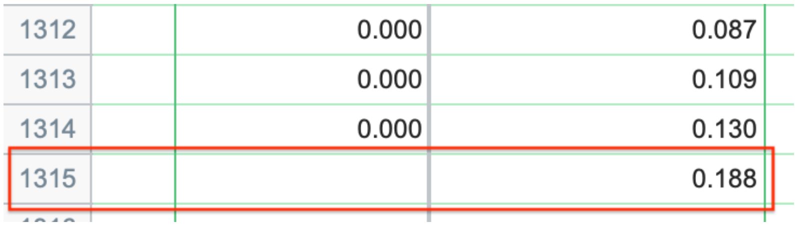
Row prediction

The next section that we'll discuss is actually not on the main results tab. At the top of the main results sheet, you'll see a tab titled "Row Prediction". When clicking on this tab, Prism will provide a complete list of predicted probabilities for all observations with complete information for each variable:



		X	A
		Actual outcome	Predicted probability
		X	Y
1		0.000	0.127
2		1.000	0.498
3		0.000	0.070
4		1.000	0.889
5		1.000	0.824
6		0.000	0.093
7		1.000	0.296
8		1.000	0.858
9		1.000	0.341
10		0.000	0.112
11		1.000	0.105
12		1.000	0.924
13		1.000	0.801
14		0.000	0.127
15		1.000	0.090
16		0.000	0.507
17		1.000	0.625
18		0.000	0.308

This table provides predicted probabilities for all complete observations in the table. This includes both observations from the data that were fit as well as observations that were entered without a Y value. In this example, we could imagine a hypothetical passenger who was a male, age 34, with a second class ticket. Surprisingly, no such passenger exists in the current data set. However, we could enter the corresponding values for this hypothetical passenger into the original table on row 1315 for the corresponding variables, leaving the cell for "Survived?" blank. Since the survival cell is blank, adding this row to the table won't affect most of the results of logistic regression. but Prism will show you the predicted probability of this hypothetical individual surviving.

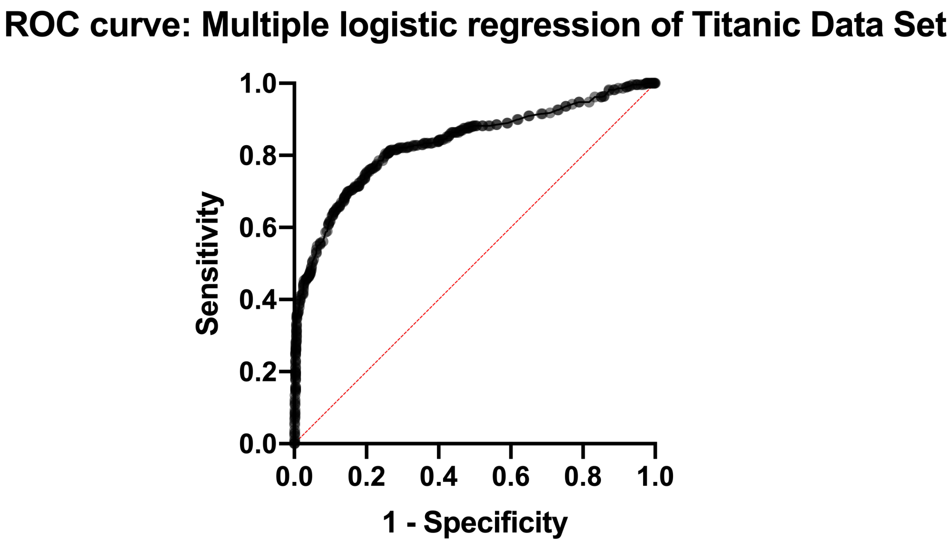


1312		0.000	0.087
1313		0.000	0.109
1314		0.000	0.130
1315			0.188

This result tells us that - based on the observations of the other 1314 passengers - our new hypothetical passenger (34 year old male with 2nd class ticket) would have only an 18.8% probability of surviving!

The ROC curve and the area under the ROC curve

Going back to the "Tabular results" section of the results sheet, the next section of the results deals exclusively with something called an ROC curve. The ROC curve for this analysis is provided in the Graphs section of the Navigator, and looks like this:

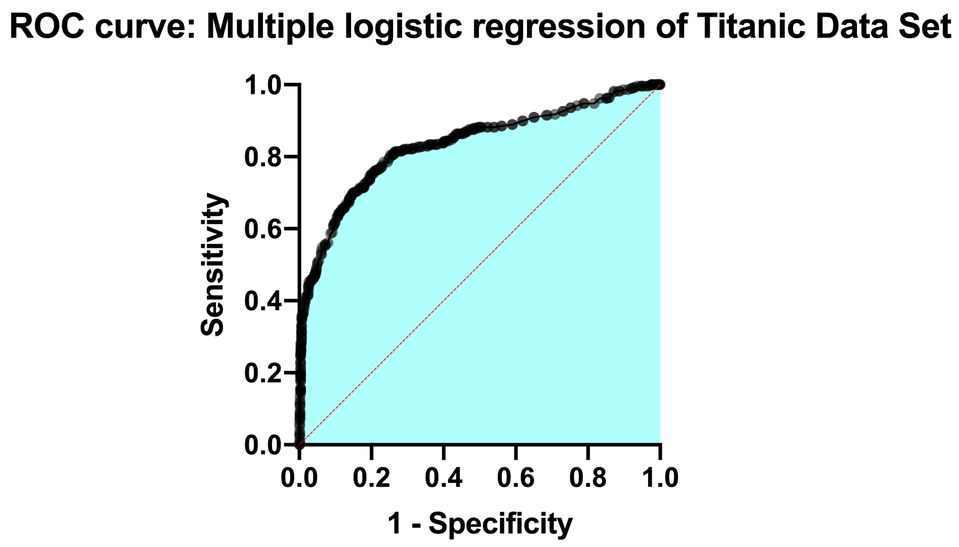


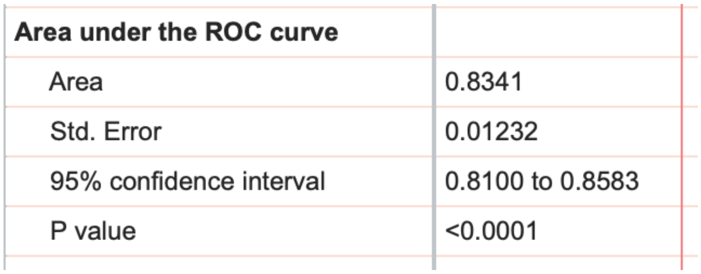
Understanding ROC curves takes a bit of experience, but ultimately what these graphs are showing you is the relationship between the model's ability to correctly classify successes correctly and its ability to classify failures correctly. The way the model classifies observations is by setting a cutoff value. Any predicted probability greater than this cutoff is classified as a 1, while any predicted probability less than this cutoff is classified as a 0. If you set a very low cutoff, you would almost certainly classify all of your observed successes correctly. The proportion of observed successes correctly classified is referred to as the Sensitivity, and is plotted on the Y axis of the ROC curve (a Y value of 1 represents perfect classification of successes, while a Y value of 0 represents complete mis-classification of successes).

However, with a very low cutoff, you would also likely *incorrectly* classify many failures as successes as well. Specificity is the proportion of correctly classified failures, and "1-Specificity" is plotted on the X axis (so that an X value of 0 represents perfect classification of failures, and an X value of 1 represents complete mis-classification of failures).

You can imagine that as the cutoff is varied (from 0 to 1), there will be a trade-off of the observed successes and failures that are correctly (and incorrectly) classified. That trade-off is what the ROC curve shows: as sensitivity increases, specificity must decrease (i.e. 1-specificity must increase). Each point along this ROC curve represents a different cutoff value with corresponding sensitivity and specificity values.

The area under the ROC curve (AUC) is a measure of how well the fit model correctly classifies successes/failures. This value will always be between 0 and 1, with a larger area representing a model with better classification potential. In our case, the AUC for the the ROC curve (depicted below) is 0.8889, and is listed in the results table along with the standard error of the AUC, the 95% confidence interval, and P value (null hypothesis: the AUC is 0.5). Read more about ROC curves for logistic regression 239 for even more information and some of the math involved.



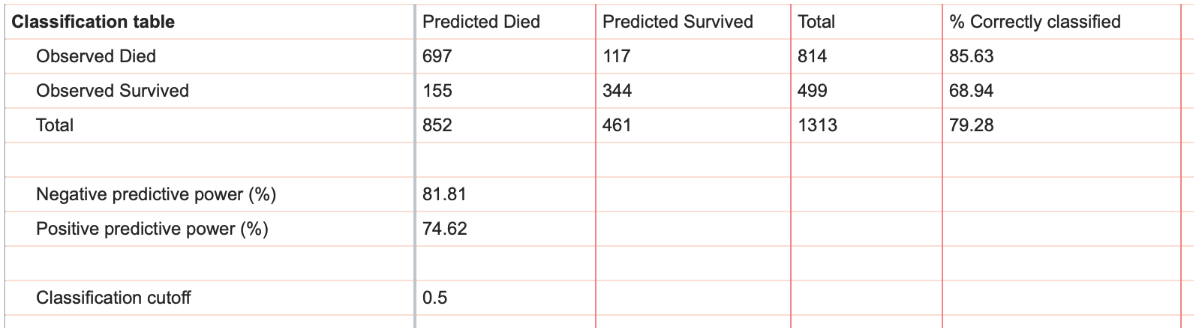


Area under the ROC curve	
Area	0.8341
Std. Error	0.01232
95% confidence interval	0.8100 to 0.8583
P value	<0.0001

Classification table

As discussed in the previous section, the area under the ROC curve considers every possible cutoff value for distinguishing if an observation is predicted to be a "success" or a "failure" (i.e. predicted to be a 1 or a 0). In this section of the results, a single cutoff value is used to generate a table of results that provide the number of observed 1s and 0s along with the predicted number of 1s and 0s. The default cutoff value used to generate this table is 0.5, but this can be changed manually in on the "Goodness-of-fit" tab of the multiple logistic regression parameters dialog 206.

Below is the table containing our results for a cutoff of 0.5:



Classification table	Predicted Died	Predicted Survived	Total	% Correctly classified
Observed Died	697	117	814	85.63
Observed Survived	155	344	499	68.94
Total	852	461	1313	79.28
Negative predictive power (%)	81.81			
Positive predictive power (%)	74.62			
Classification cutoff	0.5			

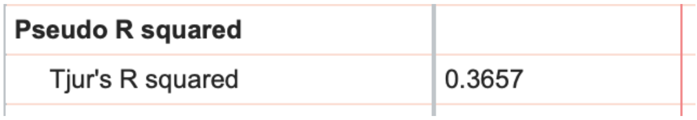
From this table, you can quickly see the total number of observed died (814), total number of observed survived (499), the predicted total number of died (852), and the predicted total number of survived (461). Additionally, this table provides the breakdown for how model predicted each of the the observed survived and died, along with the percent of observed survived and died that were classified correctly.

Finally, the classification table provides information on the negative and positive predictive power, which are other ways that the performance of a model can be assessed.

Pseudo R squared

Another way to quantify how well a logistic regression model fits to the given data is by using metrics known as pseudo R squared values. It's important to note right away that although "R squared" may be in their name, these metrics simply cannot be interpreted in the same way that R squared for linear and nonlinear regression is interpreted. Instead, these values provide different kinds of information about the model fit, and will take on values between 0 and 1, with higher values indicating a better fit of the model to the data.

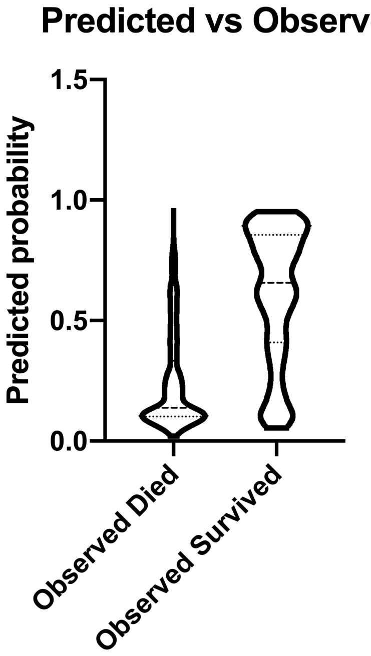
Of the various pseudo R squared values that Prism can calculate, Tjur's R squared is arguably the easiest to interpret, and is the only one reported by default. To calculate Tjur's R squared, find the average predicted probability of success for the observed successes and the average predicted probability for the observed failures. Then calculate the absolute value of the difference between these two values. That's Tjur's R squared! Values close to 1 would indicate that there is a clear separation between observed 0s and observed 1s, while a a Tjur's R squared close to zero would indicate that the average predicted probability of success for both groups were almost the same (i.e. the model didn't do a good job separating the observed 0s and 1s).



Pseudo R squared	
Tjur's R squared	0.3657

Predicted vs Observed graph

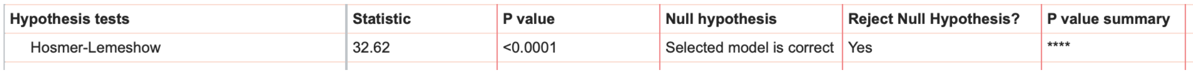
Another way to visually inspect how well the selected model does at predicting successes and failures is to look at the Predicted vs Observed graph provided in the Graphs section of the Navigator by default. The graph for our data looks like this:



Interpretation of this graph is fairly straightforward. In this graph, we can see that there are two groups (one for the group of individuals that survived and one for the group of individuals that did not). We can also see the distribution of predicted probabilities for both of those groups. Looking at the violin plot for the group of individuals that died, we can see that the majority of them had predicted probabilities of survival well below 0.5 (with a median of 0.1383 and mean of 0.2411). We can also see that the model did not perform as well with classifying the group of observed survivors. For this group, we see that the predicted probabilities are more uniformly distributed (with a median of 0.6564 and a mean of 0.6068). Of course, the predictions are based only on the independent variables (age, gender and class of service) and ignore the actual outcome.

Hypothesis tests

If we click back to the "Tabular results" tab of the results sheet, we can continue investigating the other results reported by multiple logistic regression. By default, the next section of the results provides one of two ways that Prism can test how well the model fits the data 237. This test is known as the Hosmer-Lemeshow test, and tests the null hypothesis that the selected model is correct. The specifics of this test are a bit complicated, but based on our data, we would likely elect to reject this null hypothesis that the specified model is correct.



Hypothesis tests	Statistic	P value	Null hypothesis	Reject Null Hypothesis?	P value summary
Hosmer-Lemeshow	32.62	<0.0001	Selected model is correct	Yes	****

Given this result, we *may* choose to investigate additional factors that could have influenced the probability of survival that weren't included in our original model. However, it's very important to note that the Hosmer-Lemeshow test has been shown to be sensitive to slight changes in its calculation method, and as such is included in Prism's results primarily as a means to validate findings in Prism with other software applications.

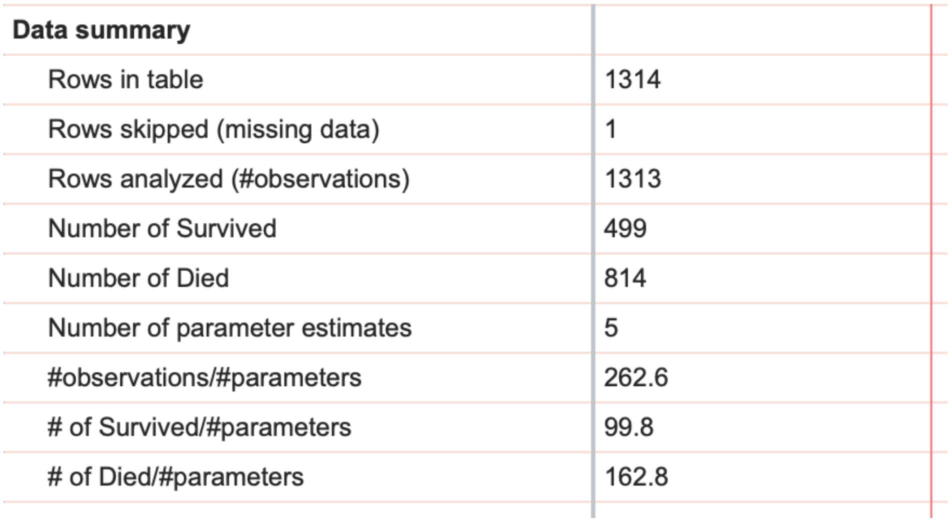
A small P value does NOT necessarily mean that your model is somehow inherently "bad". After all,

"All models are wrong, but some are useful..."

Data Summary

The final pieces of information that Prism provides from multiple logistic regression are given in the form of a data summary that includes the number of rows in the data table, the number of rows that were skipped, and the difference of these two values providing the number of observations in the analysis. Note that in our data set, we have 1314 rows in the table (1315 if you added the example for interpolation), but only 1313 rows analyzed. This difference is due to the fact that the outcome (survived or didn't) was unknown for one (or more) passengers, and so these rows were skipped when fitting the model.

Additionally in the data summary, the total number of 1s and the total number of 0s is given as "Number of Survived" and "Number of Died". Finally, three ratios are provided: number of observations to number of parameters, number of survived to number of parameters, and number of died to number of parameters (we recommend that these last two ratios should be at least 10 for logistic regression).



Data summary	
Rows in table	1314
Rows skipped (missing data)	1
Rows analyzed (#observations)	1313
Number of Survived	499
Number of Died	814
Number of parameter estimates	5
#observations/#parameters	262.6
# of Survived/#parameters	99.8
# of Died/#parameters	162.8

4.2.5 Results of multiple logistic regression

Logistic regression was added with Prism 8.3.0. This section of the guide will provide you with information on how to interpret multiple logistic regression results generated by Prism.

Error messages from logistic regression 247

Perfect separation, Quasi-perfect separation, and Linear dependence 247

Logistic models with no intercept 253

A simple example of perfect separation 255

4.2.5.1 Parameter Estimates

Prism reports the parameter estimates in two ways.

- The parameter estimates' effects on the log odds (remember that log odds = β0 + β1 * X1 + β2 * X2 + ...).

- An interpretation of the multiplicative change on the log odds in the form of odds ratios 228

Additional information on P values calculated for these parameter estimates 230 is provided elsewhere.

Parameter interpretations

Interpreting parameter estimates for logistic regression is more complicated than for linear regression. The reason is that we have transformed Y to model the log odds. The beta coefficient estimates listed under "Parameter estimates" in the output have the following interpretation:

If we are talking about β2, we can say that for a 1 unit increase in X2, the log odds of Y increases by β2, when all the other X values are held constant.

Since most people don’t intuitively think in terms of log odds, Prism also offers interpretation based on odds ratios (link to next page).

Standard errors and confidence intervals

Similar to many other analyses (such as multiple linear regression), the only way to know the actual, true value of a given parameter is to collect information on the entire population. For example, if you wanted to know average human weight, you could (hypothetically), measure the weight of every single human and calculate the average. However, since you can’t actually collect data from every person, you collect a sample. The average that you collect from that sample will have some error due to random variability in the subjects that you selected. For multiple logistic regression, Prism reports two values that provide an idea as to the amount of error in the estimates of the provided parameter coefficients: standard error and profile likelihood confidence intervals.

The standard error of a coefficient can be difficult to interpret, but in simple terms, it provides an idea for how precise the parameter estimate is.

Another way to look at this concept of precision is with confidence intervals, which provide you with some idea of how sure you can be of the provided coefficient estimate. The general concept is that if the experiment were repeated a huge number of times and confidence intervals constructed for the parameter coefficient each time, 95% of these intervals (for a 95% confidence interval) would contain the true parameter coefficient for the population. Note that some software reports symmetric confidence intervals calculated using the standard error described above. Prism actually calculates more accurate (but also slightly more complicated) profile likelihood confidence intervals. These can be (usually are) asymmetric around the parameter value.

4.2.5.2 Odds ratios

Odds ratios are simply a transformation of the parameter estimates 227 calculated for the logistic regression model. However, many find odds ratios to be much more useful when interpreting the results of logistic regression. The reason for this is that the parameter estimates tell you how much the "log odds" change when the independent (X) variable changes. But "log odds" are hard to explain. In contrast, odds ratios tell you how much the odds change when the independent (X) variable associated with that odds ratio changes. Before reading on, be sure you can tell the difference between probability and odds.

The standard form of the equation that multiple logistic regression fits is:

$$\ln[P(Y=1)/P(Y=0)] = \beta0 + \beta1*X1 + \beta2*X2 + \ldots, \text{ or}$$

$$\ln(Odds) = \beta0 + \beta1*X1 + \beta2*X2 + \ldots$$

That is, the log odds can be expressed as a linear equation. If we exponentiate both sides, we get the following relationship:

$e^{\ln(\text{Odds})} = e^{\beta 0 + \beta 1*X1 + \beta 2*X2 + \dots}$, or

Odds = $e^{\beta 0 + \beta 1*X1 + \beta 2*X2 + \dots}$, or

Odds = $(e^{\beta 0})*(e^{\beta 1*X1})*(e^{\beta 1*X2})*\dots$

If we replace $(e^{\beta 0}) = \mathit{\beta 0}$, we get

Odds = $(\mathit{\beta 0})*(\mathit{\beta 1}^{X1})*(\mathit{\beta 1}^{X2})*\dots$

This illustrates the relationship of the parameter estimates and the odds ratios mentioned before: if you exponentiate β1 in the parameter estimate section, you'll obtain the value reported for β1 in the odds ratio section of the results. Using this knowledge, it can be seen that the estimates for the odds ratios have the following interpretation:

For *β2*, we can say that a 1 unit increase in X2 has a multiplicative effect equal to the odds ratio estimate of *β2* on the odds of Y when all the other X values are held constant.

As a simple example, let's say for a given set of values, you calculated the odds of a "success" to be 3 (sometimes called "3:1 odds" or "three to one odds"). If *β2* was estimated to be 2, then an increase of X2 by 1 would result in an increase of the odds to 6 (or "6:1 odds").

Prism also provides confidence intervals for the odds ratio parameters. These are often misunderstood statistical concepts because they aren't quite what our intuition wants them to be. A proper interpretation of a confidence interval would read: We are 95% confident that the range between lowerVal and upperVal includes the true odds ratio for thisParameter

Note that calculated P values 230 will be the same for both the parameter estimates and the odds ratios for a given logistic regression model.

4.2.5.3 P values

Prism provides the option to calculate a P value for each parameter estimate (and odds ratio) of a logistic regression model.

Although the statistical test has a different distribution, the interpretation of P values when used to assess model parameters for logistic regression is the same as it is with multiple linear regression 189.

Specifically, the null hypothesis tested is that the true population value of the coefficient/parameter is zero. Note that if the value of the coefficient were actually zero, this would mean that an increase (or decrease) in the associated variable (X value) would have no effect on the log odds (or odds).

For each coefficient, a calculated P value answers the question: if the null hypothesis (above) were true, what is the probability of observing this coefficient estimate or a more extreme one? If this probability is small enough (a P value smaller than the selected alpha level, generally 0.05), we reject the null hypothesis.

For each parameter, Prism reports

- The absolute value of Z, calculated as the coefficient estimate divided by its standard error
- The P value which is determined from Z
- A P value summary, reported as “ns” or as one or more asterisks

4.2.5.4 Model diagnostics

Prism provides a number of options that it can report that provide information on how well the specified model compares to other models fit to the same data. Specifically, Prism will report values for Akaike’s corrected information criterion (AICc), the negative log likelihood value, and the model deviance for both the specified model as well as an intercept-only model. Each of these values are described briefly below.

Akaike's corrected Information Criterion (AICc)

This value is derived from an information theory approach that attempts to determine how well the data fit the model. It depends both on the model deviance (described below) as well as the number of parameters in the model. Note that many other software packages will report simply AIC. This is an uncorrected form of Akaike's Information Criterion, and has been shown to select models with too many parameters (i.e. it will overfit) when sample sizes are small. Below are the equations needed to calculate AIC and AICc, and how to convert between the two.

AIC = 2*k + Deviance

where k is the number of parameters in the model (reported by Prism in the Data summary section of the results)

AICc = AIC + [(2k2 + 2k)/(n – k – 1)]

where n is the sample size/number of observations (reported by Prism in the Data summary section of the results)

Note that the equation for AIC and AICc is a bit different for nonlinear regression. Nonlinear regression (and multiple linear regression) essentially fits the value of the sum of squares, so k in the equations above is replaced by k+1.

Negative log likelihood

Although the concept of likelihood is somewhat complex, this is simply another metric by which you can assess how well the specified model fits to the entered data. Mathematically, the log likelihood is:

Log likelihood = Σ(ln(Predicted probabilities for values entered as 1)) + Σ(ln(1 - Predicted probabilities for values entered as 0))

Because the predicted probabilities are all less than 1 (and 1 – predicted probabilities are also less than 1), the natural log of these values are all negative. Thus, the sum of all of these negative values is also negative. Therefore, we take the negative of this value to get a positive value for the "Negative log likelihood" (confusing, I know).

In general, when comparing two models fit to the same data, the model with the larger log likelihood (smaller negative log likelihood) is considered to be a better "fit".m

Model Deviance

This final metric that Prism reports can be used to assess how well a model fits to the data, and also uses the likelihood of the model. As seen above, deviance is also used directly in calculating AIC (and AICc) for logistic regression. Fortunately, once you've calculated the negative log likelihood, calculating the model deviance is simple:

Model deviance = 2*(negative log likelihood)

Sometimes you'll see this as

Deviance = -2*ln(Likelihood), or even as

Deviance = -ln(Likelihood^2)

This is also why deviance is also sometimes referred to as “G squared”

4.2.5.5 Multicollinearity in logistic regression

Strongly correlated predictors, or more generally, linearly dependent predictors, cause estimation instability. What is meant by “linearly dependent predictors”? This simply means that one variable can be written as a linear function of the other. For example, variables X1 and X2 would be linearly dependent if X2 = 3*X1 + 6. This is a very simple example of linear dependence, but with this you can see that simply by knowing the value of X1, you automatically know the value of X2. Thus, as a predictor, X2 adds no new information to the model if X1 is already included.

With predictive modeling, this problem is called multicollinearity. In the extreme case, if two X columns in the model are exactly equal, the optimization algorithm can't determine the coefficient estimates for either column. This is because there are an infinite number of solutions. To see this more clearly, consider a simple case, where the estimated logistic regression model is logit(Y) = 1 + 2*X1. Now, say that we create X2, which is a duplicate of X1 and attempt to refit the model with both predictors. The predicted model could be represented in a number of equivalent ways, such as:

logit(Y) = 1 + X1 + X2

logit(Y) = 1 + 2*X1

logit(Y) = 1 + 0.5 * X1 + 1.5 * X2

In fact, there are an infinite number of ways this equation could be re-written with different coefficients. In statistics, this model is said to be non-identifiable. In this extreme situation, standard errors, confidence intervals and P values can't be calculated.

More common in practice is to have predictor columns that are strongly, but not perfectly, correlated. Although Prism will produce estimates in this case, a similar problem occurs. The multicollinearity increases uncertainty in the parameter estimates, and thus increases confidence intervals and P values.

If your only concern is prediction, then the standard errors being large are not actually a problem. However, if you are interested in interpreting the magnitude of the coefficient estimates (e.g. the larger X1 is, the higher the probability of a success), then multicollinearity is a problem.

In Prism, you can evaluate multicollinearity using variance inflation factors (VIFs). The general rule of thumb is that VIFs greater than 10 indicate strong multicollinearity. In that event, you probably want to remove one of the columns with a high VIF, refit the model and repeat as necessary. VIFs are described in more detail here 193.

You can also choose to have Prism output a correlation matrix. This presents the pairwise correlation between the predictors in matrix form. Variables that are highly correlated with other variables in the model will cause problems with estimating standard errors, confidence intervals and P values.

4.2.5.6 Classification

Often, the goal of logistic regression is to simply classify observations as one of the two possible outcomes that the model describes. We've already discussed how, based on a set of values for the independent (X) variables, the probability of "success" (Y=1) can be calculated. Classification works by using that predicted probability and applying the following set of rules:

1. Define a cutoff value (between 0 and 1)

2. Determine the probability of success for an observation and compare it to the cutoff value

3. If the determined probability is greater than the cutoff value, classify the observations as a success. If the determined probability is less than the cutoff value, classify the observation as a failure

Prism provides two methods that report results based on the logistic model's classification of the data. These include the area under the ROC curve (AUC) and the 2x2 Classification table, both described below.

Area under the ROC curve (AUC)

Area under the ROC curve (AUC) provides an aggregate value of how well the model correctly classifies the 0s and 1s with all possible cutoff values. AUC values range between 0.5 and 1, where an area of 0.5 means that the model predicts which outcomes will be 1 or 0 no better than flipping a coin, and an area of 1 means that the model predicts perfectly. To understand reported AUC values in more detail, look at some examples of various extremes for ROC curves from logistic regression 239.

Classification table (2x2)

The classification table reports a 2x2 table that displays the numbers of correctly classified values at the user-specified cutoff. This table has four entries that report the number of observed 0s (and 1s) that were correctly (and incorrectly) predicted. Additionally, the classification table will provide information on total number of observed 1s and 0s, total number of predicted 1s and 0s, the percent of correctly classified 1s and 0s, the percent of total correctly classified observations, and the positive and negative predictive power.

- The total number of observed (entered) 0s = A + B

- The total number of observed (entered) 1s = C + D

- The total number of predicted 0s = A + C
- The total number of predicted 1s = B + D
- The percent of observed 0s correctly classified = (A/(A+B))*100
- The percent of observed 1s correctly classified = (D/(C+D))*100
- The percent of all observations correctly classified = ((A+D)/(A+B+C+D))*100
- The negative predictive power (%) = (A/(A+C))*100
- The positive predictive power (%) = (D/(B+D))*100

	Predicted 0	Predicted 1
Observed 0	A	B
Observed 1	C	D

There are a number of other values that can be obtained from the classification table that Prism doesn't report directly. For example, common values obtained from this sort of data include the False Discovery Rate (B/(B+D)), the False Negative Rate (C/(C+D)), and many others. For additional information, read more about values that can be calculated from classification tables.

Note also that two common values – Sensitivity and Specificity – can be calculated from the classification table. For the selected cutoff value (default 0.5), the sensitivity and specificity can be calculated from the values in the classification table as:

Sensitivity = D/(C+D)

Specificity = A/(A+B)

Prism reports these in the "percent correctly classified" column. The “percent of observed 1s correctly classified” is the sensitivity, and “percent of observed 0s correctly classified” is the specificity. Prism reports these as percentages.

4.2.5.7 Pseudo R squared

Prism offers four pseudo R squared values. When evaluating these values, it’s important to remember that these cannot be interpreted in the same way as R squared in linear regression. This can be challenging at first due to the fact that – although they do not provide the same information about the model - they've been developed to have some analogies to that popular metric, such as the fact that they are constrained between 0 and 1.

Tjur’s R squared

Tjur's R squared has an appealing intuitive definition. For all of the observed 0s in the data table, calculate the mean predicted value. Similarly, for all of the observed 1s in the data table, calculate that mean predicted value. Tjur's R squared is the distance (absolute value of the difference) between the two means. Thus, a Tjur's R squared value approaching 1 indicates that there is clear separation between the predicted values for the 0s and 1s. Additionally, Tjur’s R squared (like R squared in linear regression) is actually bound between 0 and 1.

Tjur’s R squared = |Average Predicted value for 0s – Average Predicted value for 1s|

McFadden’s R squared

McFadden's (and Cox-Snell’s and Nagelkerke’s) R squared are calculated using likelihoods. The concept of likelihood and log likelihood are briefly discussed in the model diagnostics section 230 of this guide. However, it’s not critical to understand how log likelihood is calculated to get an idea for what this pseudo R squared metric is telling you. Briefly, the likelihood (and log likelihood) give you an idea of how well a model fits to the data. McFadden’s R squared calculates the ratio of the log likelihood for the specified model and an intercept-only model, and subtracts this ratio from 1. In other words:

McFadden’s R squared = 1 – (LogLikelihood(Specified Model)/LogLikelihood(Intercept-only Model)

If the specified model fits the data well, the ratio or log likelihoods will be small, and McFadden’s R squared will be close to 1. If the intercept only model more closely fits the data, the ratio will be closer to 1, and McFadden’s R squared will be closer to zero.

Cox-Snell’s R squared

Similar to McFadden’s R squared, Cox-Snell’s R squared uses the likelihood of the selected model and an intercept-only model fit to the same data (McFadden’s R squared uses the log likelihood). In this case,

Cox-Snell’s R Squared = 1 – [(Likelihood(Intercept-only Model)/(Likelihood(Specified Model)]2/n, where n is the number of observations.

It’s worth noting that while Cox-Snell’s R squared takes a similar approach to McFadden’s R squared, the upper limit of Cox-Snell’s R squared isn’t 1; in fact, the upper limit in many cases can be much less than 1. That means that even if the specified model fits the data perfectly, Cox-Snell’s R squared might be less than 1!

Nagelkerke’s R squared

Nagelkerke’s R squared can be thought of as an “adjusted Cox-Snell’s R squared” mean to address the problem described above in which the upper limit of Cox-Snell’s R squared isn’t 1. This is done by dividing Cox-Snell’s R squared by its largest possible value. In other words:

Nagelkerke’s R squared = (Cox-Snell’s R squared)/(1 – Likelihood(Intercept-only Model)2/n), where n is the number of observations

This website contains more information on these and other pseudo R squared values, while this paper provides a good assessment of these and other goodness of fit metrics.

4.2.5.8 Hypothesis tests

A reminder of how hypothesis tests work

Two hypothesis tests are offered by Prism for assessing how well a model fits the entered data. Like other hypothesis-based tests that you’ve likely encountered, these two tests start by defining a null hypothesis (H0). Each test then calculates a statistic and corresponding P value. The calculated P value represents the probability of obtaining a test statistic as large as the one calculated (or larger) if the null hypothesis were true . This test also requires that a threshold be set for how small a P value must be to make a decision about whether to reject the null hypothesis. This – admittedly arbitrary – threshold is typically set to 0.05, and is also referred to as the alpha (ɑ) level. If the obtained P value is smaller than ɑ, then we reject the null hypothesis.

Before reading about each of the tests below, it's also important to understand what is meant by the terms "specified model" and "intercept-only model". The specified model is simply the model that was fit to the data (the one that you defined on the Model tab of the analysis). It includes the effects of the predictor variables on the probability of observing a success. The intercept-only model is a model that assumes that the contributions from the predictor variables is zero. In other words, the intercept-only model assumes that none of the independent variables help predict the outcome.

Hosmer-Lemeshow (HL) Test

The Hosmer-Lemeshow test is a classic hypothesis test for logistic regression. The null hypothesis is that the specified model is correct (that it fits well). The way the test works is by first sorting the observations by their predicted probability, and splitting them into 10 groups of equal numbers of observations (N). For each group, the average predicted probability is calculated, and that average is multiplied by N to get an expected number of 1s for that group (and in turn the expected number of 0s for that group). It then calculates a Pearson goodness of fit statistic using the observed and expected numbers of 0s and 1s, and summing across all of the groups. It uses a chi-squared distribution to then calculate a P value.

As mentioned, the null hypothesis is that the specified model fits well, so contrary to many tests, a small P value indicates a poor fit of the model to the data. Another way to think about this is that a small P value indicates that there is more deviation from the expected number of 0s and 1s (given 10 bins) than you'd expect by chance. Thus, there may be some additional factor, interaction, or transform that is missing from the model.

This test has received criticism for the arbitrary number of bins (10), since it has been shown that changing this number can influence the result of the test. The test is included as an option in Prism so you can compare results obtained in Prism with results calculated elsewhere, even though this test is not recommended.

Log likelihood ratio test (LRT)

The log likelihood ratio test is also a classic test that compares how well the model selected fits compared to the intercept only model. In this case, the null hypothesis is that the intercept-only model fits best, so a small P value here indicates that you would reject this null hypothesis (or that the specified model outperforms the intercept only model). As the name implies, this test uses the log likelihood of the specified model and the intercept-only model to calculate the associated statistic and P value. Although this test specifically looks at the defined model and an intercept-only model, it is the same test that is provided on the Compare tab of the multiple logistic regression parameters dialog 203 to compare any two nested models.

The likelihood ratio tells you how much more likely the data were to be generated by the fit model than by the intercept-only model. If the independent variables do a good job of predicting the outome, the likelihood ratio will be high and the corresponding P value will be small.

Note the different meaning of the P values

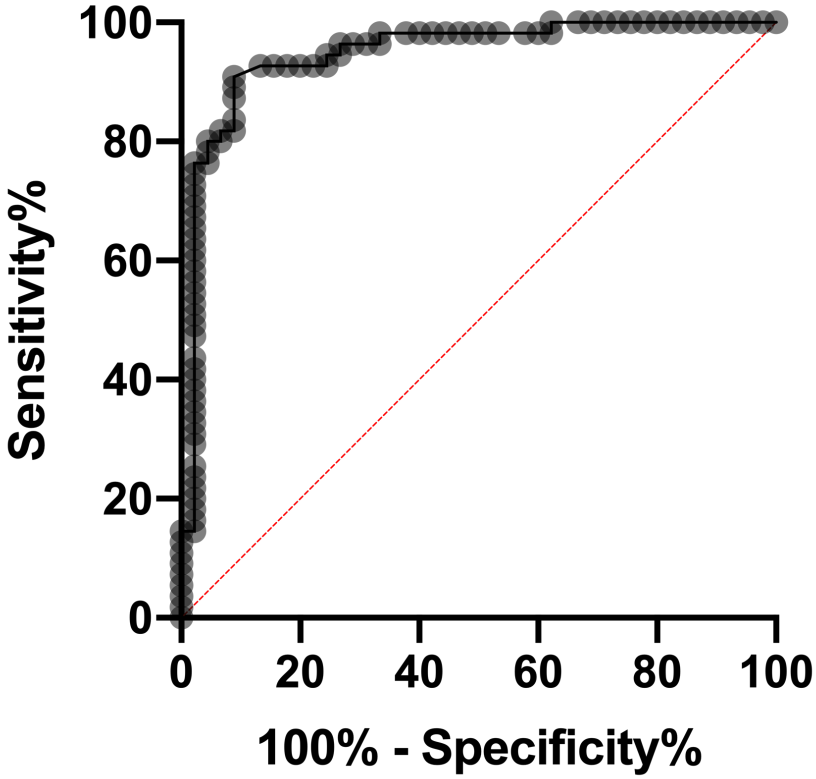
A small P value has opposite meanings with the two tests:

- A small P value from the Hosmer-Lemeshow test means that the *specified model* doesn't do a good job of predicting the data. Consider whether you need additional independent variables or interactions included in the model.

- A small P value from the likelihood ratio test means that the *intercept-only model* doesn't do a good job of predicting the data. The specified independent variables and interactions improve the fit of the model to the data.

4.2.5.9 Interpreting Logistic ROC Curves

ROC curves in logistic regression are used for determining the best cutoff value for predicting whether a new observation is a "failure" (0) or a "success" (1). If you're not familiar with ROC curves, they can take some effort to understand. An example of an ROC curve from logistic regression is shown below.



First, let's cover what a classification cutoff is actually doing. When you choose a classification cutoff (let's say you choose 0.5), you're saying that you would like to classify every observation with a predicted probability from the model equal to or greater than 0.5 as a "success". Note that you will classify observations meeting this criteria as a success regardless if that outcome was actually observed to be a success. Confused? Don't worry, it's less complicated than it sounds. Your observed outcome in logistic regression can ONLY be 0 or 1. The predicted probabilities from the model can take on all possible values between 0 and 1. So, for a given observation, the predicted probability from the model may have been 0.51 (51% probability of success), but your observation was actually a 0 (not a success). We'll discuss the importance of correctly or incorrectly classifying your observations in a minute. For now, let's focus back on the ROC curve.

Each dot on the curve represents a different possible cutoff value for classifying predicted values. You could feasibly pick any value between 0 and 1 as the cutoff, but doing this manually for every possible meaningful cutoff value would be exhausting. So what an ROC curve does is looks at every possible cutoff value that results in a change of classification of any observation in your data set (if stepping the classification cutoff up from 0.5 to 0.6 doesn’t result in a change in how the observations are classified, well then it’s not an interesting step to consider). For every classification cutoff that results in a change of classification, a dot is placed on the plot. But where does that dot go? To answer that, let’s go back to the outcome of classifications to understand a bit more about what classification is doing and the classification table.

Whatever cutoff you choose, a certain number of the rows of data will be correctly classified (you predicted the correct value for that row), and a certain number will be misclassified. Sensitivity and specificity are two metrics for evaluating the proportion of true positives and true negatives, respectively. In other words, sensitivity is the proportion of 1s that you correctly identified as 1s using that particular cutoff value, or the true positive rate. Conversely, specificity is the proportion of 0s that you correctly identified as 0s, or the true negative rate.

Mathematically these are represented as:

- Sensitivity = (number correctly identified 1s)/(total number observed 1s)

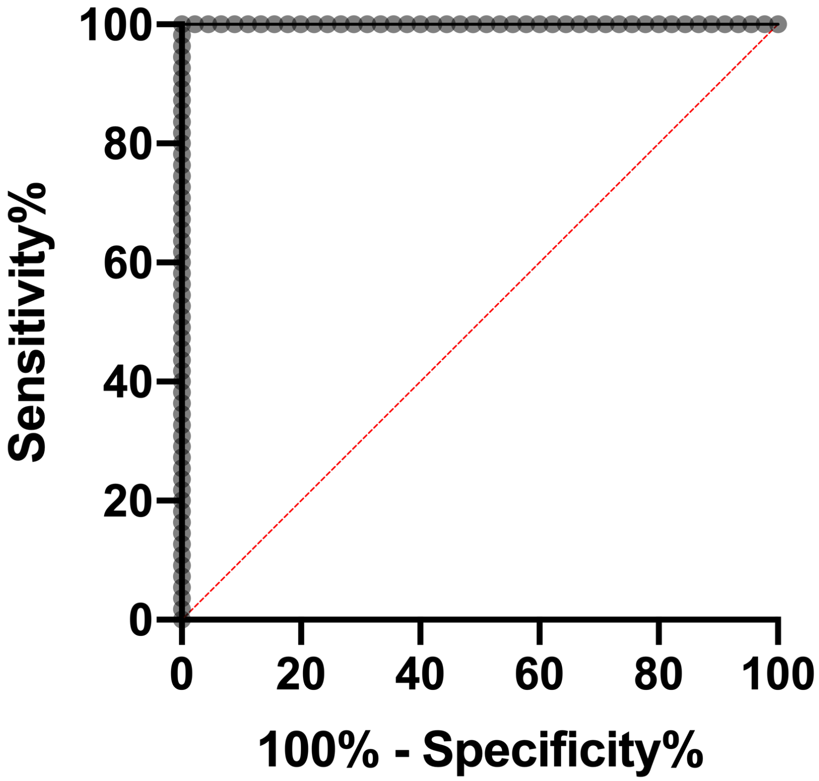
- Specificity = (number correctly identified 0s)/(total number observed 0s)

Given this information, we can put everything together to understand ROC curves. First, we identify the axes of an ROC curve: the Y axis is just sensitivity (or true positive rate), while the X axis is 1-specificity. Although it takes a little extra math (and brainpower) to prove, it can be shown that 1-specificity is equivalent to the false positive rate.

For every point on the ROC curve (representing a different cutoff value), the location of that point is plotted as the sensitivity at that cutoff value on the Y axis, and 1 – specificity at that cutoff value on the X axis. As such, the ROC curve shows graphically the tradeoff that occurs between trying to maximize the true positive rate vs. trying to minimize the false positive rate. In an ideal situation, you would have sensitivity and specificity near 100% at all cutoffs, meaning you predict perfectly in all cases. If you have that, you don't need statistics, because your "success" and "failures" are very easy to tell apart. In fact, with logistic regression, it wouldn’t even be possible to fit this model.

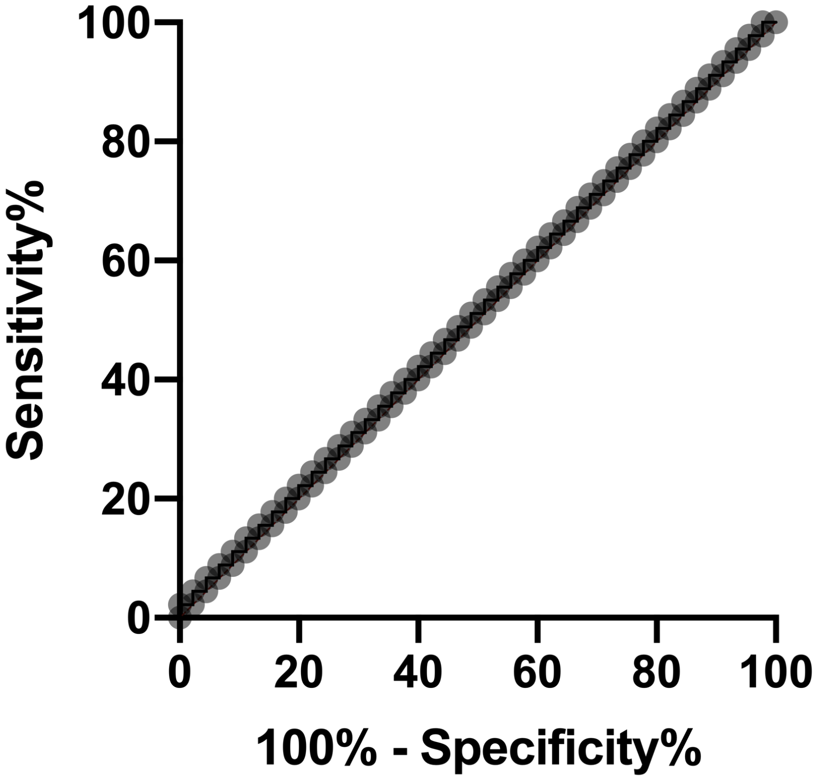
Best-case ROC curve

A best-case ROC would look like a 90 degree angle. If you have this curve, then you probably don't need statistics, since it is trivial to discriminate between the 0s and 1s. Note that at every point, either sensitivity or specificity are at 100% (meaning 1-specificity is at 0%). In fact, this curve shows that there is a cutoff for which both sensitivity and specificity are at 100%. Another way to state this is that there are no false positives and no false negatives. The AUC of this ROC curve is 1.



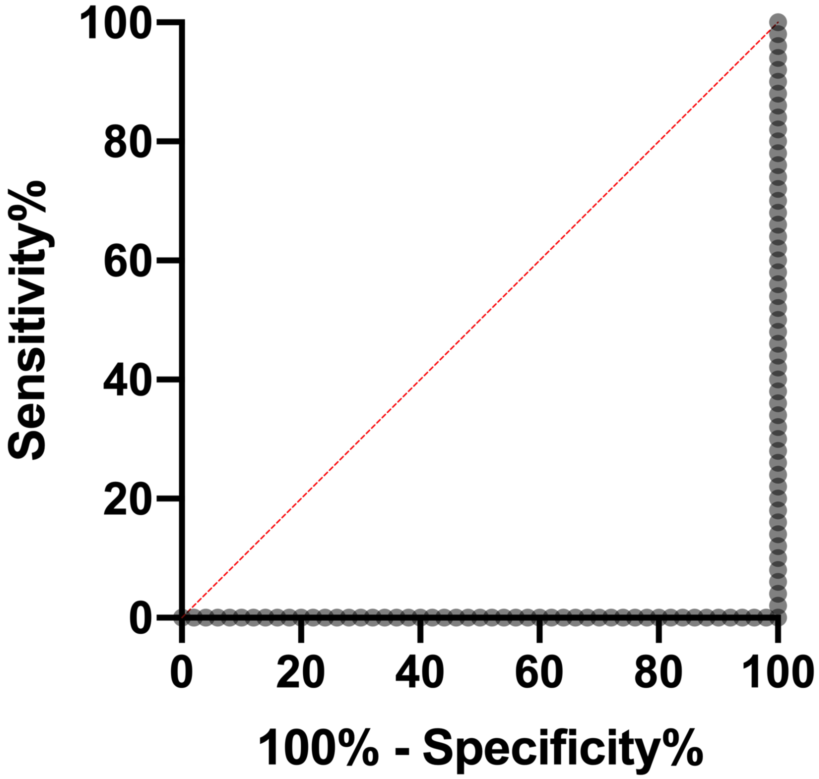
ROC curve with no predictive power:

Alternatively, the worst possible ROC curve (in Prism) predicts no better than by chance, which shows up in an ROC curve as a straight line at 45 degrees. The fit model predicts outcome no better than flipping a coin. Another way to think about this is that the only way to increase the true positive rate (sensitivity) is to also increase the false positive rate (1 – specificity) by the same amount: not a great method at all. The AUC of this ROC curve is 0.5.



Worst-case ROC curve:

Note that there is an additional situation in which a model could (in theory) perform worse than random chance. Recall that the ROC curve plots the sensitivity and specificity of a model, and that both of these values are based on the classification of subjects. You could probably imagine a model in which “successes” (or 1s) were more commonly predicted to be “failures” (or 0s) than what would be expected by random chance. In this case, the model would still be able to identify different groups of outcomes, but would classify them incorrectly (1s would be classified as 0s and vice versa). In the most extreme case, a model could perfectly predict all of your observed 1s to be 0s, and all of your observed 0s to be 1s. In contrast to the “best-case ROC curve”, the graph below shows that for every cutoff value, either sensitivity or specificity (or both) are at 0%. The AUC of this ROC curve is 0!



Area Under the ROC curve

The Area Under the ROC curve (AUC) is an aggregated metric that evaluates how well a logistic regression model classifies positive and negative outcomes at all possible cutoffs. It can range from 0.5 to 1, and the larger it is the better. People will sometimes use the AUC as a means for evaluating predictive performance of a model, although because it represents all possible cutoff values, which isn't feasible in practice, the interpretation is difficult. We recommend interpreting the ROC curve directly as a way to choose a cutoff value.

Choosing a cutoff value

In reality, you will only be able to pick one cutoff value for your model. How do you determine which cutoff to use? It depends on your specific scenario. If false negatives are worse than false positives, then choose a cutoff with high sensitivity (a value higher on the Y axis ofthe ROC graph). Alternatively, if false positives are worse, then pick a cutoff with high specificity (values to the left in the ROC graph).

4.2.5.10 Analysis checklist: Multiple logistic regression

To check that multiple logistic regression is an appropriate analysis for these data, ask yourself these questions.

✓ Is the outcome (Y) variable binary (dichotomous)?

The independent (Y) variable may only take on two values and in Prism, these must be coded as 0 and 1.

✓ Are the rows of Y independent observations?

One of the fundamental assumptions of logistic regression is that each row of data is a unique, independent observation. An example of independent observations is a study on 100 randomly selected people where a 1 indicates a positive outcome and a 0 a negative outcome, and each person is recorded on a single row. If each person was measured more than once (say at various time points in the study), then the observations are not independent and logistic regression isn't appropriate. If the study was of 50 married couples, it is not fair to treat the data as 100 independent observations.

✓ Does the model fit and predict the data well?

All models are wrong, but some are useful…

Prism offers a variety of metrics to evaluate how well the specified model fits to the entered data. However, you should keep in mind that fitting models to data and interpretation of model fits is – to some extent – subjective. Some possibilities to consider when evaluating a given model include:

- Does the logistic model classify data well? In other words, given an appropriate cutoff value (such as 0.5), does the model correctly predict the observed 0s and 1s? You can evaluate this in Prism using the Predicted vs. Observed graph, a classification table, Tjur's R-squared, an ROC plot, and the row classification table.

- Does the logistic model outperform an intercept-only model? You can test this with the likelihood ratio hypothesis test. You may also want to run the Hosmer-Lemeshow test.

Are the X variables linearly dependent?

If the X variables have high multicollinearity 232, the estimated P values and standard errors will be meaningless. Read more about multicollinearity for more information.

Do you have sufficient data to trust your results?

As with all stats modeling, the more data the better. At the bottom of the Tabular results sheet of the analysis results, Prism will report how many observations were in the model (Rows analyzed), how many parameters were included in the model, and the ratio of these two values (#observations/#parameters). One rule of thumb is to have at least ten rows of 0s and ten rows of 1s for each independent (X) variable.

Are you overfitting?

Do changes in your variables actually contribute to changes in the log(odds) of success? If not, do you still want to include those covariates in the model? Sometimes it's important to keep variables in the model for their explanation or because of how an experiment was designed based on your knowledge of the experiment and the science involved. However, if a variable isn't necessary, perhaps it's enough to say that it doesn't help predict the outcome in the presence of the other X predictors and remove it. But removing variables is controversial 45, so don't do so without a lot of thought.

Are you underfitting?

If your prediction performance isn't as good as desired, then perhaps you are missing some key variables that you either didn't measure or didn’t include in the model. If the key variable is one that you didn’t measure, there isn’t much you can do but go back out and collect more data. However, if you left some variables out of the model, you might want to examine how those affect model performance when they’re included. You can also explore fitting models with interactions and transformations on the X variables.

4.2.5.11 Error messages from logistic regression

There are two common types of errors that you may encounter when performing logistic regression within Prism. These categories are listed below, and each is explained in greater detail on their respective pages:

- Problems with model convergence due to separation or linear dependence 247
- Warning due to model selection choices (specifically, omitting the intercept term) 253

4.2.5.11.1 Perfect separation, Quasi-perfect separation, and Linear dependence

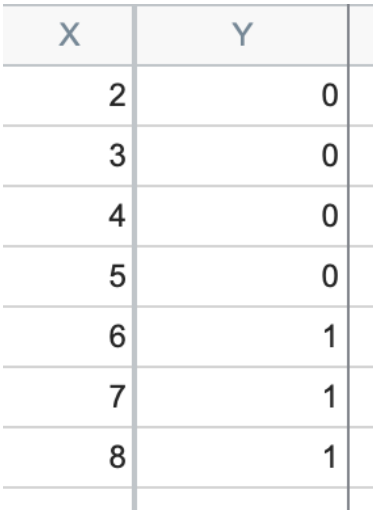
Review of how the logistic model is fit

The process of fitting a logistic regression model to a set of data involves identifying a set of “best-fit” values for the parameters of the model. The way this works is by using an iterative algorithm to maximize the likelihood function for the logistic regression model. One way to think about this process is that it’s trying to discover the values of the parameters for the model that were “most likely” to have created the observed data. This results in a couple of important concepts. First, this means that - in general - logistic regression models will perform better at fitting (or classifying) the entered data than they will at correctly predicting outcomes of new data.

The other implication of using this method is that there are some cases in which it is impossible for the algorithm to determine the parameter values, and thus it is impossible in these cases to define a logistic regression model. Three common cases with logistic regression in which this occurs include: perfect separation, quasi-perfect separation, and linear dependence of X variables.

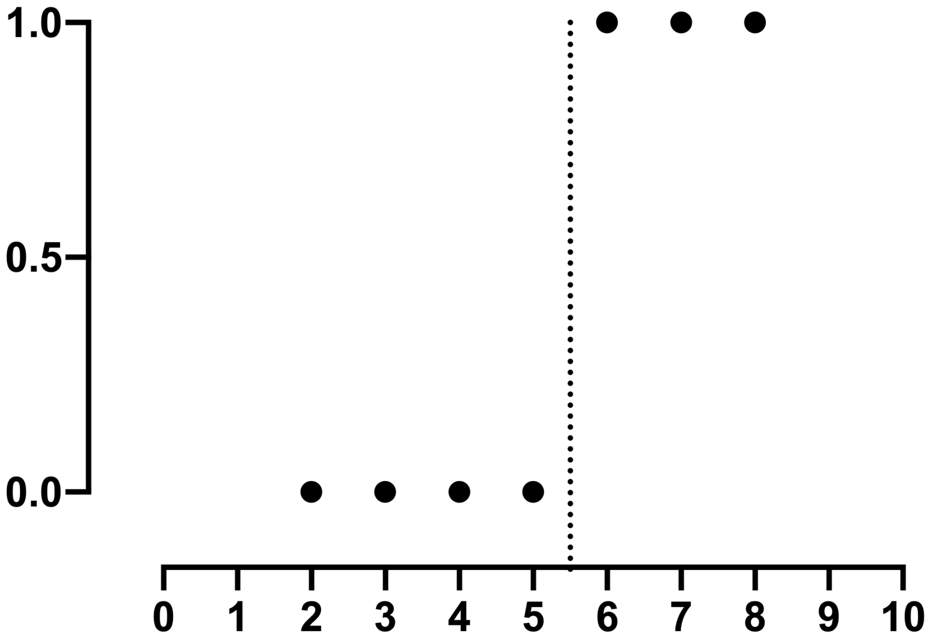
Perfect separation

Perfect separation, sometimes also referred to as complete separation, is the term that is used when a model that predicts the data perfectly. Another way of stating this is that for a given predictor (or some linear combination of predictors), one outcome always occurs above a certain value of the predictor, while the other outcome always occurs below that value. That might sound a bit confusing, but in practice what it means is that the model would correctly classify every point entered as a 0 as a negative outcome and every point entered as a 1 as a positive outcome. On the surface, this doesn't seem like a major problem as one of the goals of logistic regression is to classify the observed outcomes. However, another goal of logistic regression is to find the best-fit estimates of the model parameters, The problem with perfect separation is that, in these situations, there is no single set of best fit values that maximize the likelihood. Let's look at a simple example:



X	Y
2	0
3	0
4	0
5	0
6	1
7	1
8	1

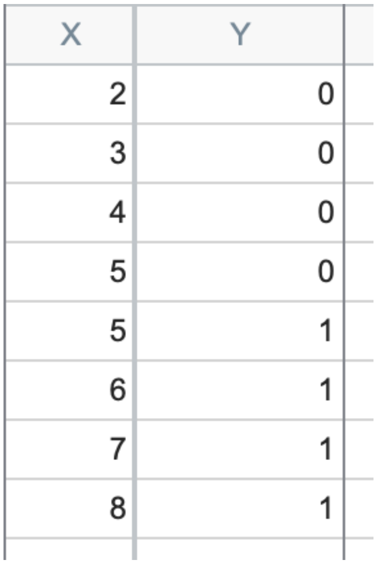
You can see that for every value of X equal or less than 5, the Y value is 0. And for every value of X greater than 5, the Y value is 1. When there's only one predictor (such as with simple logistic regression), we can see what perfect separation means graphically:



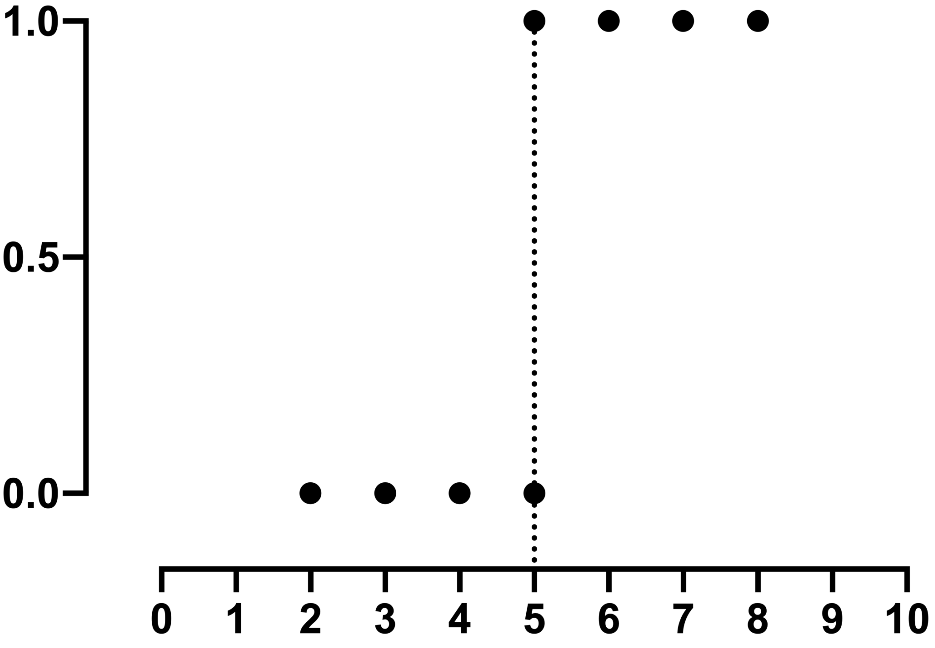
It’s clear that a vertical line can be drawn between X=5 and X=6 for which all points to the left of the line are at Y=0 and all points to the right of the line are at Y=1. So it is easy to predict the outcome from X. But the S shaped logistic curve can't do that job, because the data give no clue as to where that curve should be between 5 and 6. If X=5.5, what is the predicted probability?

Quasi-perfect separation

Quasi-perfect separation is a closely related issue to perfect separation. It occurs when a predictor (or linear combination of predictors) classifies the outcome correctly except at a single value or point. Again, let’s look at some data to better understand this issue:



X	Y
2	0
3	0
4	0
5	0
5	1
6	1
7	1
8	1



You can see now that for all values of X less than 5, the Y value is 0. And for all values of X greater than 5, the Y value is 1. However, AT X=5, we have both Y=0 AND Y=1. Once again, the likelihood cannot be maximized, so best-fit estimates of the parameters cannot be determined.

Finally, if all of your X values are the same value, it will be impossible for Prism to fit a logistic model to the data or to maximize the likelihood function (see section below describing linearly dependent X variables).

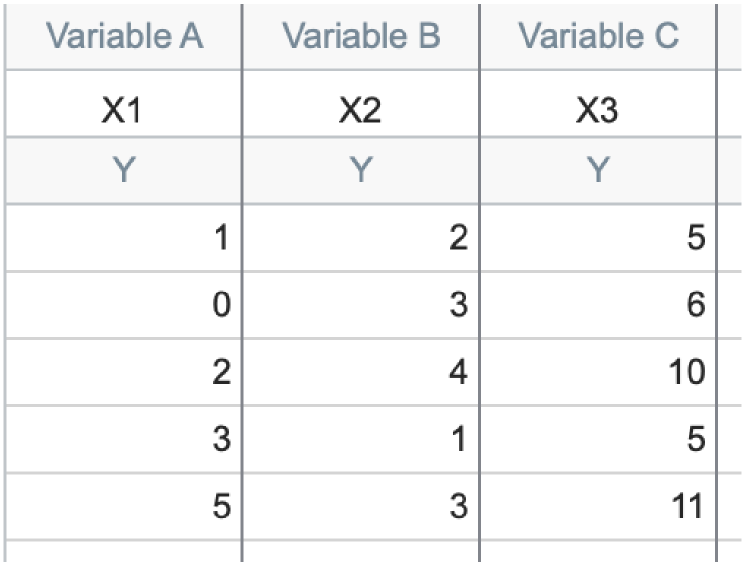
Keep in mind, however, that if you're getting this error, it doesn’t necessarily represent a problem in your experimental design or your data. It could simply mean that you have identified the X variable (or variables) that are capable of perfectly (or quasi-perfectly) predicting your outcome!

Linearly dependent X variables

Another possible problem when attempting to fit a logistic model to your data is the presence of linearly dependent X variables. When a model contains linearly dependent predictors, the algorithm to fit the model will fail because the maximum of the likelihood function won’t exist. To understand this, let’s first define what linearly dependent means. If any of your X variables can be expressed as a linear combination of other variables, then your variables are said to have linear dependence. That didn’t help much unless we also define what a linear combination of variables means. A linear combination of variables refers to the sum of a given set of variables, each multiplied by a constant. This is probably explained easiest via an example. Let’s say you have three variables: X1, X2, and X3. Now, we’ll write a formula:

X3 = A*X1 + B*X2

If there are any values of A and B that could make that expression true, then X3 is a linear combination of X1 and X2. Let’s look at some data to make this concept even more clear:

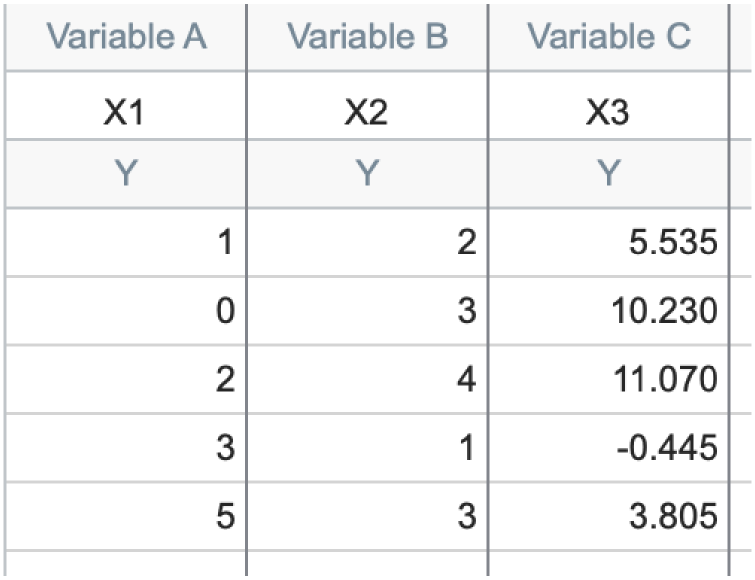


Variable A	Variable B	Variable C
X1	X2	X3
Y	Y	Y
1	2	5
0	3	6
2	4	10
3	1	5
5	3	11

This data shows our three variables (the variables could be anything: age, height, number of steps taken to get to the bathroom…). For this data, the linear relationship is fairly easy to spot:

X3 = 1*X1 + 2*X2

In other words, in every row, the third value is the sum of the first value plus twice the second value. Other linear relationships might not be so easy to spot:



Variable A	Variable B	Variable C
X1	X2	X3
Y	Y	Y
1	2	5.535
0	3	10.230
2	4	11.070
3	1	-0.445
5	3	3.805

The values in the first two columns are the same, but here the linear relationship is given by:

X3 = -1.285*X1 + 3.41*X2

Another way that the linear dependence issue can present itself is if you have duplicated X columns. This may happen unintentionally when using categorical predictors that have been coded. Coded categorical predictors can only take on a limited number of values, and so it’s easy to see how two otherwise unrelated categorical variables could happen to end up with the same values for each observation (this is even more possible with variables that are in some way related). Of course, the odds of this happening by chance decrease with an increasing number of observations.

One easy to understand example of two variables that would be identical could be from the analysis of football (soccer) games. Imagine that one variable was used and represented “goals made”, while a second variable could be “final score”. It seems obvious in this case, but you should always be wary of duplicate (or linearly dependent) variables that are disguised as independent variables in your model.

If you have duplicated X columns, X columns that are perfectly correlated, or X columns that have a more complicated linear relationship, then the optimization algorithm will fail. To evaluate this problem, use the multicollinearity option in the multiple logistic regression dialog. See the multicollinearity page for more details.

Summary of (some) potential problems with Logistic Regression Model Convergence

1. You have too few 0s or 1s (we recommend, if possible, at least 10 rows of each for each independent (X) variable)

    a. In the extreme case, if you have all 0s (or all 1s), you are guaranteed perfect separation, since there is no way for the algorithm to distinguish between the two outcomes.

    b. Another extreme case is if you have as many or more X variables as rows of data. There is no way to estimate model error in this case.

2. One of your X variables (or a linear combination of X variables) results in perfect separation or quasi-perfect separation of your Y variable.

3. Your X variables exhibit linear dependence.

4.2.5.11.2 Logistic models with no intercpet

When performing logistic regression, it’s quite uncommon to choose a model that lacks an intercept (β0) term, so uncommon that Prism displays a warning to alert you to make sure you made that decision for good reasons.

If you exclude the intercept term from your model, **you're making the assumption that the probability of observing a success (P(Y=1)) is equal to 0.5 when all predictor variables are zero**. This is rarely ever a reasonable assumption, and as such, the intercept term is almost *always* kept in the logistic regression model. Let's look at the model for logistic regression, and explore some of the math behind this issue:

Ln(Odds) = β0 + β1*X1 + …

If we set the value of Xi to zero in the equation above, it's reduced to:

Ln(Odds) = β0

Thus, the intercept β0 is equal to the log odds when all other predictors are zero. Excluding the intercept from the model is mathematically equivalent to setting the intercept term equal to zero. If we set eβ0 to zero and exponentiate the equation above, we obtain:

$e^{\mathrm{Ln(Odds)}} = e^{\beta 0}$

$\mathrm{Odds} = e^{\beta 0}$

$\mathrm{Odds} = e^{0}$

Odds = 1

If we now use the fact that the odds can be written as the ratio of the "probability of Y=1" and the "1 - probability of Y=1".

Odds = 1

P(Y=1)/[1 - P(Y=1)] = 1

Solving for P(Y=1), we obtain:

P(Y=1) = (1)*[1 - P(Y=1)]

P(Y=1) = 1 - P(Y=1)

2*P(Y=1) = 1

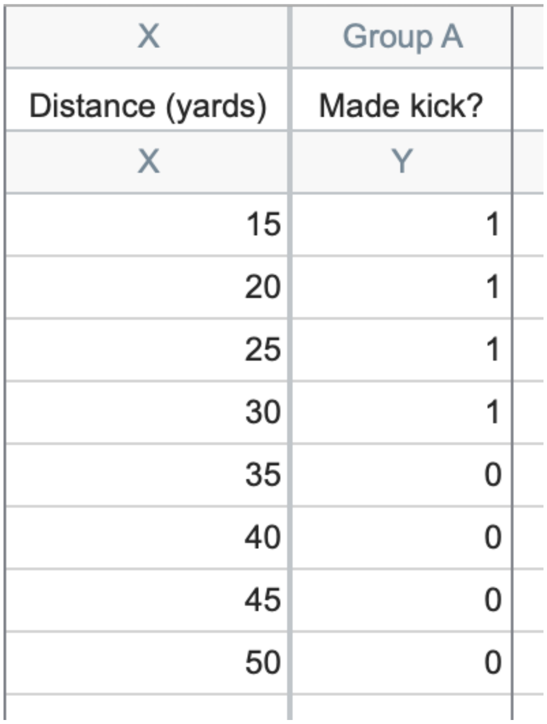
P(Y=1) = 0.5

Thus, by excluding the intercept from the logistic regression model, you have made the assumption that when all predictors are zero, the probability of observing a success is 50%. This assumption is rarely applicable, and so the intercept term is almost always included in the logistic regression model.

4.2.5.11.3 A simple example of perfect separation

The field goal kicker

Let's imagine you're interested in making predictions about whether or not a football kicker will be able to make a given kick based on the distance that they'll have to kick. You know that the distance the ball has to be kicked will play an important part in determining if the kick will be good or not, so to start, you might collect some initial data on the kicker's ability from a variety of distances. To do this, you have the kicker attempt to kick one field goal from a variety of distances (we'll assume for the sake of this example that no one attempt has an impact on any other attempt). The results of this initial test might look something like the following:

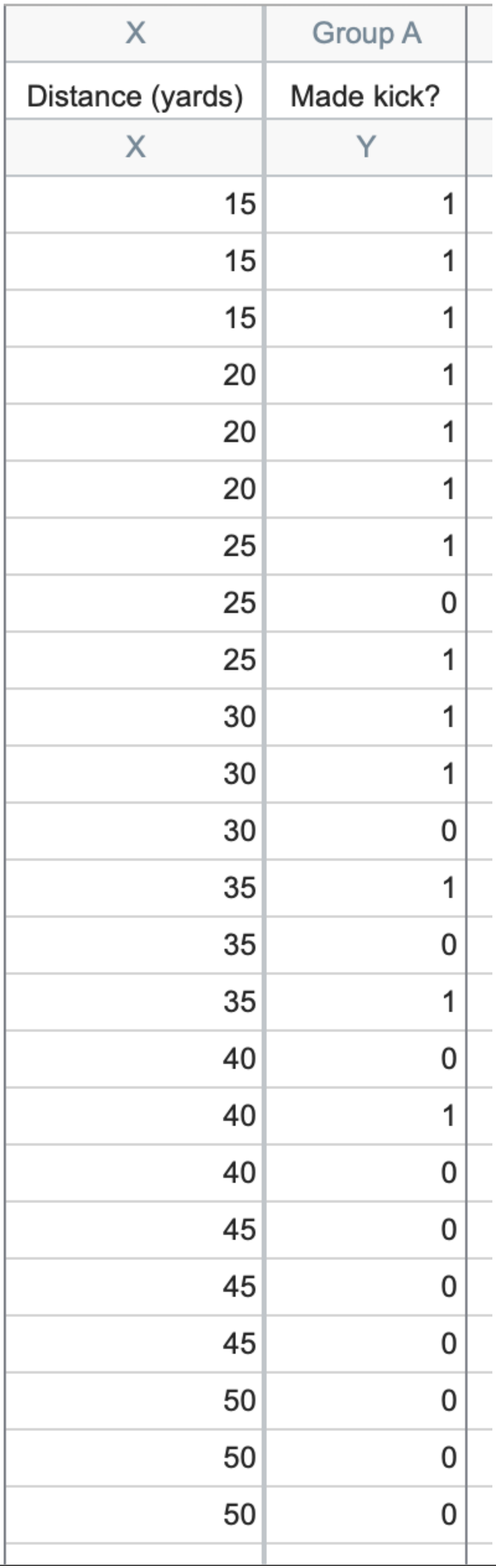


X	Group A
Distance (yards)	Made kick?
X	Y
15	1
20	1
25	1
30	1
35	0
40	0
45	0
50	0

You can see that the data seem to support your hypothesis that distance plays a part in whether or not the field goal attempt will be successful: shorter kicks were successful while longer attempts were missed. In fact, you can see that all attempts over 30 yards were missed. This data represents perfect separation. When describing perfect separation on another page 247, it's pointed out that perfect separation isn't *always* necessarily a bad thing. It could simply mean that you have a variable (in this case distance) that you can use to perfectly predict the outcome (in this case whether or not the field goal attempt was successful).

But there's a problem...

In this example, it's entirely possible that our field goal kicker CAN make field goals from over 30 yards, and we simply didn't collect enough data. If we repeated this experiment, but collected results from multiple attempts at each distance (again, assuming each attempt is actually independent of the others), our data might look something like this:



X	Group A
Distance (yards)	Made kick?
X	Y
15	1
15	1
15	1
20	1
20	1
20	1
25	1
25	0
25	1
30	1
30	1
30	0
35	1
35	0
35	1
40	0
40	1
40	0
45	0
45	0
45	0
50	0
50	0
50	0

These data no longer present an issue with perfect separation, and provide us with a much better idea of how well the kicker might perform at various distances. In fact, using this data, we can determine that our kicker has even odds (1:1 odds or a probability of success = 50%) of successfully kicking a field goal at just over 34 yards (this distance is reported by Prism for this data as "X at 50%").

4.3 Interpolating from a standard curve

Prism makes it very easy to interpolate unknown values from a standard curve. Enter the standards with both X and Y values, fit a line or curve, and Prism will tell you which X values correspond to Y values you entered on the same data table.

- Key concept: Interpolating 258
- How to interpolate 260
- Equations used for interpolating 266
- Where are the interpolated results? 269
- When X values are logarithms 273
- Analysis checklist: Interpolating 274
- Q&A: Interpolating 279
- Standard Addition Method 284

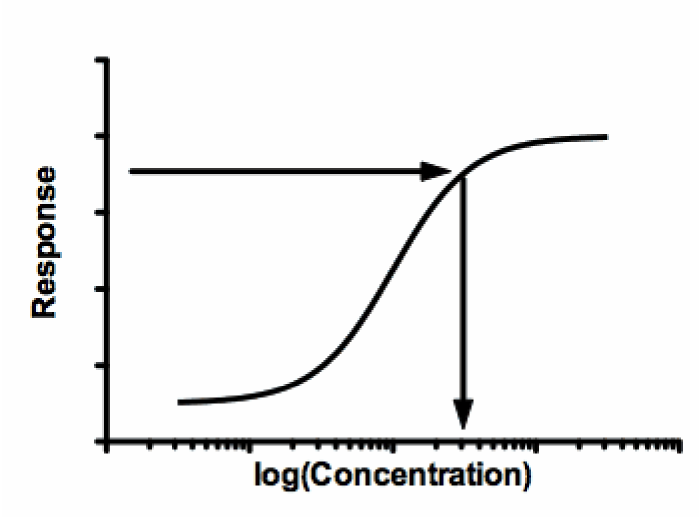
4.3.1 Key concept: Interpolating

A common use of nonlinear (and linear) regression is interpolating.The basic idea is simple:

1. Enter a standard curve, also called a calibration curve. Each point consists of a concentration (or logarithm of concentration) and a response. The responses are commonly radioactivity, fluorescence, or optical density. Standard curves are sometimes called *calibration curves*.

2. Fit a line or curve through the standards. For this purpose, you don't much care about the meaning of the model, or the interpretation of the best-fit values of the parameters (EC50, etc.).

3. For unknown samples, you know the response, but not the concentration. So read the concentration from the curve:



Notes:

- This graph shows the response going up as the concentration gets larger. In many cases, the response goes down as the concentration gets larger. Interpolation works in either case.

- Sigmoid shaped dose-response curves are commonly used with many assays. If the X values analyzed by the analysis are the logarithm of concentration, then so are the interpolated values. You need to take the antilogarithm of these values to return to concentration units.

- This example shows the most common use of interpolating. Finding an X value (concentration) for a given Y value. Prism can also interpolate the other way: Finding a Y value for an entered X value.

4.3.2 How to interpolate

Prism is not designed to analyze any particular kind of assay

Prism is a very general program, used by scientists in many fields. It is easy for you to interpolate standard curves using Prism, but you must make some decisions. Unlike some specialized programs, Prism is not set up so you can blindly enter data and get results without thinking.

Entering data for interpolation

Enter the standard curve (also inown as a calibration curve) in the top rows of a table, and then enter the unknown values right below those standards. In most cases, as shown below, you'll enter Y values and leave X blank. Prism can also interpolate the other way. In this case, enter X values and leave Y blank.

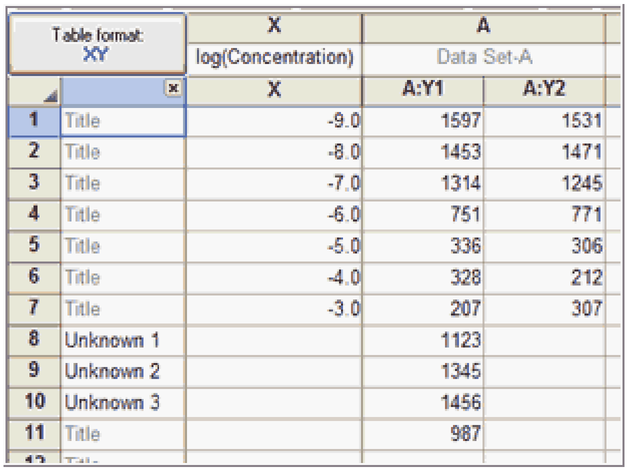


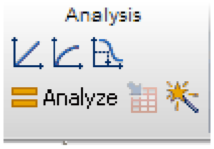
Table format: XY		X	A	
		log(Concentration)	Data Set-A	
		X	A:Y1	A:Y2
1	Title	-9.0	1597	1531
2	Title	-8.0	1453	1471
3	Title	-7.0	1314	1245
4	Title	-6.0	751	771
5	Title	-5.0	336	306
6	Title	-4.0	328	212
7	Title	-3.0	207	307
8	Unknown 1		1123	
9	Unknown 2		1345	
10	Unknown 3		1456	
11	Title		987	

Entering labels for the unknowns, as shown below, is optional.

Note that it is essential that the unknowns go below the standard curve values. If you place the unknowns above the standard curve, Prism will not interpolate. Of course, the unknown Y values must be in the same units as the Y values you entered for the standard curve.

Fitting a standard curve and interpolating

Four analyses in Prism let you interpolate values from curves. The top row of buttons in the analysis block provide shortcuts to linear regression, nonlinlear regression, and the simplified analysis for interpolating a standard curve.



Interpolate a standard curve

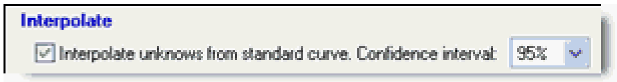
This analysis (new to Prism 6) is a simple way to interpolate using nonlinear regression.

The choices on the dialog are simple. Choose a model to fit to your data. The most commonly used models for standard curves 266 are shown in the dialog. Next choose how to handle outliers. You can choose to use robust regression 94, so outliers have little impact. Or you can ask Prism to identify, and perhaps eliminate, outliers using the ROUT method 92. Finally, if the scatter among replicates is larger when the Y values are larger, consider using relative weighting.

Note that this simplified analysis to interpolate is simply a subset of the nonlinear regression analysis, with fewer choices on the dialog. If you want to select options that aren't available, click the "More.." button at the bottom of the dialog to convert the analysis to the full nonlinear regression analysis, with many more options (six tabs worth). Once you switch to nonlinear regression analysis, it is not possible to switch back (except by starting over).

Nonlinear regression

Nonlinear regression offers the interpolation option at the bottom of the first (Fit) tab of the dialog. This is a good choice if you want to use options not available on the simpler interpolation analysis.


Interpolate

☑ Interpolate unknowns from standard curve. Confidence interval: 95%

Linear regression

Linear regression 120 offers a checkbox to interpolate the top of the dialog.

Spline

Prism can also interpolate from a standard curve in the spline analysis 289. Splines go through every point. The advantage of a spline is that you don't have to choose a model to fit to your data. The disadvantage is that the curve may wiggle too much.

An alternative approach to interpolating one value

If you enter your own (user-defined) equation, or clone an existing one, you can specify that Prism report the value of Y at a particular X, or the X value at a particular Y. 715

4.3.3 How to graph the interpolated values

The table that includes the interpolated values has a green grid which means it can be used to make graphs. While looking at that table, drop New... Graph of existing data. Choose an XY graph and you are done.

4.3.4 Example: Interpolating from a sigmoidal standard curve

1. Create the data table

From the Welcome or New Table dialog, choose to create an XY data table, and select the sample data set: RIA or ELISA. Interpolate unknowns from sigmoidal curve.

2. Inspect the data

The sample data may be partly covered by a floating note explaining how to fit the data (for people who are not reading this help page). You can move the floating note out of the way, or minimize it.

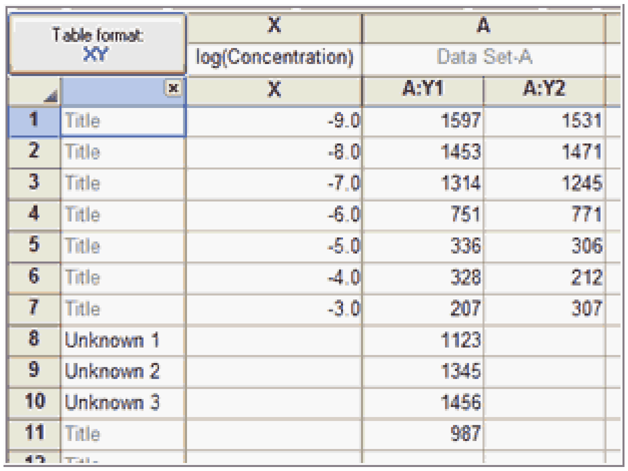
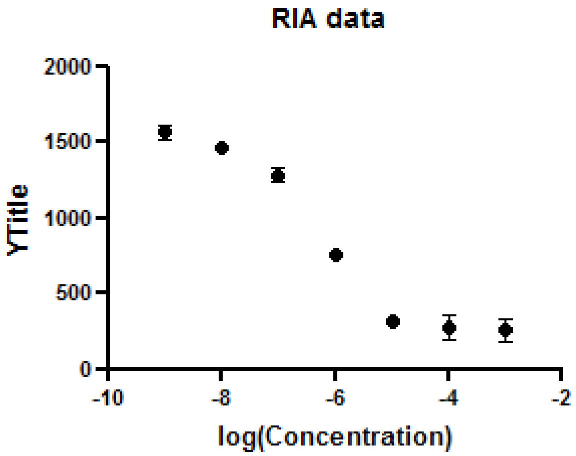


Table format: XY		X	A	
		log(Concentration)	Data Set-A	
		X	A:Y1	A:Y2
1	Title	-9.0	1597	1531
2	Title	-8.0	1453	1471
3	Title	-7.0	1314	1245
4	Title	-6.0	751	771
5	Title	-5.0	336	306
6	Title	-4.0	328	212
7	Title	-3.0	207	307
8	Unknown 1		1123	
9	Unknown 2		1345	
10	Unknown 3		1456	
11	Title		987	

The first seven rows contain the standard curve, in duplicate. Below that are three unknown values. These have a Y values that you measured, but no X. The goal of this analysis is to interpolate the corresponding X values (concentrations) for these unknowns. Note that three of the four unknowns are labeled, so you can later match up the results with the labels.

Why are X values negative? Because in this example, the X values are the logarithm of concentrations expressed in molar. So a concentration of 1 micromolar (10^{-6} Molar) would be entered as -6.

3. View the graph



The graph Prism makes automatically is fairly complete. You can customize the symbols, colors, axis labels, etc. You can also choose to plot the individual duplicates rather than plot the means. Since the unknowns have no X value, they are not included on the graph.

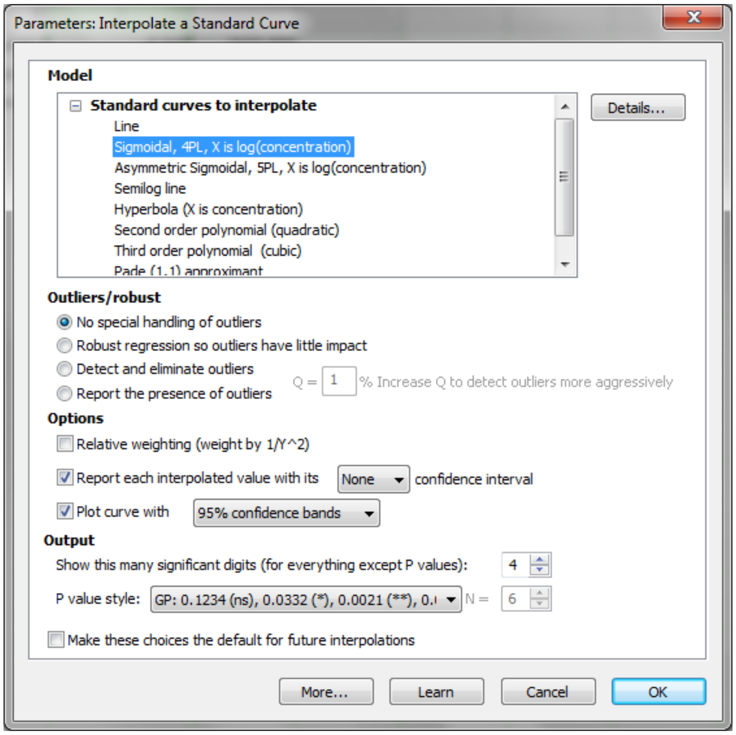
4. Choose the standard curve analysis

Click the Analyze button and from the list of XY analyses choose: Interpolate a Standard Curve.

Alternatively, you can click the “Interpolate a standard curve” button right on top of the Analyze button.

5. Choose a model

Choose the equation: Sigmoidal, 4PL, X is log(Concentration). Note that 4PL means four parameter logistic, which is another name for this kind of equation.



Parameters: Interpolate a Standard Curve

Model

⊟ **Standard curves to interpolate**
- Line
- Sigmoidal, 4PL, X is log(concentration)
- Asymmetric Sigmoidal, 5PL, X is log(concentration)
- Semilog line
- Hyperbola (X is concentration)
- Second order polynomial (quadratic)
- Third order polynomial (cubic)
- Pade (1,1) approximant

Details...

Outliers/robust

● No special handling of outliers
○ Robust regression so outliers have little impact
○ Detect and eliminate outliers
○ Report the presence of outliers
Q = 1 % Increase Q to detect outliers more aggressively

Options

□ Relative weighting (weight by 1/Y^2)
☑ Report each interpolated value with its None confidence interval
☑ Plot curve with 95% confidence bands

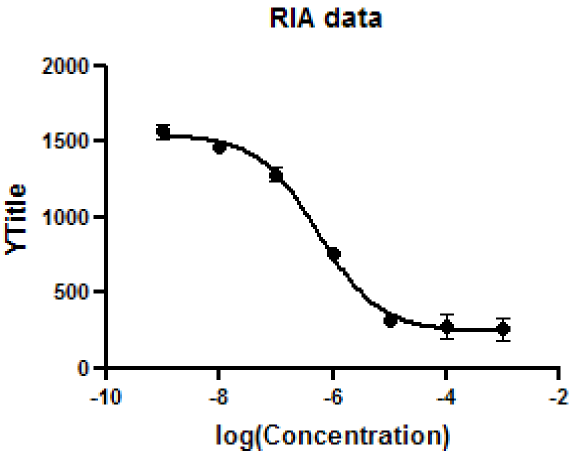
Output

Show this many significant digits (for everything except P values): 4
P value style: GP: 0.1234 (ns), 0.0332 (*), 0.0021 (**), 0.(N = 6

□ Make these choices the default for future interpolations

For this example, leave all the other settings to their default values.

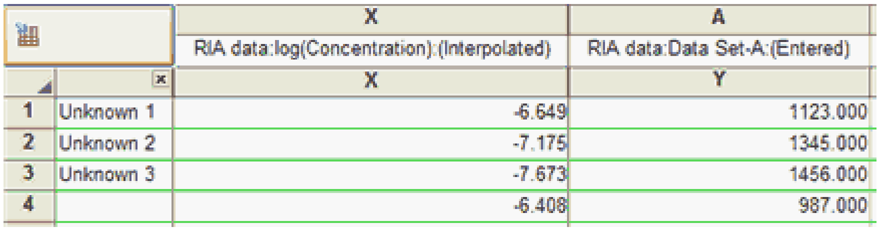
Click OK to see the curves superimposed on the graph.



6. Inspect the results

The results appear on several pages.

The first page shows you the interpolated values.



		X	A
		RIA data:log(Concentration):(Interpolated)	RIA data:Data Set-A:(Entered)
		X	Y
1	Unknown 1	-6.649	1123.000
2	Unknown 2	-7.175	1345.000
3	Unknown 3	-7.673	1456.000
4		-6.408	987.000

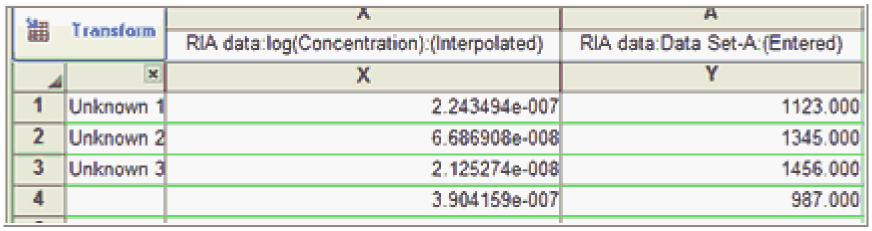
The second page is the table of results for the overall curve fit. It tabulates the best-fit values of the parameters and much more. For this example, we aren't too interested in these results.

7. Transform the results

The X column of the results table has the interpolated values we want. These are in the same units as the X values, so are the logarithm of concentration. Prism can transform these values to concentration units.

Click the Analyze button and choose Transform at the top of the Analyze dialog.

On the Transform dialog check the option to transform X values and choose the transform X=10^X



Transform		X	A
		RIA data:log(Concentration):(Interpolated)	RIA data:Data Set-A:(Entered)
		X	Y
1	Unknown 1	2.243494e-007	1123.000
2	Unknown 2	6.686908e-008	1345.000
3	Unknown 3	2.125274e-008	1456.000
4		3.904159e-007	987.000

Now the X column is in molar concentration units. Note that the column title hasn't changed. Prism isn't smart enough to adjust the column titles when you transform data.

Click and edit the column title to "Concentration (M)".

4.3.5 Equations used for interpolating

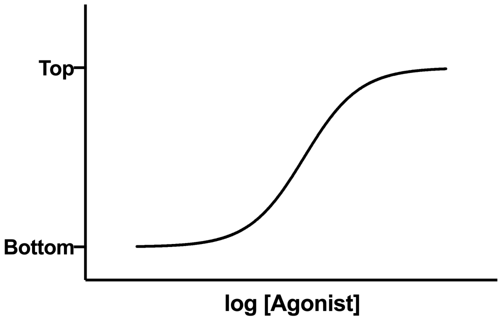
You can interpolate from any curve fit through a set of standards. However, the equations listed below are used most commonly. These equations are instantly available from the interpolation analysis. You'll also find this set of equations in the equation tree used in nonlinear regression.

Notes:

- These graphs show the curve going up (increasing Y) as X increases. But all will work for downward sloping curves too.

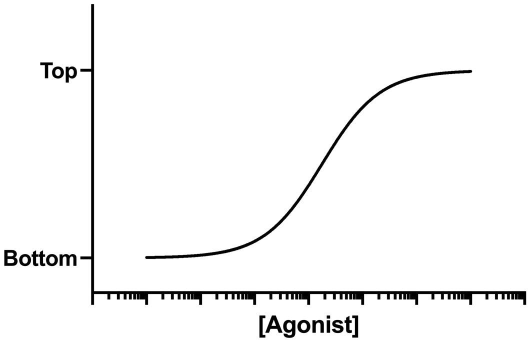
- All of these equations (except Pade) are also present in other equation folders in the nonlinear regression dialog.

Sigmoidal, 4PL, X is log(concentration) 473



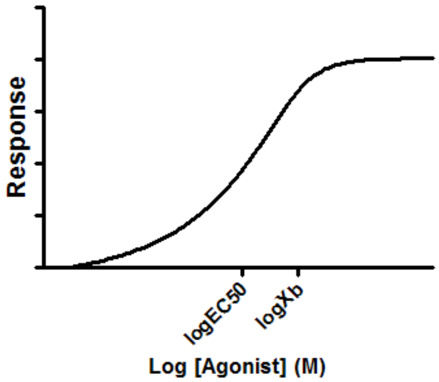
Note that in nonlinear regression this equation is called "log(agonist) v

Sigmoidal, 4PL, X is concentration 481



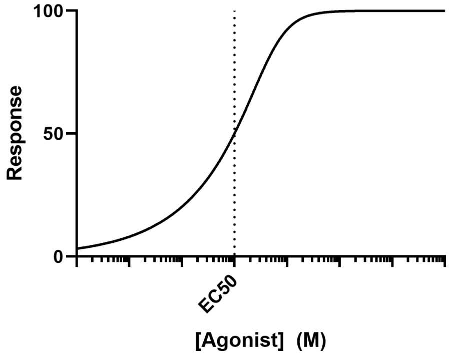
Note that in nonlinear regression this equation is called "[Agonist] vs. r

Asymmetric Sigmoidal, 5PL, X is log(concentration) 529



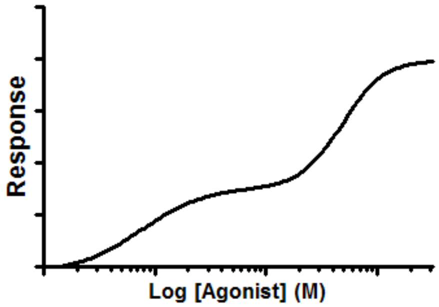
Note that in nonlinear regression this equation is called "Asymmetrical

Asymmetric Sigmoidal, 5PL, X is concentration 502

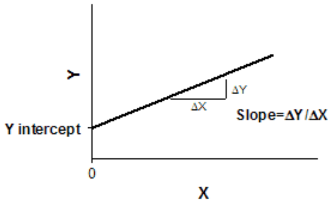


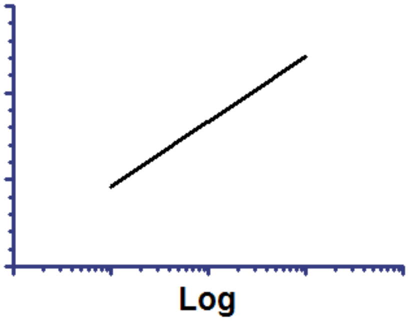
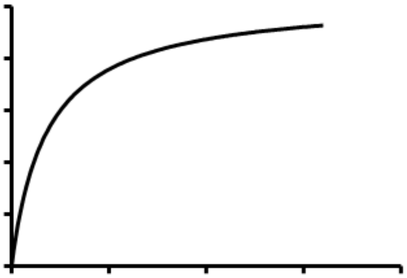
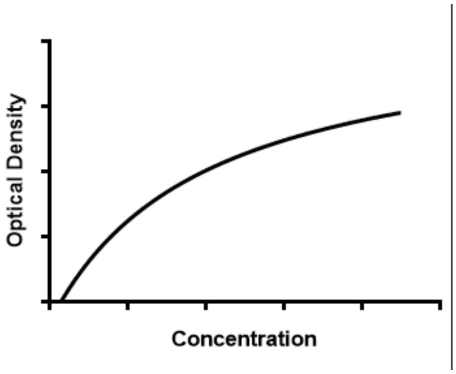
Note that in nonlinear regression this equation is called "Asymmetrical

Biphasic 532



Straight line 653

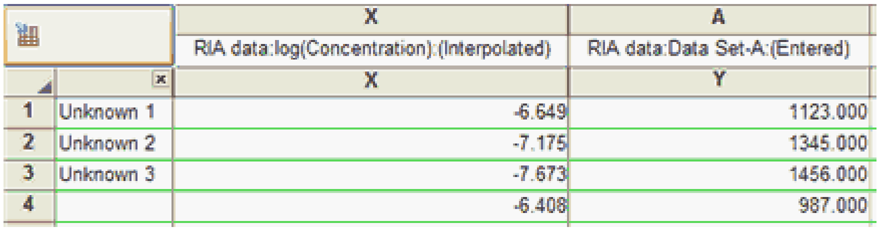


Semilog line -- X is log, Y is linear 661	
Hyperbola (X is concentration) 568	
Padé (1,1) approximant	
Second order polynomial (quadratic) 672	Y=B0 + B1*X + B2*X^2
Third order polynomial (cubic) 672	Y=B0 + B1*X + B2*X^2 + B3*X^3

4.3.6 The results of interpolation

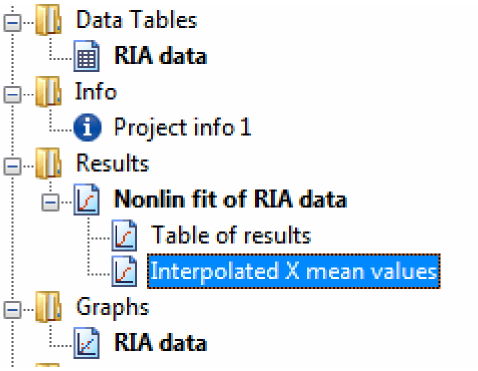
Where are the interpolated results?

Prism places the interpolation results into its an additional results page (or two additional pages).



		X	A
		RIA data:log(Concentration):(Interpolated)	RIA data:Data Set-A:(Entered)
		X	Y
1	Unknown 1	-6.649	1123.000
2	Unknown 2	-7.175	1345.000
3	Unknown 3	-7.673	1456.000
4		-6.408	987.000

To view this table, click the interpolated results page of the results. In some cases you may first need to click the + in front of the results main page to see all the pages within.



- Data Tables
  - **RIA data**
- Info
  - Project info 1
- Results
  - **Nonlin fit of RIA data**
    - Table of results
    - Interpolated X mean values
- Graphs
  - **RIA data**

What units are the results in?

The interpolated results are expressed in the same units as the standard curves. If you entered a Y value, and Prism interpolated an X value, then that X value is in the same units as the standards you entered.

If the X values in the table are the logarithm of concentration (common), then the interpolated results are also logarithms of concentration.

In some assays, different samples are diluted differently. Prism does not correct for dilution, so you'll need to make those corrections manually.

4.3.7 Interpolating with replicates in side-by-side subcolumns

Choices when entering data with replicates in subcolumns

It is common to enter replicate values for the unknowns, as shown below.

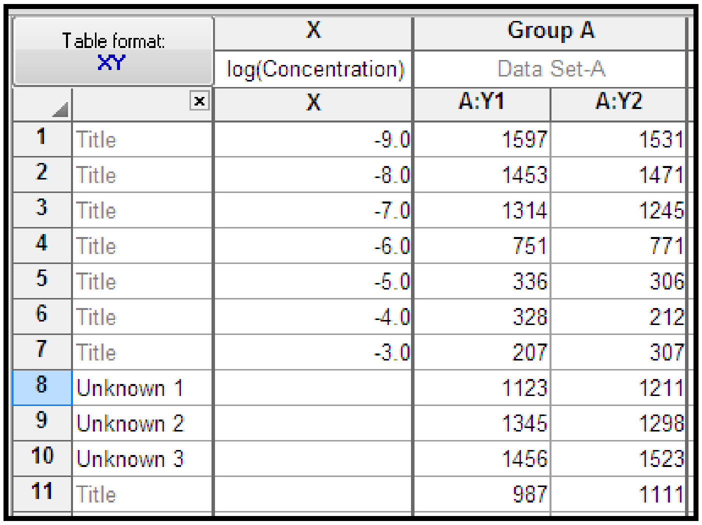
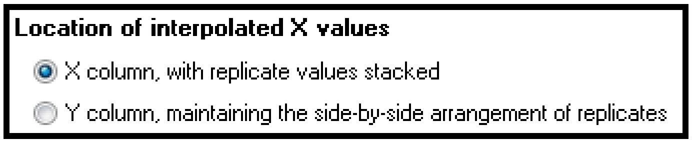


Table format: XY		X	Group A	
		log(Concentration)	Data Set-A	
	☒	X	A:Y1	A:Y2
1	Title	-9.0	1597	1531
2	Title	-8.0	1453	1471
3	Title	-7.0	1314	1245
4	Title	-6.0	751	771
5	Title	-5.0	336	306
6	Title	-4.0	328	212
7	Title	-3.0	207	307
8	Unknown 1		1123	1211
9	Unknown 2		1345	1298
10	Unknown 3		1456	1523
11	Title		987	1111

Since Prism's results tables cannot be configured to put side-by-side replicates in the X column, you have two choices for how to organize the results. These choices are available at the bottom of the Output tab of the nonlinear regression dialog.



Location of interpolated X values

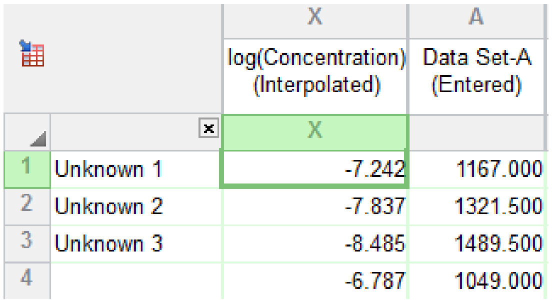
● X column, with replicate values stacked

○ Y column, maintaining the side-by-side arrangement of replicates

Place computed unknown values into the X column, with replicate values stacked

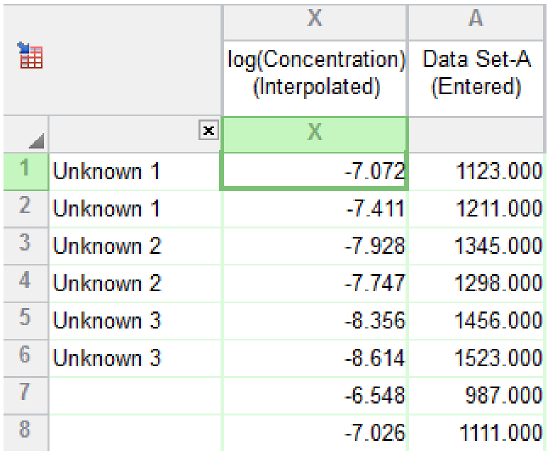
In this case, Prism creates two tabs of results. The interpolation is the same for both tables. Prism interpolates each replicate value individually.

The first tab, Interpolated Mean X Values, shows the mean of the Y values you entered and the mean of the interpolated X values. The X column shows the concentrations (or logarithm of concentrations) to match the data you entered.



		X	A
		log(Concentration) (Interpolated)	Data Set-A (Entered)
		X	
1	Unknown 1	-7.242	1167.000
2	Unknown 2	-7.837	1321.500
3	Unknown 3	-8.485	1489.500
4		-6.787	1049.000

The second tab, Interpolated X replicates, shows individual replicate values for the interpolated results. The X column maintains the same meaning as it does in the data table you entered. To do so, with only a single X column, requires that the replicates be stacked.

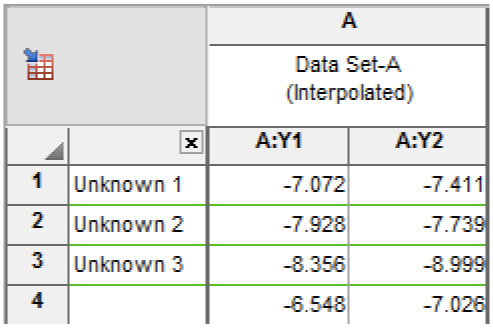


		X	A
		log(Concentration) (Interpolated)	Data Set-A (Entered)
		X	
1	Unknown 1	-7.072	1123.000
2	Unknown 1	-7.411	1211.000
3	Unknown 2	-7.928	1345.000
4	Unknown 2	-7.747	1298.000
5	Unknown 3	-8.356	1456.000
6	Unknown 3	-8.614	1523.000
7		-6.548	987.000
8		-7.026	1111.000

Place computed unknown values into a Y data set, maintaining the side-by-side arrangement of replicates.

When you make this choice, the interpolated values appear in the Y column, and there is no X column. You can view the interpolated values, with their name (if you entered row titles). But the Y values you entered are not shown.

The first page shows the mean values. The second page shows the interpolated values in side-by-side replicates, to match the format you entered them. Even though these values are interpolated X values, note that the results are in the Y column.

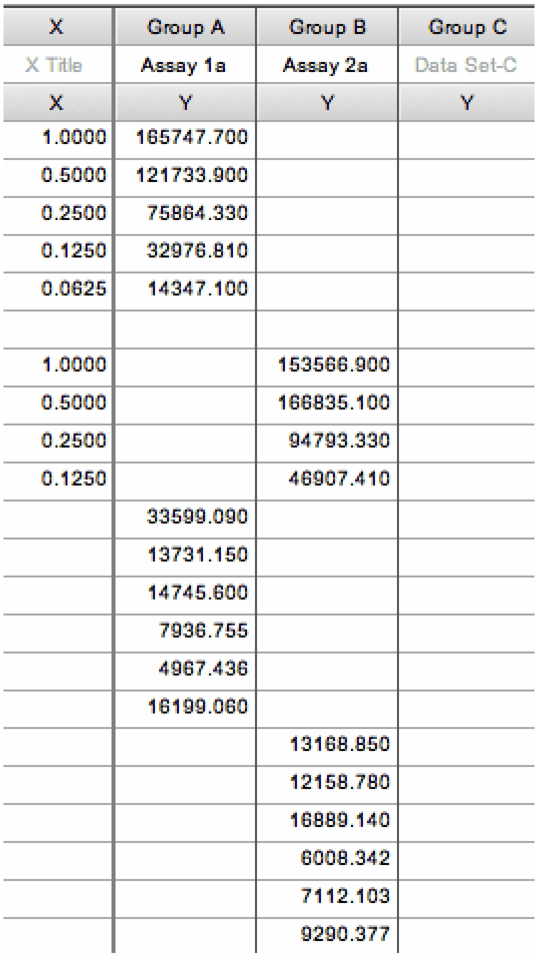


		A	
		Data Set-A (Interpolated)	
		A:Y1	A:Y2
1	Unknown 1	-7.072	-7.411
2	Unknown 2	-7.928	-7.739
3	Unknown 3	-8.356	-8.999
4		-6.548	-7.026

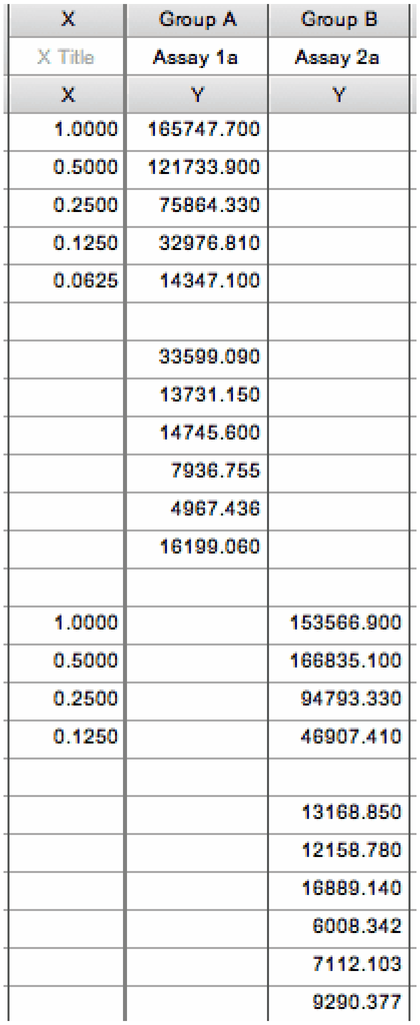
4.3.8 Interpolating several data sets at once

Prism can interpolate from several data sets at once. To do this, put all the unknowns on lower rows on the data table than all the knowns (left panel). Prism won't be able to interpolate past the first data set if you put the second data set below the first set of knowns (right panel).

Yes! All unknowns at the bottom. **No! Unknowns everywhere.**



X	Group A	Group B	Group C
X Title	Assay 1a	Assay 2a	Data Set-C
X	Y	Y	Y
1.0000	165747.700		
0.5000	121733.900		
0.2500	75864.330		
0.1250	32976.810		
0.0625	14347.100		
1.0000		153566.900	
0.5000		166835.100	
0.2500		94793.330	
0.1250		46907.410	
	33599.090		
	13731.150		
	14745.600		
	7936.755		
	4967.436		
	16199.060		
		13168.850	
		12158.780	
		16889.140	
		6008.342	
		7112.103	
		9290.377	



X	Group A	Group B
X Title	Assay 1a	Assay 2a
X	Y	Y
1.0000	165747.700	
0.5000	121733.900	
0.2500	75864.330	
0.1250	32976.810	
0.0625	14347.100	
	33599.090	
	13731.150	
	14745.600	
	7936.755	
	4967.436	
	16199.060	
1.0000		153566.900
0.5000		166835.100
0.2500		94793.330
0.1250		46907.410
		13168.850
		12158.780
		16889.140
		6008.342
		7112.103
		9290.377

Why can't Prism handle the kind of data shown in the right panel above? Because then it might get mixed up about which values are unknowns to be interpolated, and which are simply adjacent to values left blank because they are missing.

4.3.9 When X values are logarithms

In many situations, you'll fit the standard curve to a log(dose) vs. response curve. In these cases, you'll enter the X values for the standard curve as the logarithm of concentration or you'll transform the data to logarithms before running nonlinear regression.

The interpolated X values are in the same units as the X values the analysis sees. If you are analyzing data entered or transformed so X is the logarithm of concentration, then the interpolated values will also be logarithms.

If you want to transform these results back to concentrations, that is easy to do:

1. From the interpolated results page of results, click Analyze.

2. Choose Transform at the very top of the Analyze dialog.

3. Choose the standard list of transforms.

4. Depending on whether your interpolated values are in the X or Y column, 269 choose to transform that column.

5. From the list of transforms, choose Y=10^Y or X=10^X. These choices are about one third of the way down the long list of transforms.

6. The results sheets will have the interpolated values as concentrations, rather than logs.

Why 10^Y or 10^X? Those are abbreviations for 10^Y or 10^X , which are the antilogarithm functions. Review logarithms and antilogarithms.

4.3.10 Analysis checklist: Interpolating

Your approach in evaluating nonlinear regression depends on your goal.

In many cases, your goal is to learn from the best-fit values. If that is your goal, view a different checklist 422.

If your goal is to create a standard curve from which to interpolate unknown values, your approach depends on whether this is a new or established assay.

Established assay

If the assay is well established, then you know you are fitting the right model and know what kind of results to expect. In this case, evaluating a fit is pretty easy.

✓ **Does the curve go near the points?**

✓ Is the R^2 'too low' compared to prior runs of this assay?

If so, look for outliers, or use Prism's automatic outlier detection.

✓ Are the confidence bands too wide?

The confidence bands 352 let you see how accurate interpolations will be, so we suggest always plotting prediction bands when your goal is to interpolate from the curve. If your are running an established assay, you know how wide you expect the prediction bands to be.

New assay

With a new assay, you also have to wonder about whether you picked an appropriate model.

✓ Does the curve go near the points?

Look at the graph. Does it look like the curve goes near the points.

✓ Are the confidence bands too wide?

How wide is too wide? The prediction bands show you how precise interpolations will be. Draw a horizontal line somewhere along the curve, and look at the two places where that line intercepts the confidence bands. This will be the confidence interval for the interpolation.

✓ Does the scatter of points around the best-fit curve follow a Gaussian distribution?

Least squares regression is based on the assumption that the scatter of points around the curve follows a Gaussian distribution. Prism offers three normality tests (in the Diagnostics tab) that can test this assumption (we recommend the D'Agostino test). If the P value for a normality test is low, you conclude that the scatter is not Gaussian.

✓ Could outliers be impacting your results?

Nonlinear regression is based on the assumption that the scatter of data around the ideal curve follows a Gaussian distribution. This assumption leads to the goal of minimizing the sum of squares of distances of the curve from the points. The presence of one or a few outliers (points much further from the curve than the rest) can overwhelm the least-squares calculations and lead to misleading results.

You can spot outliers by examining a graph (so long as you plot individual replicates, and not mean and error bar). But outliers can also be detected automatically. GraphPad has developed a new method for identifying outliers we call the ROUT method. You can either ask Prism to simply identify outliers or to eliminate them. These choices are adjacent on the Interpolate a Standard Curve dialog. But they are in separate tabs in the Nonlinear Regression dialog. The option to count outliers is on the Diagnostics tab, and the the option to exclude outliers is on the Fit tab.

✓ Does the curve deviate systematically from the data?

If either the runs test or the replicates test yields a low P value, then you can conclude that the curve doesn't really describe the data very well. You may have picked the wrong model.

4.3.11 Reasons for blank (missing) interpolated results

Reasons for all interpolated values to be blank

If the nonlinear regression procedure did not converge[437] on a best-fit curve, then Prism will leave all interpolated values blank. It doesn't make sense to interpolate unless Prism finds a good curve.

Reasons for some interpolated values to be blank

When you ask Prism to interpolate X values from Y values you enter, it will leave the result blank when it is unable to find a result. This can happen when:

- The input value for Y falls out of the vertical range of the standard curve. This happens when you fit a sigmoidal log(dose) vs response curve. If you enter an unpaired Y value that is larger than the best-fit value of Top, or smaller than the best-fit value of Bottom, there simply is no corresponding X value, and Prism will simply leave that part of the Results sheet blank. Note that "top" and "bottom" apply to the best-fit values for those fitted parameters, not the maximum and minimum values of the input data. It simply is impossible to interpolate a X value if the Y value is too high, or too low, to be on the curve.

- The interpolated X value falls way outside the output range of X for the fitted curve. Prism will try to extrapolate a bit beyond the range of X values for which it calculates and draws the curve. But it won't extrapolate far from the ends of the curve. You can extend this range. From the Nonlinear regression Parameters dialog, choose the Range tab and enter a lower value for the minimum X and/or a higher value for the maximum X.

- You entered the unknowns in rows of the data table above (smaller row numbers) than the standard curve values. The unknowns must be entered in rows below the standards.

- You mistakenly entered the unknowns in the wrong units. The unknown values must be in the same units as the standards. If you enter unknown Y values and expect Prism to interpolate an X value, then the unknown Y values must be entered in the same units as the Y values for the standards.

Reasons for the confidence limits of interpolated value is blank

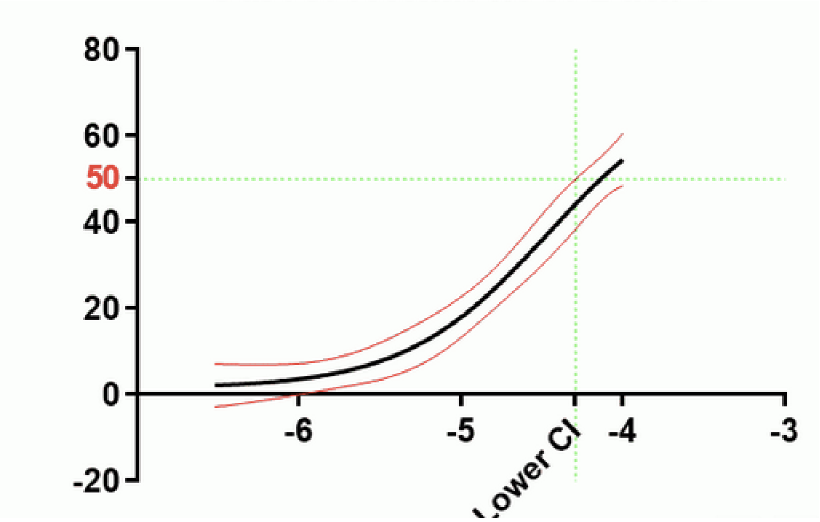
Prism will leave the confidence interval of the interpolated value blank in two situations:

- The fit is ambiguous 437.

- If you chose a robust fit.

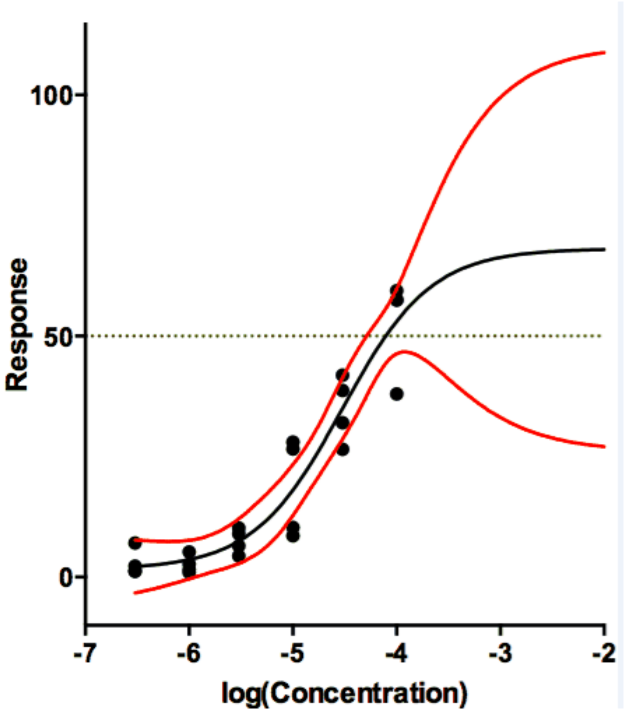
In some cases, Prism will report confidence limits for some interpolated values, but not all of them.

Prism can only report the confidence interval for an interpolated value when the curve is drawn far enough along the X axis. The graph below shows an example where Prism can report the lower confidence limit but not the upper one. The horizontal green line shows the interpolation. You asked Prism for the X value when Y=50. The interpolation works fine as does the lower confidence limit. But notice that the lower confidence band (in red) does not extend far enough to intersect the horizontal green line. That is why Prism can't report the upper confidence limit which would be the X coordinate where the lower confidence band of the curve intersects the horizontal line at Y=50. To fix this problem, go to the Range tab 341 and extend the curve to larger values of X. In the example below, extending the range of the curve to -3 would solve the problem.



Reason why one confidence limit of an interpolated value is blank

In some cases, the bottom confidence band bends downwards, so the upper confidence limit of the interpolated value is simply undefined. The figure below shows one such example. It doesn't matter how far you extend the X axis, the upper confidence limit of the interpolated value is simply not defined by the data. The dotted horizontal line at Y=50 will never intersect the lower of the confidence bands. The only fix would be to collect data at higher X concentrations, or perhaps more precise Y values.



4.3.12 Q&A: Interpolating

Expand all

Collapse all

⊟ **Can Prism interpolate several data sets at once?**

Yes, Prism can interpolate X values from Y values entered into several data sets. Note the following:

- Prism has to figure out which values are unknowns. It uses this rule: X(or Y) values are considered as unpaired (and are used for calculation of unknowns) when they are lower in the table than all XY pairs in the data table. It is not enough that the unpaired Y values (from which you want Prism to interpolate X) is lower on the table than the paired XY values for that particular data set. They must be on lower rows in the table than all the paired XY values of all data sets in the table.

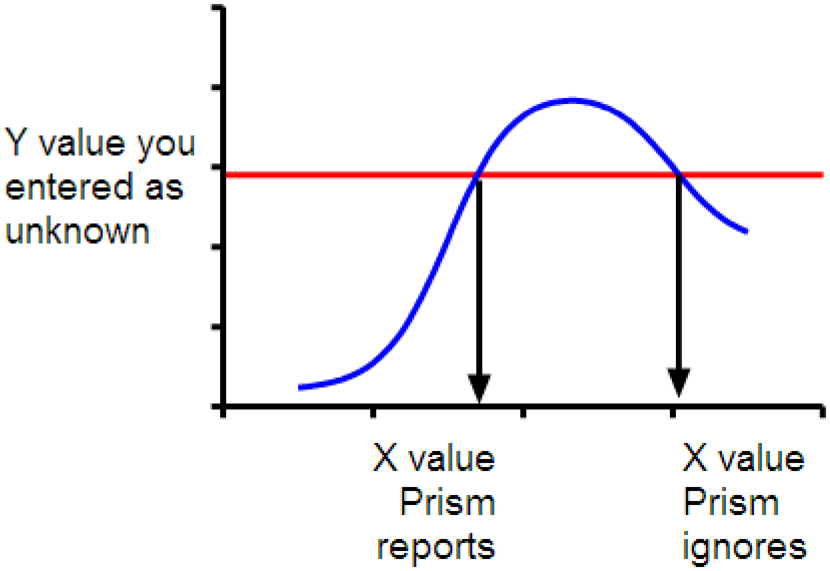
- Prism offers an option in the Output tab 269 of nonlinear regression that let you choose where the interpolated X values should go (into the X column staggered, or onto Y columns). If you have replicates and have unknowns in more than one data set, choose the option to show the unknowns in Y columns.

⊟ What units are the interpolated values expressed in?

When you enter a Y value, and ask Prism to interpolate the X value, that X value is in the same units as the X values you entered. Usually this is concentration. If it is the logarithm of concentration, you can transform it back to a concentration scale. 273

⊟ What happens when there are two X values corresponding to the entered Y value?

In some cases, there are two X values that correspond to a Y value you enter. To understand how Prism handles this situation, you need to know how it interpolates. It starts at the minimum X value and works its way to the largest X value. It stops when it finds the Y value you entered. If it doesn't find a Y value matching the one you entered, it then looks at X values lower than the minimum shown on the graph and to X values larger than the maximum. If there are two points on the curve that have the Y value you entered as an unknown, Prism will find the lower of the two corresponding X values.



⊟ Can Prism plot the interpolated values?

Sure. The interpolated values are tabulated on a results table. If you want to plot only these values, start from this table, click the New button, and choose New Graph of Existing data. If you want to add these interpolated values to the graph of the unknowns and its fit, go to that graph, double-click to go to Format Graph, go to the middle (Data on graph) tab, and choose to add the table of interpolated values.

4.3.13 Why Prism doesn't attempt to find a 'linear range'

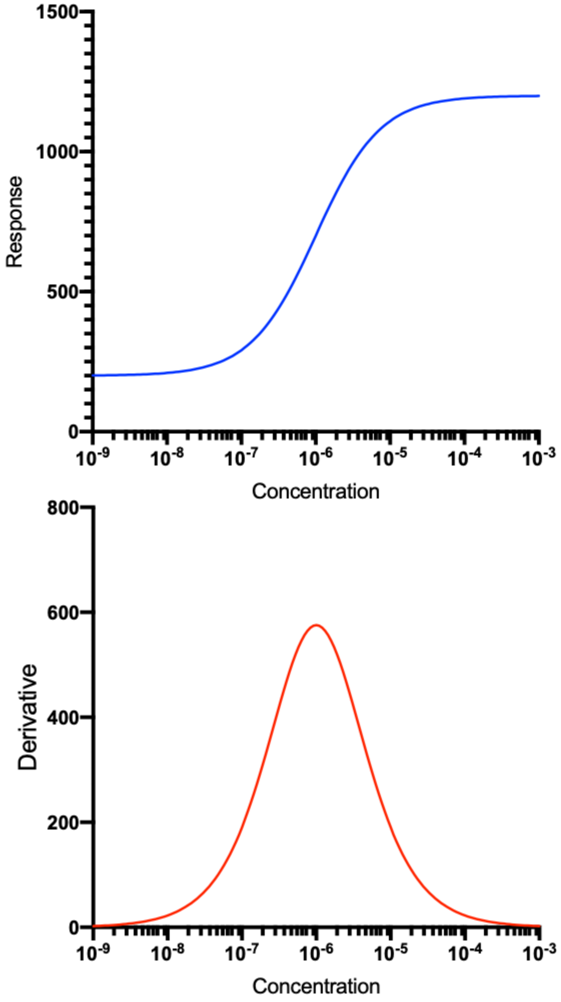
Prism can interpolate values from linear or nonlinear standard curves (calibration curve). When the standard curve is nonlinear, people sometimes ask whether Prism can determine a section in the middle of the curve that is linear -- the "linear range".

Prism does not help you determine a "linear range" for several reasons.

- There is no need to restrict analysis to a linear range. If you fit a curve with nonlinear regression, you can interpolate from all parts of the curve. It doesn't make sense to define a somewhat arbitrary linear range, and then analyze just those values with linear regression.

- You can't use a "linear range" as a way to make sure that the interpolated value is precise. Even with linear regression, interpolating in the middle of the concentrations of the standards is more precise than interpolating near the highest or lowest concentrations. With both linear and nonlinear regression, Prism can report a confidence interval of the interpolated values.

- There is no clear definition of a "linear range". Any such definition is arbitrary. The graph below shows an ideal log(concentration) vs. response curve. It looks like there is a "linear range" in the middle of the curve. But the lower graph shows the derivative (slope) of the curve. If there were a linear part of the curve, the derivative curve would be flat (horizontal) where the slope of the curve doesn't change (because it is "linear"). Because there is no flat portion of the derivative curve, any definition of "linear portion" would be arbitrary.



4.3.14 How Prism interpolates

If you ask Prism to compute Y for a given X, it does so by evaluating the equation that defines the curve. That is very straightforward.

If you ask Prism to compute X for a given Y, it is more complicated. Prism does not try to solve the equation algebraically but rather does the interpolation numerically. The results are accurate to at least 6-7 decimal places.

Prism decides on the range of X values to consider. To allow for extrapolation a bit beyond the range of the data, Prism creates an interpolation/extrapolation range that includes the range of the data, and extends in each direction by a distance equal to half the difference between Xmax and Xmin. There are two special cases. When all the data are positive (or zero), that range is clipped to exclude negative numbers. Similarly, when the data are all negative (or zero), the interpolation/extrapolation range is clipped to exclude positive numbers.

Prism then divides that interpolation/extrapolation range into 1000 line segments.

For each value to be interpolated or extrapolated, Prism first tries to interpolate within the range of the X values of the data.

1. It starts with the lowest X value (scans from left to right on the graph). If more than one line segment includes the Y value, Prism only finds the first (lowest X value).

2. Prism then interpolates within that line segment to determine X as accurately as possible. In most cases, it does this by binary bisection. It divides the segment in half and figures out which contains Y. Then it divides that half in half again. And again. This continues until X is determined as accurately as possible given the numerical precision of the computer. In rare cases, it is possible that Y is not monotonic within the range determined in step 1. In this case, Prism uses linear interpolation rather than binary bisection.

If Prism is not able to interpolate a Y value that corresponds to the entered X value within the X range of the data, it will try to extrapolate a value to the extended range mentioned above. It first looks at X values below the minimum X value in the data, and then at X values above the maximum X value in the data.

When Prism interpolates a value (the X value is in the range of the data), it will also compute a confidence interval if you request it to do so. It does this by determining where the two confidence bands intersect the Y value you entered. When Prism extrapolates a value from the curve, it will not compute a confidence interval for that extrapolated value.

4.3.15 Standard Addition Method

Prism can easily interpolate from a linear or nonlinear standard curve. You perform the assay at a number of known concentrations, fit a line or curve, and interpolate the unknown values.

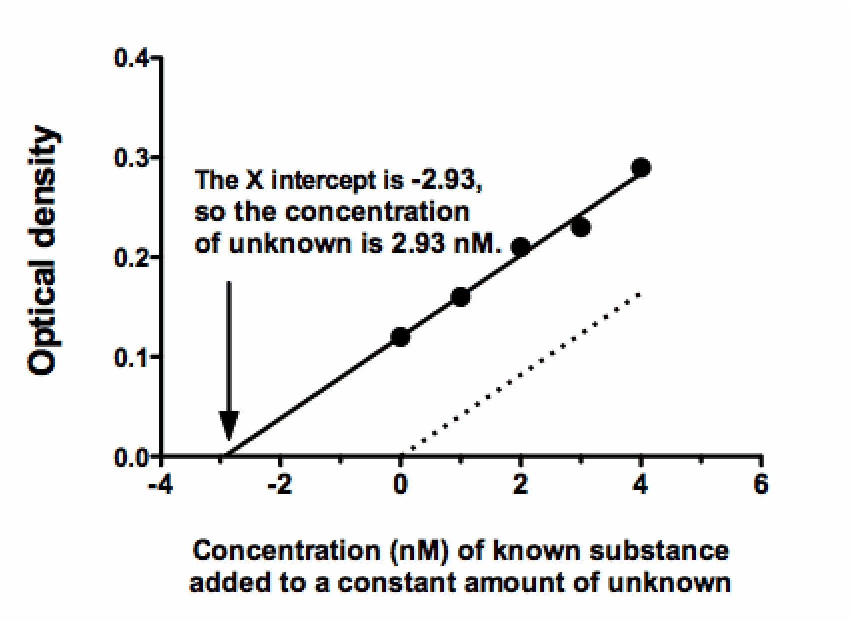
But there is a problem with interpolating from a standard curve. The results can be incorrect when the unknown sample are contaminated with other substances that alter the assay. This is known as the *matrix effect problem*.

The *Standard Addition Method* is a way to bypass this problem. Add various known concentrations (including zero) of known substance to a constant amount of the unknown, and perform the assay. This ensures that all the samples have the same amount of unknown, including any substances that interfere with the assay.

Fit the data with linear regression. The Standard Addition Method only works when the assay measurement (Y) is proportional to the concentration of substance you are measuring.

The dotted line in the figure below shows how the assay results (usually optical density) varies as you change the concentration of the substance you are assaying. The circles and solid line show the assay results in the presence of an unknown concentration of the substance you are assaying. This pushes the line up and to the left.

The value you want to know is how far the solid line is shifted to the left of the dotted line. There is an easy way to find out: Extrapolate the line down to Y=0. One of the parameters that Prism reports is the X intercept, which will be negative. Take the absolute value, and that is the concentration of the unknown substance. The confidence interval for the X intercept gives you the confidence interval for the concentration of the unknown. Simply multiply both confidence limits by -1.



To plot the data in Prism, you'll want to extend the linear regression line to start at an X value equal to the X intercept (a choice in the Linear regression parameters dialog). You may also want to move the origin to the lower left, a choice on the first tab of the Format Axis dialog. Prism file.

4.3.16 Back calculating from a standard curve

What is back calculating?

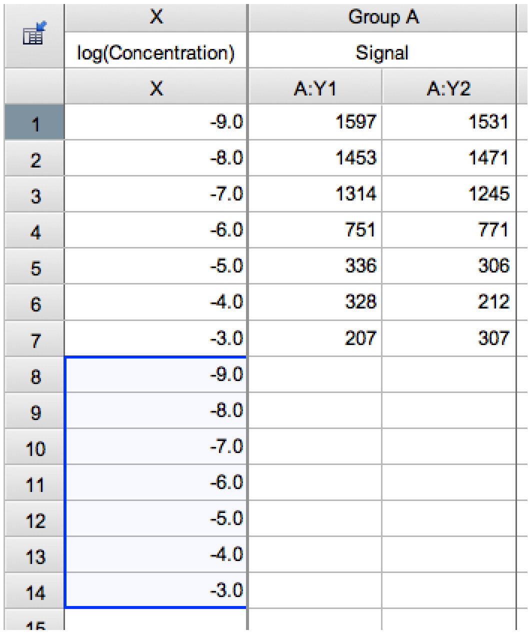
When it "interpolates" an X value from a standard curve, Prism finds the X value corresponding to the entered Y value on the best-fit curve. The line or curve doesn't go exactly through every experimentally measured point, so the interpolated X value for a Y value won't exactly match the observed X value for that Y. Some find it helpful to look at the patterns of how the actual X values differ from the back calculated X values. Prism does not do this automatically, but use these steps to get the job done.

In other words, the goal of back calculating is to interpolate each of the Y values which the model was fit to to find out what X value they correspond to. Then the goal is to compare those interpolated (back calculated) X values with the actual X values.

How to back calculate with Prism

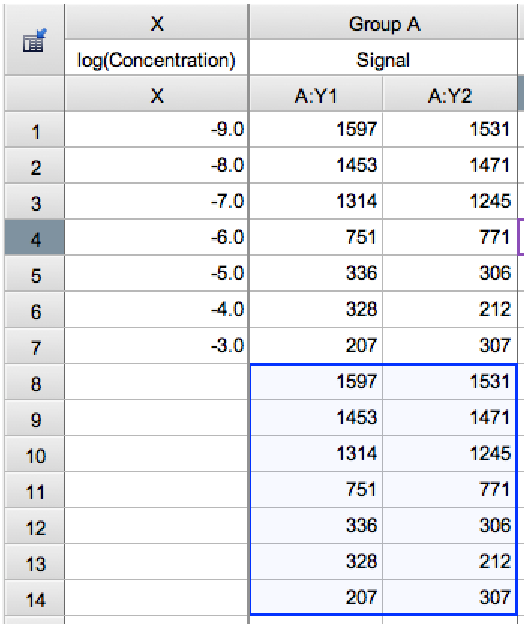
1. Enter the X and Y values for the standard curve.

2. If you want to back calculate Y values from entered X values: Select and copy the X values, and paste them lower on the table. If using Prism 6 or later, use Paste link so these pasted values will update if you edit or replace the original X values. When you paste link, the pasted values are surrounded by a blue box. Note that these rows will have only X values and no Y values.



	X	Group A	
	log(Concentration)	Signal	
	X	A:Y1	A:Y2
1	-9.0	1597	1531
2	-8.0	1453	1471
3	-7.0	1314	1245
4	-6.0	751	771
5	-5.0	336	306
6	-4.0	328	212
7	-3.0	207	307
8	-9.0		
9	-8.0		
10	-7.0		
11	-6.0		
12	-5.0		
13	-4.0		
14	-3.0		
15			

If you want to back calculate X values from entered Y values. Select and copy the Y values, and paste them lower on the table. If using Prism 6 or later, use Paste link so these pasted values will update if you edit or replace the original Y values. When you paste link, the pasted values are surrounded by a blue box. Note that these rows will have only Y values and no X values.



	X	Group A	
	log(Concentration)	Signal	
	X	A:Y1	A:Y2
1	-9.0	1597	1531
2	-8.0	1453	1471
3	-7.0	1314	1245
4	-6.0	751	771
5	-5.0	336	306
6	-4.0	328	212
7	-3.0	207	307
8		1597	1531
9		1453	1471
10		1314	1245
11		751	771
12		336	306
13		328	212
14		207	307

3. Choose nonlinear (or linear) regression and choose the option to interpolate from a standard cuve.

4. Find the results tab with the interpolated values.

5. Make whatever graph or analysis you want to compare the back calculated values with the entered values. There don't appear to be any standard ways to do this.

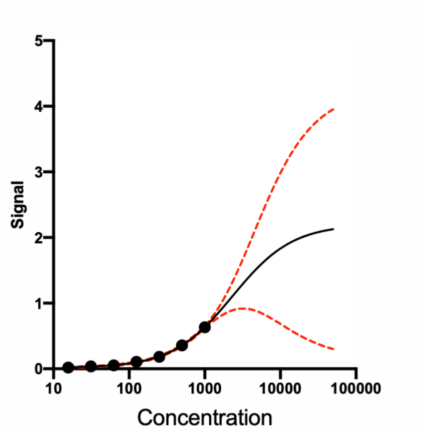
4.3.17 Extrapolating

The same method used to interpolate can also be used to extrapolate. But note:

- Prism will extrapolate only a limited distance beyond the data. Usually, it will extrapolate in each direction up to half the range of the X axis.

- To extrapolate further, extend the fit curve to a wider range of X values. For nonlinear regression, use the Range tab to extend the fit curve

- Even if a curve fits the data well, it can be a bit of a stretch to believe that the same model will be accurate when extrapolated beyond the range of the data. It all depends, of course, on the context of the experiment.

- When extrapolating, it is a good idea to ask Prism to report the 95% CI of the interpolated values (which might be very wide).
- When extrapolating, it is a good idea to plot the curve with the 95% confidence bands, so you can see how precise (or imprecise) the curve is as you get further from the data. The curve below is drawn over a range of X values way beyond the data. As the curve gets further from the data, the confidence bands (red) get wider and wider.
- 

4.4 Fitting a curve without a model

In some circumstances, your goal is simple. You don't care about models, and don't expect best-fit values that you can interpret. Instead, you just want to draw a smooth curve to make a graph look attractive, or to use as a standard curve.

- Spline and Lowess curves 289
- Using nonlinear regression with an empirical model 292

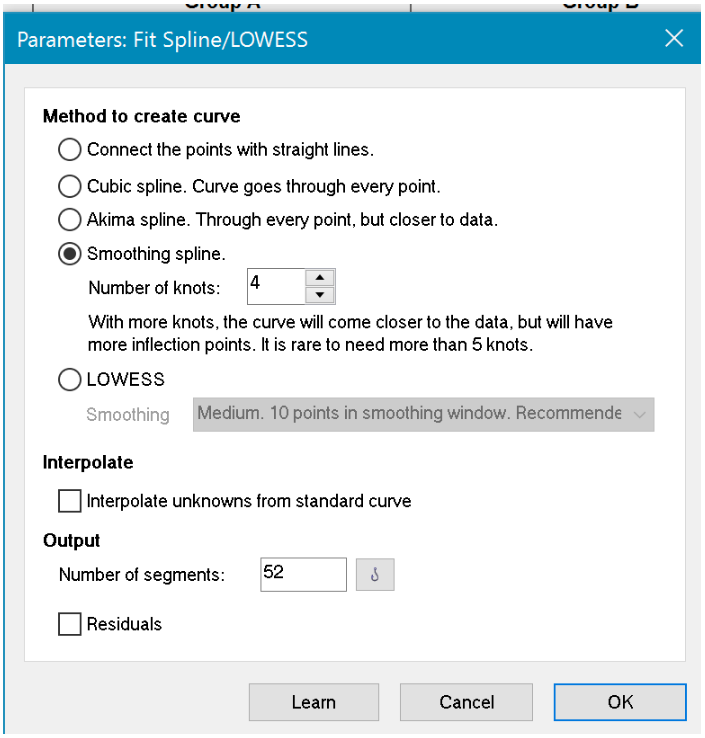
4.4.1 Spline and Lowess curves

Curve fitting without a model

The term *curve fitting* is more general than *regression*. Your approach to curve fitting depends on your goal.

In some circumstances, you just want to draw a smooth curve to make a graph look attractive, or to use as a standard curve. You don't care about models, and aren't looking for best-fit values that you can interpret.

Prism provides two approaches for fitting a curve without selecting a model. From a table or graph of XY data, click Analyze, and then choose 'Fit spline/LOWESS" from the list of XY analyses.



Parameters: Fit Spline/LOWESS

Method to create curve

○ Connect the points with straight lines.

○ Cubic spline. Curve goes through every point.

○ Akima spline. Through every point, but closer to data.

● Smoothing spline.

Number of knots: 4

With more knots, the curve will come closer to the data, but will have more inflection points. It is rare to need more than 5 knots.

○ LOWESS

Smoothing: Medium. 10 points in smoothing window. Recommende

Interpolate

□ Interpolate unknowns from standard curve

Output

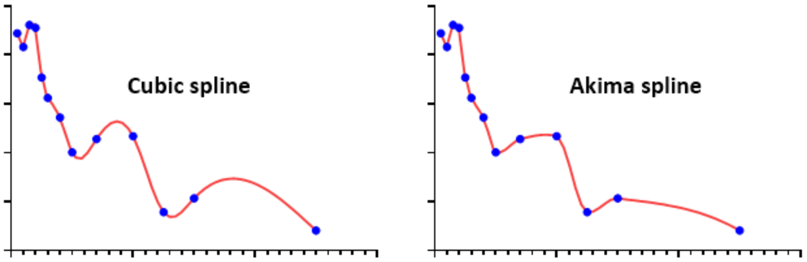
Number of segments: 52

□ Residuals

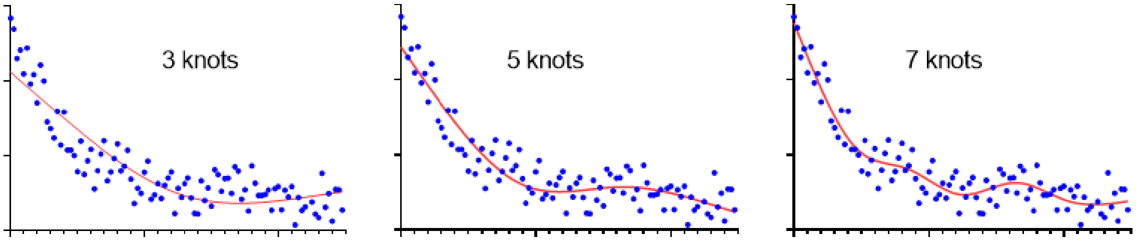
Spline curves

Up to Prism 7, Prism only offered cubic spline curves that go through every point. These tend to wiggle too much to be very useful. Prism 8 offers more kinds of splines.

In addition to the usual cubic spline that goes through every point, Prism now can also draw *Akima splines*. These also go through every point, but it makes tighter turns so some think it looks better.



Prism also can now plot *smoothing splines*. You decide the degree of smoothness by specifying the number of knots. This example shows splines with 3, 5, and 7 knots. Since the splines were derived for art (computer animation) the choice of how many knots to allow is yours to make artistically. Splines have nothing to do with data analysis.



Also check out fitting the hinge function 659. It connects two straight lines with a gentle curve.

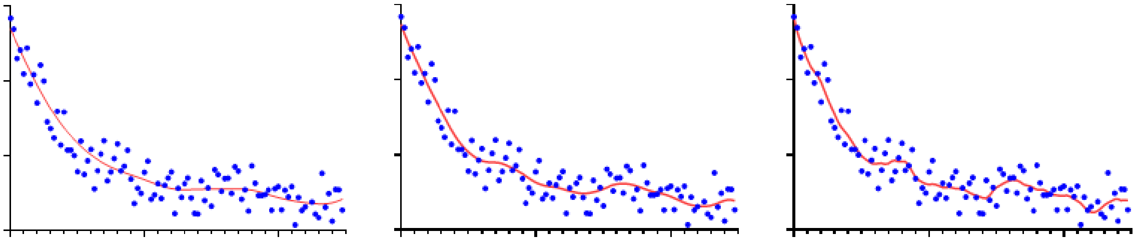
Connect points with straight lines

Prism's spline/lowess analysis can also create a point-to-point "curve" -- a series of line segments connecting all your data. Don't create a point-to-point curve just so you can connect points with a line on the graph. You can do that by checking an option on the Format Graph dialog from the Graphs section of your project. Only select the point-to-point analysis if you want to use the point-to-point line as a standard curve, or if you wish to calculate area under that curve.

Note a subtle distinction between a point-to-point line "fit" by the spline analysis and a point-to-point line added in the Format Graph dialog. The two will look the same if both axes are linear. But if you use a logarithmic axis, the two are distinct. The lines created by Format Graph will look straight, even with the log axis. Prism simply connects the points. The point-to-point "curve" created by the spline analysis is actually a set of XY coordinates (you can decide how many). With a log axis, this point-to-point "curve" will not appear straight.

Lowess curves

A **lowess** curve follows the trend of the data and tends to be a bit jagged. Lowess curves can be helpful when the data progresses monotonically, but are less helpful when there are peaks or valleys. Prism lets you choose between coarse, medium and fine lowess curves. The coarse curve (left panel below) shows only the general trend, but obscures the detail. The fine curve (right panel below) reveals the fine structure of the data, but tends to wiggle a lot. The medium curve is in between.



Prism generates lowess curves using an algorithm adapted from reference 1. Details. Don't select a lowess curve unless you have well over twenty data points.

How many line segments

Prism generates the curve as a series of line segments. Enter the number of segments you want. The default is nearly always fine. Prism always generates a lowess curve with at least four times more line segments than the number of data points, and you cannot reduce the number of segments below this value.

Check the option box if you want Prism to create an output table with the XY coordinates of each point that defines the curve.

Standard curve calculations

Check the option if you want to use the resulting Lowess, point-to-point, or spline curve as a standard curve. How to enter data 260.

References

1. John Chambers et. al., *Graphical Methods for Data Analysis*, Wadsworth and Brooks, 1983.

4.4.2 Using nonlinear regression with an empirical model

If your goal is just to plot a smooth curve, without worrying about a model, you have several choices.

Splines created by Prism go through every point, so may wiggle too much, but smoothing splines work well.

Lowess curves follow the general trend of the data, but can be too jagged.

An alternative is to use nonlinear regression.

Nonlinear regression requires you pick a model, but you don't have to pay attention to the meaning of the model or the value of the parameters. Instead, you can pick a model empirically and judge it solely on the appearance of the curve. In this case, you are using nonlinear regression as a tool to create a smooth curve, and not as a method to analyze data.

If you use nonlinear regression in this way, you can experiment with any model you want. But first try fitting polynomial models 668, which are very general (and never give fitting problems due to poor initial values). If the curve strays too far from the trend of the data, pick a higher order model. If the polynomial curve wiggles too much, pick a lower order model.

4.5 Nonlinear regression with Prism

Prism's nonlinear regression dialog has ten tabs, explained here. It can seem overwhelming at first. But you don't have to learn about all the choices when you first fit a curve. Start by focussing on the choices in the first (Fit) tab, and then learn about the other choices later.

- Before learning all the details, first understand the big picture 294 -- the basic steps required to fit a model to data.

- So many choices! Which ones are essential? 327

4.5.1 Nonlinear regression tutorials

Prism makes it quite easy to fit a model to your data. If you are new to Prism, choose from the sample XY data sets. These not only show you how to use Prism, but also review the principles of nonlinear regression, including comparing models, identifying outliers, global fitting, and more.

- Example: Fitting an enzyme kinetics curve 296

- Example: Comparing two enzyme kinetics models 303

- Example: Automatic outlier elimination (exponential decay) 307

- Example: Global nonlinear regression (dose-response curves) 312

- Example: Ambiguous fit (dose-response) 319

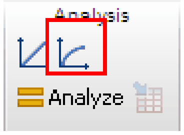
- Example: Interpolating from a sigmoidal standard curve 262

4.5.1.1 How to fit a model with Prism

Step 1. Enter data

Create an XY table, and enter data. If you have replicate Y values at each X value, format the table for entry of replicates.

From an XY table or graph, click the shortcut button to fit a model with nonlinear regression. Or click Analyze and select from the analyze dialog.



Step 2. Choose a model

Nonlinear regression fits a model to your data. You must, therefore, choose a model or enter a new model.

Why a computer program cannot pick a model for you 16.

Step 3. Choose (or review) initial values

Nonlinear regression is an iterative procedure. The program must start with estimated values for each variable that are in the right "ball park" - say within a factor of five of the actual value. It then adjusts these initial values to improve the fit. See How nonlinear regression works.

GraphPad Prism provides initial values automatically if you use a built-in equation. If you enter your own equation, you can also enter rules for initial values. For example the initial value of one parameter may be twice the maximum Y value in the data, while the initial value of another parameter may equal the average of the highest and lowest X values. Once you define these rules, Prism will compute appropriate initial values based on the range of your data.

The Initial Values 339 tab of the Nonlinear regression dialog lets you review, and override, the computed initial values.

You'll find it easy to estimate initial values if you have looked at a graph of the data, understand the model, and understand the meaning of all the parameters in the equation. Remember that you just need an estimate. It doesn't have to be very accurate. If you are having problems estimating initial values, set aside your data and simulate curves using the model. Change the variables one at a time, and see how they influence the shape of the curve. Once you have a better feel for how the parameters influence the curve, you might find it easier to estimate initial values.

When fitting a simple model to clean data, it won't matter much if the initial values are fairly far from the correct values. You'll get the same best-fit curve no matter what initial values you use, unless the initial values are extremely far from correct. Initial values matter more when your data have a lot of scatter or your model has many variables.

Step 4. Decide whether to constrain any parameters

When performing nonlinear regression, you don't have to fit each parameter in the equation. Instead, you may fix one or more of the parameters to constant values. It is often helpful to define constants when you have only a few data points. For example, you might fix the bottom plateau of a sigmoid curve or exponential decay to zero.

Remember that nonlinear regression programs have no "common sense". You need to think about how you did the experiment, and decide whether some of the parameters should be fixed. For example, if a background signal has already been subtracted, it makes sense to fix the bottom plateau of a dose-response curve or an exponential decay curve to zero.

Prism also let you constrain parameters to a certain range of values.

Step 5. If you are fitting two or more data sets at once, decide whether to share any parameters

If you enter data into two or more data set columns, Prism will fit them all in one analysis. But each fit will be independent of the others unless you specify that one or more parameters are shared. When you share parameters, the analysis is called a global nonlinear regression 70.

Step 6. Decide on a weighting scheme

Nonlinear regression programs generally weight each point equally. But there are many ways to differentially weight the points 329.

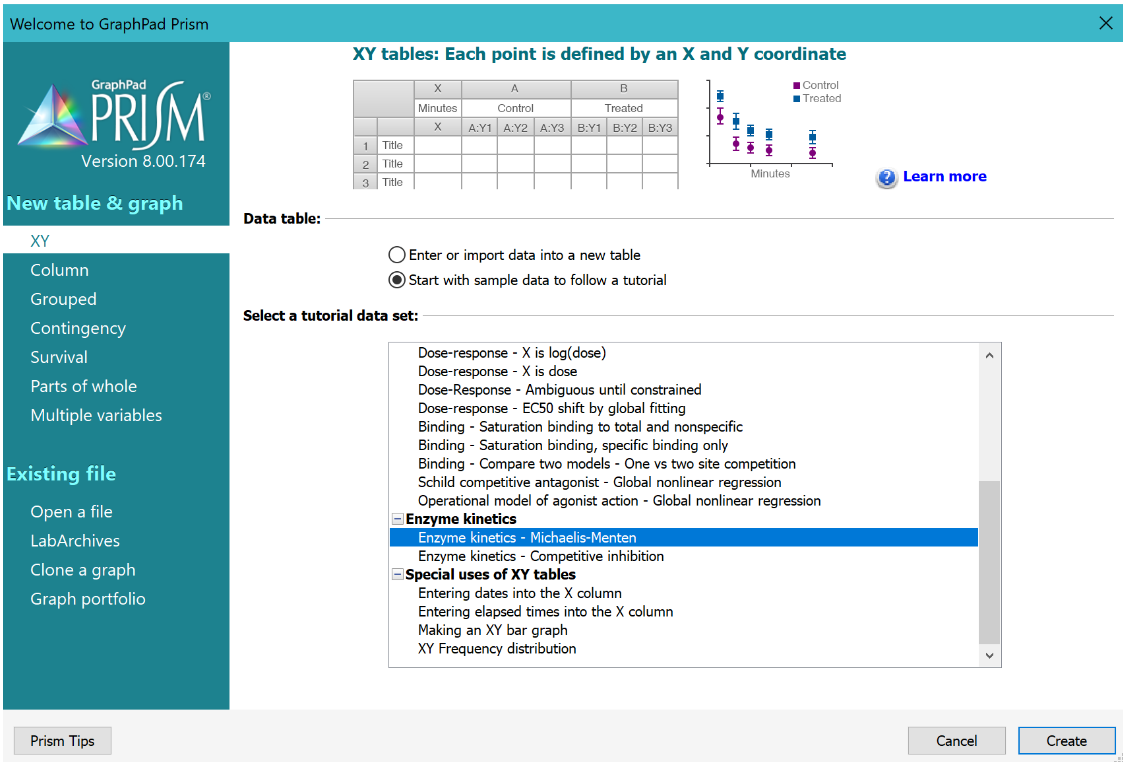
Step 7. Choose other options

Read about choices on the Range 341, Output 342 and Diagnostics 347 tab.

4.5.1.2 Example: Fitting an enzyme kinetics curve

1. Create the data table

From the Welcome or New Table dialog, choose to create an XY data table, choose tutorial data sets, and select the sample data "Enzyme kinetics -- Michaelis-Menten" from the enzyme kinetics section.



Welcome to GraphPad Prism

XY tables: Each point is defined by an X and Y coordinate

		X	A			B		
		Minutes	Control			Treated		
		X	A:Y1	A:Y2	A:Y3	B:Y1	B:Y2	B:Y3
1	Title							
2	Title							
3	Title							

Data table:

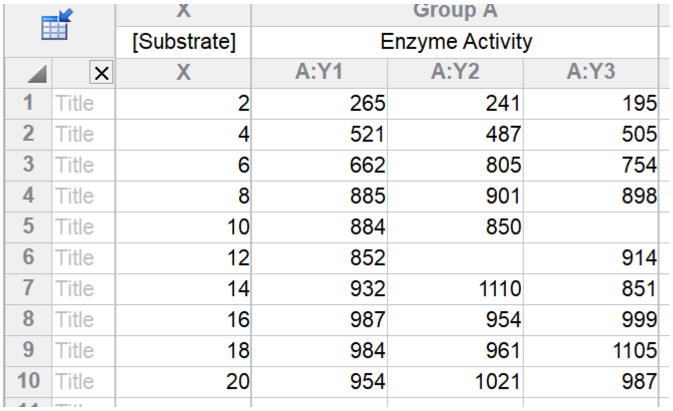
○ Enter or import data into a new table
● Start with sample data to follow a tutorial

Select a tutorial data set:

Dose-response - X is log(dose)
Dose-response - X is dose
Dose-Response - Ambiguous until constrained
Dose-response - EC50 shift by global fitting
Binding - Saturation binding to total and nonspecific
Binding - Saturation binding, specific binding only
Binding - Compare two models - One vs two site competition
Schild competitive antagonist - Global nonlinear regression
Operational model of agonist action - Global nonlinear regression
Enzyme kinetics
Enzyme kinetics - Michaelis-Menten
Enzyme kinetics - Competitive inhibition
Special uses of XY tables
Entering dates into the X column
Entering elapsed times into the X column
Making an XY bar graph
XY Frequency distribution

2. Inspect the data

The sample data will be partly covered by a floating note explaining how to fit the data (for people who are not reading this help page). You can move the floating note out of the way, or minimize it.



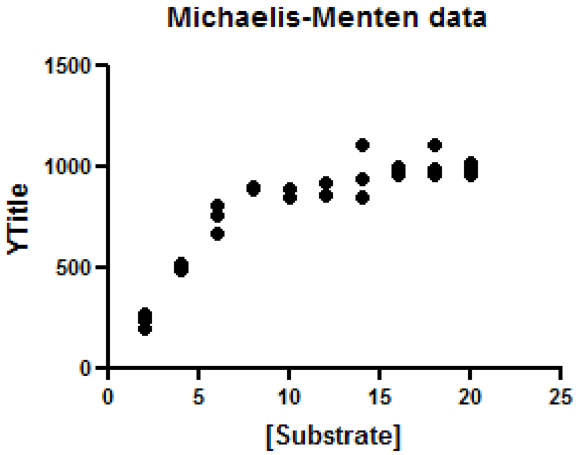
		X	Group A		
		[Substrate]	Enzyme Activity		
	☒	X	A:Y1	A:Y2	A:Y3
1	Title	2	265	241	195
2	Title	4	521	487	505
3	Title	6	662	805	754
4	Title	8	885	901	898
5	Title	10	884	850	
6	Title	12	852		914
7	Title	14	932	1110	851
8	Title	16	987	954	999
9	Title	18	984	961	1105
10	Title	20	954	1021	987

The data are in triplicate. Some values are missing, and Prism always handles these just fine.

3. View the graph

Prism automatically created a graph and gave it the same name as the data table. Click on the Michaelis-Menten graph in the graphs section.

Since this is the first time you are viewing the graph, Prism will pop up the Change Graph Type dialog. Select the third choice, to plot individual replicates rather than mean and error bars.

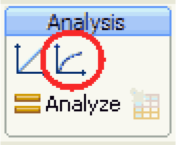


The graph Prism makes automatically is fairly complete. You can customize the symbols, colors, axis labels, position of legend, etc.

4. Choose nonlinear regression

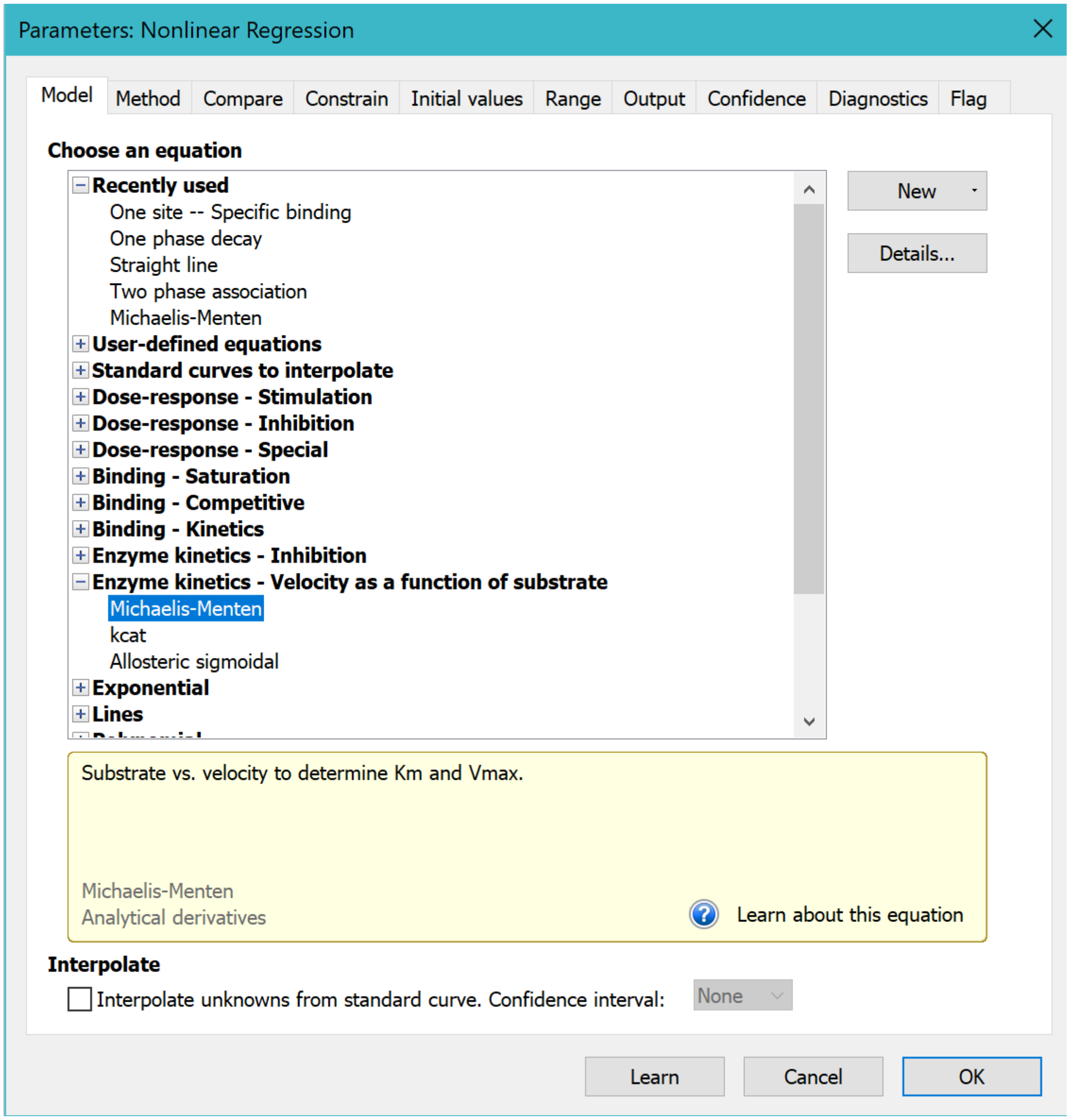
Click the Analyze button and choose Nonlinear regression from the list of XY analyses.

Even faster, click the shortcut button for nonlinear regression.



5. Choose a model

On the Fit tab of the nonlinear regression dialog, open the equation folder, Enzyme Kinetics - Substrate vs. Velocity. Then choose the Michaelis-Menten equation.



Choose an equation

- Recently used
  - One site -- Specific binding
  - One phase decay
  - Straight line
  - Two phase association
  - Michaelis-Menten
- User-defined equations
- Standard curves to interpolate
- Dose-response - Stimulation
- Dose-response - Inhibition
- Dose-response - Special
- Binding - Saturation
- Binding - Competitive
- Binding - Kinetics
- Enzyme kinetics - Inhibition
- Enzyme kinetics - Velocity as a function of substrate
  - Michaelis-Menten
  - kcat
  - Allosteric sigmoidal
- Exponential
- Lines

Substrate vs. velocity to determine Km and Vmax.

Michaelis-Menten
Analytical derivatives

Learn about this equation

Interpolate

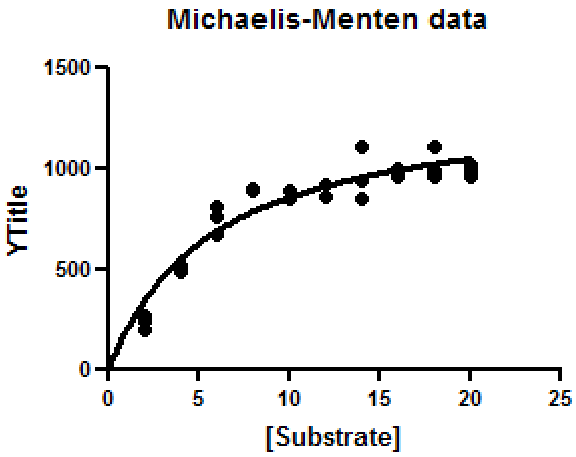
□ Interpolate unknowns from standard curve. Confidence interval: None

Learn more about the principles of enzyme kinetics 610 and about fitting Michaelis-Menten curves 613.

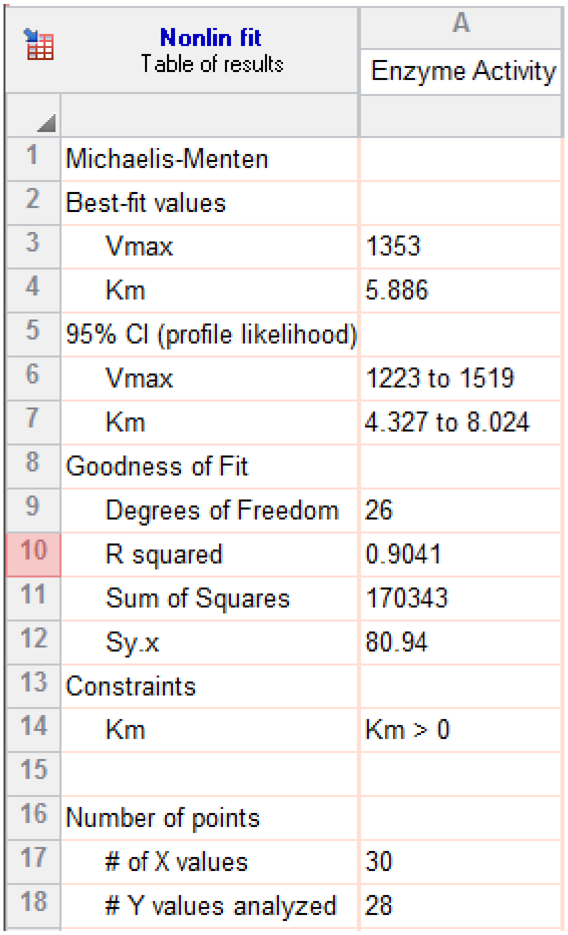
For this example, leave all the other settings to their default values.

Click OK to see the curves superimposed on the graph.

6. Inspect the graph



7. Inspect the results



	Nonlin fit Table of results	A Enzyme Activity
1	Michaelis-Menten	
2	Best-fit values	
3	Vmax	1353
4	Km	5.886
5	95% CI (profile likelihood)	
6	Vmax	1223 to 1519
7	Km	4.327 to 8.024
8	Goodness of Fit	
9	Degrees of Freedom	26
10	R squared	0.9041
11	Sum of Squares	170343
12	Sy.x	80.94
13	Constraints	
14	Km	Km > 0
15		
16	Number of points	
17	# of X values	30
18	# Y values analyzed	28

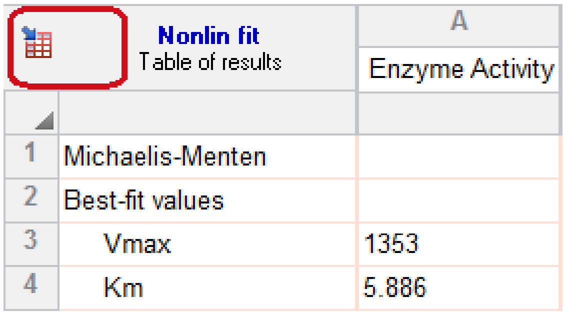
The goal of nonlinear regression is to find the best-fit values of the parameters. These are reported at the top of the table. You can't really interpret the best-fit values without knowing how precise they are, and this is reported both as standard errors and confidence intervals 372.

8. Go back and perform the replicates test

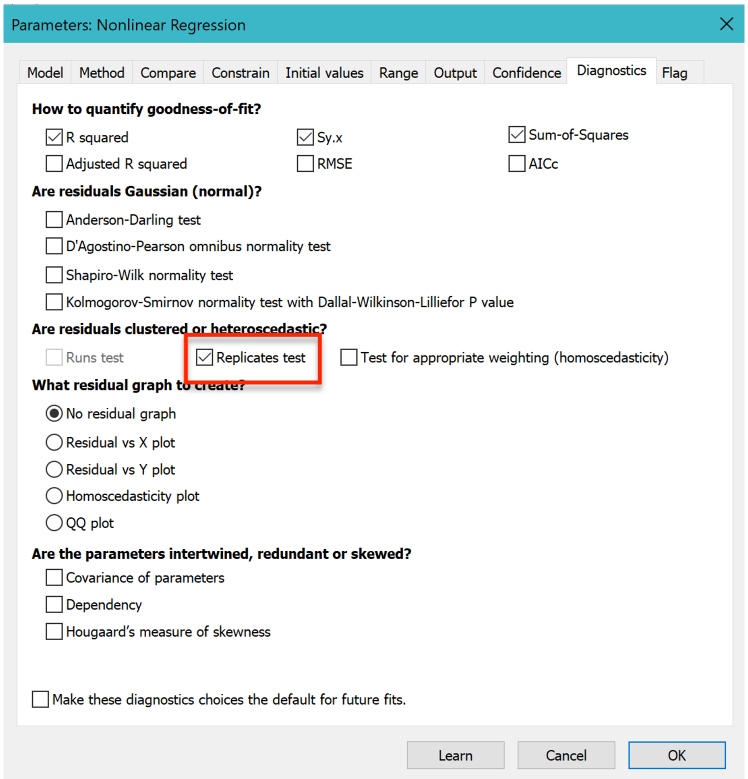
The replicates test assesses the adequacy of the fit by comparing the scatter among the triplicates with the scatter of points around the curve. It is not calculated by default, so the results do not appear in the results of step 7.

You don't have to do the fit over again. Instead click the button in the upper left corner of the results table to return to the nonlinear regression dialog.

Go to the Diagnostics tab 347, and check the option to perform the replicates test 390. Note that you can also check an option to make your settings here become the default for future fits.



Nonlin fit Table of results	A Enzyme Activity
1 Michaelis-Menten	
2 Best-fit values	
3 Vmax	1353
4 Km	5.886



Parameters: Nonlinear Regression

How to quantify goodness-of-fit?

☑ R squared ☑ Sy.x ☑ Sum-of-Squares

□ Adjusted R squared □ RMSE □ AICc

Are residuals Gaussian (normal)?

□ Anderson-Darling test

□ D'Agostino-Pearson omnibus normality test

□ Shapiro-Wilk normality test

□ Kolmogorov-Smirnov normality test with Dallal-Wilkinson-Lilliefor P value

Are residuals clustered or heteroscedastic?

□ Runs test ☑ Replicates test □ Test for appropriate weighting (homoscedasticity)

What residual graph to create?

● No residual graph

○ Residual vs X plot

○ Residual vs Y plot

○ Homoscedasticity plot

○ QQ plot

Are the parameters intertwined, redundant or skewed?

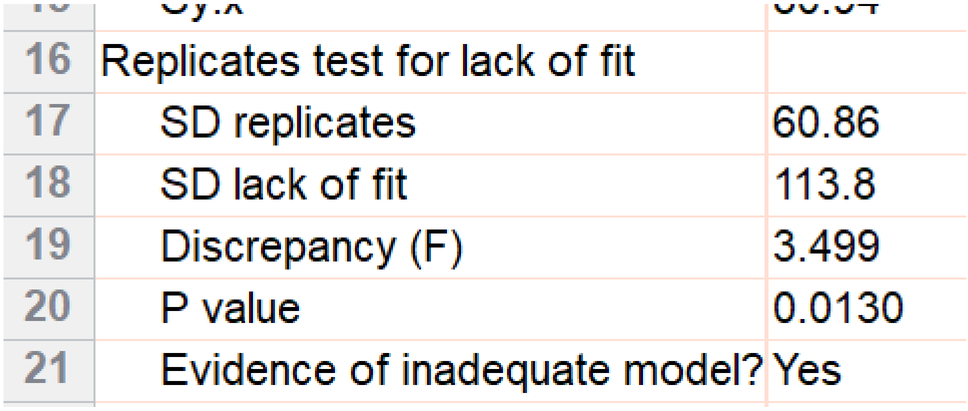
□ Covariance of parameters

□ Dependency

□ Hougaard's measure of skewness

□ Make these diagnostics choices the default for future fits.

The P value is small (0.013). This means that the scatter of the data from the curve is greater than you'd expect from the variation among triplicates. This suggests that you might want to consider fitting an alternative model, which we do in the next example 303.

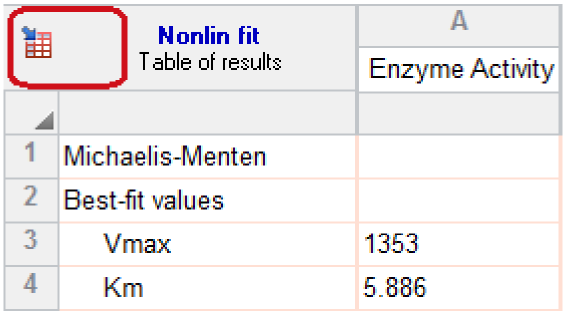


16	Replicates test for lack of fit	
17	SD replicates	60.86
18	SD lack of fit	113.8
19	Discrepancy (F)	3.499
20	P value	0.0130
21	Evidence of inadequate model?	Yes

4.5.1.3 Example: Comparing two enzyme kinetics models

1.Open analysis parameters dialog

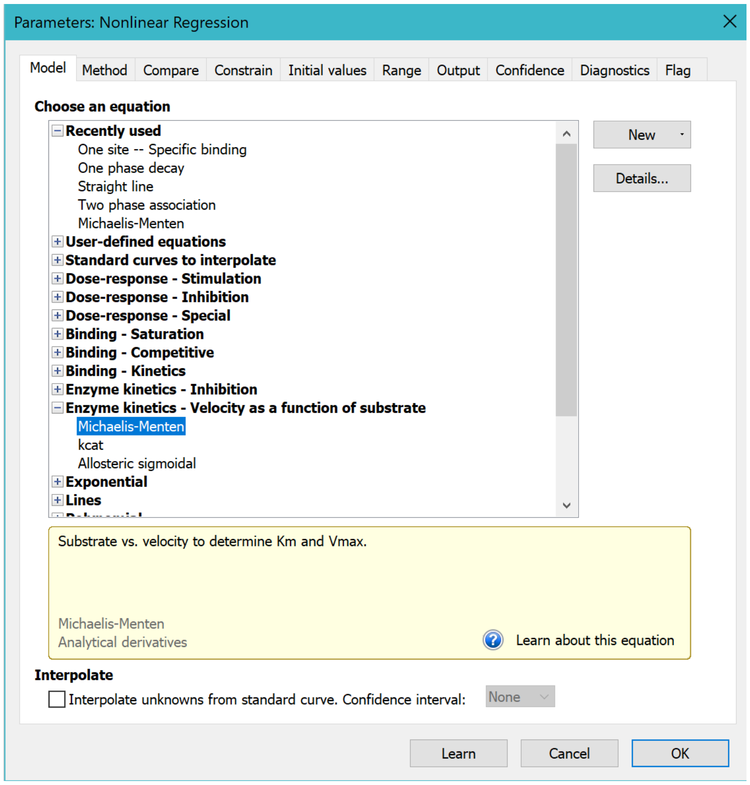
This example continues from the previous one 296. Click the analysis parameter button again.



	Nonlin fit Table of results	A Enzyme Activity
1	Michaelis-Menten	
2	Best-fit values	
3	Vmax	1353
4	Km	5.886

2. Choose the first model

On the Fit tab, choose (or make sure you still have chosen) the Michaelis-Menten equation.



Choose an equation

- Recently used
  - One site -- Specific binding
  - One phase decay
  - Straight line
  - Two phase association
  - Michaelis-Menten
- User-defined equations
- Standard curves to interpolate
- Dose-response - Stimulation
- Dose-response - Inhibition
- Dose-response - Special
- Binding - Saturation
- Binding - Competitive
- Binding - Kinetics
- Enzyme kinetics - Inhibition
- Enzyme kinetics - Velocity as a function of substrate
  - Michaelis-Menten
  - kcat
  - Allosteric sigmoidal
- Exponential
- Lines

Substrate vs. velocity to determine Km and Vmax.

Michaelis-Menten
Analytical derivatives

Interpolate

□ Interpolate unknowns from standard curve. Confidence interval: None

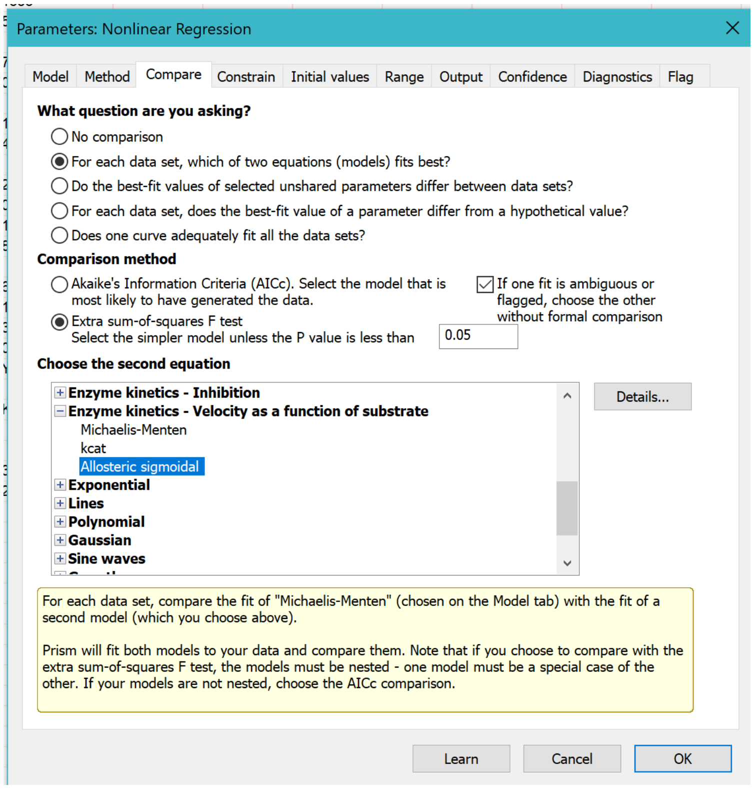
3. Choose the other model

Go to the Compare tab 334.

Choose: For each data set, which of two equations (models) fits best?

There are two ways to compare models. For this example, choose the extra sum-of-squares F test 82.

For the second equation, choose "Allosteric sigmoidal 619" from the Enzyme kinetics section.



Parameters: Nonlinear Regression

What question are you asking?

○ No comparison
● For each data set, which of two equations (models) fits best?
○ Do the best-fit values of selected unshared parameters differ between data sets?
○ For each data set, does the best-fit value of a parameter differ from a hypothetical value?
○ Does one curve adequately fit all the data sets?

Comparison method

○ Akaike's Information Criteria (AICc). Select the model that is most likely to have generated the data.
● Extra sum-of-squares F test
Select the simpler model unless the P value is less than 0.05
☑ If one fit is ambiguous or flagged, choose the other without formal comparison

Choose the second equation

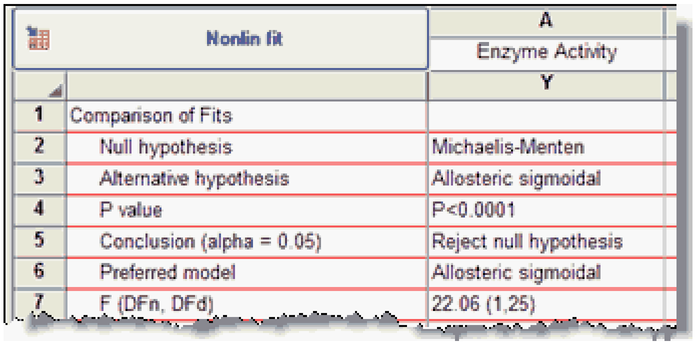
⊞ Enzyme kinetics - Inhibition
⊟ Enzyme kinetics - Velocity as a function of substrate
Michaelis-Menten
kcat
Allosteric sigmoidal
⊞ Exponential
⊞ Lines
⊞ Polynomial
⊞ Gaussian
⊞ Sine waves

For each data set, compare the fit of "Michaelis-Menten" (chosen on the Model tab) with the fit of a second model (which you choose above).

Prism will fit both models to your data and compare them. Note that if you choose to compare with the extra sum-of-squares F test, the models must be nested - one model must be a special case of the other. If your models are not nested, choose the AICc comparison.

4. View the results of the model comparison

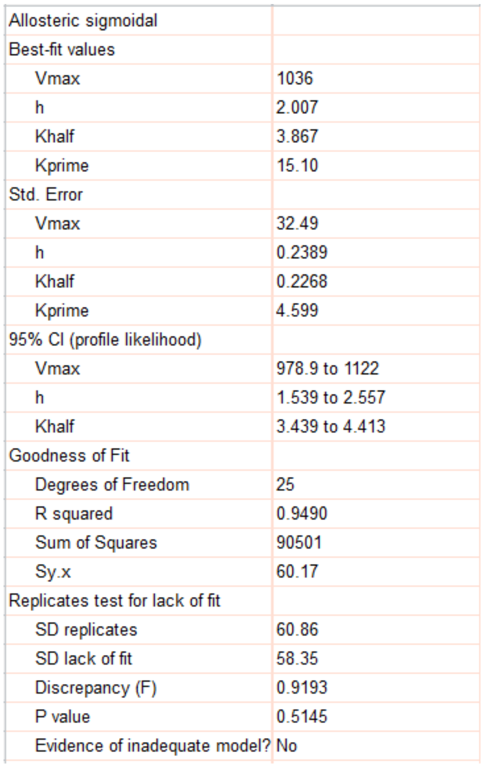
The top part of the results summarizes the comparison. The P value is low, suggesting that the simpler (Michaelis-Menten) model is too simple and should be rejected. The allosteric model fits significantly better.



	Nonlin fit	A Enzyme Activity Y
1	Comparison of Fits	
2	Null hypothesis	Michaelis-Menten
3	Alternative hypothesis	Allosteric sigmoidal
4	P value	P<0.0001
5	Conclusion (alpha = 0.05)	Reject null hypothesis
6	Preferred model	Allosteric sigmoidal
7	F (DFn, DFd)	22.06 (1,25)

5. View the numerical results of the fit to the allosteric model

Scroll down to the best-fit parameter values for the allosteric model.



Allosteric sigmoidal	
Best-fit values	
Vmax	1036
h	2.007
Khalf	3.867
Kprime	15.10
Std. Error	
Vmax	32.49
h	0.2389
Khalf	0.2268
Kprime	4.599
95% CI (profile likelihood)	
Vmax	978.9 to 1122
h	1.539 to 2.557
Khalf	3.439 to 4.413
Goodness of Fit	
Degrees of Freedom	25
R squared	0.9490
Sum of Squares	90501
Sy.x	60.17
Replicates test for lack of fit	
SD replicates	60.86
SD lack of fit	58.35
Discrepancy (F)	0.9193
P value	0.5145
Evidence of inadequate model?	No

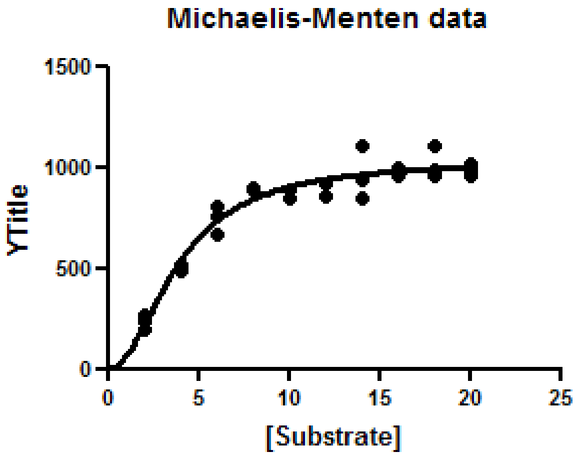
The P value of the replicates test 390 is high, which means the scatter of points around the curve is consistent with variability of replicates from each other.

The parameter H equals 2.0, with a 95% confidence interval ranging from 1.5 to 2.5. A value of 2.0 suggests that this enzyme might be a dimer. When H equals 1.0, the allosteric model is identical to the Michaelis-Menten model.

Further interpretation must, of course, be in the context of what is known about this enzyme from prior work. Statistics is only part of analyzing scientific data.

6. View the graph

Since the allosteric model fit significantly better, that is the model that Prism uses when it plots a curve on the graph. You can barely see that it has a sigmoidal shape. You can also see that you should collect more data with substrate concentrations between 0 and 5 to fully define this curve.

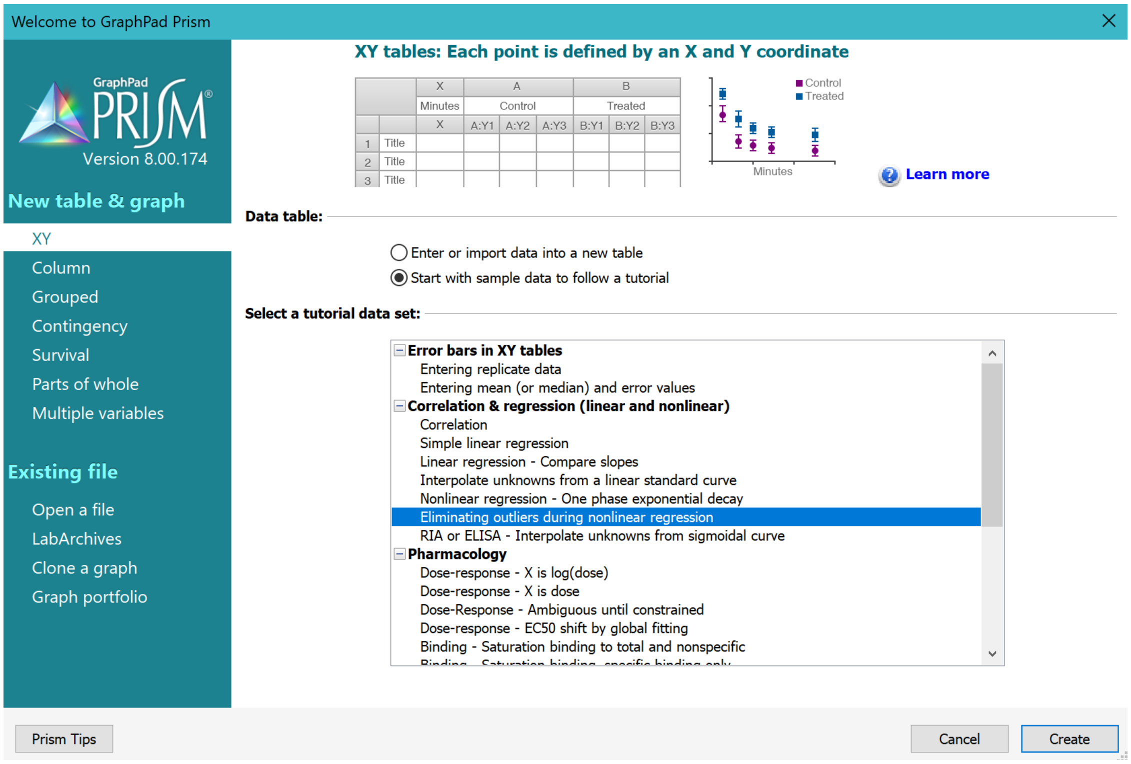


4.5.1.4 Example: Automatic outlier elimination (exponential decay)

Prism implements a unique method to identify outliers when fitting curves with nonlinear regression. This example shows you how easy it is to identify outliers with Prism. Learn more about how this method works 92, when it is useful 88, and when it should be avoided 89.

1. Create the data table

From the Welcome or New Table dialog, choose to create XY data table, choose to use tutorial data, and select the sample data "Eliminating outliers during nonlinear regression".



XY tables: Each point is defined by an X and Y coordinate

Data table:

○ Enter or import data into a new table
● Start with sample data to follow a tutorial

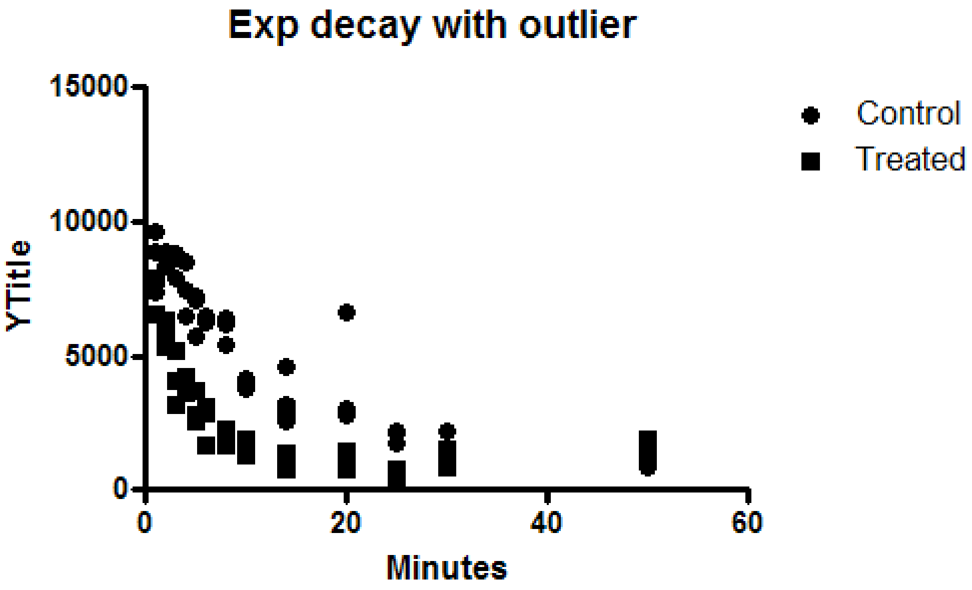
Select a tutorial data set:

- **Error bars in XY tables**
  - Entering replicate data
  - Entering mean (or median) and error values
- **Correlation & regression (linear and nonlinear)**
  - Correlation
  - Simple linear regression
  - Linear regression - Compare slopes
  - Interpolate unknowns from a linear standard curve
  - Nonlinear regression - One phase exponential decay
  - Eliminating outliers during nonlinear regression
  - RIA or ELISA - Interpolate unknowns from sigmoidal curve
- **Pharmacology**
  - Dose-response - X is log(dose)
  - Dose-response - X is dose
  - Dose-Response - Ambiguous until constrained
  - Dose-response - EC50 shift by global fitting
  - Binding - Saturation binding to total and nonspecific
  - Binding - Saturation binding - specific binding only

2. Inspect the data table and graph

The data table has values for two data sets (control and treated) with triplicate subcolumns.

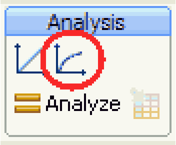
Since this is the first time you are viewing the graph, Prism will pop up the Change Graph Type dialog. Select the third choice, to plot individual replicates rather than mean and error bars.



3. Choose nonlinear regression

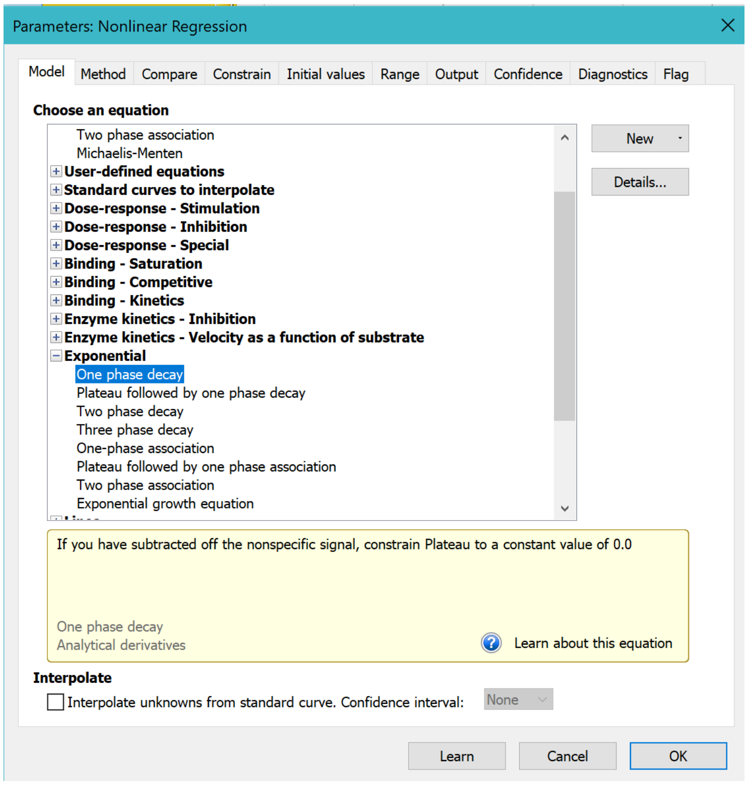
Click the Analyze button and choose Nonlinear regression from the list of XY analyses.

Alternatively, click the shortcut button for nonlinear regression.



4. Choose a model, and choose automatic outlier detection

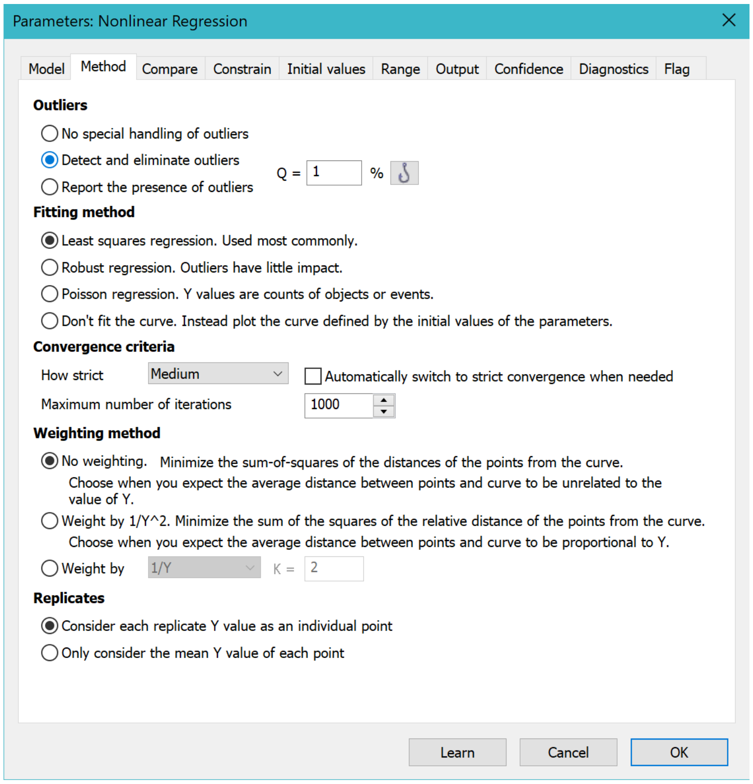
On the Fit tab of the nonlinear regression dialog, open the panel of exponential equations and choose: One phase decay.



If you have subtracted off the nonspecific signal, constrain Plateau to a constant value of 0.0

□ Interpolate unknowns from standard curve. Confidence interval: None

Then go to the Method tab and choose to identify and eliminate outliers. Learn about when outlier elimination is helpful 88, when it should be avoided 89, and how it works 92.



Parameters: Nonlinear Regression

Outliers

○ No special handling of outliers
● Detect and eliminate outliers
○ Report the presence of outliers

Q = 1 %

Fitting method

● Least squares regression. Used most commonly.
○ Robust regression. Outliers have little impact.
○ Poisson regression. Y values are counts of objects or events.
○ Don't fit the curve. Instead plot the curve defined by the initial values of the parameters.

Convergence criteria

How strict: Medium

□ Automatically switch to strict convergence when needed

Maximum number of iterations: 1000

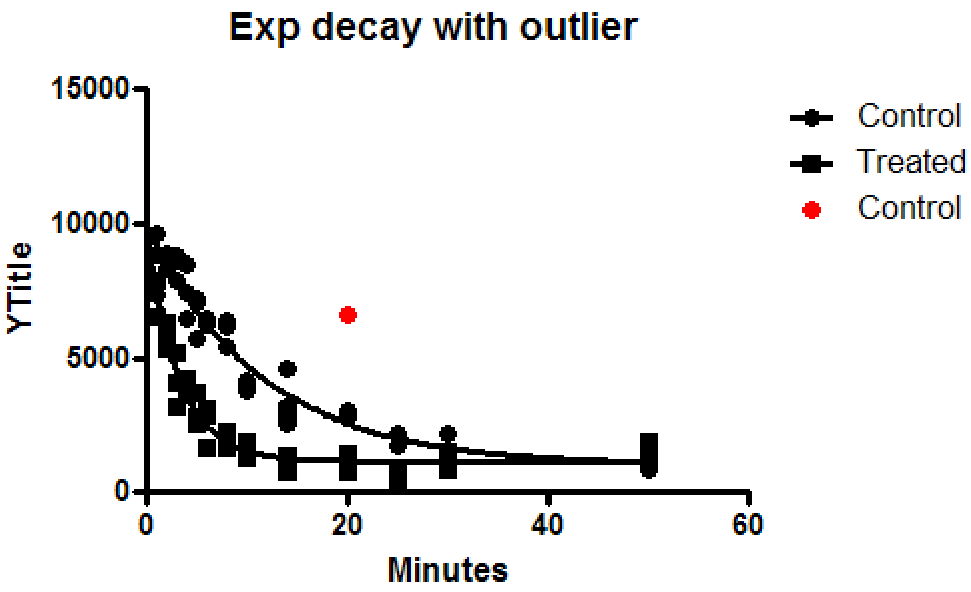
Weighting method

● No weighting. Minimize the sum-of-squares of the distances of the points from the curve. Choose when you expect the average distance between points and curve to be unrelated to the value of Y.
○ Weight by 1/Y^2. Minimize the sum of the squares of the relative distance of the points from the curve. Choose when you expect the average distance between points and curve to be proportional to Y.
○ Weight by 1/Y K = 2

Replicates

● Consider each replicate Y value as an individual point
○ Only consider the mean Y value of each point

5. View the graph



Prism identified the outlier, and plotted it in red, overlaid on top of the data graph. After identifying the outlier, Prism fit the remaining data points as if the outlier wasn't present. Before accepting the results, think about why the point was an outlier. Remember, not all outliers are "bad" points 91.

Double click on the graph to bring up the Format Graph dialog. Go to the second tab. You can see that this graph now has three data sets, the data, the curve fit, and the outliers. Read more about graphing outliers 358 .

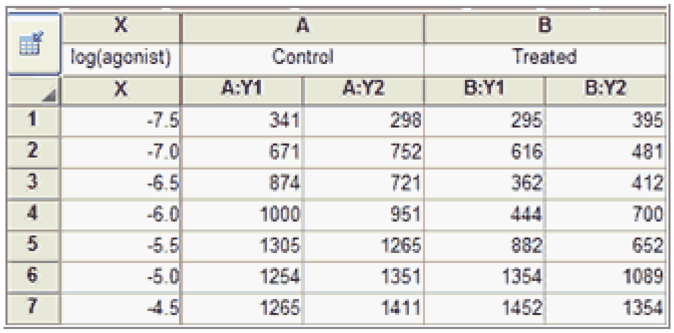
4.5.1.5 Example: Global nonlinear regression (dose-response curves)

1. Create the data table

From the Welcome or New Table dialog, choose to create an XY data table, choose to use tutorial data, and select the sample data "Dose-response: EC50 shift by global fitting" in the set of pharmacology tutorials.

2. Inspect the data

The sample data may be partly covered by a floating note explaining how to fit the data (for people who are not reading this help page). You can move the floating note out of the way, or minimize it.



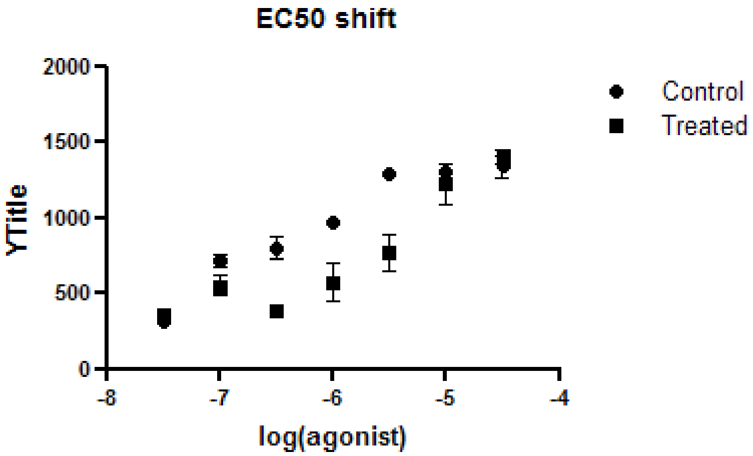
	X	A		B	
	log(agonist)	Control		Treated	
	X	A:Y1	A:Y2	B:Y1	B:Y2
1	-7.5	341	298	295	395
2	-7.0	671	752	616	481
3	-6.5	874	721	362	412
4	-6.0	1000	951	444	700
5	-5.5	1305	1265	882	652
6	-5.0	1254	1351	1354	1089
7	-4.5	1265	1411	1452	1354

The X values are the logarithm of the concentration of agonist. Note that "1e-9" is exactly the same as 0.000000001. To enter as a logarithm, enter " -9".

The Y values are responses, in duplicate, in two conditions.

3. View the graph

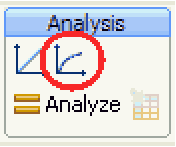
Since this is the first time you are viewing the graph, Prism will pop up the Change Graph Type dialog. Select the first choice, to plot mean and error bars, and choose SD error bars.



4. Choose nonlinear regression

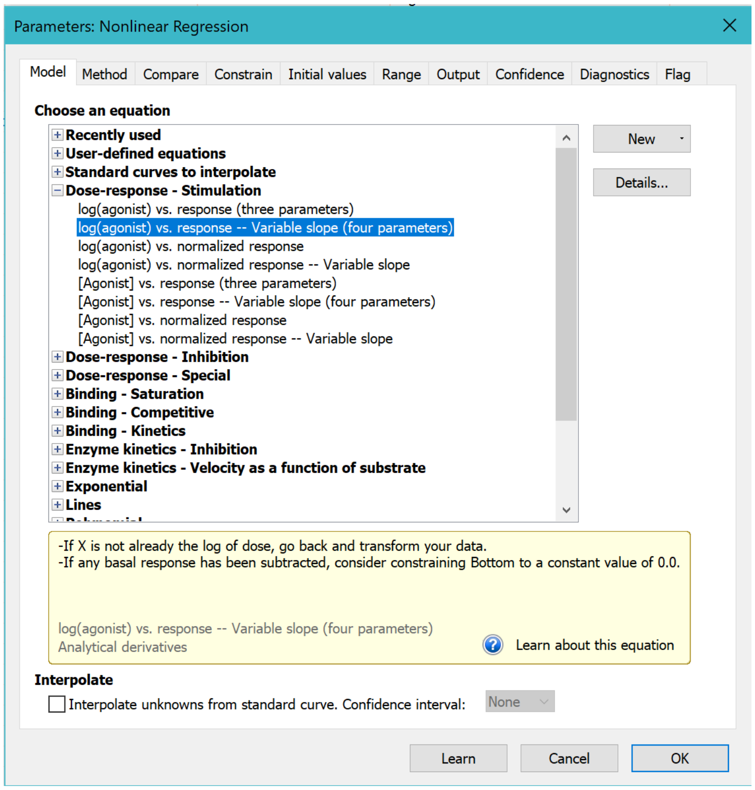
Click Analyze and choose Nonlinear regression from the list of XY analyses.

Alternatively, click the shortcut button for nonlinear regression.



5. Choose a model

On the Fit tab of the nonlinear regression dialog, go to the dose-response-stimulation models and choose: log(agonist) vs. response -- variable slope.



Parameters: Nonlinear Regression

Choose an equation

- Recently used
- User-defined equations
- Standard curves to interpolate
- Dose-response - Stimulation
  - log(agonist) vs. response (three parameters)
  - log(agonist) vs. response -- Variable slope (four parameters)
  - log(agonist) vs. normalized response
  - log(agonist) vs. normalized response -- Variable slope
  - [Agonist] vs. response (three parameters)
  - [Agonist] vs. response -- Variable slope (four parameters)
  - [Agonist] vs. normalized response
  - [Agonist] vs. normalized response -- Variable slope
- Dose-response - Inhibition
- Dose-response - Special
- Binding - Saturation
- Binding - Competitive
- Binding - Kinetics
- Enzyme kinetics - Inhibition
- Enzyme kinetics - Velocity as a function of substrate
- Exponential
- Lines

-If X is not already the log of dose, go back and transform your data.
-If any basal response has been subtracted, consider constraining Bottom to a constant value of 0.0.

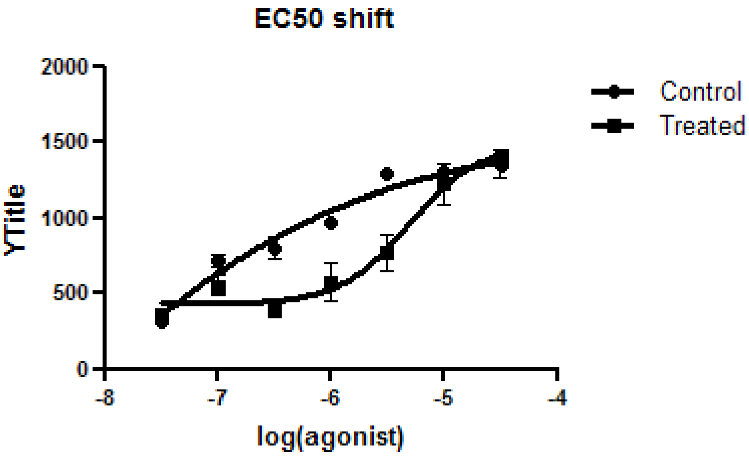
log(agonist) vs. response -- Variable slope (four parameters)
Analytical derivatives

Interpolate

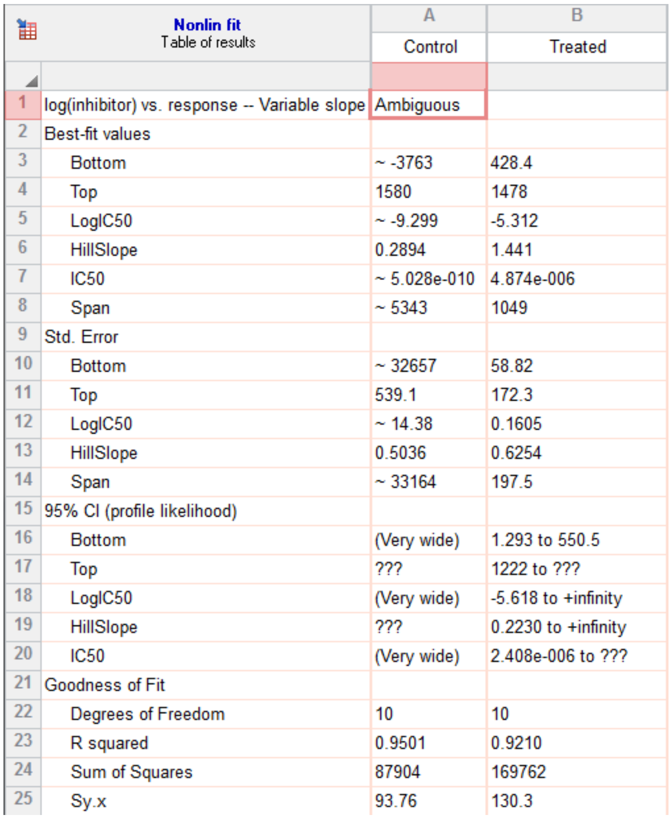
□ Interpolate unknowns from standard curve. Confidence interval: None

For now, leave all the other settings to their default values.

Click OK to see the curves superimposed on the graph.



6. Inspect the results



	Nonlin fit Table of results	A Control	B Treated
1	log(inhibitor) vs. response -- Variable slope	Ambiguous	
2	Best-fit values		
3	Bottom	~ -3763	428.4
4	Top	1580	1478
5	LogIC50	~ -9.299	-5.312
6	HillSlope	0.2894	1.441
7	IC50	~ 5.028e-010	4.874e-006
8	Span	~ 5343	1049
9	Std. Error		
10	Bottom	~ 32657	58.82
11	Top	539.1	172.3
12	LogIC50	~ 14.38	0.1605
13	HillSlope	0.5036	0.6254
14	Span	~ 33164	197.5
15	95% CI (profile likelihood)		
16	Bottom	(Very wide)	1.293 to 550.5
17	Top	???	1222 to ???
18	LogIC50	(Very wide)	-5.618 to +infinity
19	HillSlope	???	0.2230 to +infinity
20	IC50	(Very wide)	2.408e-006 to ???
21	Goodness of Fit		
22	Degrees of Freedom	10	10
23	R squared	0.9501	0.9210
24	Sum of Squares	87904	169762
25	Sy.x	93.76	130.3

The control results are labeled ambiguous. This means that Prism is unable to find a unique curve through the data. Lots of other sets of parameter values would lead to curves that fit just as well. You can see which parameters are ambiguous by looking at the 95% confidence intervals. Instead of reporting an interval, Prism reports 'very wide' for the Bottom and logEC50.

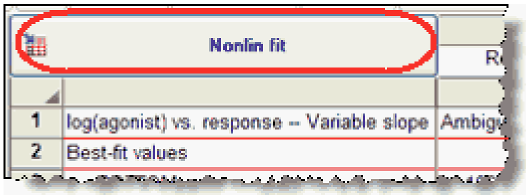
The data do not define a bottom plateau for the control (circles) data set, so its best-fit value is ambiguous. The EC50 is the concentration that gives a response half way between the bottom and top plateaus of the curve. If the bottom is ambiguous, so is the EC50.

The treated curve is not labeled 'ambiguous', but the confidence intervals are wider than you'd like.

7. Go back to the dialog, and share three parameters

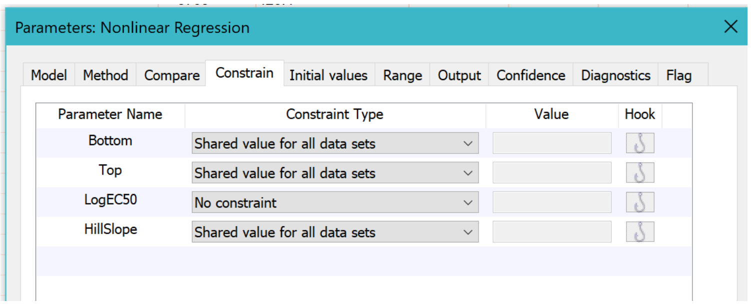
You can get much better results from this data set if you are willing to assume that that the top and bottom plateaus, and the slope, are the same under control and treated conditions. In other words, you assume that the treatment shifts the EC50 but doesn't change the basal response, the maximum response, or the Hill slope.

Return to the nonlinear regression dialog by clicking the button in the upper left of the results table.



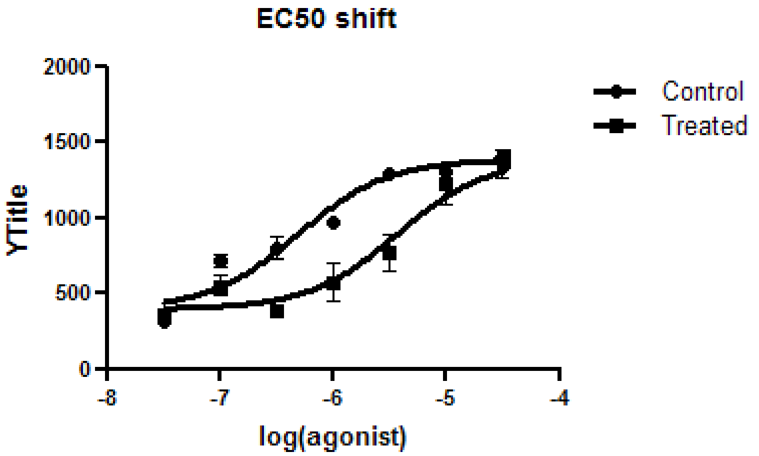
Nonlin fit		R
1	log(agonist) vs. response -- Variable slope	Ambigu
2	Best-fit values	

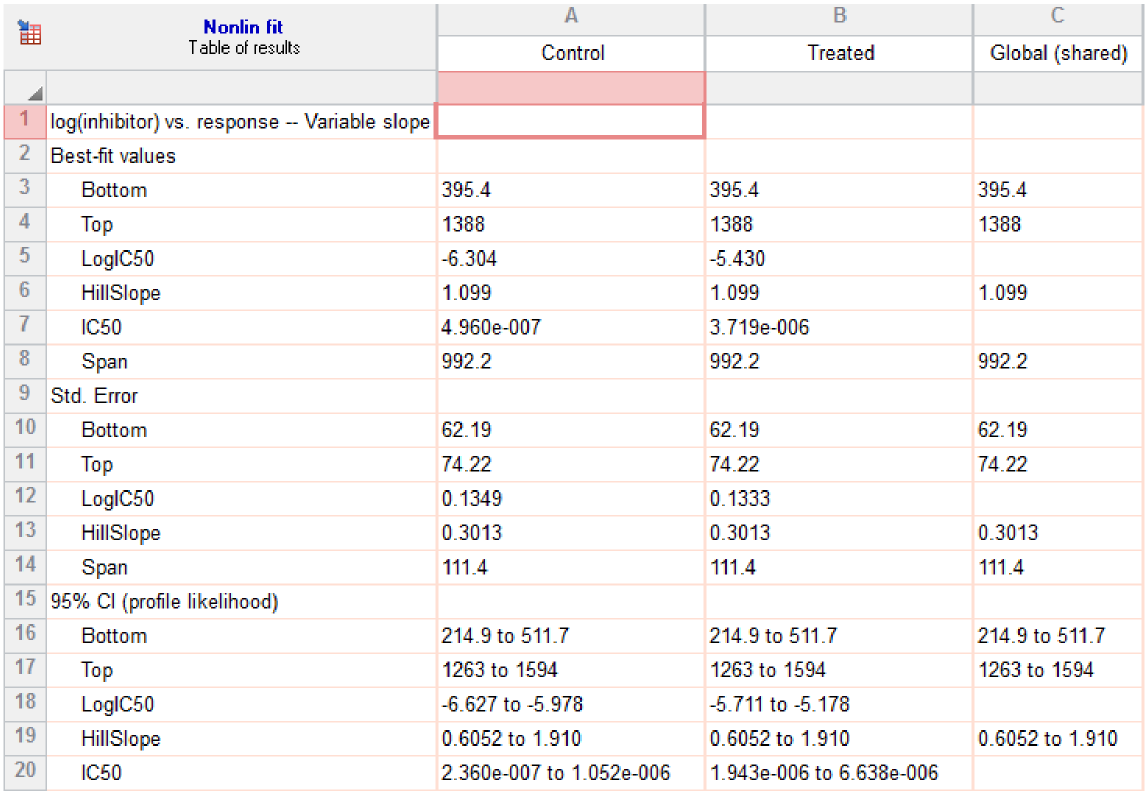
Go to the constraints tab and choose to share the value of Bottom, Top, and HillSlope. When you share these parameters, Prism fits the data sets globally to find one best-fit value for Bottom, Top and HillSlope (for both data sets) and separate best-fit values for the logEC50.



Parameter Name	Constraint Type	Value	Hook
Bottom	Shared value for all data sets		
Top	Shared value for all data sets		
LogEC50	No constraint		
HillSlope	Shared value for all data sets		

8. View the revised graph and results



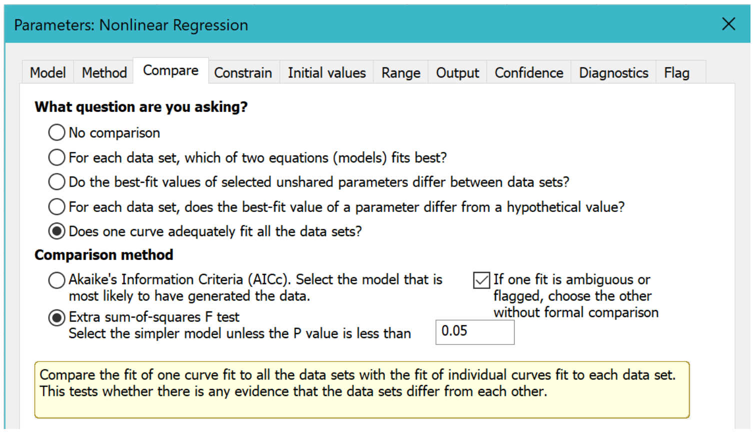


	Nonlin fit Table of results	A Control	B Treated	C Global (shared)
1	log(inhibitor) vs. response -- Variable slope			
2	Best-fit values			
3	Bottom	395.4	395.4	395.4
4	Top	1388	1388	1388
5	LogIC50	-6.304	-5.430	
6	HillSlope	1.099	1.099	1.099
7	IC50	4.960e-007	3.719e-006	
8	Span	992.2	992.2	992.2
9	Std. Error			
10	Bottom	62.19	62.19	62.19
11	Top	74.22	74.22	74.22
12	LogIC50	0.1349	0.1333	
13	HillSlope	0.3013	0.3013	0.3013
14	Span	111.4	111.4	111.4
15	95% CI (profile likelihood)			
16	Bottom	214.9 to 511.7	214.9 to 511.7	214.9 to 511.7
17	Top	1263 to 1594	1263 to 1594	1263 to 1594
18	LogIC50	-6.627 to -5.978	-5.711 to -5.178	
19	HillSlope	0.6052 to 1.910	0.6052 to 1.910	0.6052 to 1.910
20	IC50	2.360e-007 to 1.052e-006	1.943e-006 to 6.638e-006	

The fit is no longer labeled 'ambiguous' and the confidence intervals are much tighter.

9. Statistically compare the two logEC50 values

Go back to the parameters dialog for nonlinear regression and go to the Compare tab. Check the option to test whether one curve adequately fits all the data sets.



Parameters: Nonlinear Regression

What question are you asking?

○ No comparison
○ For each data set, which of two equations (models) fits best?
○ Do the best-fit values of selected unshared parameters differ between data sets?
○ For each data set, does the best-fit value of a parameter differ from a hypothetical value?
● Does one curve adequately fit all the data sets?

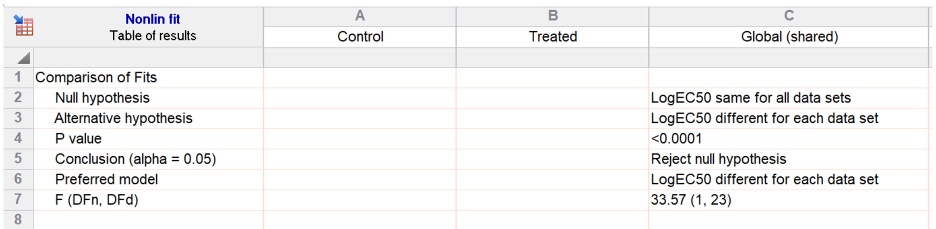
Comparison method

○ Akaike's Information Criteria (AICc). Select the model that is most likely to have generated the data.
● Extra sum-of-squares F test
Select the simpler model unless the P value is less than 0.05

☑ If one fit is ambiguous or flagged, choose the other without formal comparison

Compare the fit of one curve fit to all the data sets with the fit of individual curves fit to each data set. This tests whether there is any evidence that the data sets differ from each other.

Prism will now fit the data two ways. The first is the same as before, fitting a separate IC50 for each data set. The second fit shares all the parameters. In this case, three parameters were already shared but one wasn't. So in this second fit, all four parameters are shared, so Prism fits one curve through all the data, ignoring which treatment group they are in. It compares the sum-of-squares (really the sum-of-sum-of squares since there are two data sets fit in each case), using the extra sum-of-squares F test. The results are shown at the top of the results sheet.



	Nonlin fit Table of results	A Control	B Treated	C Global (shared)
1	Comparison of Fits			
2	Null hypothesis			LogEC50 same for all data sets
3	Alternative hypothesis			LogEC50 different for each data set
4	P value			<0.0001
5	Conclusion (alpha = 0.05)			Reject null hypothesis
6	Preferred model			LogEC50 different for each data set
7	F (DFn, DFd)			33.57 (1, 23)
8				

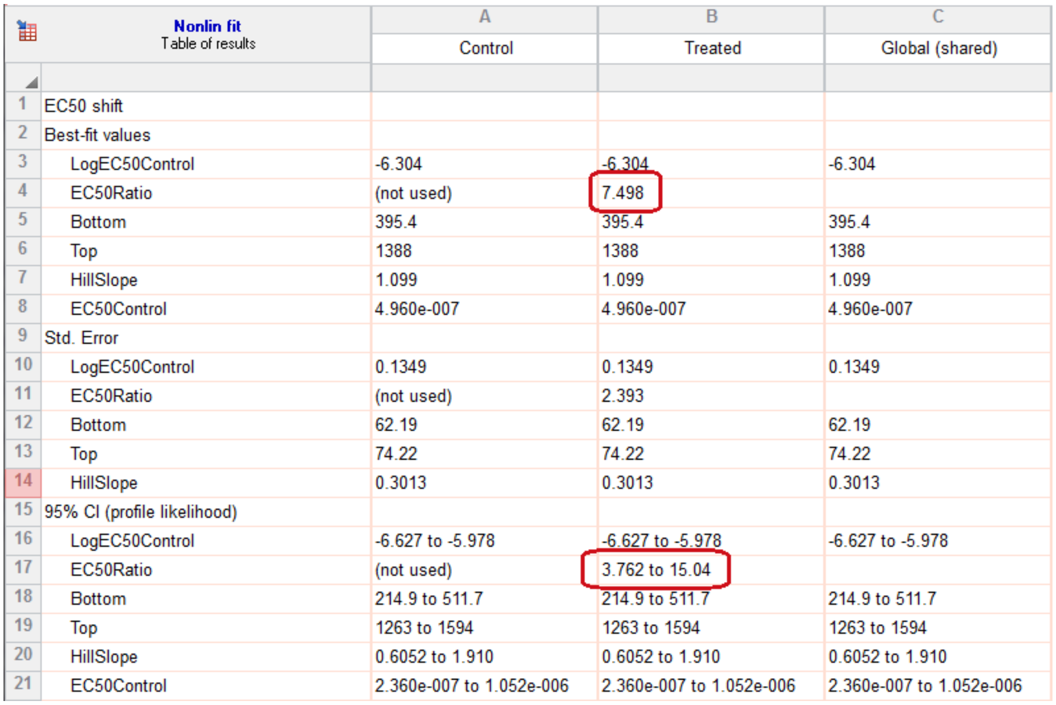
The P value is tiny, so we reject the null hypothesis that the two IC50 values are identical in the population, and instead conclude that the two IC50 values are different.

10. Fit the ratio of two IC50 values directly

From the results in step 8, you can compute what you want to know -- the ratio of the two EC50 values.

But Prism can calculate this value directly.

Go back to the analysis parameters dialog, and on the Fit tab, change the equation to "EC50 shift" from the "Dose-response - Special" group of equations. Accept all defaults and click OK. The graph will look identical, as the model is equivalent. But now, rather than fitting two logEC50 values, Prism fits one and also fits the ratio.



	Nonlin fit Table of results	A Control	B Treated	C Global (shared)
1	EC50 shift			
2	Best-fit values			
3	LogEC50Control	-6.304	-6.304	-6.304
4	EC50Ratio	(not used)	7.498	
5	Bottom	395.4	395.4	395.4
6	Top	1388	1388	1388
7	HillSlope	1.099	1.099	1.099
8	EC50Control	4.960e-007	4.960e-007	4.960e-007
9	Std. Error			
10	LogEC50Control	0.1349	0.1349	0.1349
11	EC50Ratio	(not used)	2.393	
12	Bottom	62.19	62.19	62.19
13	Top	74.22	74.22	74.22
14	HillSlope	0.3013	0.3013	0.3013
15	95% CI (profile likelihood)			
16	LogEC50Control	-6.627 to -5.978	-6.627 to -5.978	-6.627 to -5.978
17	EC50Ratio	(not used)	3.762 to 15.04	
18	Bottom	214.9 to 511.7	214.9 to 511.7	214.9 to 511.7
19	Top	1263 to 1594	1263 to 1594	1263 to 1594
20	HillSlope	0.6052 to 1.910	0.6052 to 1.910	0.6052 to 1.910
21	EC50Control	2.360e-007 to 1.052e-006	2.360e-007 to 1.052e-006	2.360e-007 to 1.052e-006

This equation was designed to do exactly what is needed for this example. Read about how this equation was set up 545, so you can construct your own equations when necessary.

4.5.1.6 Example: Ambiguous fit (dose-response)

1. Create the data table

From the Welcome or New Table dialog, choose to create XY data table, choose to use tutorial data, and select the sample data "Dose-Response - Ambiguous until constrained" from the list of Pharamcology tutorials. Note that this choice is in a drop down menu of sample data sets.

2. Inspect the data table

The X values are the logarithms of molar concentration. The Y values are responses, entered as mean and SD. With Prism, you can either enter replicate values or enter error values (here SD) computed elsewhere.

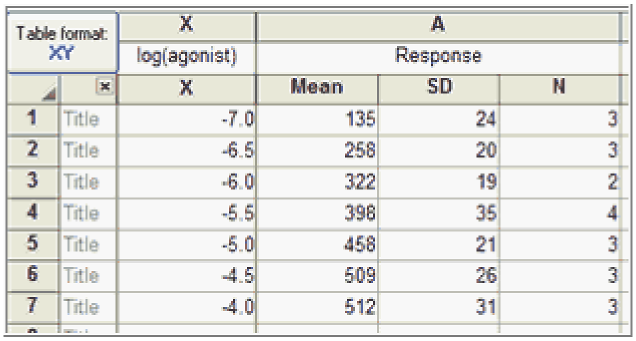
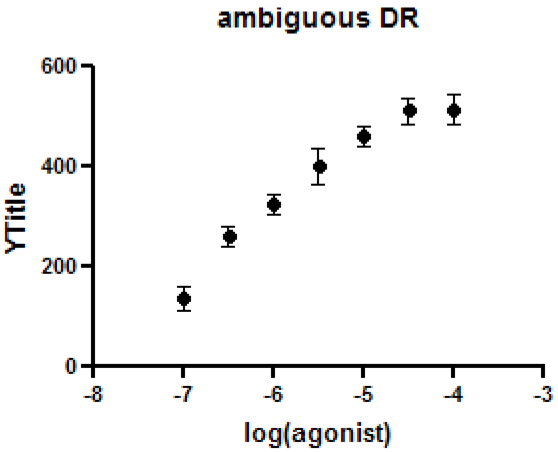


Table format: XY		X	A		
		log(agonist)	Response		
		X	Mean	SD	N
1	Title	-7.0	135	24	3
2	Title	-6.5	258	20	3
3	Title	-6.0	322	19	2
4	Title	-5.5	398	35	4
5	Title	-5.0	458	21	3
6	Title	-4.5	509	26	3
7	Title	-4.0	512	31	3

3. View the graph

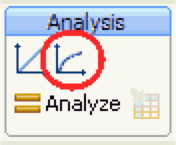
Since this is the first time you are viewing the graph, Prism will pop up the Change Graph Type dialog. Select the first choice, to plot mean and error bars, and choose the kind of error bars you prefer.



4. Choose nonlinear regression

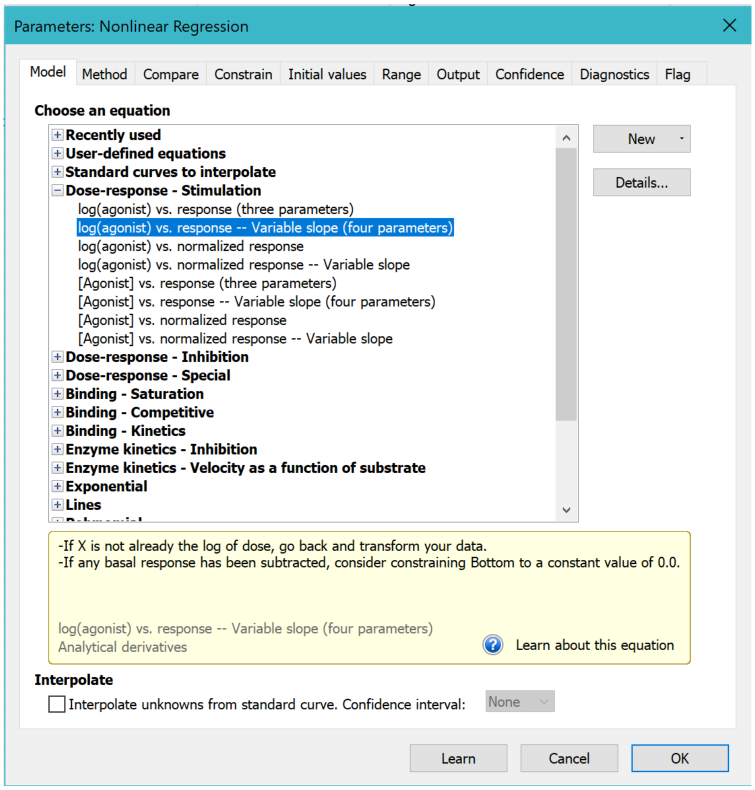
Click Analyze and choose from the list of XY analyses.

Even faster, click the shortcut button for nonlinear regression.



5. Choose a model

On the Fit tab of the nonlinear regression dialog, open the panel of Stimulatory dose-response equations and choose: log(agonist) vs. response -- Variable slope.



Choose an equation

- ⊞ **Recently used**
- ⊞ **User-defined equations**
- ⊞ **Standard curves to interpolate**
- ⊟ **Dose-response - Stimulation**
  - log(agonist) vs. response (three parameters)
  - log(agonist) vs. response -- Variable slope (four parameters)
  - log(agonist) vs. normalized response
  - log(agonist) vs. normalized response -- Variable slope
  - [Agonist] vs. response (three parameters)
  - [Agonist] vs. response -- Variable slope (four parameters)
  - [Agonist] vs. normalized response
  - [Agonist] vs. normalized response -- Variable slope
- ⊞ **Dose-response - Inhibition**
- ⊞ **Dose-response - Special**
- ⊞ **Binding - Saturation**
- ⊞ **Binding - Competitive**
- ⊞ **Binding - Kinetics**
- ⊞ **Enzyme kinetics - Inhibition**
- ⊞ **Enzyme kinetics - Velocity as a function of substrate**
- ⊞ **Exponential**
- ⊞ **Lines**

-If X is not already the log of dose, go back and transform your data.
-If any basal response has been subtracted, consider constraining Bottom to a constant value of 0.0.

log(agonist) vs. response -- Variable slope (four parameters)
Analytical derivatives

Learn about this equation

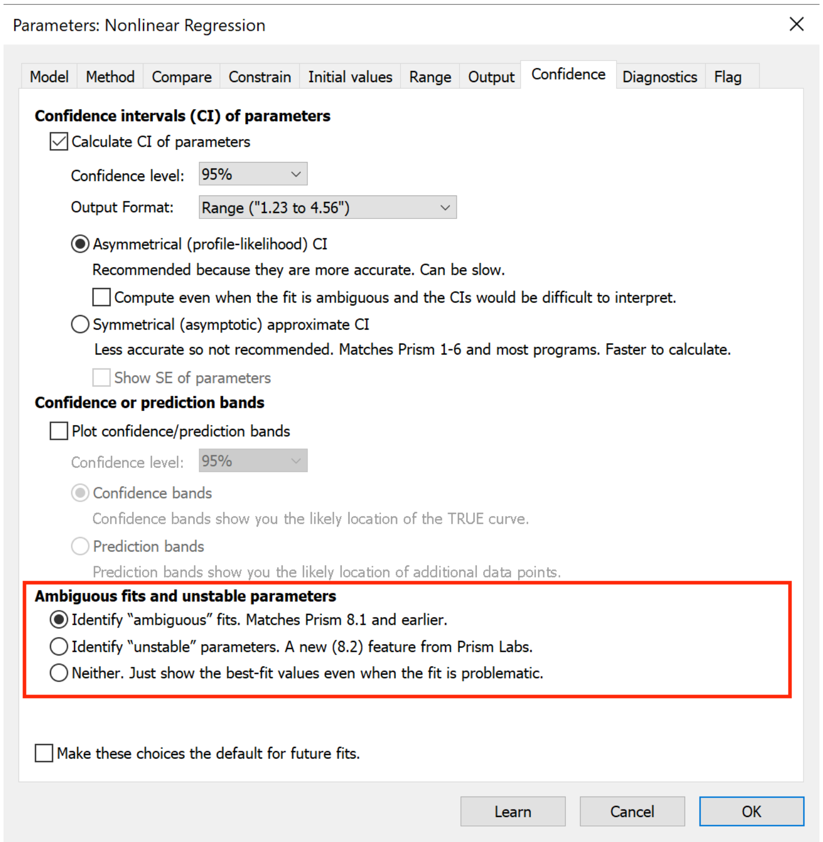
Interpolate

☐ Interpolate unknowns from standard curve. Confidence interval: None

Learn more about fitting dose-response curves 473.

6. Choose to detect "Ambiguous" model fits

On the Confidence tab of the nonlinear regression dialog, make sure that the option to "Identify "ambiguous" fits" is selected (this is the default option, but not the only method that Prism offers to detect when data don't quite fit the model).



Parameters: Nonlinear Regression

Confidence intervals (CI) of parameters

☑ Calculate CI of parameters

Confidence level: 95%

Output Format: Range ("1.23 to 4.56")

● Asymmetrical (profile-likelihood) CI

Recommended because they are more accurate. Can be slow.

□ Compute even when the fit is ambiguous and the CIs would be difficult to interpret.

○ Symmetrical (asymptotic) approximate CI

Less accurate so not recommended. Matches Prism 1-6 and most programs. Faster to calculate.

□ Show SE of parameters

Confidence or prediction bands

□ Plot confidence/prediction bands

Confidence level: 95%

● Confidence bands

Confidence bands show you the likely location of the TRUE curve.

○ Prediction bands

Prediction bands show you the likely location of additional data points.

Ambiguous fits and unstable parameters

● Identify "ambiguous" fits. Matches Prism 8.1 and earlier.

○ Identify "unstable" parameters. A new (8.2) feature from Prism Labs.

○ Neither. Just show the best-fit values even when the fit is problematic.

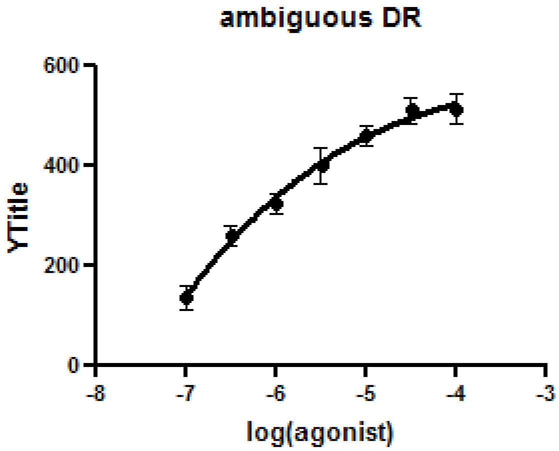
□ Make these choices the default for future fits.

For this example, leave all the other settings to their default values.

Click OK to see the curves superimposed on the graph.

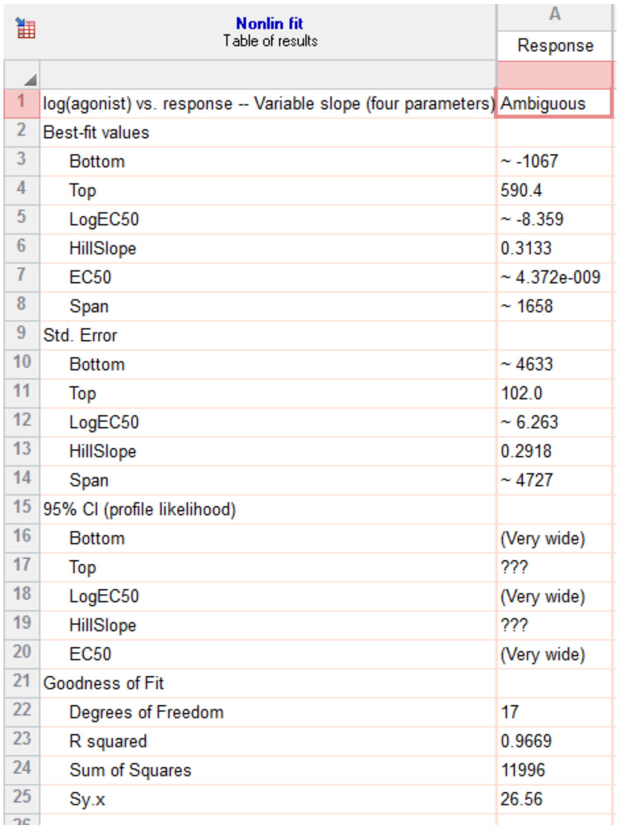
7. Inspect the graph

The curve goes through the point nicely, and looks fine.



8. Inspect the results

Go to the results page, and view the results.



	Nonlin fit Table of results	A Response
1	log(agonist) vs. response -- Variable slope (four parameters)	Ambiguous
2	Best-fit values	
3	Bottom	~ -1067
4	Top	590.4
5	LogEC50	~ -8.359
6	HillSlope	0.3133
7	EC50	~ 4.372e-009
8	Span	~ 1658
9	Std. Error	
10	Bottom	~ 4633
11	Top	102.0
12	LogEC50	~ 6.263
13	HillSlope	0.2918
14	Span	~ 4727
15	95% CI (profile likelihood)	
16	Bottom	(Very wide)
17	Top	???
18	LogEC50	(Very wide)
19	HillSlope	???
20	EC50	(Very wide)
21	Goodness of Fit	
22	Degrees of Freedom	17
23	R squared	0.9669
24	Sum of Squares	11996
25	Sy.x	26.56

Note the word 'ambiguous' at the top of the results. This means that Prism was unable to find a unique fit to these data. Lots of sets of parameter values would lead to curves that fit just as well. Learn more about ambiguous fits 437.

Prism does not report confidence intervals for the logEC50 or the Bottom of the curve, but instead simply says the intervals are 'very wide'. That tells you it was impossible to fit those parameters precisely.

While the Y values of the data rum about 100 to about 600, the best-fit value for the bottom of the curve is -1067. Furthermore, the best-fit value of the logEC50 is outside the range of the data.

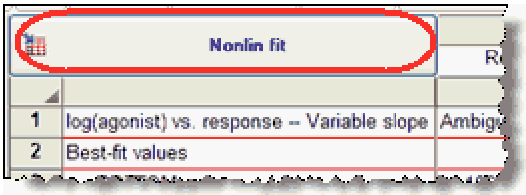
Even though the curve is close to the points (the R^2 is 0.9669), the best-fit parameter values are useless.

9. Constrain the Bottom of the curve

The problem is simple. You have asked Prism to find best-fit value for four parameters representing the bottom and top plateaus of the curve as well as the mid point and steepness. But the data simply don't define the bottom of the curve. In fact, Prism the best fit value of Bottom is very very far from the data.

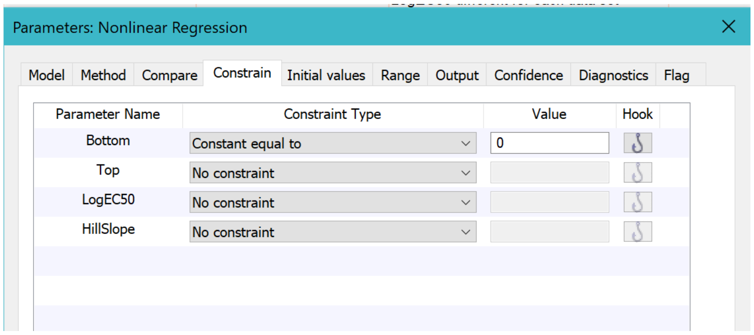
These data were calculated so the basal nonspecific response was subtracted away. This means that you know that the response (Y) at very low concentrations of agonist (very low values of X) has to be zero. Prism needs to know this to fit the data sensibly.

You don't have to do the fit over again. Instead click the button in the upper left corner of the results table to return to the nonlinear regression dialog.



Nonlin fit	R
1 log(agonist) vs. response -- Variable slope	Ambig
2 Best-fit values	

Go to the Constrain tab, and check constrain the parameter Bottom to have a constant value which you set to 0.0.

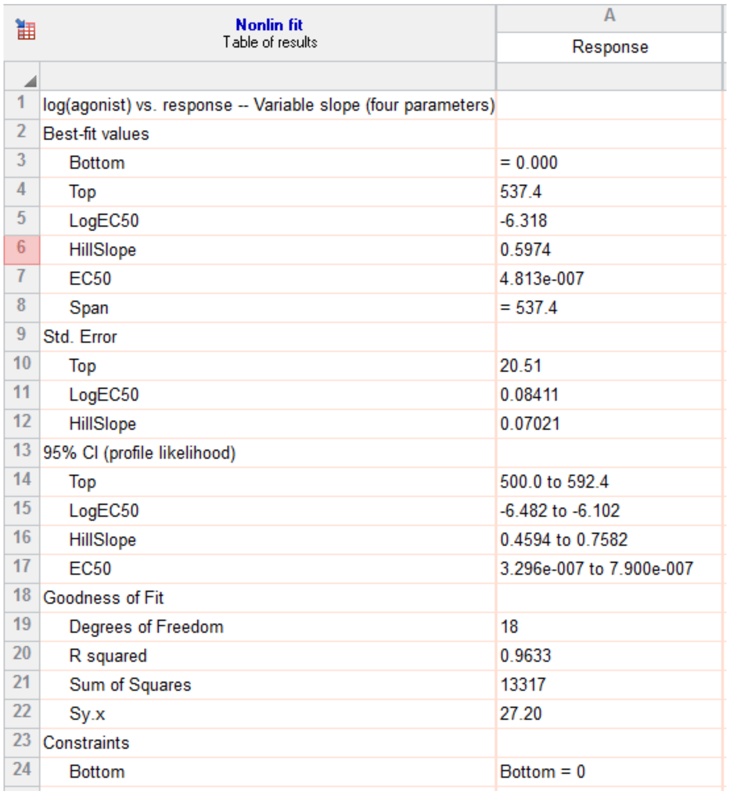


Parameters: Nonlinear Regression

Parameter Name	Constraint Type	Value	Hook
Bottom	Constant equal to	0	
Top	No constraint		
LogEC50	No constraint		
HillSlope	No constraint		

10. Inspect the revised results

Now the results make sense. The logEC50 is in the middle of the range of X values. The confidence intervals are reasonably tight. And, of course, the results are no longer 'ambiguous'.



	Nonlin fit Table of results	A Response
1	log(agonist) vs. response -- Variable slope (four parameters)	
2	Best-fit values	
3	Bottom	= 0.000
4	Top	537.4
5	LogEC50	-6.318
6	HillSlope	0.5974
7	EC50	4.813e-007
8	Span	= 537.4
9	Std. Error	
10	Top	20.51
11	LogEC50	0.08411
12	HillSlope	0.07021
13	95% CI (profile likelihood)	
14	Top	500.0 to 592.4
15	LogEC50	-6.482 to -6.102
16	HillSlope	0.4594 to 0.7582
17	EC50	3.296e-007 to 7.900e-007
18	Goodness of Fit	
19	Degrees of Freedom	18
20	R squared	0.9633
21	Sum of Squares	13317
22	Sy.x	27.20
23	Constraints	
24	Bottom	Bottom = 0

4.5.2 Nonlinear regression choices

- Choose the model (equation) on the Model tab 327. This is also where you choose to eliminate outliers and to interpolate values from the fit curve.

- Compare tab 334. Choose to compare the fit of two models to each data set, or to compare the fit of two (or more) data sets.

- Constrain tab 336. Fix parameter values to constant values and/or designate parameters to be shared among all data sets.

- There are many situations where you don't want to weight all points equally. Set unequal weighting on the Method tab 329.

- Nonlinear regression is an iterative procedure. The program must start with estimated initial values for each parameter.

It then adjusts these initial values to improve the fit. View those values on the Initial values tab 339.

- Range tab 341. Set the X range of data to fit, the starting and ending X values for the plotted curve, and whether you want to view a table of XY coordinates that define the curve.

- Output tab 342

- Diagnostics tab 347. Prism can create more detailed results tables.

4.5.2.1 Which choices are essential?

So many choices! Prism's nonlinear regression dialog has ten tabs, and can seem overwhelming at first. But you don't have to learn about all the choices when you first fit a curve.

If your goal is to fit a standard curve from which to interpolate unknown values

If your goal is to interpolate from a standard curve, you can ignore most of the choices that Prism offers. You do need to pick a model, but you should judge the adequacy of the model visually. If it goes through the points without wiggling too much, it is fine for interpolation. Note that Prism offers a simplified curve fitting analysis for interpolation 260.

If your goal is to fit a model to determine best-fit parameters

If your goal is to fit a model to determine parameter values, then choosing a model is the essential first step. You should not expect a computer (or a software company's tech support) to choose a model for you. Choosing a model, and deciding which parameters should be constrained and which should be shared among data sets, is a scientific decision that is fundamental to analyzing your data. The other choices are useful, but can be put aside when you are first learning curve fitting..

4.5.2.2 Model tab

Choose an equation

Selecting an equation is the most important step in fitting a curve. The choice cannot be automated 16.

If your goal is to fit a model in order to understand your data and make comparisons, then choosing a model is a scientific decision that you must make with care. If your goal is to interpolate unknowns from a standard curve, then it matters less which equation you pick, so long as it ends up creating a smooth curve through your data.

Some tips:

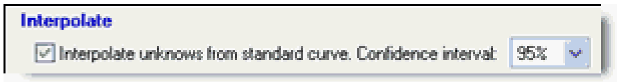
- Prism provides a long list of equations 444 that you can choose. But if these don't fit your needs, don't be afraid to create or clone an equation 726 to fit your needs.

- Part of choosing a model is choosing constraints 336. Don't skip that step. For example, if you choose a sigmoidal dose response model, you must decide whether you wish Prism to find the best fit value of the bottom plateau based on the trend of the data. The alternative is to constrain the bottom plateau to equal zero, if you have subtracted off a baseline, or some other value (defined by controls). A computer can't make these decisions for you. Choosing which constraints to apply to your model is a fundamental decision in data analysis that can have a huge impact on the results.

- If you are fitting several data sets at once, part of choosing a model is deciding which parameters you want to share between data sets. When you share a parameter (a choice on the Constrain tab 336), Prism finds one best-fit value for the parameter that applies to all the data sets. Read more 70 about shared parameters (global fitting).

Interpolate

Check this option in order to interpolate the concentration of unknown samples from the best-fit curve. Learn more.



Interpolate

☑ Interpolate unknows from standard curve. Confidence interval: 95%

With this option, Prism will report the Y value for any X values you enter, and the X value for any Y values you enter (including extrapolating in each direction a distance equal to half the length of the X axis).

4.5.2.3 Method tab

Outliers

Use with caution

Automatic outlier removal 88 is extremely useful, but can lead to invalid (and misleading) results in some situations 88, so should be used with caution. You can ask Prism to simply identify and count values it identifies as outliers. Or you can ask it to exclude identified outliers from the data set being fit.

Rout coefficient

If choose to exclude or identify outliers, set the ROUT coefficient Q 92 to determine how aggressively Prism defines outliers.

We recommend using a value of 1%. Our simulations have shown that if all the scatter is Gaussian, Prism will falsely find one or more outliers in about 2-3% of experiments. If there really are outliers present in the data, Prism will detect them with a False Discovery Rate less than 1%. See reference 1.

If you set Q to a higher value, the threshold for defining outliers is less strict. This means that Prism will have more power to detect outliers, but also will falsely detect 'outliers' more often. If you set Q to a lower value, the threshold for defining outliers is stricter. This means that Prism will have a less power to detect real outliers, but also have a smaller chance of falsely defining a point to be an outlier.

If you set Q to 0, Prism will fit the data using ordinary nonlinear regression without outlier identification.

Create table of cleaned data

Check the option (introduced with Prism 8) to create a new analysis tab with a table of cleaned data (data without outliers). Prism does not automatically graph this table of cleaned data, but it is easy to do so (New..Graph of existing data).

Prism always creates an analysis tab table of outliers, and there is no option to not show this.

Fitting method

Prism offers four choices of fitting method:

Least-squares

This is standard nonlinear regression. Prism minimizes the sum-of-squares of the vertical distances between the data points and the curve, abbreviated *least squares*. This is the appropriate choice if you assume that the distribution of residuals (distances of the points from the curve) are Gaussian.

Robust regression

Robust regression 94 is less affected by outliers, but it cannot generate confidence intervals for the parameters, so has limited usefulness. Its main use in Prism is as a first step in outlier detection. It is the baseline from which to determine if a residual is "too large" so the point should be declared an outlier. It is rarely helpful to perform robust regression on its own, but Prism offers you that choice if you want to.

Poisson regression

Choose Poisson regression 68 when every Y value is the number of objects or events you counted. These must be the actual counts, not normalized in any way. If a machines says your sample had 98.5 radioactive decays per minute, but you asked the counter to count each sample for ten minutes, then it counted 985 radioactive decays. That is the value you should enter for Poisson regression. If the Y values are normalized counts, and are not actual counts, then you should not choose Poisson regression.

Don't fit

Nonlinear regression works iteratively, and begins with initial values 339 for each parameter. Check "don't fit the curve" to see the curve generated by your initial values. If the curve is far from the data, go back to the initial parameters tab and enter better values for the initial values. Repeat until the curve is near the points. Then go back to the Methods tab and check "Fit the curve". This is often the best way to diagnose problems with nonlinear regression.

Convergence criteria

Nonlinear regression is an iterative process. It starts with initial values 339 of the parameters, and then repeatedly changes those values to increase the goodness-of-fit. Regression stops when changing the values of the parameters makes a trivial change in the goodness of fit.

How strict?

Prism lets you define the convergence criteria in three ways.

- Quick. If you are fitting huge data sets, you can speed up the fit by using the 'quick' definition of convergence. With this choice, nonlinear regression is defined to converge when two iterations in a row change the sum-of-squares by less than 0.01%.

- Medium (default). Nonlinear regression is defined to converge when five iterations in a row change the sum-of-squares by less than 0.0001%.

- Strict. If you are having trouble getting a reasonable fit, you might want to try the stricter definition of convergence. With this choice, the nonlinear regression iterations don't stop until five iterations in a row change the sum-of-squares by less than 0.00000001%. It won't help very often, but might be worth a try. The only reason not to always use the strictest choice is that it takes longer for the calculations to complete. That won't matter with small data sets, but will matter with large data sets or when you run scripts to analyze many data tables.

Maximum number of iterations

While fitting a curve, Prism will stop after that many iterations. The default is 1000, and there is little reason to enter a different value. One reason would be if you are running a script to automatically analyze many data tables, each with many data points. The fits might be slow enough that it makes sense to lower the maximum number of iterations so Prism won't waste time trying to fit impossible data.

Weighting method

It is often useful to differentially weight the data points. Learn why. 57

Prism offers seven choices on the Weight tab of nonlinear regression:

No weighting. Regression is most often done by minimizing the sum-of-squares of the vertical distances of the data from the line or curve. Points further from the curve contribute more to the sum-of-squares. Points close to the curve contribute little. This makes sense, when you expect experimental scatter to be the same, on average, in all parts of the curve.

Weight by 1/Y^2. In many experimental situations, you expect the average distance (or rather the average absolute value of the distance) of the points from the curve to be higher when Y is higher. The points with the larger scatter will have much larger sum-of-squares and thus dominate the calculations. If you expect the relative distance (residual divided by the height of the curve) to be consistent, then you should weight by 1/Y2.

Weight by 1/Y. This choice is useful when the scatter follows a Poisson distribution -- when Y represents the number of objects in a defined space or the number of events in a defined interval.

Weight by $1/Y^K$. Also called "General weighting". Read more. 59

Weight by 1/X or $1/X^2$ .These choices are used rarely. Only choose these weighting schemes when it is the standard in your field, such as a linear fit of a bioassay.

Weight by $1/SD^2$. If you enter replicate Y values at each X (say triplicates), it is tempting to weight points by the scatter of the replicates, giving a point less weight when the triplicates are far apart so the standard deviation (SD) is high. But unless you have lots of replicates, this doesn't help much. The triplicates constituting one mean could be far apart by chance, yet that mean may be as accurate as the others. Weighting needs to be based on *systematic* changes in scatter. The choice to weight by $1/SD^2$ is most useful when you want to use a weighting scheme not available in Prism. In this case, enter data as mean and SD, but enter as "SD" weighting values that you computed elsewhere for that point. In other words, the values you enter in the SD subcolumn are not actually standard deviations, but are weighting factors computed elsewhere.

Notes on weighting

- If you have normalized your data, weighting rarely makes sense.

- Simulations can show you how much difference it makes if you choose the wrong weighting scheme.

- If you choose unequal weighting, Prism takes this into account 361 when plotting residuals.

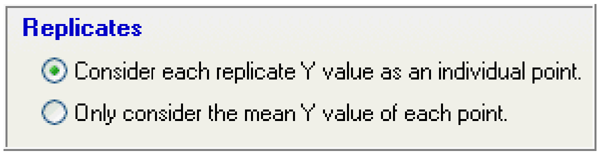
- Prism accounts for weighting when it computes R 377 2 377.

- Learn about the math of weighting 59 and how Prism does the weighting 101.

- If you choose robust regression (on the first, Fit, tab) then choices you make on the weighting tab will not influence the fitting, so will no influence the best-fit parameter values or the curve. However your choice of weighting will have an impact on the residuals Prism computes and graphs and on how it identifies outliers. Both these choices (residuals, and outlier counting) are on the Diagnostics tab of nonlinear regression.

- If you ask Prism to remove outliers, the weighting choices don't affect the first step (robust regression). Then outliers are identified by looking at the size of the weighted residuals. Finally, the cleaned data (without outliers) are fit with weighted regression.

Replicates



Replicates

● Consider each replicate Y value as an individual point.

○ Only consider the mean Y value of each point.

Choose whether to fit all the data (individual replicates if you entered them, or accounting for SD or SEM and n if you entered the data that way) or to just fit the means.

If you fit only the means, Prism "sees" fewer data points, so the confidence intervals on the parameters tend to be wider, and there is less power to compare alternative models. For these reasons,when possible you should choose to let the regression see each replicate as a point and not see means only.

The issue comes down to one of independence. An important assumption of regression is that the residuals from all data points are independent. Here is an example where the replicates are not independent, so you would want to fit only the means: You performed a dose-response experiment, using a different animal at each dose with triplicate measurements. The three measurements are not independent because if one animal happens to respond more than the others, all the replicates are likely to have a high value. Since the replicates are not independent, you should fit the means and not the individual replicates.

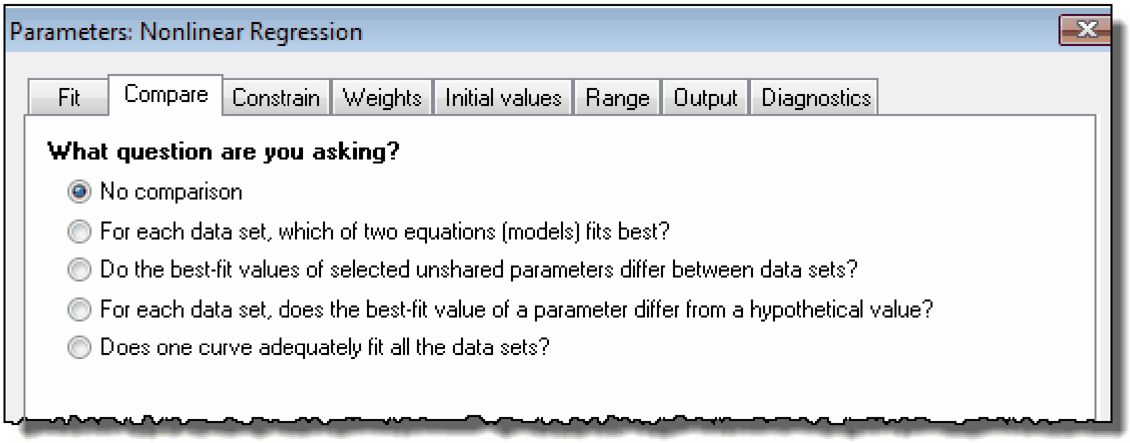
If you entered the data as mean, n, and SD or SEM Prism gives you the choice of fitting just the means, or accounting for SD and n. If you make that second choice Prism will compute exactly the same results from least-squares regression as you would have gotten had you entered raw data.

Reference

1. Motulsky HM and Brown RE, Detecting outliers when fitting data with nonlinear regression – a new method based on robust nonlinear regression and the false discovery rate, BMC Bioinformatics 2006, 7:123..

4.5.2.4 Compare tab

What question are you asking?



What question are you asking?

● No comparison

○ For each data set, which of two equations (models) fits best?

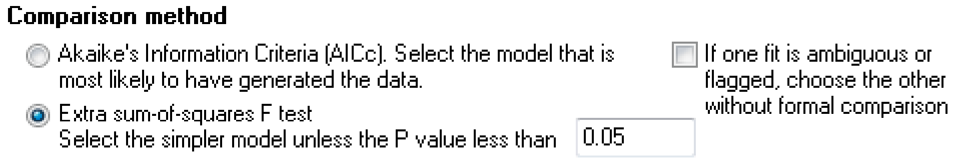
○ Do the best-fit values of selected unshared parameters differ between data sets?

○ For each data set, does the best-fit value of a parameter differ from a hypothetical value?

○ Does one curve adequately fit all the data sets?

When fitting biological data with regression, your main objective is often to discriminate between different models, to ask if an experimental intervention changed a parameter, or to ask if the best-fit value of a parameter differs significantly from a theoretical value. Learn more about these four kinds of comparisons 77. Your choice, of course, has to be based on your experimental goals.

Comparison method



Comparison method

○ Akaike's Information Criteria (AICc). Select the model that is most likely to have generated the data.

● Extra sum-of-squares F test
Select the simpler model unless the P value less than 0.05

□ If one fit is ambiguous or flagged, choose the other without formal comparison

Prism can perform the comparison using two alternative methods 78: the extra sum-of-squares F test 82, and using Akaike's information criteria 84. Use these guidelines to choose:

- In most cases, the two models will be 'nested'. This means that one model is a simpler case of the other. For example, a one-phase exponential model is a simpler case of a two-phase exponential model. Either the F test or the AICc method may be used with nested models. The choice is usually a matter of personal preference and tradition. Basic scientists in pharmacology and physiology tend to use the F test. Scientists in fields like ecology and population biology tend to use AICc.

- If the models are not nested, then the F test is not valid so you should choose AICc. Note that Prism does not enforce this. It will calculate the F test even if the models are not nested, but the results won't be useful.

The extra sum-of-squares F test only makes sense when the models being compared have different numbers of parameters, and so have different numbers of degrees of freedom. If you want to compare two models with the same number of parameters, choose the AIC method.

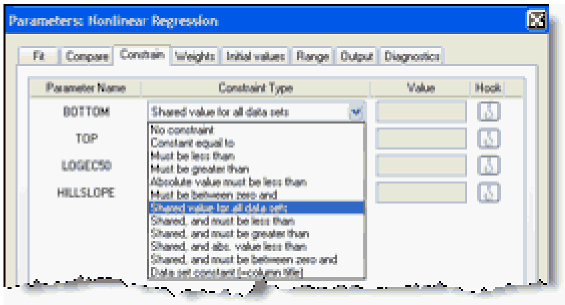
Check the option on the right, and before running the extra sum-of-squares F or AICc comparison, Prism checks whether both models actually fit the data. If one fit is ambiguous 437 or flagged 350, Prism will choose the other without any calculation.

If you are comparing the fits of three or more models, fit individually, you can ask Prism to report the AICc with each fit. This is a choice on the Diagnostics tab.

4.5.2.5 Constrain tab

The importance of constraints

Prism lets you constrain each parameter to a constant value, constrain to a range of values, share among data sets (global fit), or define a parameter to be a column constant. This is an important decision, and will influence the results.


Parameters: Nonlinear Regression
Fit
Compare
Constrain
Weights
Initial values
Range
Output
Diagnostics
No constraint
Constant equal to
Must be less than
Must be greater than
Absolute value must be less than
Must be between zero and
Shared value for all data sets
Shared, and must be less than
Shared, and must be greater than
Shared, and abs. value less than
Shared, and must be between zero and
Data set constant (=column title)

Parameter Name	Constraint Type	Value	Hook
BOTTOM	Shared value for all data sets		
TOP			
LOGEC50			
HILLSLOPE			

Constrain to a constant value

You won't always want to fit all the parameters in a model. Instead, you can fix one or more parameters to constant values. For example, if you have normalized a dose-response curve to run from 0 to 100, constrain Top to equal 100 and Bottom to 0.0. Similarly, if you have subtracted a baseline so you know that the exponential decay curve has to plateau at Y=0.0, you can constrain the Bottom parameter to equal 0.0.

Deciding to constrain a parameter to a constant value can make a huge difference in the results. Remember Prism has no common sense. It doesn't know if a curve has to plateau at zero unless you tell it.

Constrain to a range of values

Constrain to a range of values to prevent Prism from letting parameters take on impossible values. For example, you should constrain rate constants to only have values greater than 0.0, and fractions (say the fraction of binding sites that are high affinity) that have a value between 0.0 and 1.0. Setting this kind of constraint can have four outcomes:

- The constraint is irrelevant, as the parameter never would have taken on a value in the forbidden range.

- The constraint helped speed up the fit. Nonlinear regression works by iteratively changing the values of the parameters. With some complicated fits, the nonlinear regression process can 'get confused' and end up spending time exploring parameter values that make no sense. Constraining the values of one or more parameters can prevent the nonlinear regression process from being led astray. With huge numbers of data points, you might see a noticeable speeding up of the fitting process.

- The constraint helped nonlinear regression choose from several local minima. Nonlinear regression works by changing parameter values step by step until no small change affects the sum-of-squares (which quantifies goodness-of-fit). With some models, there can be two sets of parameter values that lead to local minima in sum-of-squares. Applying a constraint can ensure that nonlinear regression finds the minimum with scientifically relevant values, and ignores another minimum that fits the curve well but using parameter values that make no scientific sense (i.e. negative concentrations).

- The constraint prevents nonlinear regression from finding a minimum sum-of-squares. Instead, the best the program can do (while obeying the constraint) is set the parameter to the limit of the constrained range. Prism then reports that the fit 'hit constraint'.

In the first case, the constraint is harmless but useless.

In the next two cases, the constraint helps the nonlinear regression reach sensible results. Essentially, the constraint can give the nonlinear regression process some scientific judgment about which parameter values are simply impossible. These cases are really what constraints are for.

The last case, when the fit ends with a parameter set to one end of its constraint, it is tricky to interpret the results. 441

Constraints on constraints

Constraints can be for one parameter (Kfast must be greater than zero) or for the relationship between two parameters (Kfast must be greater than Kslow). But note that you cannot invoke a constraint between two parameters if both parameters also are themselves constrained. In a two phase exponential equation, you may want to constrain both parameters to be greater than zero, and also define one rate constant to be larger than the other (Kfast > Kslow). Prism won't let you do that. What you have to do is define one constraint that Kfast is greater than zero, and another that Kfast is greater than Kslow. But don't put in the constraint that Kslow is greater than zero. That is implied by being larger than Kfast.

Sharing parameters among data sets. Global nonlinear regression.

If you are fitting a family of curves, rather than just one, you can choose to share some parameters between data sets. For each shared parameter, Prism finds one (global) best-fit value that applies to all the data sets. For each non-shared parameter, the program finds a separate (local) best-fit value for each data set. Global fitting is a very useful tool in two situations:

- The parameter(s) you care about are determined from the relationship between several data sets. Learn more. 74
- Each dataset is incomplete, but the entire family of datasets defines the parameters. See example. 72

Data set constant

When you fit a family of curves at once, you can set one of the parameters to be a data set constant. Prism offers two data set constants.

Column title. The value comes from the column title, which can be different for every data set. This parameter becomes almost a second independent variable. It has a constant value within any one data set, but a different value for each data set. For example, when fitting a family of enzyme progress curves in the presence of various concentrations of inhibitor, the inhibitor concentration can be entered into the column title of the data table. View an example 746.

Mean X. The value is the mean of all the X values for which there are Y values in that data set. This is used in centered polynomial regression.

Different constants for different data sets

When you use the Constrain tab to set a parameter to a constant value, you do so for all data sets. It is not possible to enter a different constant value for each data set (unless you use the column title as the constant value as described in the previous section. If you want a parameter to have a different constant value (not the column title) for each data set, you'll need to write a user defined equation 714, and use special notation to assign a different value for each data set. You'll need to hard code the constants within the user-defined equation, and cannot choose them in the nonlinear regression parameters dialog. For example:

```
<A>Bottom=4.5
<B>Bottom=34.5
<C>Bottom=45.6
Y=Bottom + span*(1-exp(-1*K*X))
```

In this example, the parameter Bottom is set to 4.5 when fitting data set A, to 34.5 when fitting data set B, and to 45.6 when fitting data set C.

4.5.2.6 Initial values tab

Why initial values?

Nonlinear regression is an iterative procedure. The program must start with estimated initial values for each parameter. It then adjusts these initial values to improve the fit.

How does Prism provide initial values automatically?

Every equation built-in to Prism, as well as equations you define, include rules to compute initial values. These rules use the range of the X and Y values to come up with initial values, which become the original automatic initial values. You can change the rules 721 for user-defined equations, and can clone built-in equations to make them user-defined. The new rules will be invoked when you next choose this equation for a new analysis. It won't change the initial values for the analysis you are working on.

Are the initial values reasonable?

If nonlinear regression doesn't report any results, or if they seem wrong, the problem might be bad initial values. To check whether the initial values are reasonable, check "**Don't fit the curve. Instead plot the curve defined by initial values."** on the Diagnostics tab. When you click OK from the nonlinear regression dialog, Prism will not fit a curve but will instead generate a curve based on your initial values. If this curve is not generally in the vicinity of the data points, change the initial values before running nonlinear regression.

Changing the initial values

In most cases, the automatic initial values will work fine. If you have trouble getting a sensible fit, you might need to tweak the initial values. You can override the automatic initial values for each parameter for each data set:

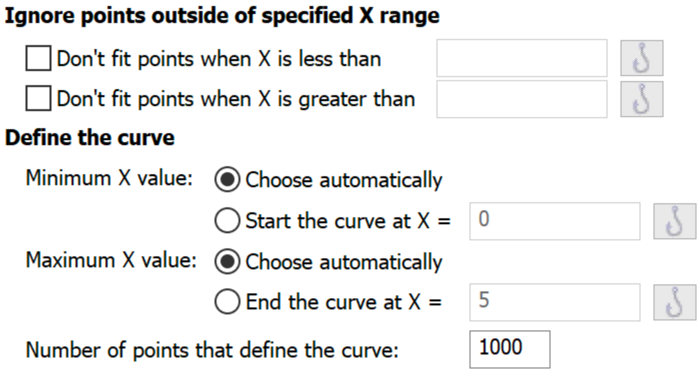
1. Select a data set on the top part of the initial values tab. Or hold down the Ctrl key and select several data sets. Or click "Select all" in the top right of the tab.
2. Uncheck "choose automatically" and enter the new initial value.

you will find it easy to estimate initial values if you have looked at a graph of the data, understand the model, and understand the meaning of all the parameters in the equation. Remember that you just need an estimate. It doesn't have to be very accurate.

How much difference do initial values make?

When fitting a polynomial model, it makes no difference what values you enter as initial values. When fitting other models, the importance of the initial values depends on how well the data define the curve and on how many parameters are in the model. Initial values matter the most when your data have a lot of scatter and don't define the model very well, and your model has many parameters.

4.5.2.7 Range tab



Ignore points outside of specified X range

☐ Don't fit points when X is less than

☐ Don't fit points when X is greater than

Define the curve

Minimum X value: ● Choose automatically ○ Start the curve at X = 0

Maximum X value: ● Choose automatically ○ End the curve at X = 5

Number of points that define the curve: 1000

Ignore points outside of a specified X range

If you choose these options nonlinear regression will completely ignore all data where the X value is less than, or larger than, the thresholds you set. This could be useful if you collect data over time, and only want to fit data within a certain range of time points.

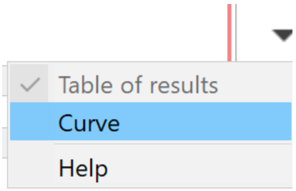
Define the curve

In addition to fitting the model to your data, Prism also superimposes the curve on your graph. Choose where the curves should begin and end, and how many equally spaced points define the curve.

Table of XY coordinates

Prior to version 8, Prism gave you an option on this tab to show (or not show) a table of XY coordinates. Prism 8 does not give you this option here.

Instead, Prism 8 always creates this results tab, but it is hidden by default. Use the menu to the right of the analysis tabs to check the option to view this tab. This will let you see the XY coordinates of the points that define the curve, and also do further analyses with that table.



4.5.2.8 Output tab

Summary table

When analyzing several data sets, the results table is rather lengthy. To display key results on a summary table, check the option box to create a summary table and select the variable you wish to summarize. Prism creates a summary table (as an additional results view) that shows the best-fit value of that parameter for each data set, and graphs this table.

Depending on your choices in the dialog, this may be a bar graph or an XY graph. It shows the best-fit value of a selected parameter for each data set on the table. In some cases, you may analyze the summary table with linear or nonlinear regression. For example, the summary graph may show the best-fit value of a rate constant as a function of concentration (obtained from the column titles of the original data). You can fit a line or curve to that graph.

When Prism compares the fits of two equations, it shows only the results for the second equation. Since this may not be helpful, we suggest that you only make summary tables when fitting a single equation.

Number of digits in output

Choose how many significant digits you want to see in the main results table. This is especially useful if you embed the results table on a graph or layout. If you choose to show many digits of precision in the results, you probably should also choose strict convergence criteria on the Diagnostics tab 347.

For the table of residuals and the curve itself, you can choose a number format by selecting column(s) and bring up the Decimal Format dialog.

Additional output (legacy for old Prism versions)

The options here (dose ratios, and Ki from IC50) are available only to be compatible with earlier versions of Prism (so Prism can open files made with older versions). They are almost always unavailable, becoming available only if you select certain equations from the classic equations list.

Since Prism 5, you can calculate a Gaddum/Schild EC50 shift 541 directly, without need to separately compute dose ratios. Similarly, you can fit competitive binding curves directly to determine the Ki for one[585] or two[590] sites, without a separate Cheng-Prusoff calculation of Ki from IC50(1).

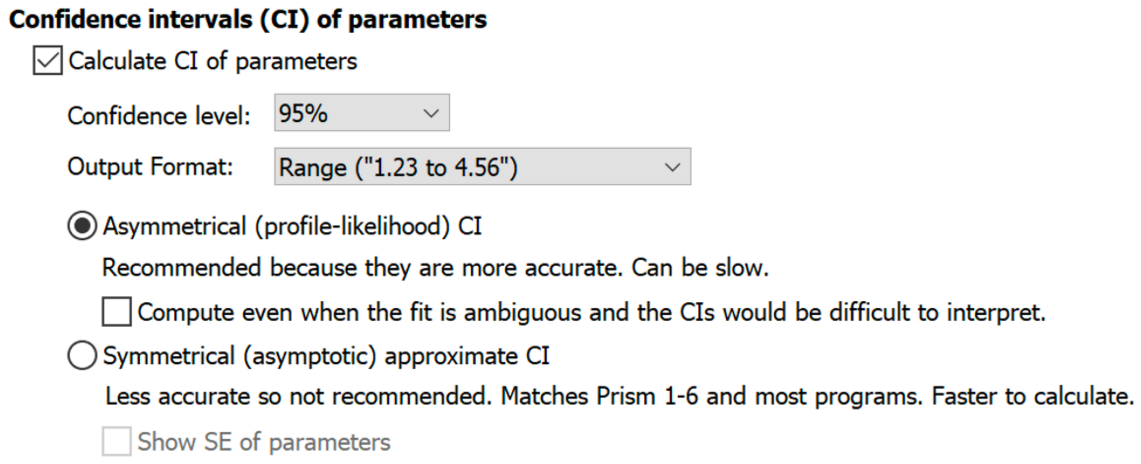
1. Cheng, Y. and Prusoff, W. H. Relationship between the inhibition constant (K1) and the concentration of inhibitor which causes 50 per cent inhibition (I50) of an enzymatic reaction. Biochem Pharmacol, 22: 3099-3108, 1973.

4.5.2.9 Confidence tab

This is an important tab, with three big choices: how to report the confidence interval of the parameters, whether to graph the confidence or prediction bands of the curve, and what to do when the data don't provide enough information to fit all the parameters.

Confidence intervals of the parameters

If the reason you want to run nonlinear regression is to interpolate unknown values, you won't really care about the values of the parameters so won't care about the confidence interval of the parameters. But if you do care about the values of the parameters, we suggest that you always ask Prism to report confidence intervals, as inspecting the confidence intervals of best-fit parameters is an essential part of evaluating any nonlinear fit.



Confidence intervals (CI) of parameters

☑ Calculate CI of parameters

Confidence level: 95%

Output Format: Range ("1.23 to 4.56")

● Asymmetrical (profile-likelihood) CI

Recommended because they are more accurate. Can be slow.

□ Compute even when the fit is ambiguous and the CIs would be difficult to interpret.

○ Symmetrical (asymptotic) approximate CI

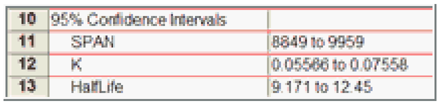
Less accurate so not recommended. Matches Prism 1-6 and most programs. Faster to calculate.

□ Show SE of parameters

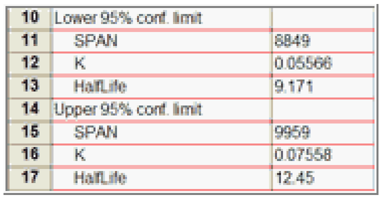
Confidence level and output format

Confidence intervals are traditionally usually computed for 95% confidence, but you can choose another confidence level.

Prism can report the confidence intervals in two ways: as a range or as separate blocks of lower and upper confidence limits (useful if you want to paste the results into another program). The former is easier to read. The latter might be better if you are tabulating the results elsewhere.



10	95% Confidence Intervals	
11	SPAN	8849 to 9959
12	K	0.05566 to 0.07558
13	HalfLife	9.171 to 12.45



10	Lower 95% conf. limit	
11	SPAN	8849
12	K	0.05566
13	HalfLife	9.171
14	Upper 95% conf. limit	
15	SPAN	9959
16	K	0.07558
17	HalfLife	12.45

How to compute the intervals

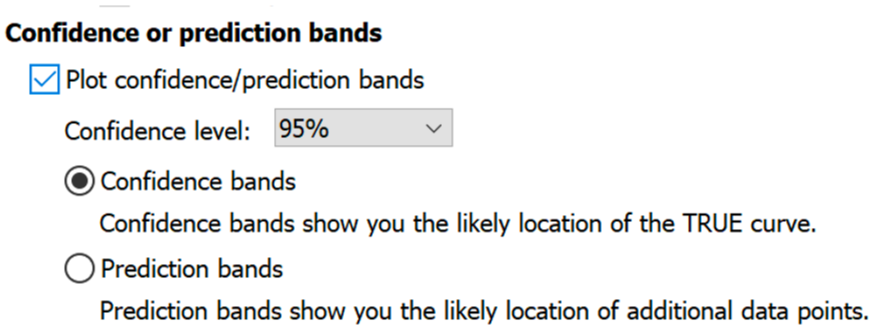
Prism offers two methods to compute the confidence intervals. Don't confuse the two choices here with the two choices for reporting the CI of parameter transforms 715.

- Asymmetrical (and thus more accurate) profile likelihood confidence intervals 104. We recommend this choice because it simply does a better job of quantifying how precisely you know the parameters value. The uncertainty often is asymmetrical, so it is better to use a method that can report asymmetrical intervals. The only disadvantage is that the calculations are more complex so are noticeably slower with huge data sets (especially with user-defined equations).

- Asymptotic approximate symmetrical confidence intervals. These are also called Wald confidence intervals. These were the only confidence intervals reported by Prism 6 and earlier and by most programs. But since the true uncertainty in the parameter's value is often asymmetrical, these symmetrical intervals are not always accurate. We suggest choosing them only when you need to compare Prism's results with other programs, when you need to be consistent with earlier work, or when you have so much data that the profile likelihood method is too slow.

If you choose the asymptotic symmetrical confidence intervals, you can also ask Prism to report the standard errors of the parameters. The standard errors are intermediate values used to compute the symmetrical confidence intervals, but are not very meaningful by themselves. You may want to include standard errors in the results to compare Prism's results to those of another program that doesn't report confidence intervals or to collaborate with colleagues who don't understand confidence intervals. But we recommend routinely turning off reporting of the standard errors, because they do a poor job of conveying the precision of the best-fit parameter values. Starting with version 8.2, Prism only offers the choice of reporting the standard error values if you choose to report the symmetrical confidence intervals.

Confidence and prediction bands



Confidence or prediction bands

☑ Plot confidence/prediction bands

Confidence level: 95%

● Confidence bands

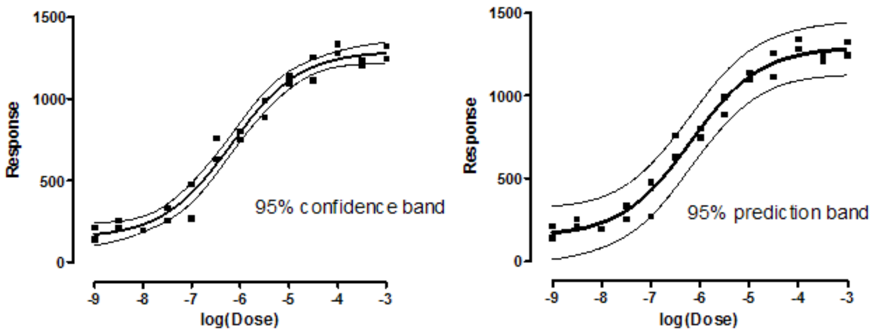
Confidence bands show you the likely location of the TRUE curve.

○ Prediction bands

Prediction bands show you the likely location of additional data points.

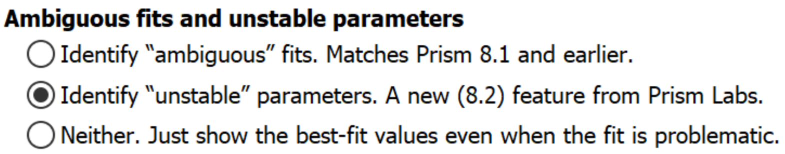
The 95% **confidence bands** enclose the area that you can be 95% sure contains the true curve. It gives you a visual sense of how well your data define the best-fit curve.

The 95% **prediction bands** enclose the area that you expect to enclose 95% of future data points. This includes both the uncertainty in the true position of the curve (enclosed by the confidence bands), and also accounts for scatter of data around the curve. Therefore, prediction bands are always wider than confidence bands. When you have lots of data points, the discrepancy is huge.



Learn more about confidence and prediction bands. 396

Ambiguous fits and unstable parameters



Ambiguous fits and unstable parameters

○ Identify "ambiguous" fits. Matches Prism 8.1 and earlier.

● Identify "unstable" parameters. A new (8.2) feature from Prism Labs.

○ Neither. Just show the best-fit values even when the fit is problematic.

Sometimes your data simply won't define the model very well. Prism offers you three ways to deal with this situation.

- Identify "ambiguous 437" fits. This was the method always used by Prism 7.0 to 8.1. It is useful if you want results to be consistent with older work. If any parameter has a dependency greater than 0.9999, Prism marks the fit "ambiguous", precedes the best fit value with a tilde (~) to show it is not to be believed, and doesn't show its confidence interval. Learn more about how Prism calculates dependency 111.

- Identify unstable parameters. This approach was introduced in Prism 8.2 as a Prism Labs (experimental) feature. We welcome feedback. In rare circumstances, a dependency can be >0.9999 and yet it is still possible to obtain a useful best-fit value of the parameter and a useful confidence interval. Our "ambiguous" method based on dependency didn't report any results for these parameters. Our new method works quite differently 114 and can find best-fit values or confidence intervals for data sets where the "ambiguous" method could not. When applicable, Prism will display the word "unstable" in place of that parameter's best fit value and confidence interval.

- Neither. Just report the best-fit values no matter what. This is what most other programs do.

4.5.2.10 Diagnostics tab

How to quantify goodness-of-fit

The choices vary depending on what fitting method you chose in the Methods tab.

If you chose least-squares regression

- R[377]2[377] is the standard way to assess goodness-of-fit.

- The adjusted R2[382] takes into account the number of parameters fit to the data, so has a lower value than R^2 (unless you fit only one parameter, in which case R^2 and adjusted R^2 are identical). It is not commonly reported with nonlinear regression.

- The sum-of-squares (or weighted sum-of-squares) is the value that Prism minimizes when it fits a curve. Reporting this value is useful only if you want to compare Prism's results to those of another program, or you want to do additional calculations by hand.

- Sy.x and RMSE[383] are alternative ways to quantify the standard deviation of the residuals. We recommend the Sy.x, which is also called Se.

- The AICc[84] is useful only if you separately fit the same data to three or more models. You can then use the AICc to choose between them. But note that it only makes sense to compare AICc between fits when the only difference is the model you chose. If the data or weighting are not identical between fits, then any comparison of AICc values would be meaningless.

If you chose robust regression

If you chose robust regression, Prism can compute a Robust Standard Deviation of the Residuals (RSDR). The goal here is to compute a robust standard deviation, without being influenced by outliers. In a Gaussian distribution, 68.27% of values lie within one standard deviation of the mean. We therefore calculate the range that contains 68.27% of the residuals. It turns out that this value underestimates the SD a bit, so the RSDR is computed by multiplying that empirical value by n/(n-K), where K is the number of parameters fit.

If you chose Poisson regression

If you chose Poisson regression, Prism offers three ways to quantify the goodness of fit 387: the pseudo-R2, the dispersion index and the model deviance. The pseudo-R2 can be interpreted pretty much like an ordinary R2. The other two values will be of interest only to those who have studied Poisson regression in depth.

Are the residuals Gaussian ?

Least-squares nonlinear regression assumes that the distribution of residuals follows a Gaussian distribution (robust nonlinear regression does not make this assumption). Prism can test this assumption by running a normality test 375 on the residuals. Prism offers four normality tests. We recommend the D'Agostino-Pearson test.

These tests make no sense and are not available if you chose robust or Poisson regression.

Are the residuals clustered or heteroscedastic ?

Does the curve follow the trend of the data? Or does the curve systematically deviate from the trend of the data? Prism offers two tests that answer these questions.

If you have entered replicate Y values, choose the replicates test 390 to find out if the points are 'too far' from the curve (compared to the scatter among replicates). If the P value is small, conclude that the curve does not come close enough to the data.

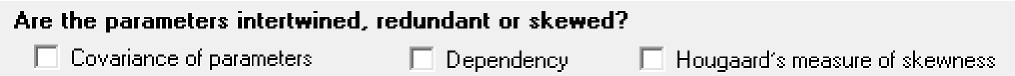
The runs test 389 is available if you entered single Y values (no replicates) or chose to fit only the means rather than individual replicates (Method tab). A 'run' is a series of consecutive points on the same side of the curve. If there are too few runs, it means the curve is not following the trend of the data. If you fit several curves at once, sharing one or more parameters (but not all the parameters) with global regression, Prism will report the runs test for each curve fit, but not for the global fit. Prior versions of Prism reported the overall runs test for the global fit, by summing the runs for each component curve, but this is not standard and Prism no longer reports this.

Nonlinear regression assumes that, on average, the distance of the points from the curve is the same all the way along the curve, or that you have accounted for systematic differences by choosing an appropriate weighting. Prism can test this assumption with a test for appropriate weighting 401. If you have chosen equal weighting (the default) this is the same as a test for homoscedasticity.

Residual plots

If you choose a residual plot 361, Prism creates a new graph. Prism offers a choice of four different residual graphs. Viewing a residual plot can help you assess whether the distribution of residuals is random above and below the curve.

Are the parameters intertwined or redundant or skewed?



Are the parameters intertwined, redundant or skewed?

□ Covariance of parameters □ Dependency □ Hougaard's measure of skewness

What does it mean for parameters to be intertwined? After fitting a model, change the value of one parameter but leave the others alone. The curve moves away from the points. Now, try to bring the curve back so it is close to the points by changing the other parameter(s). If you can bring the curve closer to the points, the parameters are *intertwined*. If you can bring the curve back to its original position, then the parameters are *redundant*. In this case, Prism will alert you by labeling the fit 'ambiguous' 437.

We suggest that you report the dependency 393, and not bother with the covariance matrix 393. When you are getting started with curve fitting, it is OK to leave both options unchecked.

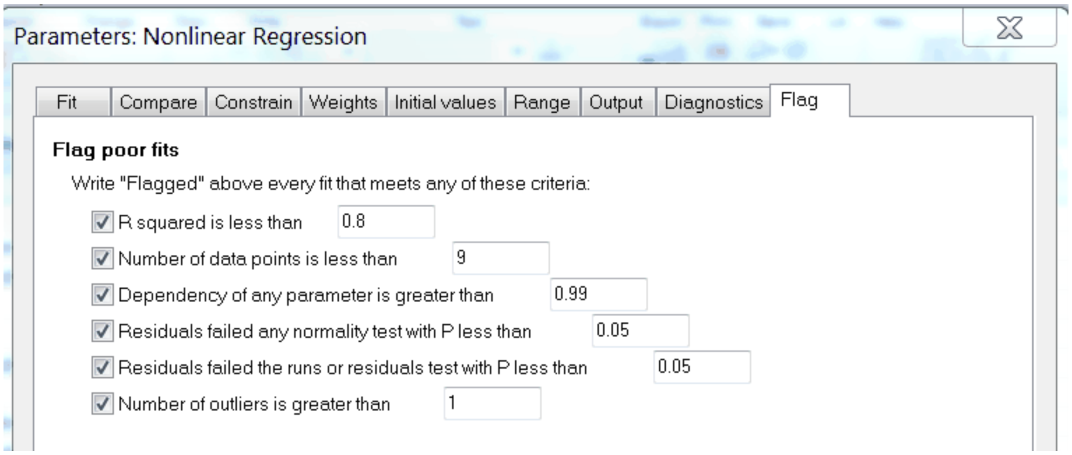
Even though nonlinear regression, as its name implies, is designed to fit nonlinear models, some of the inferences actually assume that some aspects of the model are close to linear, so that the distribution of each parameter is symmetrical. This means that if you analyzed many data sets sampled from the same system, the distribution of the best-fit values of the parameter would be symmetrical and Gaussian.

If the distribution of a parameter is highly skewed, the reported SE and CI for that parameter will not be very useful ways of assessing precision. Hougaard's measure of skewness 399 quantifies how skewed each parameter is.

4.5.2.11 Flags tab

This feature is useful if you analyze many sets of data (perhaps by running a script), and need a way to automatically flag poor fits that need to be inspected more carefully.

In the Flag tab, define the attributes that define poor fits you wish to flag. You can choose to flag a fit with any or all of the following attributes: low R², too few data points, too high a dependency for any parameter, a failed normality test of residuals, a failed runs or residual test, or too many outliers.



Flag poor fits

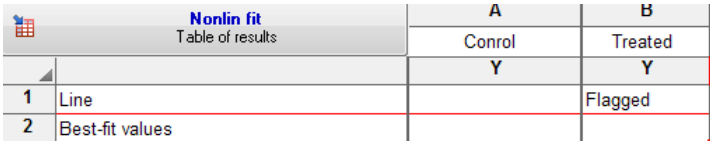
Write "Flagged" above every fit that meets any of these criteria:

- ☑ R squared is less than 0.8
- ☑ Number of data points is less than 9
- ☑ Dependency of any parameter is greater than 0.99
- ☑ Residuals failed any normality test with P less than 0.05
- ☑ Residuals failed the runs or residuals test with P less than 0.05
- ☑ Number of outliers is greater than 1

Some options may be gray if the corresponding option is gray in the Diagnostics tab. You can't flag, for example, fits with high dependency unless you chose the option to compute dependency.

What choices should you make? It depends on the kind of work your are doing. In some systems, you would only flag fits with R2 less than 0.50. In other systems, you expect tight data and so would flag any fit with R² less than 0.90. Once you have fit a model to dozens or hundreds of similar data sets, you'll have a feel for what to expect, and when to flag dubious fits. Until then, don't flag fits.

What happens if a fit is flagged? Not much! The word "Flagged" appears at the top of the fit as shown below. Use this as a cue to take a more careful look at this data set and its fit.



Nonlin fit Table of results	A Conrol Y	B Treated Y
1 Line		Flagged
2 Best-fit values		

When you run a Monte-Carlo simulation, there is a new option to ignore (not tabulate) results from fits that are flagged.

4.5.3 Graphing tips: Nonlinear regression

Prism automatically graphs the best-fit curves from nonlinear regression, and you'll rarely need to think about this step.

- Tips 351 on graphing curves
- Graphing prediction and confidence bands 352
- Adding an equation 357, or a table of best-fit parameters, to the graph.
- Graphing outliers 358.
- Graphing residuals 361.

4.5.3.1 Graphing best-fit curves

Automatic vs. manual plotting

When Prism performs nonlinear regression, it automatically superimposes the curves on the graph.

If you need to create additional graphs, or change which curve is plotted on which graph, keep in mind that from Prism's point of view, a curve generated by nonlinear regression is simply a data set. You can add curves to a graph or remove curves from a graph on the 'Data on graph' tab of the Format Graph dialog.

Plotting curves after comparing models

If you ask Prism to compare two models, only the selected model is graphed. If you want to graph both models, you will need to analyze the data twice, once with each model.

Plotting on logarithmic axes

With Prism 5 and 6, it is Ok to stretch an axis to a log scale, a choice on top of the Format Axis dialog. With prior versions, this didn't work well. The curve was defined as a series of line segments, and these would be far from equally spaced on a log axis. Prism 5 and 6 are smarter, so the curve will look smooth on either a linear or logarithmic axis.

Plotting only a portion of the curve

Let's say that the X values of your data range from 0 to 100, and you fit a curve with nonlinear regression. Now you want to create a graph showing only the data with X values between 0 and 10. The curve will look ragged. The problem is that Prism defines the entire curve using 150 line segments. In this example, you only plot 10% of the range, so the curve will have only 15 line segments.

To fix this problem, go to the Range tab of nonlinear regression and choose to define the curve with more line segments. In this example, increasing the value tenfold, to 1500, would be about right.

4.5.3.2 Graphing confidence and prediction bands

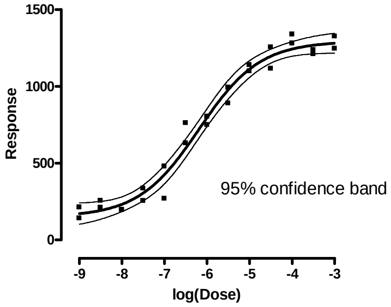
The difference between confidence and prediction bands

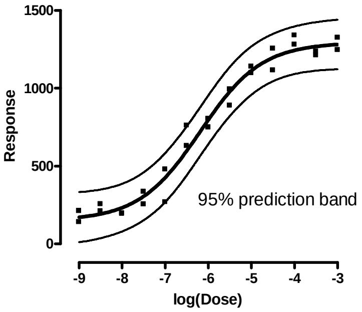
The 95% **confidence bands** enclose the area that you can be 95% sure contains the true curve. It gives you a visual sense of how well your data define the best-fit curve.

The 95% **prediction bands** enclose the area that you expect to enclose 95% of future data points. This includes both the uncertainty in the true position of the curve (enclosed by the confidence bands), and also accounts for scatter of data around the curve.

Choose in the Confidence tab[343] of the Nonlinear regression parameters dialog. Learn how they are calculated[108].

Prediction bands are always wider than confidence bands. When you have lots of data points, the discrepancy is huge.

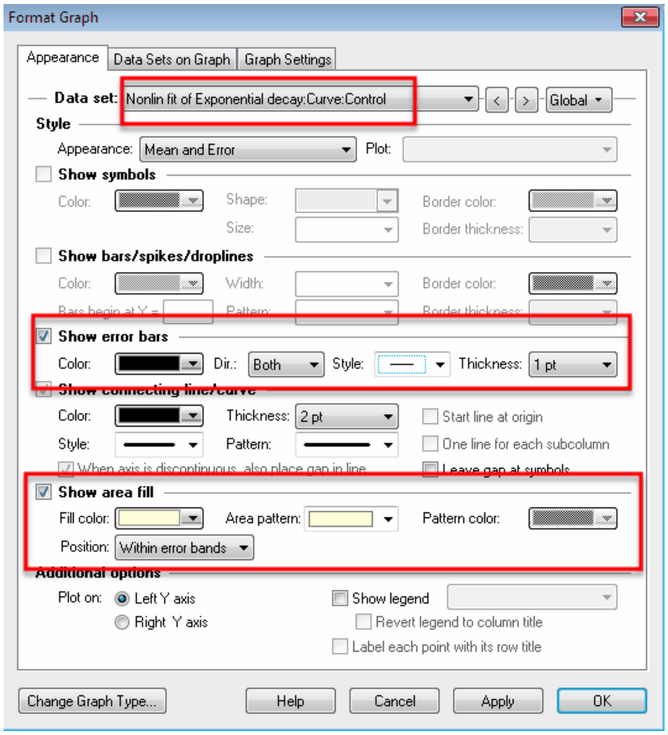




Plotting confidence or prediction bands

After you choose to create a confidence or prediction band in the diagnostics tab, it should just appear on the graph. You can fine tune its appearance on the graph. Note the following:

- The fit curve is seen by the Format Graph dialog as one more data set to plot. be sure to select one of those nonlinear fits first (top of dialog) before adjusting its properties.
- To plot the confidence bands, choose to show error bars, and set the style to ------ or ______.
- You can also choose to show an area fill within the error bands.



Data set: Nonlin fit of Exponential decay:Curve:Control

Style

Appearance: Mean and Error

□ Show symbols

□ Show bars/spikes/droplines

☑ Show error bars

Color: ■ Dir.: Both Style: —— Thickness: 1 pt

☑ Show connecting line/curve

Color: ■ Thickness: 2 pt

Style: —— Pattern: ——

□ Start line at origin

□ One line for each subcolumn

☑ When axis is discontinuous, also place gap in line

□ Leave gap at symbols

☑ Show area fill

Fill color: ▭ Area pattern: ▭ Pattern color: ▨

Position: Within error bands

Additional options

Plot on: ● Left Y axis ○ Right Y axis

□ Show legend

□ Revert legend to column title

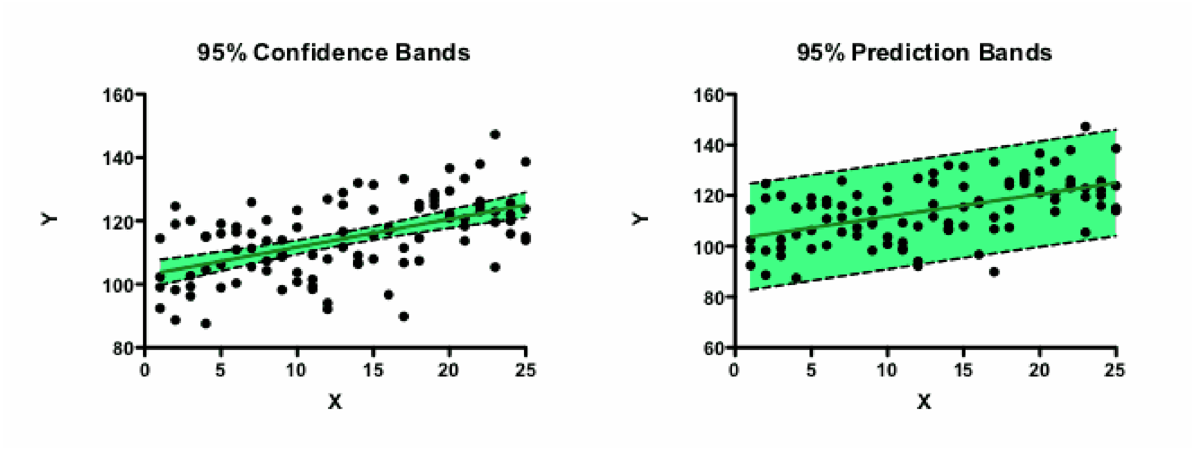
□ Label each point with its row title

Why do so many points lie outside the confidence bands?

Confidence bands show you how well you know the location of the best fit line or curve. Given all the assumptions of the analysis, you can be 95% sure that the true curve (nonlinear regression) or line (linear regression) lies within the bands.

Prediction bands show you where you can expect the data to lie. You expect 95% of all data points to lie within the prediction bands.

With many data points, you expect a large fraction of the data points to lie outside the confidence band, but 95% to lie within the prediction bands. The confidence bands aren't supposed to show you the scatter of the data, but rather the uncertainty in the position of the line or curve. With lots of data, the line or curve is known fairly precisely, so only a small fraction of data lie within the bands. The figure below demonstrates this with linear regression.



Plotting both confidence and prediction bands

Prism lets you choose either a confidence band or a prediction band as part on the Confidence tab 343. But not both. To plot both on one graph, analyze your data twice, choosing a confidence band the first time and a prediction band the second time. The curves are data sets that you can add to any graph by drag-n-drop or using the Change menu from the graph.

One-sided confidence (or prediction) bands

Confidence and prediction bands are usually two-sided. But you can get Prism to show the band on only one side.

One use for a one-sided confidence band would be on a graph that plots purity of a substance over time. You want to know, with a certain confidence, how low the purity could be at a certain time. You don't really care about quantifying how high the purity could be.

To plot a one-sided 95% confidence (or prediction) band:

1. Choose "90%" confidence bands in the Diagnostics tab of the nonlinear regression dialog. With 90% bands, there is a 5% chance that the true value is larger than the upper band, and 5% it is lower than the lower band. When you plot only one of these bands, therefore, it is really a 95% confidence band, which is probably what you want.

2. On the Format Symbols dialog, choose the best-fit line or curve and make sure that error bars are turned on with the "---" style. Then

choose either Up or Down (but not Both) in the Direction drop-down menu.

Viewing the XY coordinates of confidence or prediction bands

1. Go to the Diagnostics tab and check the option to plot confidence or prediction bands.

2. Go to the Range tab of the nonlinear regression dialog.

3. Check the option to create a table of XY coordinates. This table will include the confidence or prediction bands.

4. When viewing the results, expand the results sheet in the navigator and go to the page labeled 'Curve'. For each X value, you'll see the y value and the distance the confidence or prediction curve is above and below that value.

Situations where Prism can't plot confidence or prediction bands

Prism will not plot confidence or prediction bands in several situations:

- If the best-fit value of a parameter hit a constraint 441, the fit is unlikely to be useful. Prism does not plot confidence or prediction bands, because they would almost certainly be misleading.

- If the results of nonlinear regression are ambiguous 437, the confidence or prediction bands would be super wide, maybe infinitely wide. They would not be useful, so Prism does not plot them.

- If you choose robust 94 nonlinear regression, Prism does not compute confidence or prediction bands, as it cannot compute standard errors or confidence intervals of the parameters.

- The fit is perfect 443. If the sum-of-squares is 0.0 and R2 is 1.0, it is not possible to compute or interpret confidence or prediction bands.

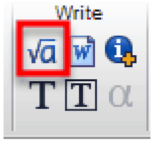
- If the fit is interrupted 435, confidence and prediction bands are not computed.

- If you chose to weight the fit by $1/SD^2$. To plot confidence and prediction bands, Prism needs to know the weighting factor at every X value in the range it plots the curve. Not a problem with most weighting schemes. But if you weight by the values you enter in the SD column, then Prism only knows those weights, and not the weights for X values between the points, so can't plot confidence or prediction bands.

4.5.3.3 Adding the equation to the graph

Prism does not automatically write equations on the graph. There are two approaches you can use to do so.

Microsoft Office comes with an equation editor. If Microsoft Equation Editor is installed on your system, you will see a quick-access button on your toolbar when you are on either a graph or a layout sheet.

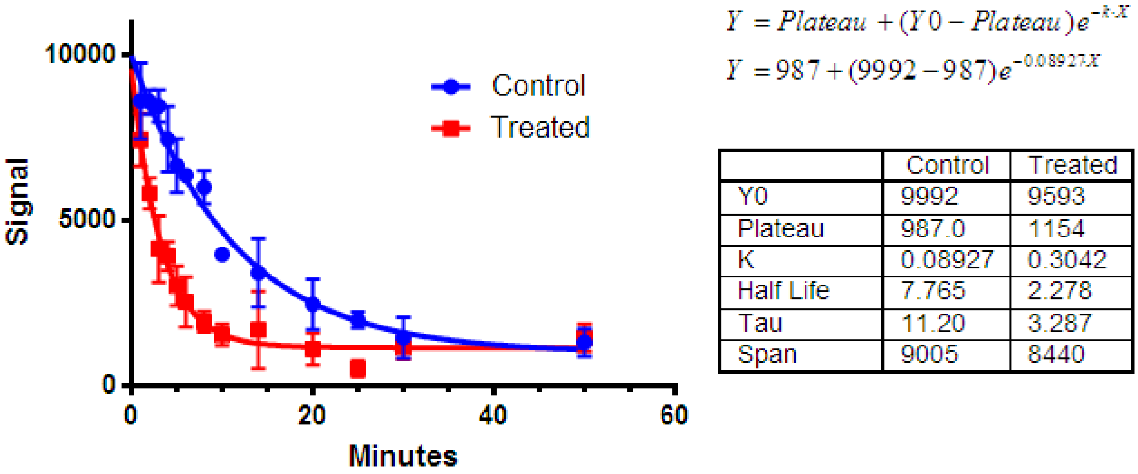


You can create either a generic equation, or an equation with the best-fit values of the parameters substituted for the variable names.

If the button doesn't appear on your toolbar, create the equation inside of Microsoft Word or some other program, and then copy and paste.

Another approach is to simply show the parameter values from the curve of best fit in tabular form. Copy selected results and then paste onto a graph. The table will be linked, so its values will change if you edit or replace the data. This doesn't show the form of the equation (exponential decay, in this example) but gives you the results of this particular experiment, which may be more important.

The example below shows use of the equation editor to write both a generic and specific equation (for the control data set) and an embedded table showing the results for both data sets.



$$Y = Plateau + (Y0 - Plateau)e^{-k \cdot X}$$

$$Y = 987 + (9992 - 987)e^{-0.08927 X}$$

	Control	Treated
Y0	9992	9593
Plateau	987.0	1154
K	0.08927	0.3042
Half Life	7.765	2.278
Tau	11.20	3.287
Span	9005	8440

4.5.3.4 Graphing outliers

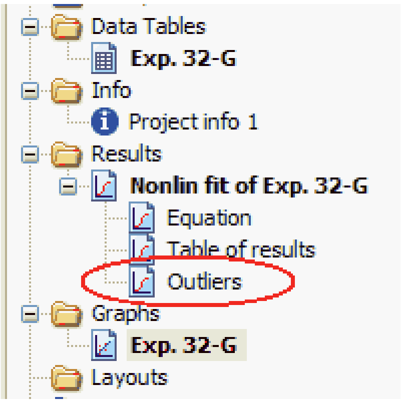
How to identify or eliminate outliers

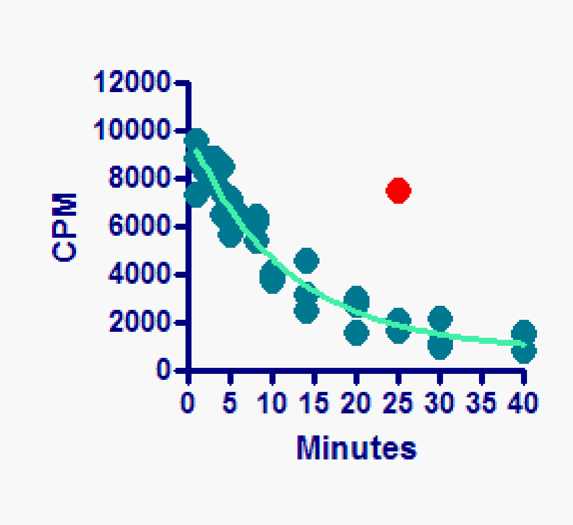
Prism can identify outliers from the curve fit, and you can choose whether these should simply be plotted or also ignored by the curve fitting process. Choose in the Methods tab 329.

You can adjust the value of Q to redefine how aggressively Prism defines outliers.

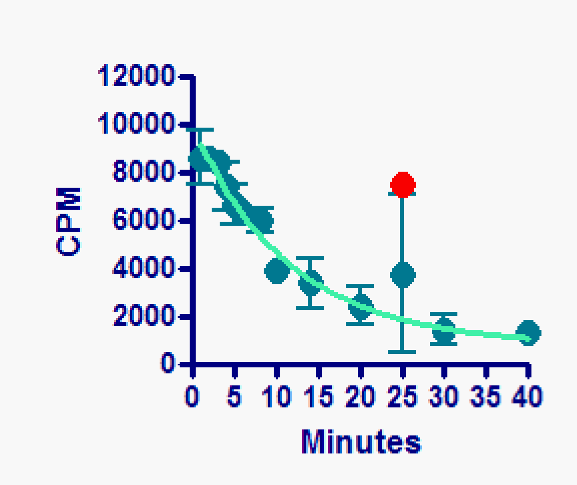
How Prism plots outliers

If you choose to either count or eliminate outliers, Prism tabulates any outliers in a page of the results sheet called 'outliers' and plots them in red (unless data are plotted in red, in which case the outliers are blue).



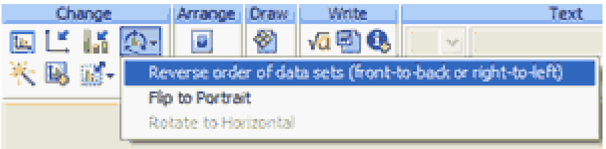


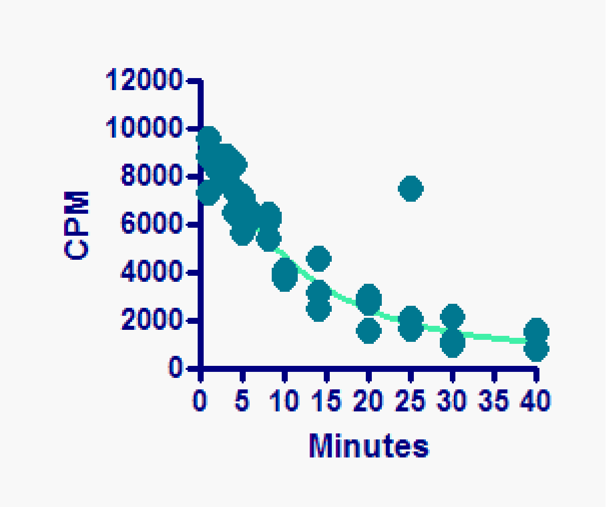
The example above shows a graph where each replicate is plotted individually. The example below shows a graph which plots mean and SD error bars. The error bars are computed from **all the data**, including any outliers. The outliers are then superimposed on the graph as well.



Note that Prism does not remove the outlier from the regular data set. When you plot mean and error bar, the outlier is included in the calculation. When you plot individual data points, the outlier is still plotted with the full dataset. Then Prism superimposes the outlier(s) as a separate dataset plotted in front.

Look back at the first example, showing each replicate. You can see that the curve goes over the data points. If you want the curve to go under the data points, click the Reverse/Flip/Rotate button.





The results, above, are not exactly what you want. Indeed the relative front-to-back order of data and curve has reversed, so the curve is now behind the points. But the relative front-to-back order of the full dataset and the outlier dataset is also reversed, so the outlier is now behind the regular data point, so is invisible. This demonstrates that Prism does not really change the color of the data points of outliers. Rather, it plots the outliers twice -- once as part of the full data set and again as part of the outlier dataset superimposed on the graph. To get the effect you want here, go to the Data on Graph tab of the Format Graph dialog, and fine-tune the back-to-front order of data sets (with the outlier(s) and curve considered to be data sets).

4.5.3.5 Residual plot

What are residuals?

Unweighted fits

A residual is the distance of a point from the curve. Least-squares regression works to minimize the sum of the squares of these residuals. A residual is positive when the point is above the curve, and is negative when the point is below the curve. Create a residual plot to see how well your data follow the model you selected. Mild deviations of data from a model are often easier to spot on a residual plot than on the plot of data with curve.

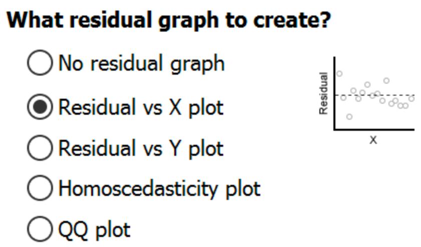
Weighted fits

If you choose to weight your data unequally, Prism adjusts the definition of the residuals accordingly.

The residual that Prism tabulates and plots equals the residual defined in the prior paragraph, divided by the weighting factor. The most common common alternative weighting is "Weight by $1/Y^2$ (minimize relative distances squared)". In this case, the residual is defined to be the distance of the point from the curve divided by the Y value of the curve. Weighted nonlinear regression minimizes the sum of the square of these residuals. Note the ambiguity in defining weighting. The Prism dialog gives the choice to weight by $1/Y^2$. This means that the squared residual is divided by Y^2. The weighted residual is defined as the residual divided by Y. Weighted nonlinear regression minimizes the sum of the squares of these weighted residuals.

Earlier versions of Prism (up to Prism 4) always plotted basic unweighted residuals, even if you chose to weight the points unequally.

Which kind of residual graph?



What residual graph to create?

○ No residual graph

● Residual vs X plot

○ Residual vs Y plot

○ Homoscedasticity plot

○ QQ plot

Starting with Prism 8, choose among four kinds of residual plots:

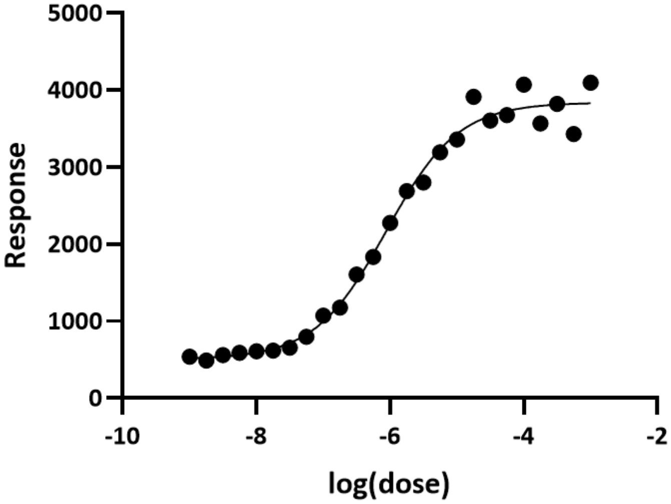
	X axis	**Y axis**
Residual vs. X plot	X values of data	Residual or weighted residual
Residual vs. Y plot	Predicted Y value	Residual or weighted residual
Homoscedasticity plot	Predicted Y value	Absolute value of residual or weighted residual
QQ plot	Actual residual	Predicted residual if residuals are sampled from a Gaussian distribution

Prism only lets you create one residual graph from nonlinear regression. If you want to make two or more kinds of residual graphs, you'll need to duplicate the results page, and then change the residual option on the new copy.

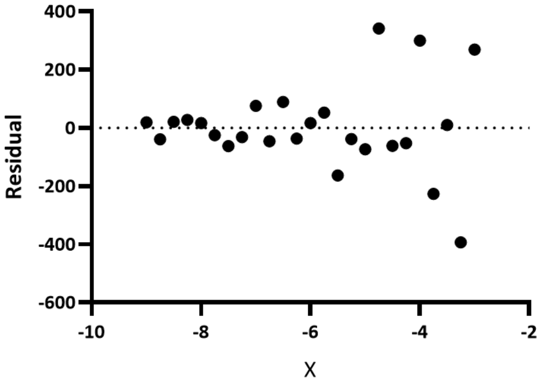
Examples

Example fit

Here is a log(dose) vs. response curve with simulated data. The random scatter was chosen so the points iwth larger Y values have larger average scatter. The fit was done the usual way without weighting. If you look carefully, you can see that the points at the top of the curve have more scatter, but it isn't so obvious.

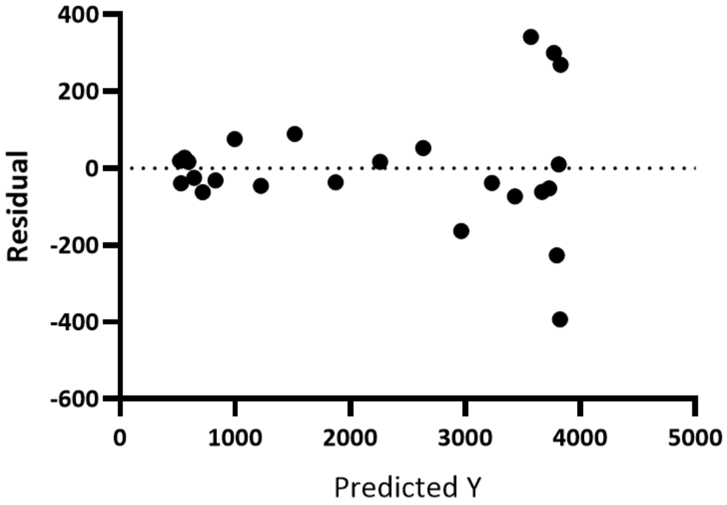


Residual vs. X



This is the most common residual graph. Prism 7 and earlier only created this kind of residual plot. You can see that the points with large X values have larger residuals (positive and negative).

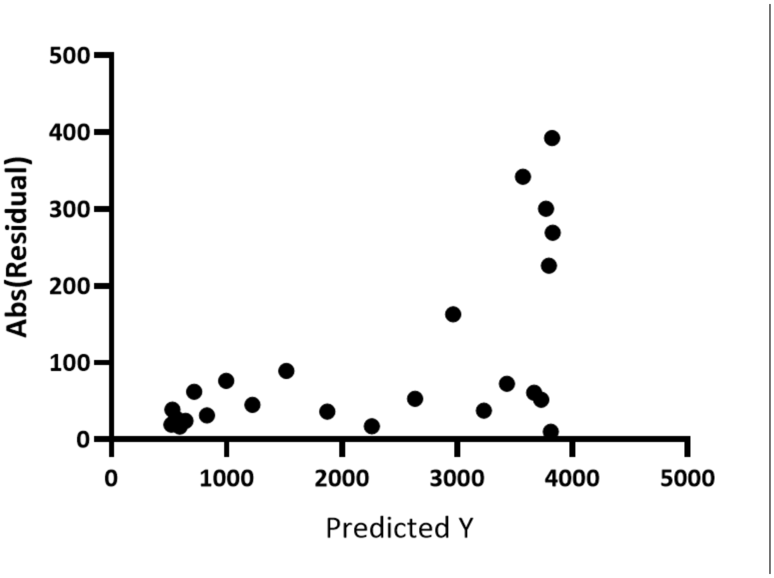
Residual vs. predicted Y



For each point, Prism calculates the Y value of the curve at that X value, and plots that Y value on the X axis of the residual plot. The Y axis of the residual plot graphs the residuals or weighted residuals. You can see that the points with larger Y values have larger residuals, positive and negative.

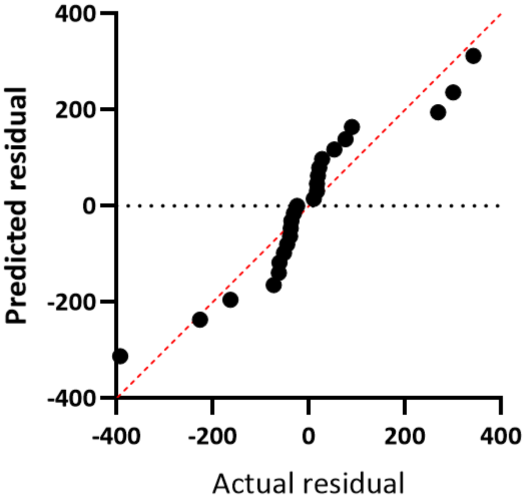
In this example the Y values get larger as X values get larger. So this graph doesn't look very different than the residual vs. X plot. But if the curve is biphasic, the two graphs will look more different.

Homoscedasticity plot



This graph is made just like the graph of predicted Y vs. residuals, except here the absolute values of the residuals are shown. Now it is really clear that the residuals get larger as Y gets larger.

QQ plot

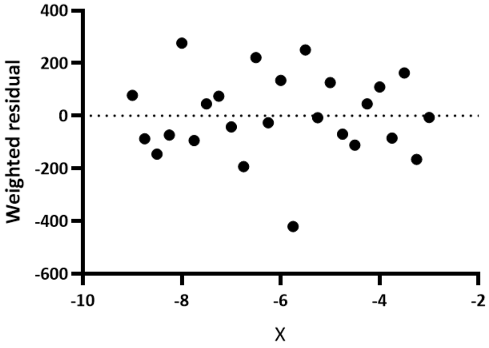


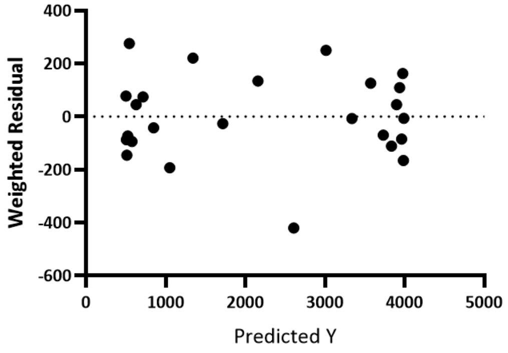
The X axis plots the actual residual or weighted residuals. The Y axis plots the predicted residual (or weighted residual) assuming sampling from a Gaussian distribution. An assumption of regression is that the residuals are sampled from a Gaussian distribution, and this plot lets you assess that assumption. If the assumption is true, the points should all be very close to the line of identity, shown in red on the graph. QQ plots sometimes plots one or both axes as percentiles are quantiles (same as percentiles but as fractions rather than percentages). Prism always plots both axes in the same units as the Y values of the data.

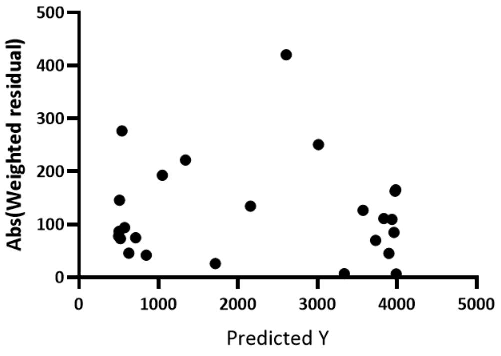
In this example, the data don't follow the line of identity very well. The data are sampled from Gaussian distributions, but the SD of that distribution was larger when the Y values were larger. The residuals were not sampled from a single Gaussian distribution, and this accounts for the systematic discrepancy between the points and the line of identity.

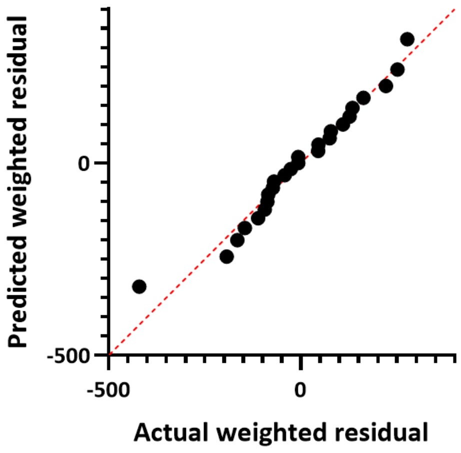
Same example fit with weighted regression

The data were then fit to the same model but with relative weighting. Here are the four residual graphs. All show that the data now comply with the assumption of the fit.









4.5.4 Interpreting nonlinear regression results

Interpreting nonlinear regression results can be confusing.

- If Prism reported error message on the top of the results column, here are the explanations 430.
- Help with graphing 351 best-fit curves, with their confidence or prediction bands and residual plots.
- Read detailed explanations of the results, and review analysis checklists. Where to go depends on your goal.

Goal: Determine best-fit values	Analysis checklist 422	Learn more. 368
Goal: Compare models	Analysis checklist 426	Learn more. 409
Goal: Interpolate unknowns from a standard curve	Analysis checklist 428	Learn more. 258

4.5.4.1 Interpreting results: Nonlinear regression

When evaluating nonlinear regression results, first consider what your goal is. If your goal is to interpolate unknowns from a standard curve, skip this page and go right to this analysis checklist 274.

If your goal is to determine the values of the best-fit parameters or to compare models, answer the three questions below before looking at parameter values, R2, etc.

- Are your results free of error messages? Prism sometimes puts a short message at the top of the table of results. Read about the meaning of "Bad initial values" 435, "Interrupted" 435, "Not converged" 437, "Ambiguous" 437, "Hit constraint" 441, "Don't fit" 442, "Too few points" 442, "Perfect fit" 443, "Impossible weights" 443, and "Equation not defined" 443.

- Does the curve go near your data points? In rare cases, the curve may be far from the data points. This may happen, for example, if you picked the wrong equation. Look at the graph to make sure this didn't happen.

- Are the best-fit values of the parameters scientifically sensible? Nonlinear regression programs have no common sense and don't know the context of your experiment. The curve fitting procedure can sometimes yield results that make no scientific sense. For example with noisy or incomplete data, nonlinear regression can report a best-fit rate constant that is negative, a best-fit fraction that is greater than 1.0, or a best-fit Kd value that is negative. All these results are scientifically meaningless. Also check whether the best-fit values of the variables make sense in light of the range of the data. The results make no sense if the top plateau of a sigmoid curve is far larger than the highest data point, or an EC50 is not within the range of your X values.If the results make no scientific sense, they are unacceptable, even if the curve comes close to the points and R2 is close to 1.0.

If the answer to all three questions is yes, then it makes sense to delve into the numerical results of nonlinear regression in detail.

- Standard errors and confidence intervals of parameters 372

- Normality tests of residuals 375

- Goodness of fit of nonlinear regression 377
- Runs test 389
- Replicates test 390
- Dependency and covariance matrix 393
- Hougaard's measure of skewness 399
- Could the fit be a local minimum? 403
- Outliers 404

4.5.4.1.1 Best-fit parameter values

Before trying to interpret the rest of the results, first look at the best-fit value of each parameter in your model. First make sure you know what units each parameter is expressed in (if there are any units; some parameters are unitless constants). Next, make sure each value makes biological or scientific sense.

What if a parameter value makes no biological or chemical sense?

If a value doesn't make any sense, ask your self these questions:

- Does your data actually define all the parameters?
- Should you be constraining a parameter (or several) to constant values? If not, should you be constraining them to have a range of values (only positive numbers, perhaps)?
- Should you be fitting a family of datasets together using global nonlinear regression?

Why does Prism say "unstable" rather than reporting a best-fit value?

On the confidence tab 343 of the nonlinear regression dialog, you choose how Prism should deal with difficult fits. If you choose the recommended method, difficult parameters (and their confidence interval) will be marked as "unstable".

Why is there a ~ symbol in front of some values?

If you choose (on the Confidence tab) to use the older method of identifying some fits as "ambiguous", then if a value is preceded by ~, it means the results are 'ambiguous' 437. Changing the value of any parameter will always move the curve further from the data and increase the sum-of-squares. But when the fit is 'ambiguous', changing other parameters can move the curve so it is near the data again. In other words, many combinations of parameter values lead to curves that fit equally well.

The parameter values preceded by ~ define the curve that Prism creates, but aren't actually useful in other ways. Other values would generate the same curve, or one that fits just as well.

When the best fit value is preceded by ~, so is the standard error. The corresponding confidence intervals are shown as "very wide" with no numerical range (the range would be infinitely wide).

4.5.4.1.2 Standard error of parameters

Interpreting the standard errors of parameters

The only real purpose of the standard errors is as an intermediate value used to compute the confidence intervals. If you want to compare Prism's results to those of other programs, you will want to include standard errors in the output. Otherwise, we suggest that you ask Prism to report the confidence intervals only (choose on the Diagnostics tab 347). The calculation of the standard errors 108 depends on the sum-of-squares, the spacing of X values, the choice of equation, and the number of replicates.

'Standard error' or 'standard deviation' ?

Prism reports the standard error of each parameter, but some other programs report the same values as 'standard deviations'. Both terms mean the same thing in this context.

When you look at a group of numbers, the standard deviation (SD) and standard error of the mean (SEM) are very different. The SD tells you about the scatter of the data. The SEM tells you about how well you have determined the mean. The SEM can be thought of as "the standard deviation of the mean" -- if you were to repeat the experiment many times, the SEM (of your first experiment) is your best guess for the standard deviation of all the measured means that would result.

When applied to a calculated value, the terms "standard error" and "standard deviation" really mean the same thing. The standard error of a parameter is the expected value of the standard deviation of that parameter if you repeated the experiment many times. Prism (and most programs) calls that value a standard error, but some others call it a standard deviation.

Why is there a ~ symbol in front of some SE values?

If a value is preceded by ~, it means the results are 'ambiguous' 437. Changing the value of any parameter will always move the curve further from the data and increase the sum-of-squares. But when the fit is 'ambiguous', changing other parameters can move the curve so it is near the data again. In other words, many combinations of parameter values lead to curves that fit equally well. The standard error of those parameters is not well defined.

When the SE value is preceded by ~, the corresponding confidence intervals are shown as "very wide" with no numerical range (the range would be infinitely wide).

Why doesn't Prism display SE values for my fit?

Starting with Prism 8.2, Prism will only compute and display the SE of parameters if you ask Prism (in the Confidence tab 343) to report symmetrical confidence intervals. These intervals are computed from the SE values, so it makes sense to display them. We recommend reporting asymmetrical profile likelihood intervals, which are more useful. These are not computed from the SE values, so it makes no sense to report both the SE and the asymmetrical confidence interval. Starting with version 8.2, Prism will only show the SE if you request symmetrical confidence intervals.

4.5.4.1.3 Confidence intervals of parameters

Do not ignore the confidence intervals

In most cases, the entire point of nonlinear regression is to determine the best-fit values of the parameters in the model. The confidence interval tells you how tightly you have determined these values. If a confidence interval is very wide, your data don't define that parameter very well. Confidence intervals are computed 108 from the standard errors of the parameters.

How accurate are the standard errors and confidence intervals?

Prism gives you two choices in the Confidence tab 343.

The asymptotic or approximate standard errors reported by Prism (and virtually all other nonlinear regression programs) are based on some mathematical simplifications. They are calculated assuming that the equation is linear, but are applied to nonlinear equations. This simplification means that the intervals can be too optimistic.

Prism also lets you choose to report asymmetrical confidence intervals (profile likelihood intervals) and these are more accurate but take longer to calculate.

Sometimes Prism reports "very wide" instead of reporting the confidence interval

If you see the phrase 'very wide' instead of a confidence interval, you may also see the phrase 'ambiguous' 437 at the top of the results tables. This means that the data do not unambiguously define the parameters. Many sets of parameters generate curves that fit the data equally well. The curve may fit well, making it useful artistically or to interpolate unknowns, but you can't rely on the best-fit parameter values. Alternatively, depending on the options selected on the Confidence tab 343 , you may also see specific parameters identified as being "unstable". Learn more about unstable parameters here 431.

When does Prism report "infinity" as a confidence limit?

Prism reports that the upper confidence limit is infinity when the model and data simply don't define the upper confidence limit. If you are aiming for 95% confidence limits, Prism will report that the limit is infinite when the confidence level is less than 95% no matter how large the algorithm sets that confidence limit.

Similarly, Prism reports -infinity when the model and data simply don't define the lower confidence limit.

When you see that a confidence limit is infinity (or -infinity), you can conclude that your data simply don't define that parameter in the model very well. One example would be if you are fitting an asymmetrical ("five parameter") log(dose)-response curve, and the confidence limit for the asymmetry parameter is infinity. This can happen when your data form a symmetrical log(dose)-response curve, so there is no information in the data to define the asymmetry parameter.

When does Prism report "???" as a confidence limit?

Prism reports ??? for a confidence limit when its calculations got interrupted, so it was unable to compute the limit. Changes in Prism 8 make this happen less often than it did with Prism 7.

Why would the calculations get interrupted? There are several reasons.

- It is conceivable that the problem is due to the data being super huge or really tiny values. If all your data look something like 1.23e-45 or 1.23e45 try transforming to different units so the values are not so incredibly large or small. It is possible that this might be helpful and let Prism fine the confidence limits.

- Prism reports ??? when the model equation is fundamentally difficult to fit, and when the data simply don't define the model very well. This could be because of too few values, no values at important ranges of X values, too much scatter, etc. Even though Prism can't explain why it reports ???, you can conclude that there is no way to compute a reasonable short useful confidence interval given those data and that model.

Confidence intervals of transformed parameters

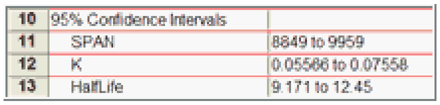
In addition to reporting the confidence intervals of each parameter in the model, Prism can also report confidence intervals for transforms 715 of those parameters. For example, when you fit an exponential model to determine the rate constant, Prism also fits the time constant tau, which is the reciprocal of the rate constant.

When you write your own equation, or clone an existing one, choose 715 between two ways to compute the confidence interval of each transformed parameter. If you pick a built-in equation, Prism always reports asymmetrical confidence intervals of transformed parameters.

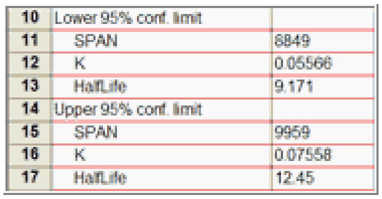
Do not mix up confidence intervals and confidence bands

It is easy to mix up confidence intervals and confidence bands. Choose both on the Confidence tab 343.

The **95% confidence interval** tells you how precisely Prism has found the best-fit value of a particular parameter. It is a range of values, centered on the best-fit value. Prism can display this range in two formats:

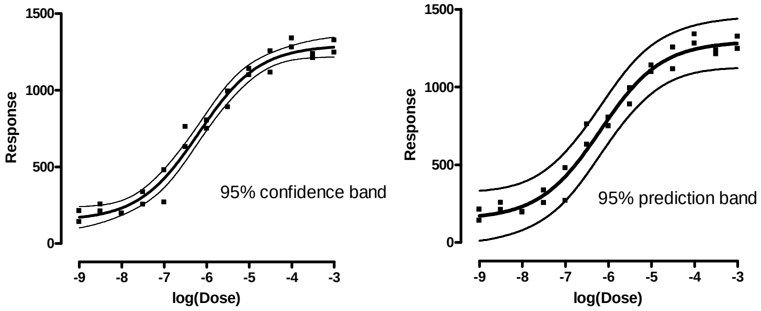


10	95% Confidence Intervals	
11	SPAN	8849 to 9959
12	K	0.05566 to 0.07558
13	HalfLife	9.171 to 12.45



10	Lower 95% conf. limit	
11	SPAN	8849
12	K	0.05566
13	HalfLife	9.171
14	Upper 95% conf. limit	
15	SPAN	9959
16	K	0.07558
17	HalfLife	12.45

The 95% **confidence bands** enclose the area that you can be 95% sure contains the true curve. It gives you a visual sense of how well your data define the best-fit curve. It is closely related to the 95% **prediction bands** , which enclose the area that you expect to enclose 95% of future data points. This includes both the uncertainty in the true position of the curve (enclosed by the confidence bands), and also accounts for scatter of data around the curve. Therefore, prediction bands are always wider than confidence bands.



4.5.4.1.4 Normality tests of residuals

Interpreting a normality test

The result of a normality test is expressed as a P value that answers this question:

> If your model is correct and all scatter around the model follows a Gaussian population, what is the probability of obtaining data whose residuals deviate from a Gaussian distribution as much (or more so) as your data does?

If the P value is large, then the residuals pass the normality test. If the P value is small, the residuals fail the normality test and you have evidence that your data don't follow one of the assumptions of the regression. Things to consider:

- Fit a different model
- Weight 329 the data differently.
- Exclude outliers 94.

A large P value means that your data are consistent with the assumptions of regression (but certainly does not prove that the model is correct). With small numbers of data points, normality tests have little power to detect modest deviations from a Gaussian distribution.

How the normality tests work

We recommend relying on the **D'Agostino-Pearson** normality test. It first computes the skewness and kurtosis to quantify how far from Gaussian the distribution is in terms of asymmetry and shape. It then calculates how far each of these values differs from the value expected with a Gaussian distribution, and computes a single P value from the sum of these discrepancies. It is a versatile and powerful (compared to some others) normality test, and is recommended. Note that D'Agostino developed several normality tests. The one used by Prism is the "omnibus K2" test.

An alternative is the **Shapiro-Wilk** normality test. We prefer the D'Agostino-Pearson test for two reasons. One reason is that, while the Shapiro-Wilk test works very well if every residual is unique, it does not work well when several residuals are identical. The other reason is that the basis of the test is hard for non mathematicians to understand.

Earlier versions of Prism offered only the **Kolmogorov-Smirnov** test. We still offer this test (for consistency) but no longer recommend it. This test compares the cumulative distribution of the data with the expected cumulative Gaussian distribution, and bases its P value simply on the largest discrepancy. This is not a very sensitive way to assess normality, and we now agree with this statement[1]: "The Kolmogorov-Smirnov test is only a historical curiosity. It should never be used."

The Kolmogorov-Smirnov method as originally published assumes that you know the mean and SD of the overall population (perhaps from prior work). When analyzing data, you rarely know the overall population mean and SD. You only know the mean and SD of your sample. To compute the P value, therefore, Prism uses the Dallal and Wilkinson approximation to Lilliefors' method (Am. Statistician, 40:294-296, 1986). Since that method is only accurate with small P values, Prism simply reports “P>0.10” for large P values.

Prism 8 adds the **Anderson Darling test**. While the Kolmogorov-Smirnov test only looks at the largest discrepancy between the actual distribution and the Gaussian distribution, the Anderson-Darling test sums all the discrepancies. Prism uses the form of the Anderson-Darling test that corrects for the fact that it uses the sample mean and sample SD, and doesn't know the population mean and SD to compare the data to.

Reference

[1] RB D'Agostino, "Tests for Normal Distribution" in *Goodness-Of-Fit Techniques* edited by RB D'Agostino and MA Stepenes, Macel Decker, 1986.

4.5.4.1.5 R squared

Q&A about R^2

What does R^2 quantify

- The value R^2 quantifies goodness of fit. It compares the fit of your model to the fit of a horizontal line through the mean of all Y values.

- You can think of R^2 as the fraction of the total variance of Y that is explained by the model (equation). With experimental data (and a sensible model) you will always obtain results between 0.0 and 1.0.

- Another way to think about R^2 is the square of the correlation coefficient between the actual and predicted Y values.

What is the range of values R^2 can have?

The simple answer is that R^2 is usually a fraction between 0.0 and 1.0, and has no units. But there are special cases:

- R^2 equals 1.00 when the curve goes through every point. But if you have replicate Y values at the same X value, it is impossible for the curve to go through every point, so R^2 has to be less than 1.00.

- When R^2 equals 0.0, the best-fit curve fits the data no better than a horizontal line going through the mean of all Y values. In this case, knowing X does not help you predict Y.

- When you choose a really inappropriate model or impose silly constraints (usually by mistake) the best-fit curve will fit worse than an horizontal line. In this case R^2 will be negative. Yes that seems odd, but R^2 is not really the square of anything and it is possible. Details at the bottom of this page.

- You may see references to R^2 possibly having a value greater than 1.0. This can only happen when an invalid equation is used so the result is simply wrong.

r^2 or R^2?

By tradition, statisticians use uppercase (R^2) for the results of nonlinear and multiple regression and lowercase (r^2) for the results of linear regression, but this is a distinction without a difference.

Why do some suggest that R^2 not be reported with nonlinear regression?

Minitab does not report R^2 with nonlinear regression because they think it is too misleading. Kvalseth(1) also cautions about its use. The problems include:

- In linear regression, R^2 compares the fits of the best fit regression line with a horizontal line (forcing the slope to be 0.0). The horizontal line is the simplest case of a regression line, so this makes sense. With many models used in nonlinear regression, the horizontal line can't be generated at all from the model. So comparing the fits of the chosen model with the fit of a horizontal line doesn't quite make mathematical sense. For this reason, SAS callls the value "Pseudo R^2".

- It is tempting to use R^2 to compare fits of alternative models. You shouldn't do this. The adjusted R2 382 is better for that purpose, but not ideal. Prism offers two better ways 78 to compare fits of alternative models. Model selection has to assess the tradeoff -- more complicated models usually fit better but they have more parameters. Both methods Prism offers assess this tradeoff. R^2 does not. R^2 almost always gets larger with a more complicated model, even if the model is less likely to be correct. Don't base model selection on R^2.

- Two models with the same number of parameters can fit the data quite differently with the AICc method telling you that one of those models is much more likely to be true. But the R^2 values may only vary in the third or fourth digit after the decimal point.

- A high R^2 tells you that the curve came very close to the points. That doesn't mean the fit is "good" in other ways. The best-fit values of the parameters may have values that make no sense (for example, negative rate constants) or the confidence intervals may be very wide.

The fit may be ambiguous 437. You need to look at all the results to evaluate a fit, not just the R^2.

Why does Prism report the R^2 of nonlinear regression?

Prism computes and reports R^2 by default. You can turn off reporting of R^2 in the Diagnostics tab of the nonlinear regression dialog and set that preference as a default for future fits.

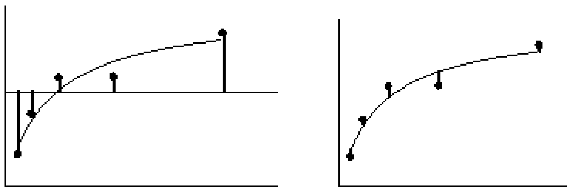
Many people find R^2 useful in this way: When you run a series of experiments, you want to make sure that today's experiment is consistent with other runs of the experiment. For example, if you always get R^2 between 0.90 and 0.95 but today you got R^2=0.75, then you should be suspicious and look carefully to see if something went wrong with the methods or reagents used in that particular experiment. And if a new employee brings you results showing R^2 of 0.99 using that same system, you should look carefully at how many"outliers" were removed, and whether some data were made up.

I think that is really the only way in which R^2 is useful, but it can be quite useful for this purpose.

How R^2 is calculated -- unweighted fits

R^2 is computed from the sum of the squares of the distances of the points from the best-fit curve determined by nonlinear regression. This sum-of-squares value is called SSreg, which is in the units of the Y-axis squared. To turn R^2 into a fraction, the results are normalized to the sum of the square of the distances of the points from a horizontal line through the mean of all Y values. This value is called SStot. If the curve fits the data well, SSres will be much smaller than SStot.

The figure below illustrates the calculation of R^2. Both panels show the same data and best-fit curve. The left panel also shows a horizontal line at the mean of all Y values, and vertical lines showing how far each point is from the mean of all Y values. The sum of the square of these distances (SStot) equals 62735. The right panel shows the vertical distance of each point from the best-fit curve. The sum of squares of these distances (SSres) equals 4165.



R^2 is calculated using this equation:

R^2 = 1.0 - (SSresiduals/SStotal) =1.0-4165/62735= 0.9336

This is method 1 from Kvalseth(1).

If you share parameters (perform global nonlinear regression) SSResiduals in the equation above is the Sum of Squares reported by Prism in the Global result column for the model with shared parameters being fitted, and SSTotal is the Sum of Squares of each Y value (from each data set) around the mean of ALL Y values (from all data sets).

How R^2 is calculated -- weighted fits

There does not seem to be an established method for computing the R^2 of a weighted nonlinear fit. Prism (since 6.00) calculates as detailed below. But note that earlier versions of Prism computed the R^2 of weighted fits differently.

1. Fit the model using relative ($1/Y^2$) or Poisson (1/Y) or general ($1/X^K$) weighting to compute the weighted sum-of-squares (wSSmodel). This is the weighted sum-of-squares of the residuals from your model.

2. Remember the weight each point received.

3. Fit the data to a horizontal line model (Y=Mean + 0*X) using the same weights for each point as used in step 1. This is the weighted sum-of-squares of the residuals from a horizontal line (wSShorizontal).

The weighted R^2 is:

1.0 - (wSSmodel/wSShorizontal)

Note that weights are only computed when fitting the model. Prism does not compute weights for the fit of the horizontal line but rather uses exactly the same weights as were used to fit the model. This ensures that the sum of the weights is identical in both fits.

For more detail, Prism uses equation 4 in Willett and Singer (2).

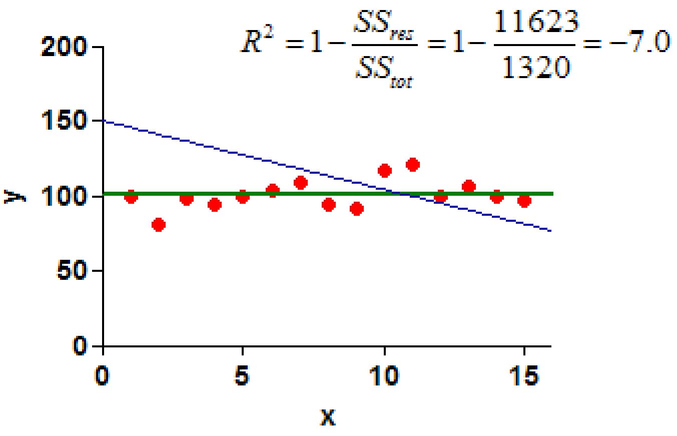
How can R^2 be negative?

How can something squared be negative? Well, R^2 is not really the square of anything. It is computed by subtracting two values. If SSres is larger than SStot, R^2 will be negative (see equation above).

How can this happen? SSres is the sum of the squares of the vertical distances of the points from the best-fit curve (or line). SStot is the sum of the squares of the vertical distances of the points from a horizontal line drawn at the mean Y value. SSres will exceed SStot when the best-fit line or curve fits the data even worse than does a horizontal line.

R^2 will be negative when the best-fit line or curve does an awful job of fitting the data. This can only happen when you fit a poorly chosen model (perhaps by mistake), or you apply constraints to the model that don't make any sense (perhaps you entered a positive number when you intended to enter a negative number). For example, if you constrain the Hill slope of a dose-response curve to be greater than 1.0, but the curve actually goes downhill (so the Hill slope is negative), you might end up with a negative R^2 value and nonsense values for the parameters.

Below is a simple example. The blue line is the fit of a straight line constrained to intercept the Y axis at Y=150 when X=0. SSres is the sum of the squares of the distances of the red points from this blue line. SStot is the sum of the squares of the distances of the red points from the green horizontal line. Since Sres is much larger than SStot, the R^2 (for the fit of the blue line) is negative.



$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = 1 - \frac{11623}{1320} = -7.0$$

Prism file

If R^2 is negative, check that you picked an appropriate model, and set any constraints correctly.

1. Kvalseth, T.O. (1985) Cautionary Note about R^2. American Statistician, 39, 279-285.
2. Willett, J.B., and Singer, J.D. (1988). Another Cautionary Note about R^2 : Its Use in Weighted Least-Squares Regression Analysis. The American Statistician 42: 236.

4.5.4.1.6 Adjusted R squared

The R^2 quantifies how well a model fits the data. When you compare models, the one with more parameters can bend and twist more to come nearer the points, and so almost always has a higher R2. This is a bit misleading.

The adjusted R^2 accounts for the number of parameters fit by the regression, and so can be compared between models with different numbers of parameters. The equations for the regular and adjusted R^2 are compared here (SSresiduals is the sum-of-squares of the discrepancy between the Y value of the curve and the data; SStotal is the sum-of-squares of the differences between the overall Y mean and each Y value; n is the number of data points, and K is the number of parameters fit):

$$R^2 = 1 - \frac{SS_{residuals}}{SS_{total}}$$

$$\text{Adjusted R}^2 = 1 - \frac{SS_{residuals} \big/ (n-K)}{SS_{total} \big/ (n-1)}$$

The adjusted R^2 is smaller than the ordinary R^2 whenever the number of parameters (K) is greater than 1.

Using the adjusted R2 to compare the fits of alternative models is better than using R2 (which should never be used) but not as good as using the extra sum-of-squares F test or the AICc methods built into Prism.

4.5.4.1.7 Sum-of-squares

Nonlinear regression finds the curve that minimizes the sum of square of the distances of the points from the curve. So Prism reports that sum-of-square value. This is useful if you want to compare Prism with another program, or compare two fits manually. Otherwise, the value is not very helpful.

If you choose to differentially weight your data, Prism reports both the absolute and the weighted sum-of-squares.

4.5.4.1.8 Standard deviation of the residuals

Prism can quantify goodness of fit by reporting the standard deviation of the residuals, computed in three distinct ways. Remember that the residual is the vertical distance (in Y units) of the point from the fit line or curve. If you have n data points, after the regression, you have n residuals.

What units? How to interpret?

All three values (RMSE, Sy.x, and RSDR) are expressed in the same units as Y and all can be interpreted in roughly the same way as the typical deviation of the points from the line or curve.

RMSE

If you simply take the standard deviation of those n values, the value is called the root mean square error, RMSE. The mean of the residuals is always zero, so to compute the SD, add up the sum of the squared residuals, divide by n-1, and take the square root:

$$RMSE = \sqrt{\frac{\sum(residual^2)}{n-1}}$$

Prism will report the RMSE when you check the appropriate option in the Diagnostics tab, because some fields use it. But we recommend reporting the Sy.x (below).

Sy.x or Se

The Sy.x is computed in a very similar way, but the denominator is n-K, where K is the number of parameters fit by regression. The value n-K is the number of degrees of freedom of the regression. If you only fit one parameter, then the RMSE and Sy.x are the same. If you fit two or more parameters, the Sy.x is larger and is a better estimate of goodness-of-fit. We recommend reporting it rather than the RMSE.

$$Sy.x = \sqrt{\frac{\sum(residual^2)}{n-K}}$$

RSDR

If you chose robust regression, Prism computes a different value we call the Robust Standard Deviation of the Residuals (RSDR). The goal here is to compute a robust standard deviation, without being influenced by outliers. In a Gaussian distribution, 68.27% of values lie within one standard deviation of the mean. We therefore calculate this value, which we call P68. It turns out that this value underestimates the SD a bit, so the RSDR is computed by multiplying the P68 by n/(n-K), where K is the number of parameters fit.

.

4.5.4.1.9 Why Prism doesn't report the chi-square of the fit

Some programs report a chi-square value with the results of nonlinear regression. Prism doesn't. This page explains how the chi-square value could be useful and why we don't report it.

Is the sum-of-squares too high?

Nonlinear regression minimizes the sum of the squared vertical distances between the data point and the curve. But how can you interpret the sum-of-squares, abbreviated SS? You can't really, as it depends on the number of data points you collected and the units you used to express Y.

The chi-square approach is to compare the observed scatter of the points around the curve (SS) with the amount of experimental scatter you expect to see based theory. This is done by computing chi-square using this equation:

$$\chi^2 = \sum_{i=1}^{i=n} \left(\frac{Y_i - Y_{curve}}{\sigma_i} \right)^2$$

Chi-square is the sum of the square of the ratio of the distance of a point from the curve divided by the predicted standard deviation at that value of X. Note that the denominator is the predicted standard deviation, not the actual standard deviation computed in this particular experiment.

If you know that the SD is the same for all values of X, this simplifies to:

$$\chi^2 = \frac{\sum_{i=1}^{i=n} \left(Y_i - Y_{curve} \right)^2}{SD^2}$$

The standard deviation value must be computed from lots of data so the SD is very accurate. Or, better, the SD can come from theory.

If you assume that replicates are scattered according to a Gaussian distribution with the SD you entered, and that you fit the data to the correct model, then the value of chi-square computed from that equation will follow a known chi-square distribution. This distribution depends on the number of degrees of freedom, which equals the number of data points minus the number of parameters. Knowing the value of chi-square and the number of degrees of freedom, a P value can be computed.

How can you interpret a small P value? If you are quite sure the scatter really is Gaussian, and that predicted SD is correct, a small P value tells you that your model is not right -- that the curve really doesn't follow the data very well. You should seek a better model.

But often a low P value just tells you that you don't predict the SD as well as you thought you would. It is hard to determine the SD values precisely, so hard to interpret the chi-square value. For this reason, Prism doesn't attempt the chi-square computation. We fear it would be more misleading than helpful.

Alternatives

Several approaches have been devised to answer the question of whether the SS is too high:

- The value of sum-of-squares can be used to compute R^2. This value is computed by comparing the sum-of-squares (a measure of scatter of points around the curve) with the total variation in Y values (ignoring X, ignoring the model). What values of R^2 do you expect? How low a value is too low? You can't really answer that in general, as the answer depends on your experimental system.

- If you have collected replicate Y values at each value of X, you can compare the SS with a value predicted from the scatter among replicates. Prism calls this the replicates test 390. This is very useful, but only if you have collected replicate Y measurements at each X.

- You can propose alternative models, and compare their fit to the data

Summary

Chi-square compares the actual discrepancies between the data and the curve with the expected discrepancies (assuming you selected the right model) based on the known SD among replicates. If the discrepancy is high, then you have some evidence that you've picked the wrong model. The advantage of the chi-square calculation is that it tests the appropriateness of a single model, without having to propose an alternative model and without having to have replicate values. The disadvantage is that the calculation depends on knowing the SD values with sufficient precision, which is often not the case.

Recommended alternatives are to compare the fits of two models 77, or use the replicates test 390.

4.5.4.1.10 Goodness of fit with Poisson regression

Prism can compute goodness-of-fit of Poission in four ways, selectable in the Diagnostics tab.

Pseudo R-Squared

It is not possible to compute R^2 with Poisson regression models. Instead, Prism reports the **pseudo R^2.** You can interpret it as you do a regular R^2. This is the simplest goodness-of-fit measure to understand, so we recommend it.

Pseudo $R^2$is computed from log-likelihoods of three models: *LLo*, the log-likelihood of horizontal-line model; *LLfit*, the log-likelihood of the model you chose; and *LLmax*, the maximum log-likelihood possible, which would occur when the actual responses exactly equal the predicted responses so the curve goes through every point and all the residuals equal 0.0. The equation that computes the pseudo R^2 is:

```
R2 = (LLfit - LLo) / (LLmax - LLo)
```

Negative log likelihood

Least squares regression minimizes the sum-of-squares, which Prism reports. Poisson regression maximizes the negative log of the likelihood, which Prism can report.

Deviance or G^2

The deviance is twice the difference between the maximum possible log-likelihood (see above) and the log-likelihood of the fitted model. The formula for the deviance is D=2(*LLmax* - *LLfit*). This is also called G^2.

Dispersion ratio

When data are sampled from the Poisson distribution, the variance equals the mean. Prism can report the variance-to-mean ratio (VMR), called the *dispersion ratio*. Prism reports the degree of overdispersion with a value phi. . If phi is much greater than 1.0, then the actual variance of points around the curve is greater than the mean, and the Poisson model may not be appropriate. This is called *overdispersion*. Some programs offer extensions to Poisson regression to deal with overdispersion, but Prism does not (let us know if you need this).

AICc

The AICc 84 is useful only if you separately fit the same data to three or more models. You can then use the AICc to choose between them. But note that it only makes sense to compare AICc between fits when the only difference is the model you chose. If the data are not identical between fits, then any comparison of AICc values would be meaningless. It is also essential that the weighting, or regression method, be the same for all the fits. If you use Poisson regression for one fit, you need to use it for all.

4.5.4.1.11 Runs test

Goal of the runs test

The runs test asks whether the curve deviates systematically from your data. The runs test is useful only if you entered single Y values (no replicates) or chose to fit only the means rather than individual replicates (weighting tab). If you entered and analyzed replicate data, use the replicates test 390 instead.

A **run** is a series of consecutive points that are either all above or all below the regression curve. Another way of saying this is that a run is a consecutive series of points whose residuals are either all positive or all negative. After fitting a curve, Prism counts the actual number of runs and calculates the predicted number of runs (based on number of data points). The runs test compares these two values.

How the runs test works

If the data points are randomly distributed above and below the regression curve, it is possible to calculate the expected number of runs. If there are Na points above the curve and Nb points below the curve, the number of runs you expect to see equals [(2NaNb)/(Na+Nb)]+1.

If the model fits the data poorly, you will tend to see clusters of points on the same side of the curve. This means you will have fewer runs than predicted from sample size, and the runs test will produce a low P value.

Interpreting the P value from a runs test

The P value answers this question:

> If the data are randomly scattered above and below the curve, what is the probability of observing as few runs (or even fewer) than actually observed in this analysis?

If the runs test reports a low P value, conclude that the curve doesn't describe the data very well. The problem might be that some of the errors are not independent, that outliers are mucking up the fit, or that you picked the wrong model.

Note that the P value is one-tailed. If you observed more runs than expected, the P value will be higher than 0.50.

Runs test with global fits

Prism reports the runs test for each data set, but does not report a global runs test. Prior versions of Prism reported a global runs test, but it is not clear if that result was actually meaningful.

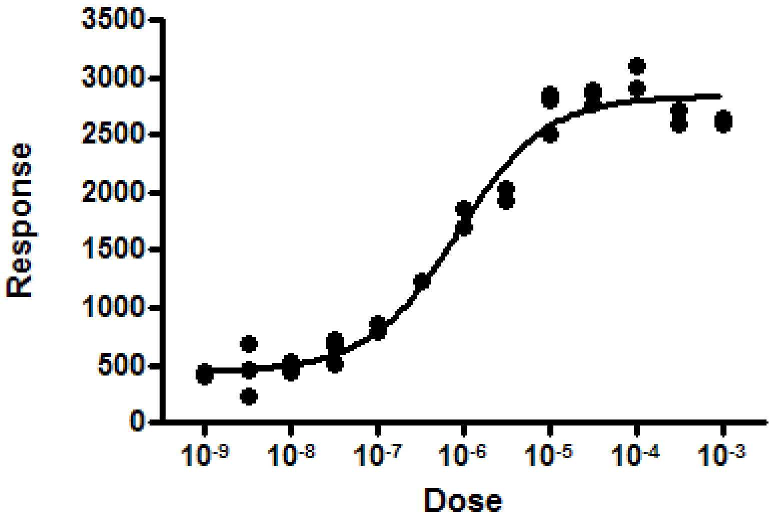
4.5.4.1.12 Replicates test

Goal of the replicates test

When evaluating a nonlinear fit, one question you might ask is whether the curve is 'too far' from the points. The answer, of course, is another question: Too far compared to what? If you have collected one Y value at each X value, you can't really answer that question (except by referring to other similar experiments). But if you have collected replicate Y values at each X, then you can ask whether the average distance of the points from the curve is 'too far' compared to the scatter among replicates.

If you have entered replicate Y values, choose the replicates test to find out if the points are 'too far' from the curve (compared to the scatter among replicates). If the P value is small, conclude that the curve does not come close enough to the data.

Example

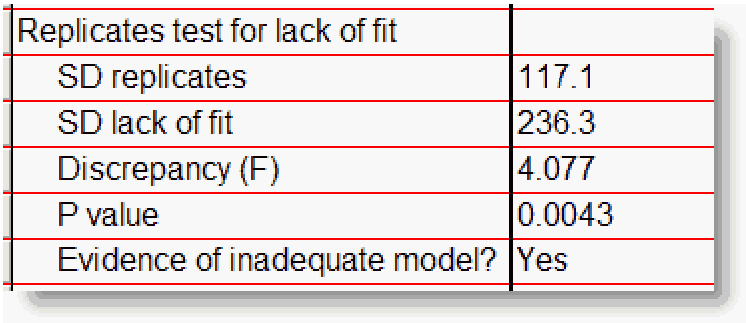


The response at the last two doses dips down a bit. Is this coincidence? Or evidence of a biphasic response?

One way to approach this problem is to specify an alternative model, and then compare the sums-of-squares of the two fits. In this example, it may not be clear which biphasic model to use as the alternative model. And there probably is no point in doing serious investigation of a biphasic model in this example, without first collecting data at higher doses.

Since replicate values were obtained at each dose, the scatter among those replicates lets us assess whether the curve is ‘too far’ from the points.

After checking the option (on the Diagnostics tab) to perform the replicates test, Prism reports these results:



Replicates test for lack of fit	
SD replicates	117.1
SD lack of fit	236.3
Discrepancy (F)	4.077
P value	0.0043
Evidence of inadequate model?	Yes

The value in the first row quantifies the scatter of replicates, essentially pooling the standard deviations at all X values into one value. This value is based only on variation among replicates. It can be computed without any curve fitting. If the replicates vary wildly, this will be a high value. If the replicates are all very consistent, this value will be low.

The value in the second row quantifies how close the curve comes to the mean of the replicates. If the curve comes close the the mean of the replicates, the value will be low. If the curve is far from some of those means, the value will be high.

The third row (F) is the square of the ratio of those two SD values.

If the model is correct, and all the scatter is Gaussian random variation around the model, then the two SD values will probably be similar, so F should be near 1. In this example, F is greater than 1 because the SD for lack of fit is so much greater than the SD of replicates. The P value answers this question:

If the model was chosen correctly and all scatter is Gaussian, what is the chance of finding a F ratio so much greater than 1.0?

A small P value is evidence that the data actually follow a model that is different than the model that you chose. In this example, this suggests that maybe some sort of biphasic dose-response model is correct -- that the dip of those last few points is not just coincidence.

How the replicates test calculations are done

This test for lack of fit is discussed in detail in advanced texts of linear regression (1,2) and briefly mentioned in texts of nonlinear regression (3, 4). Here is a brief explanation of the method:

The SD of replicates is computed by summing the square of the distance of each replicate from the mean of that set of replicates. Divide that sum by its degrees of freedom (the total number of points minus the number of X values), and take the square root. This value is based only on variation among replicates. It can be computed without any curve fitting.

The SD lack of fit is a bit trickier to understand. Replace each replicate with the mean of its set of replicates. Then compute the sum of square of the distance of those points from the curve. If there are triplicate values, then each of the three replicate values is replaced by the mean of those three values, the distance between that mean and the curve is squared, and that square is entered into the sum three times (one for each of the replicates). Now divide that sum of squares by its degree of freedom (number of points minus number of parameters minus number of X values), and take the square root.

The F ratio is the square of the ratio of the two SDs.

The P value is computed from the F ratio and the two degree of freedom values, defined above.

References

1. *Applied Regression Analysis*, N Draper and H Smith, Wiley Interscience, 3rd edition, 1998 page 47-56.

2. *Applied Linear Statistical Models* by M Kutner, C Nachtsheim, J Neter, W Li, Irwin/McGraw-Hill; 5th edition (September 26, 2004), pages 119-127

3. *Nonlinear Regression Analysis & its applications*, DM Bates and DG Watts, Wiley Interscience, 1988, pages 29-30.

4. *Nonlinear Regression,* CAF Seber and CJ Wild, Wiley Interscience, 2003, pages 30-32.

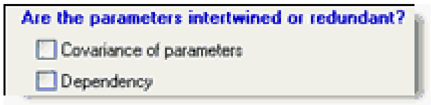
4.5.4.1.13 Dependency of each parameter

Intertwined parameters

When your model has two or more parameters, as is almost always the case, the parameters can be intertwined.

What does it mean for parameters to be intertwined? After fitting a model, change the value of one parameter but leave the others alone. this will move the curve away from the points. Now change the other parameter(s) in an attempt to to move the curve close to the data points. If you can bring the curve closer to the points, the parameters are intertwined. If you can bring the curve back to its original position, then the parameters are completely redundant.

Prism can quantify the relationships between parameters in two ways. If you are in doubt, we suggest that you focus on the dependency values and not bother with the covariance matrix.


Are the parameters intertwined or redundant?

□ Covariance of parameters

□ Dependency

Dependency

What is dependency and how do I ask Prism to compute it?

Dependency is reported for each parameter, and quantifies the degree to which that parameter is intertwined with others. Check a check box on the Diagnostics tab of nonlinear regression to view dependencies for each parameter.

Interpreting dependency

The value of dependency always ranges from 0.0 to 1.0.

A dependency of 0.0 is an ideal case when the parameters are entirely independent (mathematicians would say *orthogonal*). In this case the increase in sum-of-squares caused by changing the value of one parameter cannot be reduced at all by also changing the values of other parameters. This is a very rare case.

A dependency of 1.0 means the parameters are redundant. After changing the value of one parameter, you can change the values of other parameters to reconstruct exactly the same curve. If any dependency is greater than 0.9999, GraphPad labels the fit 'ambiguous 437'.

With experimental data, of course, the value will almost always lie between these extremes. Clearly a low dependency value is better. But how high is too high? Obviously, any rule-of-thumb is arbitrary. But dependency values up to 0.90 and even 0.95 are not uncommon, and are not really a sign that anything is wrong.

A dependency greater than 0.99 is really high, and suggests that something is wrong. This means that you can create essentially the same curve, over the range of X values for which you collected data, with multiple sets of parameter values. Your data simply do not define all the parameters in your model. If your dependency is really high, ask yourself these questions:

- Can you fit to a simpler model?
- Would it help to collect data over a wider range of X, or at closer spaced X values? It depends on the meaning of the parameters.
- Can you collect data from two kinds of experiments, and fit the two data sets together using global fitting?
- Can you constrain one of the parameters to have a constant value based on results from another experiment?

If the dependency is high, and you are not sure why, look at the covariance matrix (see below). While the dependency is a single value for each parameter, the covariance matrix reports the normalized covariance for each pair of parameters. If the dependency is high, then the covariance with at least one other parameter will also be high. Figuring out which parameter that is may help you figure out where to collect more data, or how to set a constraint.

How dependency is calculated

This example 111 will help you understand how Prism computes dependency.

4.5.4.1.14 Covariance matrix

What is the covariance matrix and how do I ask Prism to compute it?

The normalized covariance is reported for each pair of parameters, and quantifies the degree to which those two parameters are intertwined. Check a check box on the Diagnostics tab of nonlinear regression to view this covariance matrix.

Interpreting the normalized covariance matrix

Each value in the normalized covariance matrix ranges from -1.0 to 1.0. A value equal to -1.0 or 1.0 means the two parameters are redundant. A value of 0.0 means the parameters are completely independent or orthogonal -- if you change the value of one parameter you will make the fit worse and changing the value of the other parameter can't make it better. You can interpret a normalized covariance much as you interpret a correlation coefficient.

Note the difference between covariance and dependency 393. Each value in the covariance matrix tells you how much two parameters are intertwined. In contrast, each dependency value tells you how much that parameter is intertwined with all other parameters.

How to convert to the nonnormalized variance/covariance matrix

Some other programs report the actual (not normalized) variance-covariance matrix. Compute the actual covariance -- cov(i,j) -- of any two parameters (so i does not equal j) from the normalized matrix Prism reports -- NormCov(i,j) -- and the standard errors of the parameters using this equation:

Cov(i, j) = NormCov(i, j) * SE(i) * SE(j)

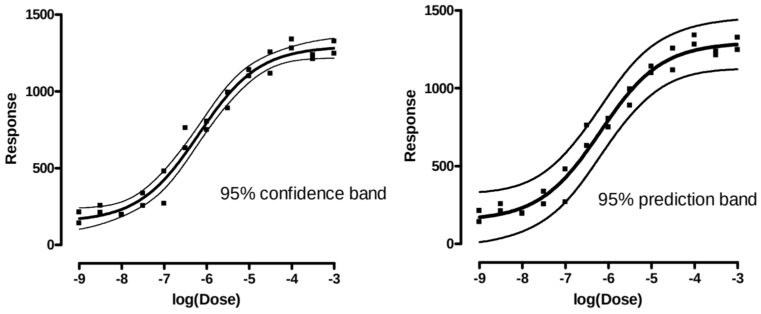
Prism does not report the normalized covariance matrix for a parameter with itself, because the normalized covariance of any parameter with itself equals, by definition, 1.0. The covariance of any parameter with itself is better called its variance. You can calculate the variance of any parameter (a diagonal value in the variance-covariance matrix) using this equation:

Cov(i, i) = SE(i)2

4.5.4.1.15 Confidence and prediction bands

What are confidence and prediction bands?

The 95% **confidence bands** enclose the area that you can be 95% sure contains the true curve. It gives you a visual sense of how well your data define the best-fit curve. It is closely related to the 95% **prediction bands** , which enclose the area that you expect to enclose 95% of future data points. This includes both the uncertainty in the true position of the curve (enclosed by the confidence bands), and also accounts for scatter of data around the curve. Therefore, prediction bands are always wider than confidence bands. How confidence and prediction bands are computed. 108



How to plot confidence bands, prediction bands or both

Choose to plot confidence or prediction bands by an checking option on the Confidence tab 343.

Prism lets you choose either a confidence band or a prediction band, but not both. To plot both on one graph, you need to analyze your data twice, choosing a confidence band the first time and a prediction band the second time.

The regression lines or curves are data sets that you can add to any graph by drag-n-drop or using the Change menu from the graph.

Prism can also show you the XY coordinates that define the curve and confidence bands (or prediction bands). To see this table, click the icon to the right of the results tabs to show a list of all results tabs. View the one labeled 'Curve'. For each X value, you'll see the Y value and the distance the confidence or prediction curve is above and below that value.

How to plot a one-sided confidence or prediction band

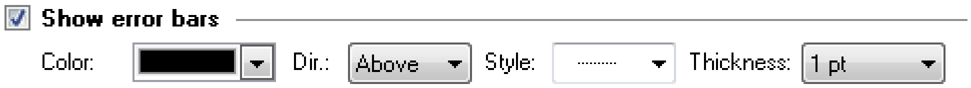
In some situations, it makes sense to plot only one side of the confidence or prediction band. One use for a one-sided confidence band would be on a graph that plots purity of a substance over time. You want to know, with a certain confidence, how low the purity could be at a certain time. But you don't really care about quantifying how high the purity could be.

To plot a one sided 95% confidence or prediction band:

1. Check the option (on the Diagnostics tab) to plot "90%" confidence (or prediction) bands. When you plot in only one direction, this is really a 95% confidence band.

2. On the Format Graph dialog, choose the data set that defines regression curve and make sure that error bars are turned on with the "---" style.

3. Choose to plot those error bands in one direction.



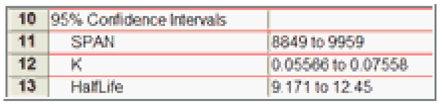
☑ Show error bars

Color: ■ Dir.: Above Style: ········· Thickness: 1 pt

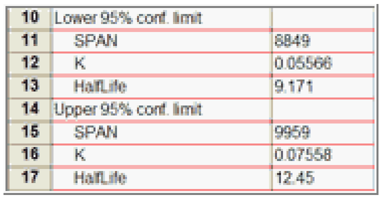
Do not mix up confidence intervals and confidence bands

It is easy to mix up confidence intervals and confidence bands. Choose both on the Confidence tab 343.

The **95% confidence interval** tells you how precisely Prism has found the best-fit value of a particular parameter. It is a range of values, centered on the best-fit value. Prism can display this range in two formats:



10	95% Confidence Intervals	
11	SPAN	8849 to 9959
12	K	0.05566 to 0.07558
13	HalfLife	9.171 to 12.45



10	Lower 95% conf. limit	
11	SPAN	8849
12	K	0.05566
13	HalfLife	9.171
14	Upper 95% conf. limit	
15	SPAN	9959
16	K	0.07558
17	HalfLife	12.45

Don't mix up confidence bands and prediction bands

Note the difference between confidence and prediction bands:

The 95% confidence bands enclose the area that you can be 95% sure contains the true curve. If you have many data points, the confidence bands will be near the line or curve, and most of your data will lie outside the confidence bands.

The 95% prediction bands enclose the area that you expect to enclose 95% of future data points. They are wider than confidence bands -- much wider with large data sets.

Situations where Prism won't plot confidence or prediction bands

Prism will not plot confidence or prediction bands in several situations:

- If the best-fit value of a parameter hits a constraint 441, the fit is unlikely to be useful. Prism does not plot confidence or prediction bands, because they would almost certainly be misleading.

- If the results of nonlinear regression are ambiguous 437, the confidence or prediction bands would be super wide, maybe infinitely wide. They would not be useful, so Prism does not plot them.

- If you choose robust nonlinear regression, Prism does not compute confidence or prediction bands, as it cannot compute standard errors or confidence intervals of the parameters.

- The fit is perfect 443. If the sum-of-squares is 0.0 and R2 is 1.0, it is not possible to compute or interpret confidence or prediction bands.

- If the fit is interrupted 435, confidence and prediction bands are not computed.

- If you chose to weight by 1/SD^2. To compute confidence or prediction bands, Prism needs to be able to compute the weights for any value of X. When you weight by 1/SD^2, weights are only known at the X values of the data, but not for other X values, so Prism cannot compute prediction or confidence bands.

4.5.4.1.16 Hougaard's measure of skewness

Consequences of an asymmetrical parameter

Even though nonlinear regression, as its name implies, is designed to fit nonlinear models, some of the inferences actually assume that some aspects of the model are close to linear, so that the uncertainty about each parameter's value is symmetrical. This means that if you analyzed many data sets sampled from the same system, the distribution of the best-fit values of the parameter would be symmetrical and Gaussian.

If the distribution of a parameter is highly skewed, there are two consequences 752:

- The SE of that parameter will not be a very useful measure of uncertainty. The SE is interpreted as a plus-minus assessment of how sure you are of the parameter value. But if the parameter is very asymmetrical, then a single SE cannot really describe the uncertainty.
- Symmetrical confidence intervals for that parameter cannot be interpreted at face value. If the parameter is very asymmetrical, then that symmetrical confidence interval does not give a accurate picture of the uncertainty. Note that Prism (starting with version 7) can compute asymmetrical profile likelihood confidence intervals, and these work fine with asymmetrical parameters.

Hougaard's measure of skewness

Hougaard (1) developed a way to assess the skewness of a parameter used in nonlinear regression without doing any simulations. Prism will compute this value for each parameter when you check the option the Diagnostics tab of nonlinear regression in the section labeled "Are the parameters intertwined, redundant or skewed?". The results are tabulated along with the other results of nonlinear regression.

Ratkowsky has proposed the following interpretation:

Absolute value	**Interpretation**
<0.10	Ideal. Almost linear. Confidence intervals can be interpreted at face value
0.10 - 0.25	Adequate
0.25 - 1.00	Noticeable skewness. Consider alternative parameterizations of the equation
> 1.00	Considerable skewness. Strongly consider alternatives

Note that these interpretations apply to the absolute value of the Hougaard's measure.

Hougaard's skewness with unequal weighting

While Prism 6 and 7 calculated Hougaard's skewness correctly for unweighted fits, they computed it incorrectly if you chose unequal weighting. This is fixed in Prism 8.

Notes

- Hougaard's measure of skewness is measured for each parameter in the equation (omitting parameters fixed to constant values).
- Prism does not compute Hougaard's skewness if you chose a robust fit, because the method is not defined for this situation.
- The values depend on the equation, the number of data points, the spacing of the X values, and the Y values.
- Hougaard's measure of skewness has no units.

- A positive value means that the asymmetry is to the right, with a longer confidence interval above the best-fit value than below it. A negative value means that the asymmetry is to the left.

- The SAS documentation does a great job of explaining Hougaard's measure (3).

- Prism can compute asymmetrical profile likelihood confidence intervals. These show the asymmetry directly, so reduce the need to ask Prism to compute Hougaard's skewness.

References

1. P. Hougaard. The appropriateness of the asymptotic distribution in a nonlinear regression model in relation to curvature. Journal of the Royal Statistical Society. Series B (Methodological) (1985) pp. 103-114

2. David A. Ratkowsky, Nonlinear Regression Modeling: A Unified Practical Approach (Statistics: a Series of Textbooks and Monogrphs). IBSN:0824719077

3.SAS documentation about Hougaard's measure.

4.5.4.1.17 Test for appropriate weighting/homoscedasticity

Overview

If you don't change the default weighting, nonlinear regression assumes that, on average, the vertical distance of the points from the curve is the same all the way along the curve. This assumption goes by the name *homoscedasticity*, and Prism can test this assumption with a *test for homoscedasticity*.

If you have chosen to differentially weight the points, Prism assumes that the weighted distance of the points from the curve is the same all the way along the curve. Prism tests this assumption with the *test for appropriate weighting*.

The null hypothesis is that you picked the right weighting scheme, so there is no correlation between the Y value of the curve and the absolute value of the weighted residual. A high P value is consistent with this hypothesis. A small P value suggests that your data violate that assumption. In this case, it might make sense to choose a more appropriate weighting scheme.

Details of the method

To run these tests, Prism follows these steps.

1. For each point, compute the difference between the actual Y value and the Y value of the fit curve at that value of X. This is called a *residual*.

2. If you chose a weighting scheme, apply this scheme to each residual. If you chose relative weighting, divide each residual by the predicted value of Y. Note a confusing point. In the weighting tab, you choose how to weight the square of the residuals. So the relative weighting is shown in the dialog as dividing by Y^2. Here we are weighting the residual itself, not the square of the residual, so divide by Y.

3. Compute the absolute values of all the weighted residuals.

4. Create a new table (not shown) where the X values are the absolute values of the weighted residuals and the Y values are the predicted Y values of the curve. There will be one row in this table for each point entered into the regression.

5. Compute the Spearman rank correlation and compute the corresponding P value.

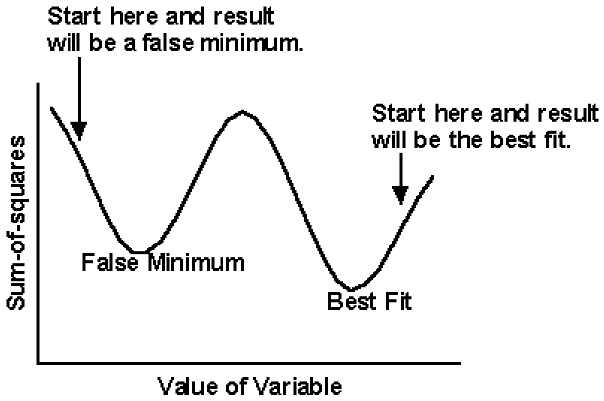
This method has not been published as far as we know, but it is also used by SigmaPlot (where we learned about it). Note that SigmaPlot does exactly what we say above. Even though their manual says they use the observed Y value in step 4, they actually use the predicted value. We don't know of any publication that defines this test. If you search for the Spearman test for hereoscedasticity you'll find that is usually used with linear regression, and where the residuals are correlated with the X value, rather than the predicted Y value.

(Note there was a bug in Prism 7.00-7.03 and 7.0a-7.0c.)

4.5.4.1.18 Could the fit be a local minimum?

What is a false minimum?

The nonlinear regression procedure adjusts the variables in small steps in order to improve the goodness-of-fit. If Prism converges on an answer, you can be sure that altering any of the variables a little bit will make the fit worse. But it is theoretically possible that large changes in the variables might lead to a much better goodness-of-fit. Thus the curve that Prism decides is the "best" may really not be the best.


Start here and result will be a false minimum.
Start here and result will be the best fit.
Sum-of-squares
False Minimum
Best Fit
Value of Variable

Think of latitude and longitude as representing two variables Prism is trying to fit. Now think of altitude as the sum-of-squares. Nonlinear regression works iteratively to reduce the sum-of-squares. This is like walking downhill to find the bottom of the valley. When nonlinear regression has converged, changing any variable increases the sum-of-squares. When you are at the bottom of the valley, every direction leads uphill. But there may be a much deeper valley over the ridge that you are unaware of. In nonlinear regression, large changes in variables might decrease the sum-of-squares.

This problem (called finding a local minimum) is intrinsic to nonlinear regression, no matter what program you use. You will rarely encounter a local minimum if your data have little scatter, you collected data over an appropriate range of X values, and you have chosen an appropriate equation.

Testing for a false minimum

To test for the presence of a false minimum:

1. Note the values of the variables and the sum-of-squares from the first fit.

2. Make a large change to the initial values of one or more parameters and run the fit again.

3. Repeat step 2 several times.

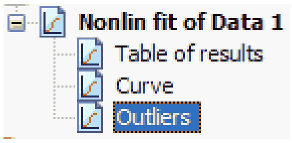
Ideally, Prism will report nearly the same sum-of-squares and same parameters regardless of the initial values. If the values are different, accept the ones with the lowest sum-of-squares.

4.5.4.1.19 Outliers

The outliers table

Choose to exclude or count outliers as part of nonlinear regression on the Method tab 327.

If you do this, one of the results tables will list all outliers.



Keep in mind an important distinction:

- If you chose to count outliers , then the outliers are still included in the fit.

- If you chose to exclude outliers , then the outliers are ignored by the fit, but are still included on the graph 358.

In both cases, the aggressiveness of the outlier hunt is determined by the ROUT coefficient Q that you enter.

What does it mean to be an outlier

Prism uses the ROUT algorithm 88, using the value of Q that you can adjust to decide when to declare a point to be an outlier. Some considerations:

- The first thing to do is make sure it isn't simply a problem in data entry. If so, fix it.

- Review the list of situations where the outlier identification procedure should not be used 89.

- Be alert to biological variation. In some experimental situations, outliers can only arise from lab mistakes. These should be excluded. In other situations, an outlier can be the result of biological variation. You may have discovered a new polymorphism in some gene. It could be missing a big opportunity if you exclude and ignore those values.

What to do once you have identified an outlier

After you have identified one or more outliers, you have several choices.

- While outliers are ignored by the nonlinear regression calculation, they are still plotted on the graph. If you want the outlier to be entirely removed from the graph, including error bars on the graph, go back to the data table and exclude the outlier(s). Prism will then refit the curve, and replot the graph without any outliers.

- If you want the outlier to remain on the graph, simply polish the graph Prism provides. Keep in mind that the outliers are a dataset, so you can separately adjust the size, color and symbol of outliers and all the data points. Use the "Datasets on graph" tab of Format Graph to adjust the back-to-front order of data sets to keep the outlier in front of the other data sets. The dialog shows the datasets in back to front order, so you want to keep the outlier data sets lower in the list.

- If you want to remove the special formatting of outliers on the graph, use the "Datasets on graph" tab of Format Graph. Select the outlier dataset(s) and click Remove. The outlier will still be plotted as part of the full data set, but will no longer be plotted separately as an outlier data set.

- If you asked Prism to count the outliers (Diagnostics tab) but now wish to fit the data with the outlier ignored, go back to the nonlinear regression dialog and choose "Automatic outlier elimination" (on the Fit tab).

4.5.4.1.20 Troubleshooting nonlinear regression

The #1 tip for troubleshooting

Nonlinear regression starts with initial values for each parameter. If these initial values are very far from the correct values, nonlinear regression might go astray. To check this, go to theDiagnostics tab of the nonlinear regression dialog 347 and choose the option at the top of the dialog: Don't fit a curve. Instead plot the curve defined by the initial values of the parameters.

If the curve defined by the initial values does not generally follow the shape of the data, and go near the data points, you should change the initial values (Initial Values tab of nonlinear regression 339) until they do.

If Prism reports an error code

Prism reports error codes as a single phrase that appears over the column of results, for example "Bad initial values" or "Impossible weights". Learn more about each error message. 430

Common problems in nonlinear regression

Here is a short list of potential curve fitting problems with suggested solutions.

The equation simply does not describe the data.

Try a different equation.

The initial values are too far from their correct values.

Enter different initial (estimated) values for the parameters. If you entered your own equation, check that you entered sensible rules for generating initial values.

The range of X values is too narrow to define the curve completely.

If possible, collect more data. Otherwise, hold one of the parameters to a constant value.

You have not collected enough data in a critical range of X values.

Collect more data in the important regions.

Your data are very scattered and don't really define a curve.

Try to collect less scattered data. If you are combining several experiments, consider normalizing the data for each experiment to an internal control.

The equation includes more than one component, but your data don't follow a multicomponent model.

Use a simpler equation.

Your numbers are too large or too small.

If your X or Y values are huge, divide by a constant to change the units. Avoid values greater than, say, 100,000.

If your X or Y values are tiny, multiply by a constant change the units. Avoid values less than about 0.00001.

Changing the units is unlikely to solve the problem, but it is worth a try.

You've set a parameter to an inappropriate constant value.

Check that you haven't made a simple mistake like setting a maximum plateau to 1.0 when it should be 100, or a Hill slope to +1.0 when it should be -1.0.

Consult an analysis check list

Prism has three different analysis checklists for nonlinear regression. Use the one that matches your goal.

Analysis checklist: Fitting a model 422

Analysis checklist: Comparing nonlinear fits 426

Analysis checklist: Interpolating from a standard curve 428

4.5.4.1.21 Why results can differ in various Prism versions

Prism 8.2

On the confidence tab 343 of the nonlinear regression dialog, you choose how Prism should deal with difficult fits. If you choose the recommended method (new in 8.2), difficult parameters (and their confidence interval) will be marked as "unstable". If you choose the other method, entire fits may be flagged as "ambiguous" and certain parameters will be preceded with a tilde/squiggle (~). The The method used to identify an unstable parameter is very different than the method to identify an ambiguous parameter, so the results will vary based on your choice.

Prism 8.0

Prism 8 improved some algorithms, so sometimes fits curves faster than Prism 7 did. The results should be the same or similar in almost all cases. Prism 8 fixed a bug so now reports Hougaard's skewness properly even if with unequal weighting. Prism 6 and 7 reported incorrect results for Hougaard's skewness with unequal weighting.

Prism 5-7

Prism 5 and 6 use the same algorithms, so should always report identical results. Prism 7 introduced slightly improved algorithms, so the results might vary a bit, but the differences are trivial.

Prism 4

Prism 4 used slightly different algorithms, so curve fitting results can differ from results with later Prism versions in these cases:

- If your fit is labeled "Ambiguous" by Prism 5 or later, you know that some of the parameters are not determined precisely. Prism 4 presented a full set of results in this case, but the results are not useful when the fit is ambiguous.

- If you chose no weighting, check the sum-of-squares from the two programs. The goal of regression is to minimize that sum of squares, so see which version of Prism found a fit with the smaller sum-of-squares. Prism 5 and later have a few improvements in the fitting algorithm, so occasionally it can find a better fit than did Prism 4. The differences, if any, are usually trivial.

- If you chose to weight by the Y values (or the Y values squared), Prism 5 and later handle weighting differently than did Prism 4. Prism now weights by the Y value of the curve, while Prism 4 (and earlier releases) weighted by the Y value of the data. Weighting by the Y value of the curve is better, so the results of Prism 5 and later are more correct. Since the weighting is computed differently, you can't directly compare the weighted sum-of-square values reported by the two versions of Prism.

- When you compare two models, Prism now does an extra step. If one of the models is ambiguous, then Prism chooses the other model, without doing the F test or AIC comparison.

- Prism now offers more rules for defining initial parameter values. If your equation uses one of these new rules, Prism 4 might not be able to find a reasonable fit until you tweak those initial values. In particular, Prism now has smarter rules for fitting sigmoidal log(dose) vs. response curves.

- If you entered data as mean, SD (or SEM) and N, then Prism 4 (by default) fits the means and weights by the sample size (N). This is one of the two options on the weighting tab (the other option is to fit means only, ignoring N). Prism is now smarter by default (although you can choose to just fit the means and ignore N and SD values). It accounts not only for sample size N but also for the SD (or SEM) values you enter. With Prism 5 or later (but not Prism 4), you'll get exactly the same results from data entered as mean, SD and N as you would have by entering raw data. Prism 4 only accounts for differences in N, but not SD. The best fit values of the parameters, and thus the appearance of the curve, is the same with Prism 4 and 5. But Prism 5 and later does a smarter job with standard errors, confidence intervals, and comparisons of models.

4.5.4.2 Interpreting results: Comparing models

In many cases, your primary goal in fitting curves is to compare models, or compare fits of two (or more) data sets.

- First review the general principles 410 of comparing models.

- Next review the method you choose to use: Extra sum-of-squares 411 or AIC model comparison 417.

- Review how Prism compares models when it detects outliers 418.

4.5.4.2.1 Interpreting comparison of models

Reality check before statistics

Apply a common sense reality check before looking at any statistical approach to comparing fits. If one of the fits has results that are scientifically invalid, then accept the other model. Only when both fits make scientific sense should you use statistical method to compare the two fits .

Prism partially automates this 'reality check' approach. If the fit of either model is ambiguous 437, then Prism chooses the other model without performing any statistical test.

Statistical approaches balance the change in sum-of-squares with the change in numbers of degrees of freedom

The more complicated model (the one with more parameters) almost always fits the data better than the simpler model. Statistical methods are needed to see if this difference is enough to prefer the more complicated model. Prism can do this via the extra sum-of-squares F test 411 or using information theory and computation of AIC 417. Don't use R^2 or the adjusted R^2 to compare models (1).

Both these methods only make sense when the models being compared have different numbers of parameters, and so have different numbers of degrees of freedom. If you want to compare two models with the same number of parameters, there is no need to use either the F test or AIC. Simply choose the model that fits the data the best with the smallest sum-of-squares.

How do these methods work to compare data sets?

The Compare tab 334 of Prism lets you ask "Do the best-fit values of selected unshared parameters differ between data sets?" or "Does one curve adequately fit all data sets?". Applying the F test or Akaike's method to answering these questions is straightforward. Prism compares the sum-of-squares of two fits.

- In one fit, the model is separately fit to each data set, and the goodness-of-fit is quantified with a sum-of-squares. The sum of these sum-of-square values quantifies the goodness of fit of the family of curves fit to all the data sets.

- The other fit is a global fit to all the data sets at once, sharing specified parameters. If you ask Prism whether one curve adequately fits all data sets, then it shares all the parameters.

These two fits are nested (the second is a simpler case of the first, with fewer parameters to fit) so the sums-of-squares (actually the sum of sum of squares for the first fits) can be compared using either the F test 411 or Akaike's method 417.

1. Spiess, A.-N. & Neumeyer, N. An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach. BMC Pharmacol 10, 6–6 (2010).

4.5.4.2.2 Interpreting the extra sum-of-squares F test

The extra-sum-of-squares F test compares nested models

The extra-sum-of-squares F test compares the goodness-of-fit of two alternative nested models. "Nested" means that one model is a simpler case of the other. Let's consider what this means in different contexts:

- If you asked Prism to test whether parameters are different between treatments, then the models are nested. You are comparing a model where Prism finds separate best-fit values for some parameters vs. a model where those parameters are shared among data sets. The second case (sharing) is a simpler version (fewer parameters) than the first case (individual parameters).

- If you asked Prism to test whether a parameter value is different than a hypothetical value, then the models are nested. You are comparing the fit of a model where a parameter is fixed to a hypothetical value to the fit of a model where Prism finds the best-fit value of that parameter. The first case (fixed value) is a simpler version (fewer parameters to fit) than the second.

- if you are comparing the fits of two equations you chose, and both models have the same number of parameters, then the two models cannot be nested. With nest models, one model has fewer parameters than the other. When the two models have the same number of parameters, Prism reports that it cannot compute the F test because the two models have the same number of degrees of freedom. In this case, Prism does not report a P value, and chooses to plot the model whose fit has the lower sum-of-squares. Prism reports the error message, "Models have the same DF."
- if you are comparing the fits of two equations you chose with different numbers of parameters, the models may or may not be nested. Prism does not attempt to do the algebra necessary to make this determination. If you chose two models that are not nested, Prism will report results for the extra-sum-of-squares F test, and these results will not be useful.

Interpreting the P value

The extra-sum-of-squares F test is based on traditional statistical hypothesis testing. The F test compares the improvement of SS with the more complicated model vs. the loss of degrees of freedom.

The null hypothesis is that the simpler model (the one with fewer parameters) is correct. The improvement of the more complicated model is quantified as the difference in sum-of-squares. You expect some improvement just by chance, and the amount you expect by chance is determined by the number of degrees of freedom in each model. The F test compares the difference in sum-of-squares with the difference you'd expect by chance. The result is expressed as the F ratio, from which a P value is calculated.

The P value answers this question:

> If the null hypothesis is really correct, in what fraction of experiments (the size of yours) will the difference in sum-of-squares be as large as you observed, or even larger?

If the P value is small, conclude that the simple model (the null hypothesis) is wrong, and accept the more complicated model. Usually the threshold P value is set at its traditional value of 0.05.

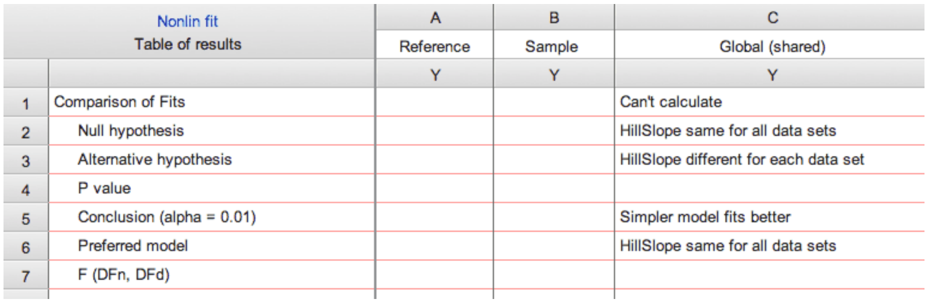
If the P value is high, conclude that the data do not present a compelling reason to reject the simpler model.

Prism names the null and alternative hypotheses, and reports the P value. You set the threshold P value in the Compare tab of the nonlinear regression dialog. If the P value is less than that threshold, Prism chooses (and plots) the alternative (more complicated) model. It also reports the value of F and the numbers of degrees of freedom, but these will be useful only if you want to compare Prism's results with those of another program or hand calculations.

When Prism won't report a P value when comparing models

Prism skips the extra sum of squares test and does not report a P value in these situations:

- If the simpler model fits the data better than (or the same) as the more complicated model. The whole point of the F test is to deal with a tradeoff. The model with more parameters fits the data better fit, but that may just be due to chance. The F test asks if that improvement in fit (decrease in sum of squares) is large enough to be "worth" the loss in degrees of freedom (increase in number of parameters). In the rare cases where the simpler model fits better (or the same) as the more complicated model (the one with more parameters), Prism will choose the simpler model without computing the F test and report "Simpler model fits better":



	Nonlin fit Table of results	A	B	C
		Reference	Sample	Global (shared)
		Y	Y	Y
1	Comparison of Fits			Can't calculate
2	Null hypothesis			HillSlope same for all data sets
3	Alternative hypothesis			HillSlope different for each data set
4	P value			
5	Conclusion (alpha = 0.01)			Simpler model fits better
6	Preferred model			HillSlope same for all data sets
7	F (DFn, DFd)			

- If the fit of either model is ambiguous 437 or flagged 350, then Prism chooses the other model without performing any statistical test. You have a choice in the Compare tab 334 of nonlinear regression to turn off this criterion.

- If the fit of one model did not converge 437, then Prism chooses the other model without doing the F test. Since the fit of one model didn't converge, it makes no sense to compare the sum-of-squares of the two models.

- If one model fits the data perfectly, so the sum of squares equals zero. If one model fits perfectly, Prism chooses it without doing the F test.

- If the two models have the same number of degrees of freedom. The idea of the F test is to balance the improvement of sum-of-squares (better fit) with the decrease in degrees of freedom (more parameters). The F test makes no sense (and is mathematically impossible due to division by zero) if the two models have the same number of degrees of freedom. In this case, Prism picks the model that fits best.

4.5.4.2.3 Intepreting the Likeihood Ratio test (Poisson regression)

The likelihood test compares nested models

The likelihood test compares the goodness-of-fit of two alternative nested models. "Nested" means that one model is a simpler case of the other. Let's consider what this means in different contexts:

- If you asked Prism to test whether parameters are different between treatments, then the models are nested. You are comparing a model where Prism finds separate best-fit values for some parameters vs. a model where those parameters are shared among data sets. The second case (sharing) is a simpler version (fewer parameters) than the first case (individual parameters).

- If you asked Prism to test whether a parameter value is different than a hypothetical value, then the models are nested. You are comparing the fit of a model where a parameter is fixed to a hypothetical value to the fit of a model where Prism finds the best-fit value of that parameter. The first case (fixed value) is a simpler version (fewer parameters to fit) than the second.

- if you are comparing the fits of two equations you chose, and both models have the same number of parameters, then the two models cannot be nested. With nest models, one model has fewer parameters than the other. When the two models have the same number of parameters, Prism reports that it cannot compute the F test because the two models have the same number of degrees of freedom. In this case, Prism does not report a P value, and chooses to plot the model whose fit has the lower sum-of-squares. Prism reports the error message, "Models have the same DF."

- if you are comparing the fits of two equations you chose with different numbers of parameters, the models may or may not be nested. Prism does not attempt to do the algebra necessary to make this determination. If you chose two models that are not nested, Prism will report results for the extra-sum-of-squares F test, and these results will not be useful.

Interpreting the P value

The likelihood test is based on traditional statistical hypothesis testing. It compares the improvement of fit (the likelihood ratio) with the more complicated model vs. the loss of degrees of freedom (more parameters).

The null hypothesis is that the simpler model (the one with fewer parameters) is correct. The improvement of the more complicated model is quantified by the likelihood ratio. You expect some improvement just by chance, and the amount you expect by chance is determined by the number of degrees of freedom in each model. A P value is calculated from the likelihood ratio and the difference in df beween the two models.

The P value answers this question:

> If the null hypothesis is really correct, in what fraction of experiments (the size of yours) will the likelihood ratio be as large as you observed, or even larger?

If the P value is small, conclude that the simple model (the null hypothesis) is wrong, and accept the more complicated model. Usually the threshold P value is set at its traditional value of 0.05.

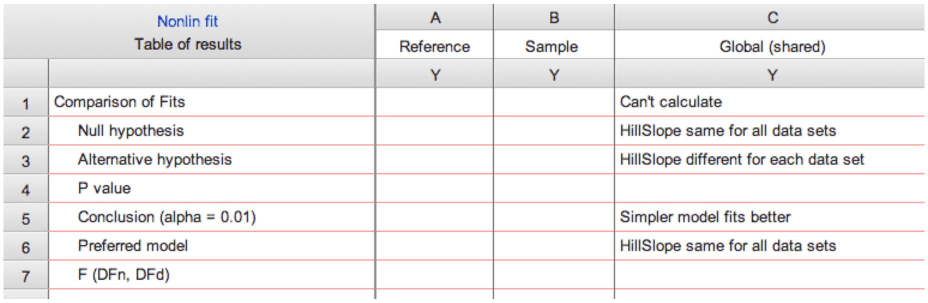
If the P value is high, conclude that the data do not present a compelling reason to reject the simpler model.

Prism names the null and alternative hypotheses, and reports the P value. You set the threshold P value in the Compare tab of the nonlinear regression dialog. If the P value is less than that threshold, Prism chooses (and plots) the alternative (more complicated) model. It also reports the value of F and the numbers of degrees of freedom, but these will be useful only if you want to compare Prism's results with those of another program or hand calculations.

When Prism won't report a P value when comparing models

Prism skips the likelihood ratio test and does not report a P value in these situations:

- If the simpler model fits the data better than (or the same) as the more complicated model. The whole point of the test is to deal with a tradeoff. The model with more parameters fits the data better fit, but that may just be due to chance. The test asks if that improvement in fit (decrease in sum of squares) is large enough to be "worth" the loss in degrees of freedom (increase in number of parameters). In the rare cases where the simpler model fits better (or the same) as the more complicated model (the one with more parameters), Prism will choose the simpler model without computing the test and report "Simpler model fits better":



	Nonlin fit Table of results	A	B	C
		Reference	Sample	Global (shared)
		Y	Y	Y
1	Comparison of Fits			Can't calculate
2	Null hypothesis			HillSlope same for all data sets
3	Alternative hypothesis			HillSlope different for each data set
4	P value			
5	Conclusion (alpha = 0.01)			Simpler model fits better
6	Preferred model			HillSlope same for all data sets
7	F (DFn, DFd)			

- If the fit of either model is ambiguous 437 or flagged 350, then Prism chooses the other model without performing any statistical test. You have a choice in the Compare tab 334 of nonlinear regression to turn off this criterion.

- If the fit of one model did not converge 437, then Prism chooses the other model without doing the F test. Since the fit of one model didn't converge, it makes no sense to compare the sum-of-squares of the two models.

- If one model fits perfectly, Prism chooses it without doing the F test.

- If the two models have the same number of degrees of freedom. The idea of the likelihood ratio test is to balance the improvement of sum-of-squares (better fit) with the decrease in degrees of freedom (more parameters). The test makes no sense (and is mathematically impossible) if the two models have the same number of degrees of freedom. In this case, Prism picks the model that fits best.

Relationship to the extra sum-of-square F test

The extra sum-of-squares F test is equivalent to the likelihood ratio test when you choose least-squares regression. We present the results as a F test with least squares regression because that is more familiar to most biologists. The P value would be the same if the results were expressed as a likelihood ratio.

4.5.4.2.4 Interpreting AIC model comparison

This alternative approach is based on information theory, and does not use the traditional “hypothesis testing” statistical paradigm. Therefore it does not generate a P value, does not reach conclusions about “statistical significance”, and does not “reject” any model.

The method determines how well the data supports each model, taking into account both the goodness-of-fit (sum-of-squares) and the number of parameters in the model. The results are expressed as the probability that each model is correct, with the probabilities summing to 100%. If one model is much more likely to be correct than the other (say, 1% vs. 99%), you will want to choose it. If the difference in likelihood is not very big (say, 40% vs. 60%), you will know that either model might be correct, so will want to collect more data.

Of course, these probabilities are meaningful only in the context of comparing those two models. It is possible a third model you didn't test fits much better so is much more likely to be correct.

Prism names the null and alternative hypotheses, and reports the likelihood that each is correct It also reports the difference between the AICc values (as the AICc of the simple model minus the AICc of the more complicated model), but this will be useful only if you want to compare Prism's results with those of another program or hand calculations. Prism chooses and plots the model that is more likely to be correct, even if the difference in likelihoods is small.

4.5.4.2.5 How Prism compares models when outliers are eliminated

Comparing models only makes sense when both models 'see' exactly the same set of data. That makes it tricky to combine outlier elimination with model comparison. Prism handles this situation automatically.

If you select automatic outlier elimination (on the Method tab 329), Prism first fits each model separately to determine which points (if any) are outliers. Then it removes all the points that are defined as outliers from either of the models, and fits both models again. It uses the sum-of-squares and df from these fits to perform the comparison. This approach ensures that exactly the same points are fit to both models, and that outliers from either model are excluded.

4.5.4.2.6 Interpreting the adjusted R2

Why you should not use R^2 to compare models

R^2 quantifies how well a model fits the data, so it seems as though it would be an easy way to compare models. It sure sounds easy -- pick the model with the larger R^2. The problem with this approach is that there is no penalty for adding more parameters. So the model with more parameters will bend and twist more to come nearer the points, and so almost always has a higher R^2. If you use R^2 as the criteria for picking the best model, you'd almost always pick the model with the most parameters.

The adjusted R^2 accounts for the number of parameters fit

The adjusted R^2 always has a lower value than R^2 (unless you are fitting only one parameter). The equations below show why.

$$R^2 = 1 - \frac{SS_{residuals}}{SS_{total}}$$

$$\text{Adjusted R}^2 = 1 - \frac{SS_{residuals}/(n-K)}{SS_{total}/(n-1)}$$

The equations above show how the adjusted R^2 is computed. The sum-of-squares of the residuals from the regression line or curve have n-K degrees of freedom, where n is the number of data points and K is the number of parameters fit by the regression. The total sum-of-squares is the sum of the squares of the distances from a horizontal line through the mean of all Y values. Since it only has one parameter (the mean), the degrees of freedom equals n-1.

When K=1, adjusted R^2 and the ordinary R^2 are identical. When K>1, The adjusted R^2 is smaller than the ordinary $R^{2.}$

Using adjusted R^2 and a quick and dirty way to compare models

A quick and easy way to compare models would seem to be to choose the one with the smaller adjusted R^2. Choose to report this value on the Diagnostics tab[347].

Comparing models with adjusted R^2 is a standard method for comparing models fit with multiple linear regression. It is not standard for nonlinear regression for good reason. Speis and Neumeyer(1) did simulations that showed that using the adjusted R^2 is a horrible way to choose among competing models. We suggest that you use the extra-sum-of-square F test[411] or comparing AICc[417] instead. If you do compare models by comparing adjusted R^2, make sure that identical data, weighted identically, are used for all fits. But even then, know that models that fit very differently as assessed by AICc may have R^2 values that differ only in the third to fifth digit after the decimal (1).

Adjusted R^2 in linear regression

Prism doesn't report the adjusted R^2 with linear regression, but you can fit a straight line with nonlinear regression.

If X and Y are not linearly related at all, the best fit slope is expected to be 0.0. If you analyzed many randomly selected samples, half the samples would have a slope that is positive and half the samples would have a negative slope. But in all these cases, R^2 would be positive (or zero). R^2 can never be negative (unless you constrain the slope or intercept so it is forced to fit worse than a horizontal line). In contrast, the adjusted R^2 can be negative. If you analyzed many randomly selected samples, you'd expect the adjusted R^2 to be positive in half the samples and negative in the other half.

Here is a simple way to think about the distinction. The R^2 quantifies the linear relationship in the sample of data you are analyzing. Even if there is no underlying relationship, there almost certainly is some relationship in that sample. The adjusted R^2 is smaller than R^2 and is your best estimate of the degree of relationship in the underlying population.

1.   Spiess, A.-N. & Neumeyer, N. An evaluation of R2 as an inadequate measure for nonlinear models in pharmacological and biochemical research: a Monte Carlo approach. BMC Pharmacol 10, 6–6 (2010).

4.5.4.2.7 Comparing fits with ANOVA

What if you wish to test for differences in best-fit parameters between three or more data sets, on the basis of a single experiment? You don't just want to know if all the curves are the same. You want to use multiple comparisons to compare pairs of curves, focusing on a particular parameter. Prism lets you do this in two ways:

ANOVA approach -- statistical significance

1. Perform the nonlinear regression analysis. Record from the Results sheet the best-fit values for the parameter you are comparing , perhaps the logEC50 of a dose response curve.

2. Also record the standard errors for those parameters and the degrees of freedom for each curve (which equals the number of data points minus the number of variables fit).

3. Create a new Grouped table in Prism, formatted for entry of "Mean, Standard Error, N". You will enter values only into the first row of this table.

4. For each data set, enter the best-fit value of the parameter (i.e. logEC50) in the "Mean"column.

5. Enter the standard error of the best-fit value in the "SEM" column.

6. For N, enter one more than the degrees of freedom for that fit. (Why enter df+1 into the "N" column? The ANOVA calculations don't actually care about the value of N. Instead, they are based on df. Prism subtracts 1 from the value you enter as N, and uses that as dfF. Since you enter df+1, Prism ends up using the correct df value.)

7. Click Analyze and choose one-way ANOVA along with an appropriate multiple comparisons test.

Compare two curves at a time with nonlinear regression

You can rerun the analysis comparing two data sets (curves) at a time. The easiest way to do this is to duplicate the results of the main analysis (New..Duplicate sheet) and then remove all but two data sets from that new analysis. Another approach is to keep one data table, click Analyze, choose nonlinear regression and on the right side of that dialog choose which two data sets to compare.

There are two approaches to use when comparing fits, the extra sum-of-squares F test and the AICc approach.

With statistical significance (extra sum-of-squares F test) approach, there is a traditional (albeit totally arbitrary) cutoff at P=0.05. But if you are doing many comparisons, you should correct for the multiple comparisons. Divide 0.05 (or whatever overall value you want) by the number of pairs of analyses you are comparing, to come up with a new stricter cut off for declaring a P value to be small enough that you can call the comparison "significant".

The AIC approach to comparing curves is not based on statistical hypothesis testing, and is not confused by multiple comparisons. There are two ways to use this approach:

- In the Diagnostics tab of nonlinear regression of Prism, check the option to report the AICc of each curve.Then you can do manual calculations or comparisons with those AICc values.

- Run the nonlinear regression with two data sets at a time, and use the AIC approach to ask how strong the evidence is that the parameter you care about (logEC50) differs between data sets. The AIC calculations just give you the likelihood that the parameter is the same in both data sets vs. different. You need to decide when those likelihood are far enough apart that you will believe the parameters are different.

4.5.4.3 Analysis checklists: Nonlinear regression

To understand your results, review the appropriate analysis checklist, depending on your experimental goal.

- Analysis checklist: Fitting a model 422
- Analysis checklist: Comparing nonlinear fits 426
- Analysis checklist: Interpolating from a standard curve 428

4.5.4.3.1 Analysis checklist: Fitting a model

Your approach in evaluating nonlinear regression depends on your goal.

In many cases, your goal is to create a standard curve from which to interpolate unknown values. We've created a different checklist 428 for this purpose.

More often, your goal is to determine the best-fit values of the model. If that is your goal, here are some questions to ask yourself as you evaluate the fit:

Curve

✓ Does the graph look sensible?

Your first step should be to inspect a graph of the data with superimposed curve. Most problems can be spotted that way.

✓ Does the runs or replicate test tell you that the curve deviates systematically from the data?

The runs 389 and replicates 390 tests are used to determine whether the curve follows the trend of your data. The runs test is used when you have single Y values at each X. It asks if data points are clustered on either side of the curve rather than being randomly scattered above and below the curve. The replicates test is used when you have replicate Y values at each X. It asks if the points are 'too far' from the curve compared to the scatter among replicates.

If either the runs test or the replicates test yields a low P value, you can conclude that the curve doesn't really describe the data very well. You may have picked the wrong model, or applied invalid constraints.

Parameters

✓ Are the best-fit parameter values plausible?

When evaluating the parameter values reported by nonlinear regression, check that the results are scientifically plausible. Prism doesn't 'know' what the parameters mean, so can report best-fit values of the parameters that make no scientific sense. For example, make sure that parameters don't have impossible values (rate constants simply cannot be negative). Check that EC50 values are within the range of your data. Check that maximum plateaus aren't too much higher than your highest data point.

If the best-fit values are not scientifically sensible, then the results won't be useful. Consider constraining the parameters to a sensible range, and trying again.

✓ How precise are the best-fit parameter values?

You don't just want to know what the best-fit value is for each parameter. You also want to know how certain that value is. Therefore an essential part of evaluating results from nonlinear regression is to inspect the 95% confidence intervals for each parameter.

If all the assumptions of nonlinear regression are true, there is a 95% chance that the interval contains the true value of the parameter. If the confidence interval is reasonably narrow, you've accomplished what you wanted to do – found the best fit value of the parameter with reasonable certainty. If the confidence interval is really wide, then you've got a problem. The parameter could have a wide range of values. You haven't nailed it down. How wide is 'too wide' depends on the scientific context of your work.

✓ Are the confidence bands 'too wide'?

Confidence bands visually show you how precisely the parameters have been determined. Choose to plot confidence bands by checking an option on the Fit tab of the nonlinear regression dialog. If all the assumptions of nonlinear regression have been met, then there is a 95% chance that the true curve falls between these bands. This gives you a visual sense of how well your data define the model.

Residuals

✓ Does the residual plot look good?

A residual plot shows the relationship between the X values of your data and the distance of the point from the curve (the residuals). If the assumptions of the regression are met, the residual plot 361 should look bland, with no trends apparent.

✓ Does the scatter of points around the best-fit curve follow a Gaussian distribution?

Least squares regression is based on the assumption that the scatter of points around the curve follows a Gaussian distribution. Prism offers three normality tests (in the Diagnostics tab) that can test this assumption (we recommend the D'Agostino test). If the P value for a normality test is low, you conclude that the scatter is not Gaussian.

✓ Could outliers be impacting your results?

The presence of one or a few outliers (points much further from the curve than the rest) can overwhelm the least-squares calculations and lead to misleading results.

You can spot outliers by examining a graph (so long as you plot individual replicates, and not mean and error bar). But outliers can also be detected automatically. GraphPad has developed a new method for identifying outliers we call the ROUT method. Check the option on the diagnostics tab to count the outliers, but leave them in the calculations. Or check the option on the Fit tab to exclude outliers from the calculations.

Models

✓ Would another model be more appropriate?

Nonlinear regression finds parameters that make a model fit the data as closely as possible (given some assumptions). It does not automatically ask whether another model might work better.

Even though a model fits your data well, it may not be the best, or most correct, model. You should always be alert to the possibility that a different model might work better. In some cases, you can't distinguish between models without collecting data over a wider range of X. In other cases, you would need to collect data under different experimental conditions. This is how science moves forward. You consider alternative explanations (models) for your data, and then design experiments to distinguish between them.

✓ If you chose to share parameters among data sets, are those datasets expressed in the same units?

Global nonlinear regression (any fit where one or more parameter is shared among data sets) minimizes the sum (over all datasets) of the sum (over all data points) of the squared distance between data point and curve. This only makes sense if the Y values for all the datasets are expressed in the same units.

Goodness of fit

✓ Is the R^2 'too low' compared to prior runs of this experiment?

While many people look at R^2 first, it really doesn't help you understand the results very well. It only helps if you are repeating an experiment you have run many times before. If so, then you know what value of R^2 to expect. If the R^2 is much lower than expected, something went wrong. One possibility is the presence of outliers.

✓ Are the values of sum-of-squares and sy.x 'too low' compared to prior runs of this experiment?

These values are related to the R2, and inspecting the results can only be useful when you have done similar experiments in the past so know what values to expect.

4.5.4.3.2 Analysis checklist: Comparing nonlinear fits

Approach to comparing models

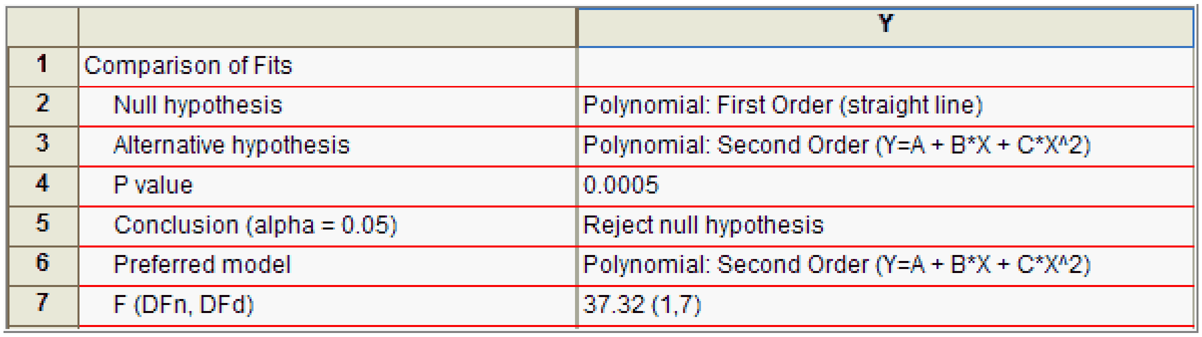
Which model is 'best'? At first, the answer seems simple. The goal of nonlinear regression is to minimize the sum-of-squares, so the model with the smaller sum-of-squares clearly fits best. But that is too simple.

The model with fewer parameters almost always fits the data worse. When you compare two models, the simpler model (with fewer parameters) will have a higher sum-of-squares. When you compare datasets, the global fit will have a higher sum-of-squares than the sum of the individual fits. And when you compare fits with and without a parameter constrained to a particular value, will have a higher sum-of-squares than a model where you don't constrain the parameter.

Both the F test and the AIC method take into account both the difference in goodness-of-fit but also the number of parameters. The question is whether the decrease in sum of squares (moving from simpler to more complicated model) is worth the 'cost' of having more parameters to fit. The F test and AICc methods 78 approach this question differently.

How Prism reports model comparison

Prism reports the results for the comparison at the top of the nonlinear regression table of results. It clearly states the null and alternative hypotheses. If you chose the F test, it reports the F ratio and P value. If you chose Akaike's method, it reports the difference in AICc and the probability that each model is correct.



		Y
1	Comparison of Fits	
2	Null hypothesis	Polynomial: First Order (straight line)
3	Alternative hypothesis	Polynomial: Second Order (Y=A + B*X + C*X^2)
4	P value	0.0005
5	Conclusion (alpha = 0.05)	Reject null hypothesis
6	Preferred model	Polynomial: Second Order (Y=A + B*X + C*X^2)
7	F (DFn, DFd)	37.32 (1,7)

Finally, Prism reports the 'preferred' model. You should understand how Prism decides which model is 'preferred' as you may 'prefer' the other model.

If you chose the extra sum-of-squares F test, then Prism computes a P value that answers this question:

> If the null hypothesis is really correct, in what fraction of experiments (the size of yours) will the difference in sum-of-squares be as large as you observed or larger?

In the Compare tab, you also tell Prism which P value to use as the cut off (the default is 0.05). If the P value is lower than that value, Prism chooses the more complicated model. Otherwise it chooses the simpler model.

If you chose Akaike's method, Prism chooses the model that is more likely to be correct. But you should look at the two probabilities. If they are similar in value, then the evidence is not persuasive and both models fit pretty well.

Analysis checklist for comparing models

✓ Was it possible to fit both models?

Before running the extra sum-of-squares F test or computing AICc values, Prism first does some common sense comparisons. If Prism is unable to fit either model, or if either fit is ambiguous 437 or perfect 443, then Prism does not compare models.

✓ Are both fits sensible?

Apply a reality check before accepting Prism's results. If one of the fits has results that are scientifically invalid, then accept the other model regardless of the results of the F test or AIC comparison.

For example, you would not want to accept a two-phase exponential model if the magnitude of one of the phases is only a tiny fraction of the total response (especially if you don't have many data points). And you wouldn't want to accept a model if the half-life of one of the phases has a value much less than your first time point or far beyond your last time point.

Analysis checklist for the preferred model

Before accepting the 'preferred' model, you should think about the same set of questions as you would when only fitting one model. You can consult the detailed checklist 422 or the abbreviated version below.

✓ How precise are the best-fit parameter values?

✓ Are the confidence bands 'too wide'?

✓ Does the residual plot look good?

✓ Does the scatter of points around the best-fit curve follow a Gaussian distribution?

4.5.4.3.3 Analysis checklist: Interpolating from a standard curve

Your approach in evaluating nonlinear regression depends on your goal.

In many cases, your goal is to learn from the best-fit values. If that is your goal, view a different checklist 422.

If your goal is to create a standard curve from which to interpolate unknown values, your approach depends on whether this is a new or established assay.

Established assay

If the assay is well established, then you know you are fitting the right model and know what kind of results to expect. In this case, evaluating a fit is pretty easy.

✓ **Does the curve go near the points?**

✓ **Is the R2 'too low' compared to prior runs of this assay?**

If so, look for outliers, or use Prism's automatic outlier detection.

✓ **Are the confidence bands too wide?**

The confidence bands[352] let you see how accurate interpolations will be, so we suggest always plotting prediction bands when your goal is to interpolate from the curve. If your are running an established assay, you know how wide you expect the prediction bands to be.

New assay

With a new assay, you also have to wonder about whether you picked an appropriate model.

✓ **Does the curve go near the points?**

Look at the graph. Does it look like the curve goes near the points.

✓ **Are the prediction bands too wide?**

How wide is too wide? The prediction bands show you how precise interpolations will be. Draw a horizontal line somewhere along the curve, and look at the two places where that line intercepts the prediction bands. This will be the confidence interval for the interpolation.

✓ **Does the scatter of points around the best-fit curve follow a Gaussian distribution?**

Least squares regression is based on the assumption that the scatter of points around the curve follows a Gaussian distribution. Prism offers three normality tests (in the Diagnostics tab) that can test this assumption (we recommend the D'Agostino test). If the P value for a normality test is low, you conclude that the scatter is not Gaussian.

✓ Could outliers be impacting your results?

Nonlinear regression is based on the assumption that the scatter of data around the ideal curve follows a Gaussian distribution. This assumption leads to the goal of minimizing the sum of squares of distances of the curve from the points. The presence of one or a few outliers (points much further from the curve than the rest) can overwhelm the least-squares calculations and lead to misleading results.

You can spot outliers by examining a graph (so long as you plot individual replicates, and not mean and error bar). But outliers can also be detected automatically. GraphPad has developed a new method for identifying outliers we call the ROUT method. check the option on the diagnostics tab to count the outliers, but leave them in the calculations. Or check the option on the Fit tab to exclude outliers from the calculations.

✓ Does the curve deviate systematically from the data?

If either the runs test or the replicates test yields a low P value, then you can conclude that the curve doesn't really describe the data very well. You may have picked the wrong model.

4.5.4.4 Error messages from nonlinear regression

When Prism hits a problem, it will present a very brief message in the top row of the results tables. Read the pages in this section to learn what these messages mean.

"Bad initial values" 435

"Interrupted" 435

"Not converged" 437

"Ambiguous" 437

"Hit constraint" 441

"Don't fit" 442

"Too few points" 442

"Perfect fit" 443

"Impossible weights" 443

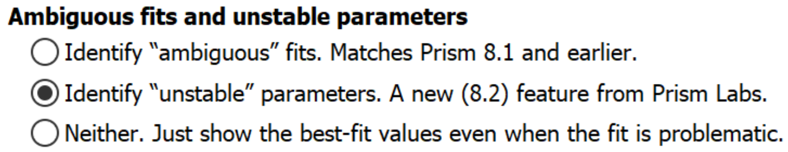
"Equation not defined" 443

4.5.4.4.1 Messages that apply to parameters or confidence limits

Enter topic text here.

4.5.4.4.1.1 "Unstable"

You have a choice, in the Confidence tab 343, to either mark entire fits as "ambiguous 437", to mark identified parameters as "unstable", or neither. The option to mark parameters as "unstable" is a new (in 8.2) feature from Prism Labs.



Ambiguous fits and unstable parameters

○ Identify "ambiguous" fits. Matches Prism 8.1 and earlier.

● Identify "unstable" parameters. A new (8.2) feature from Prism Labs.

○ Neither. Just show the best-fit values even when the fit is problematic.

The designation of "unstable" is based on the matrix manipulations that make nonlinear regression work. But it boils down to the fact that you can change the value of that parameter a bit and the goodness of fit will barely change. A wide range of values for unstable parameters would lead to essentially the same curve. The data simply don't define all the parameters in the model (at least not with the constraints and initial values you chose). Accordingly, the best-fit value determined by Prism is a bit arbitrary. If you chose to identify unstable parameters, Prism will display the word "Unstable" in place of the best fit value for any parameter meeting this criteria, and will display "Very wide" in place of the parameter's confidence interval.

If you really want to know the value that Prism determined as best-fit (even though it is unstable), go back the the Confidence tab and tell Prism to not report if parameters are unstable or ambiguous. Then it will give you the "best-fit" value with no warning.

Note that while the designation of "ambiguous" applies to an entire fit, the label "unstable" is applied to one or more parameters. For those that want to know more of the details, read about how Prism identifies unstable parameters 114.

4.5.4.4.1.2 "Very wide"

You have a choice, in the Confidence tab 343, to either mark entire fits as "ambiguous 437", to mark selected parameters as "unstable", or neither. The option to mark parameters as "unstable" is a new (in 8.2) feature from Prism Labs.

If you choose "unstable", Prism will identify any parameters that are unstable and write "(Very wide)" in place of the confidence intervals for those parameters.

If you choose "ambiguous", if the entire fit is ambiguous, all confidence intervals will be replaced with "(Very wide)".

4.5.4.4.1.3 Question marks "???"

A confidence interval consists of two confidence limits. In some cases, Prism can only determine one confidence limit and reports the other limit as "???".

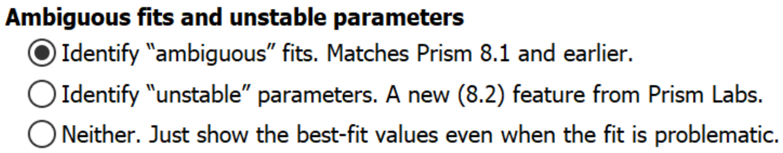
If the upper limit is shown this way, you can say with 95% (or whatever confidence level you chose) that the parameter is greater than the lower limit, but cannot say anything about how large the parameter could be.

If the lower limit is shown this way, you can say with 95% (or whatever confidence level you chose) that the parameter is smaller than the upper limit, but cannot say anything about how small the parameter could be.

It's important to note that Prism reports "???" for one or both confidence limits when the calculation to determine these values fails for any reason. Thus, the most appropriate interpretation of this result is that the confidence limit replaced by "???" is undefined.

4.5.4.4.1.4 Tilde "~"

You have a choice, in the Confidence tab 343, to either mark entire fits as "ambiguous 437", to mark selected parameters as "unstable", or neither. The option to mark parameters as "unstable" is a new (in 8.2) feature from Prism Labs.



Ambiguous fits and unstable parameters

● Identify "ambiguous" fits. Matches Prism 8.1 and earlier.

○ Identify "unstable" parameters. A new (8.2) feature from Prism Labs.

○ Neither. Just show the best-fit values even when the fit is problematic.

If you choose the older approach to identify ambiguous fits, Prism places a tilde ("~") in front of the parameters that have super high dependencies. The parameter value is what is used to calculate the Y values for the fit curve. But with super high (>0.9999) dependency, you could get a curve that fits just as well with very different parameter values. These values that are preceded with tildes are really not useful.

4.5.4.4.1.5 Incomplete Confidence Intervals

Occasionally when calculating asymmetric (profile likelihood) confidence intervals for parameters in nonlinear regression, Prism is unable to calculate a "complete" interval. This issue can take a number of forms:

- Prism could calculate a lower limit to the confidence interval, but not an upper limit

- Prism could calculate an upper limit to the confidence interval, but not a lower limit

- Prism could not calculate a lower or an upper limit to the confidence interval

Remember that a 95% confidence interval is a range of values within which you can be 95% certain that the true value can be found. How does this apply to each of the situations above?

- Calculated lower, but no upper limit: you can be 95% certain that the "true" parameter value is greater than the calculated lower confidence limit. However, you have no real certainty about how large this value could be

- Calculated upper, but no lower limit: you can be 95% certain that the "true" parameter value is smaller than the calculated upper confidence limit. However, you have no real certainty about how small this value could be

- No upper or lower limit: you really can't have any certainty about the "true" parameter value at all. Based on your data, the given parameter estimate is the best you can do, but you can't be certain at all just how big or small the "true" parameter value is.

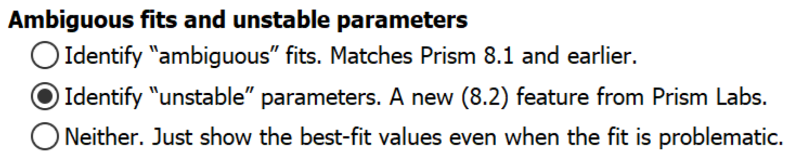
It’s important to note that the inability to report a confidence interval is rare, and occasionally it’s possible to calculate a wider confidence interval when a narrower confidence interval cannot be reported (for example, obtaining a 90% confidence interval when a 95% confidence interval cannot be reported). There are a number of different ways that Prism will indicate that a confidence interval is incomplete, and they have subtly different meanings. Read more about ways that incomplete confidence intervals can be displayed on these specific error pages:

Confidence intervals reported as "Very wide" 432

Confidence intervals reported using Question marks "???" 432

Choices in the Nonlinear Regression parameters dialog that affect Confidence Intervals

You have a choice, in the Confidence tab 343, to either mark entire fits as "ambiguous 437", to mark identified parameters as "unstable", or neither. The option to mark parameters as "unstable" is a new (in 8.2) feature from Prism Labs.



Ambiguous fits and unstable parameters

○ Identify “ambiguous” fits. Matches Prism 8.1 and earlier.

● Identify “unstable” parameters. A new (8.2) feature from Prism Labs.

○ Neither. Just show the best-fit values even when the fit is problematic.

Note that if you choose to identify "ambiguous" fits, Prism may identify one or more parameters with a tilde (~). When this happens, Prism simply will not attempt to calculate a confidence interval, and will report "Very wide" instead.

4.5.4.4.1.6 "Not used"

In some cases, user-defined models can fit different parameters to different data sets in the same data table. When this happens, Prism will report each parameter that was fit for at least one data set on its own row. If the model specified that a given parameter not be fit to a specific data set, the estimate for that data set will be replaced by the text "Not used".

4.5.4.4.2 Messages that apply to entire fits

Enter topic text here.

4.5.4.4.2.1 "Bad initial values"

What does "bad initial values" mean?

Nonlinear regression works iteratively. Prism starts with initial estimated values for each parameter. It then gradually adjusts these until it converges on the best fit. In rare circumstances, Prism cannot compute a curve based on the initial values you entered. When it tries, it encounters a math problem such as division by zero or taking the logarithm of a negative number. In this case, Prism does not report any results and instead reports "bad initial values".

Checklist

✓ Did you enter or choose the right equation?

✓ Did you enter sensible values for initial values?

✓ Is the range of X values invalid? Focus on the first and last X value (and on X=0 if included in the range). Can the equation be evaluated at those X values with the initial values you entered?

4.5.4.4.2.2 "Interrupted"

What does 'interrupted' mean?

Prism reports 'interrupted' in two situations:

- Nonlinear regression is iterative and the maximum number of iterations is specified in the Diagnostics tab, with 1000 being the default. With data that doesn't really define all the parameters in the model, Prism can just keep iterating until it hits the limit. In that case it stops and reports "interrupted". You can lift the limit in the Diagnostics tab of the nonlinear regression dialog, but usually the problem is that the data simply doesn't enough information for the model to fit.

- The fit was slow, and you clicked "Interrupt" on the progress dialog.

Checklist

If the maximum number of iterations was set to a low value, set it to a higher value and try again. If you have lots of data points and lots of parameters, nonlinear regression can sometimes require hundreds of iterations.

If the maximum number of iterations was already set to a high value, you can try a still higher value, but most likely Prism is still not going to be able to find a best-fit curve. Things to check:

✓ Did you enter the right model?

✓ Does your data provide enough information to define that model. For example if you fit a log(dose) response curve, does your data show a sigmoidal shape, enough to define the top or bottom plateaus? If your data doesn't define all the parameters, consider constraining one or more parameters to constant values.

✓ Does the curve defined by your initial values come near your data? Check the option on the diagnostics tab 347 to plot that curve.

✓ If you entered constraints, were they entered correctly?

✓ If you didn't enter any constraints, consider whether you can constrain one or more parameters to a constant value? For example, in a dose-response curve can you constrain the bottom plateau to be zero?

✓ Can you share a parameter over all the data sets (global fitting)?

4.5.4.4.2.3 "Not converged"

What does "not converged" mean?

Nonlinear regression works iteratively. Prism starts with initial estimated values for each parameter. It then gradually adjusts these until it converges on the best fit. "Converged" means that any small change in parameter values creates a curve that fits worse (higher sum-of-squares). But in some cases, it simply can't converge on a best fit, and gives up with the message 'not converged'. This happens in two situations:

- The model simply doesn't fit the data very well. Perhaps you picked the wrong model, or applied the wrong constraints.

- The initial values generated a curve that didn't come close to the points. In that case, Prism may not be able to figure out how to change the parameters to make the curve fit well.

Checklist

✓ Did you enter the right model?

✓ Does the curve defined by your initial values come near your data? Check the option on the diagnostics tab 347 to plot that curve.

✓ If you entered constraints, were they entered correctly?

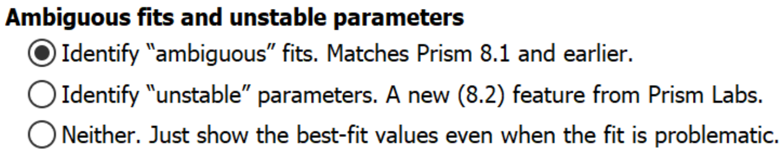
✓ If you didn't enter any constraints, consider whether you can constrain one or more parameters to a constant value? For example, in a dose-response curve can you constrain the bottom plateau to be zero?

✓ Can you share a parameter over all the data sets (global fitting)?

4.5.4.4.2.4 "Ambiguous"

An alternative to the concept of "ambiguous" fits

Starting with Prism 8.2, we offer a number of choices regarding how Prism should handle cases when the chosen model isn't fully defined by the available data. These options are to identify "ambiguous" fits, to identify "unstable" parameters, or to do neither. Prior to Prism 8.2, Prism would always identify "ambiguous" fits. We've found that the option to identify "unstable" parameters seems to work better, but we offer these all of these options as choices, so you can get results that match prior versions. The choice is in the Confidence tab[343] of the nonlinear regression dialog.



Ambiguous fits and unstable parameters

● Identify "ambiguous" fits. Matches Prism 8.1 and earlier.

○ Identify "unstable" parameters. A new (8.2) feature from Prism Labs.

○ Neither. Just show the best-fit values even when the fit is problematic.

What does "ambiguous" mean?

"Ambiguous" is a term coined by GraphPad to describe a fit that doesn't really nail down the values of all the parameters.

Changing the value of any parameter will always move the curve further from the data and increase the sum-of-squares. But when the fit is 'ambiguous', changing other parameters can move the curve so it is near the data again. In other words, many combinations of parameter values lead to curves that fit equally well.

If you check the option in the diagnostics tab to report dependency, the value will be >0.9999 for all the 'ambiguous' parameters (that is how we define 'ambiguous'; the threshold value is arbitrary). If you look at the covariance matrix, most likely some of the values will be very close to 1.0 or -1.0.

If the fit is 'ambiguous' you really can't interpret the best-fit values of some parameters.

How does Prism report ambiguous results?

Prism puts the word 'ambiguous' in the top row of results. For the parameters that are 'ambiguous' (there may be one or several), Prism puts '~' before the best fit values and standard errors, and reports "very wide" for the corresponding confidence intervals.

We no longer recommend this approach because it turns out that a parameter's value can be unstable in cases where it is still possible for Prism to compute useful confidence limits.

Does it matter?

If your goal is to interpolate unknowns from a standard curve, you won't care that the parameter values are 'ambiguous'. So long as the curve goes through the points, and doesn't wiggle too much, the interpolations will be useful.

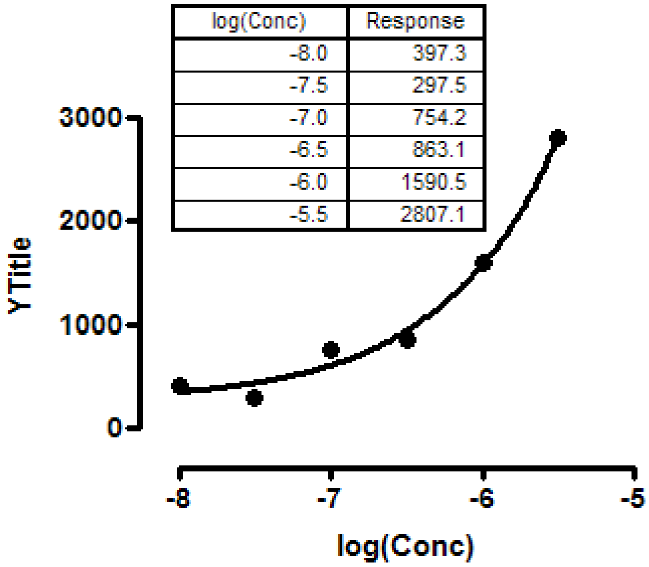
If your goal is to learn about your data by inspecting the values of the parameters, then you've got a real problem. At least one of the parameters has a best-fit value that you should not rely upon.

But the R^2 is really high...

In many cases, the R^2 will be really high, maybe 0.99. That just means that the curve comes close to the data points. It doesn't mean the data define all the parameters. If the fit is 'ambiguous', you can get an equally well-fitting curve with a different set of values of the parameters.

Reasons for ambiguous fits

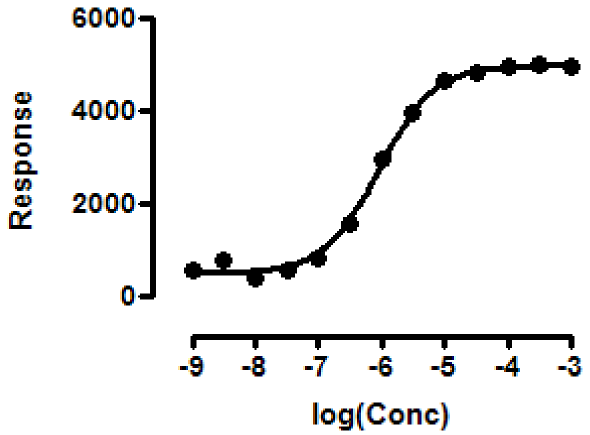
Data not collected over a wide enough range of X values



log(Conc)	Response
-8.0	397.3
-7.5	297.5
-7.0	754.2
-6.5	863.1
-6.0	1590.5
-5.5	2807.1

The data above show a fit of a dose-response curve to a set of data that don't define a top plateau. Since the top plateau was not constrained to a constant value, Prism reports the fit to be ambiguous.

Model too complicated for the data



The data above fit fine to a standard dose-response curve. But if you try to fit it to a biphasic dose-response curve, Prism reports that the results are ambiguous. The data follow a standard dose-response model just fine, with no evidence of a second component. So fitting to a biphasic model -- with two EC50 values -- is ambiguous.

Model has redundant parameters

The simplest example would be fitting this model: Y= A + B + C*X. This model describes a straight line with a slope equal to C and a Y intercept equal to the sum of A and B. But there is nothing in the data to fit A and B individually. There are an infinite number of pairs of values of A and B that lead to the same sum, so the same Y intercept. If Prism attempts to fit this model, it will conclude that the fit is ambiguous.

Prism cannot tell the difference between this case and the previous one. But the two cases are distinct. The model in this example has redundant parameters. It doesn't matter what the data look like, or how many data points you collect, fitting this model will always result in 'ambiguous' results, because two of the parameters are intertwined. In the previous example, the fit was ambiguous with the example data set, but would not have been ambiguous with other data sets.

Checklist

✓ Can you constrain any parameters to a constant value?

✓ Can you share one or more parameters among datasets?

✓ Can you collect more data points, perhaps over a wider range of X values?

✓ If your model has two phases or components, consider switching to a one phase or component model.

4.5.4.4.2.5 "Hit constraint"

What does "hit constraint" mean?

Prism reports "Hit constraint" when the best-fit value of one or more parameters is right at the limit of a constraint.

When a fit hits a constraint, the results are unlikely to provide useful information. If you had a solid reason to constrain a parameter within a range of values, it ought to end up in that range. If the fit hit the constraint limit, that means the true best-fit value is some value forbidden by the constraint.

Prism does not compute confidence and prediction bands when a parameter hit a constraint. The best-fit values are not a local minimum, so any attempt to compute confidence or prediction bands would give misleading results. Prism does compute the confidence intervals for the other parameters (the ones that didn't hit a constraint) but these need to be viewed with caution.

When a fit ends up hitting a constraint, it is likely that you set the constraint incorrectly. So the first thing to do is make sure the constraint is sensible and correctly entered. For example, if you mistakenly set a constraint that a rate constant K must be *less than or equal to* zero, Prism's nonlinear regression might converge with the best-fit value equalling zero and report that you hit the constraint. Fix the constrain to be *greater than or equal to* zero, and the fit will work fine.

Another possibility is to change the constraint from an inequality (Bottom>0) to a constant constraint (Bottom=0). You'll get the same parameter values, but difference confidence intervals, and you can get confidence and prediction bands.

Checklist

✓ Did you enter the constraint correctly? Did you mix up "<" and ">".

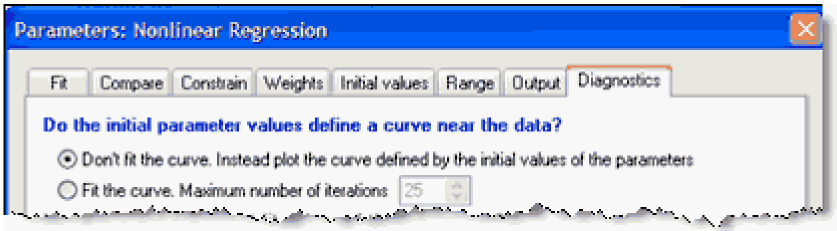
✓ Did you enter the model correctly?

✓ Are you asking Prism to do the impossible -- too many parameters for your data?

✓ If your model has two phases or components, consider switching to a one phase or component model.

4.5.4.4.2.6 "Don't fit"

The top of the Diagnostics tab offers the choice to plot the curve defined by the initial values of the parameters.



Do the initial parameter values define a curve near the data?

● Don't fit the curve. Instead plot the curve defined by the initial values of the parameters

○ Fit the curve. Maximum number of iterations 25

When you make this choice, the top of the results will show "Don't fit".

This is a very useful option to use when diagnosing problem fits. But you won't learn anything by looking at the results page. Instead, look at the graph. If the curve goes near the points, then the initial values are OK. If the curve is far from the points, or follows a different shape altogether, then you know that there is a problem with the choice of model or initial values.

If you checked this option by mistake, go back to the Diagnostics tab and check the alternative choice ("Fit the curve.").

4.5.4.4.2.7 "Too few points"

Regression requires at least as many data points as parameters. So if you fit a dose-response model with four parameters (Bottom, Top, logEC50, and HillSlope), you must have four or more points or Prism will simply report "too few points".

Having as many points as parameters is the bare minimum. Prism will try to fit the model, and will report a perfect fit 443, and so not be able to report standard errors or confidence intervals. If there are only a few more points that parameters, Prism is likely to end up with an ambiguous 437 or not converged 437 fit.

4.5.4.4.2.8 "Perfect fit"

Prism reports "perfect fit' when the curve goes through every point. The sum-of-squares is 0.0, and R^2 is 1.00.

If you are testing nonlinear regression with made up values, add some random scatter to make a better example.

If you are fitting actual data, and the fit is perfect, you either got very lucky or have very few data points. It is not possible to compute the standard errors and confidence intervals of the parameters when the fit is perfect, nor is it possible to compare models.

4.5.4.4.2.9 "Impossible weights"

Prism reports "Impossible weighting" when weighting would lead to division by zero for one or more point. This can happen when you choose to weight by the reciprocal of Y, and the Y value of the curve equals zero at some X values. Since division by zero is impossible, Prism reports "Impossible weighting". Go to the Methods tab and choose a different weighting method.

Prism also reports "impossible weights" when the Y value of the curve is negative for some values of X. If you weight by 1/Y, this would mean the weights would be negative. If you weight by 1/Y2, the weights would all be positive, but points with Y near zero would get lots of weight, and points with Y values far from zero (in either direction) would get low weights, and this doesn't really make sense.

4.5.4.4.2.10 "Equation not defined"

When you enter a user-defined equation, you can write a different model for different data sets 744. If no model is defined for certain data sets, the message "Equation not defined" will appear in the results for that data set, with no other results.

- "Bad initial values" 435

- "Interrupted" 435
- "Not converged" 437
- "Ambiguous" 437
- "Hit constraint" 441
- "Don't fit" 442
- "Too few points" 442
- "Perfect fit" 443
- "Impossible weights" 443
- "Equation not defined" 443

4.5.4.4.2.11 "Can't calculate"

If you ask Prism to use either the F test or the AICc method to compare two models, it will report "Can't calculate" if the two models have the same number of parameters. This means it can't calculate the comparison between fits, not that it can't fit the two models to the data.

Both these methods to compare models only make sense when the models being compared have different numbers of parameters, and so have different numbers of degrees of freedom. These methods are alternative methods of dealing with the tradeoff between how well the model fits and how complicated it is (assessed by the number of parameters).

If you want to compare two models with the same number of parameters, there is no need to use either the F test or AIC. Simply choose the model that fits the data the best with the smallest sum-of-squares.

4.5.5 Models (equations) built-in to Prism

To use nonlinear regression, you need to choose a model. In many cases, especially if you are fitting pharmacological or physiological data, you can choose one of Prism's built-in models described here.

4.5.5.1 Dose-response - Key concepts

What are dose-response curves?

Dose-response curves can be used to plot the results of many kinds of experiments. The X axis plots concentration (or dose) of a drug or hormone. The Y axis plots response, which could be almost any measure of biological function.

The term “dose” is often used loosely. In its strictest sense, the term only applies to experiments performed with animals or people, where you administer various doses of drug. You don't know the actual concentration of drug at its site of action—you only know the total dose that you administered. However, the term “dose-response curve” is also used more loosely to describe *in vitro* experiments where you apply known concentrations of drugs. The term “concentration-response curve” is a more precise label for the results of these types of experiments. The term “dose-response curve” is occasionally used even more loosely to refer to experiments where you vary levels of some other variable, such as temperature or voltage.

X values are logarithm of doses or concentrations

Dose-response experiments typically use around 5-10 doses of agonist, equally spaced on a logarithmic scale. For example, doses might be 1, 3, 10, 30, 100, 300, 1000, 3000, and 10000 nM. When converted to logarithms (and rounded a bit), these values are equally spaced: 0.0, 0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, and 4.0.

All the dose-response curves built in to Prism are provided in two forms. One form assumes that X is the logarithm of concentration or the logarithm of dose. The other form assumes X is concentration or dose. Make sure you pick the form of the equation to match the data you are analyzing.

The uncertainty in the value of the EC50 or IC50 is usually quite asymmetrical. So be sure to choose the profile likelihood method for computing confidence intervals that produces asymmetrical intervals 347.

Y values are responses

In a dose-response curve, the Y values are responses. For example, the response might be enzyme activity, accumulation of an intracellular second messenger, membrane potential, secretion of a hormone, change in heart rate, or contraction of a muscle.

You can transform the Y values to new units by multiplying or dividing by a constant. Use Prism's Transform analysis for this. Transforming to new units will not fundamentally change the results of a curve fit.

In some cases, the transform from experimentally observed units to practical units is nonlinear. For example, a nonlinear transform is needed to convert the ratio of two fluorescence values to concentrations of Ca++. Which Y values should be used when fitting a dose-response curve? Nonlinear regression assumes that all scatter around the curve is Gaussian, so you want to use whatever units make that assumption most true. In many cases, this may be hard to know.

4.5.5.1.2 The terms "agonist", "antagonist", and "normalized" in equation names

While the terms *agonist* and *antagonist* have rigorous meanings in pharmacology, we use the terms more loosely when naming built-in dose-response equations.

We use the term *agonist* to describe curves that go uphill, with a larger Y value as the X value gets larger.

We use the term *antagonist* to describe curves that go downhill, with a smaller Y value as the X value gets larger.

We use the term *normalized* to describe curves that go between Y=0.0 and Y=100.0. Even if you have normalized your data, you may not want to use the normalized equation. One reason not to use the normalized equation is to let Prism fit the top and/or bottom plateaus. Another reason is that with some drugs (a partial agonist) the top plateau won't be 100 (assuming you define 100 based on the response to a different drug).

4.5.5.1.3 Converting concentration to log(concentration)

Two sets o equations

Prism provides two sets of dose-response equations.

- One set assumes you have entered concentrations (or doses) as X.

- The other set assumes you entered the logarithms of concentration or dose.

It matters a lot that you pick an equation that corresponds to your data. Otherwise the fit will not be helpful.

Note that Prism's nonlinear regression "sees" a table of XY values, and fits that. The nonlinear regression part of Prism does not "see" the graph you made. If you entered X values as concentrations, they are still concentrations even if you graph those data on a graph where the X axis uses a logarithmic scale. Stretching the axis to a logarithmic scale does not change the data. The X values are still concentrations, not logarithms of concentrations.

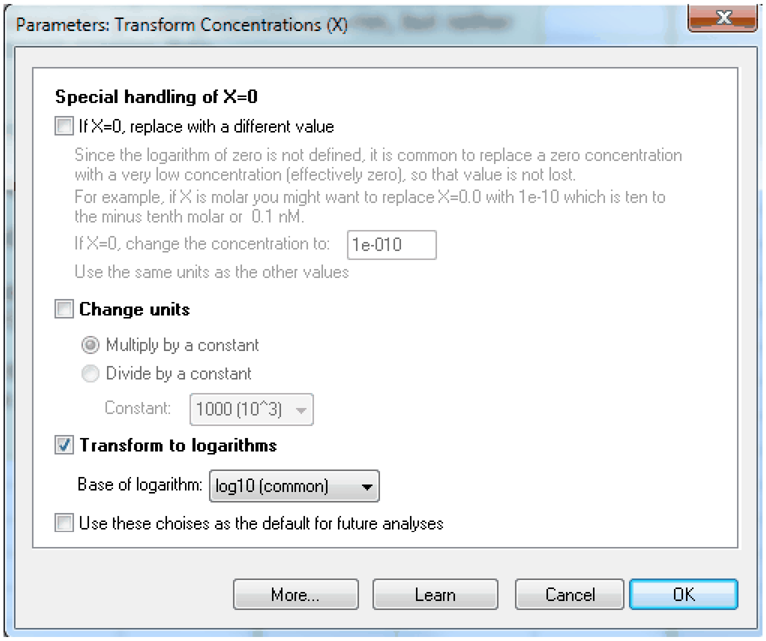
How to transform your X values from concentrations to logarithm of concentrations

If you entered actual doses or concentrations, instead of their logarithms, but want to use an equation that assumes the X values are logarithms, use Prism to transform the X values.

1. From the data table, click Analyze and then choose Transform Concentrations (X), which is the second analysis listed on the Analyze dialog.

2. On the dialog, check the option to transform X to logarithms. Base 10 (common) logarithms are used most commonly.

3. If any of your X values are 0.0, use the option at the top of the dialog to change these to some small concentration. The reason for this is that the logarithm of zero is not defined.



Special handling of X=0

□ If X=0, replace with a different value

Since the logarithm of zero is not defined, it is common to replace a zero concentration with a very low concentration (effectively zero), so that value is not lost.

For example, if X is molar you might want to replace X=0.0 with 1e-10 which is ten to the minus tenth molar or 0.1 nM.

If X=0, change the concentration to: 1e-010

Use the same units as the other values

□ **Change units**

● Multiply by a constant

○ Divide by a constant

Constant: 1000 (10^3)

☑ **Transform to logarithms**

Base of logarithm: log10 (common)

□ Use these choises as the default for future analyses

Using e notation

Don't mix up e notation with logarithms.

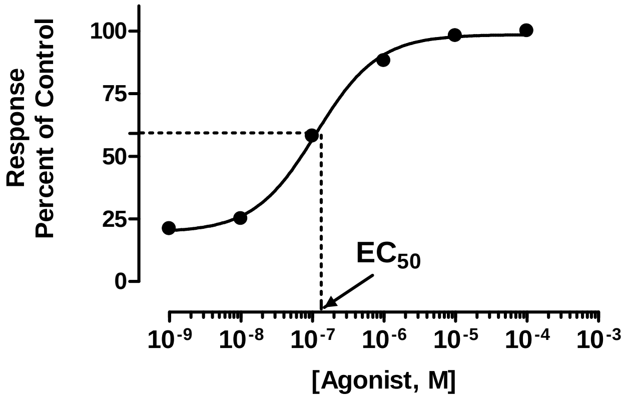
If you enter "1e-9", that is exactly the same as entering "0.000000001". To enter the logarithm, enter "-9".

4.5.5.1.4 The EC50

What is the EC50?

The EC50 is defined quite simply as the concentration of agonist that provokes a response halfway between the baseline (Bottom) and maximum response (Top). It is impossible to define the EC50 until you first define the baseline and maximum response.

Depending on how you have normalized your data, the EC50 may not be the same as the concentration that provokes a response of Y=50. For instance, in the example below, the data are normalized to percentage of maximum response, without subtracting a baseline. The baseline is about 20%, and the maximum is 100%, so the EC50 is the concentration of agonist that evokes a response of about 60% (halfway between 20% and 100%). The concentration that provokes a response of 50 in this experiment is **not** the EC50.



Don't over interpret the EC50. It is simply the concentration of agonist required to provoke a response halfway between the baseline and maximum responses. Because the EC50 defines the location of the dose-response curve for a particular drug, it is the most commonly used measure of an agonist’s potency. However, the EC50 is usually not the same as the Kd for the binding of agonist to its receptor -- it is not a direct measure of drug affinity.

The pEC50

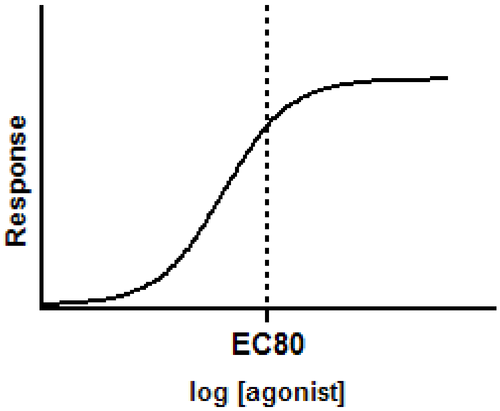
The pEC50 is defined as the negative logarithm of the EC50. If the EC50 equals 1 micromolar (10^{-6} molar), the log EC50 is –6 and the pEC50 is 6. There is no particular advantage to expressing potency this way, but it is customary in some fields.

The IC50

In many experiments, you vary the concentration of an inhibitor. With more inhibitor, the response decreases, so the dose-response curve goes downhill. With such experiments, the midpoint is often called the IC50 ("I" for inhibition) rather than the EC50 ("E" for effective). This is purely a difference in which abbreviation is used, with no fundamental difference.

ECanything

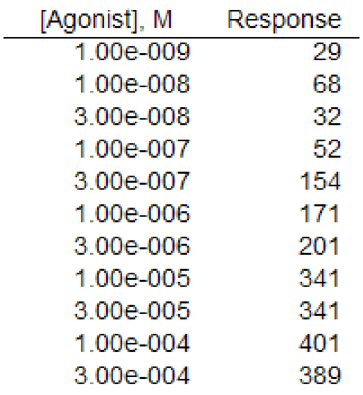
A simple rearrangement of the equation lets you fit EC80 (or EC90 or ECanything) instead of the EC50. Prism includes an equation 550 that fits any EC value directly.



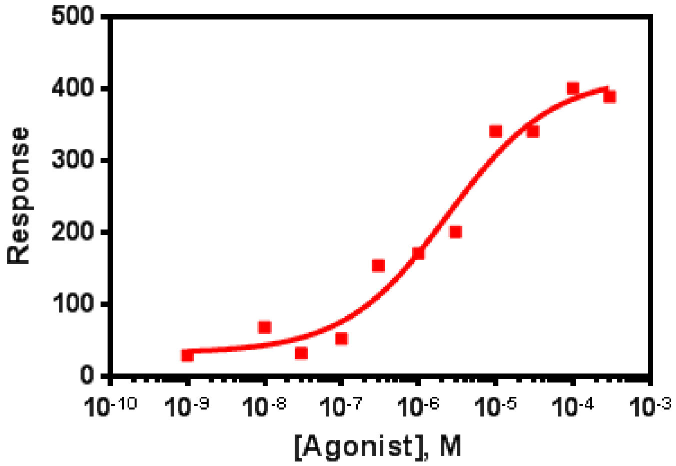
4.5.5.1.5 Confidence intervals of the EC50

The 95% confidence interval of the EC50 is not symmetrical

Sample data



[Agonist], M	Response
1.00e-009	29
1.00e-008	68
3.00e-008	32
1.00e-007	52
3.00e-007	154
1.00e-006	171
3.00e-006	201
1.00e-005	341
3.00e-005	341
1.00e-004	401
3.00e-004	389



Asymmetrical confidence intervals

The sample data above were fit to the model: [Agonist] vs. response -- variable slope 481 with the Confidence tab set to compute confidence intervals using the asymmetrical (likelihood) method which we recommend. The best fit value of the EC50 (multiplied by 10^6 to convert to micromolar) is 2.4 μM with the 95% confidence interval ranging from 0.85 μM to 15.3 μM.

Prism also reports these values on a log scale. The logEC50 (log molar) is -5.62 with a 95% confidence interval ranging from -6.07 to -4.82.

Three things to notice:

- The confidence interval is fairly wide. This is because the data have some scatter and also because we are fitting the EC50 plus three other parameters (the top and bottom plateaus and the slope). If you fix one or more of those parameters to constant values, the confidence interval for the EC50 would get narrower.
- The confidence interval, expressed on a concentration scale, is quite asymmetrical.

- The confidence interval, expressed on a log scale, is close to symmetrical.

An alternative way to fit the data are to transform the X values to their logarithm, and then fit the results of this transform to the model: log[Agonist] vs. response -- variable slope 473 with the Confidence tab set to compute confidence intervals using the asymmetrical (likelihood) method which we recommend using always. You'll get exactly the same results this way.

Symmetrical confidence intervals

Prism offers two methods to compute the confidence intervals. The results above were computed using the new method that produces asymmetrical intervals. If you use other method (the only method offered in prior versions of Prism), the confidence intervals for the parameter you fit are always symmetrical.

If you fit the model that fits the logEC50, the confidence interval ranges from -6.07 to -4.82. The "asymmetrical" confidence interval of the logEC50 shown above is nearly symmetrical, so the symmetrical confidence interval is nearly the same (is the same to three digits).

But what if you fit the EC50 and ask for symmetrical confidence interval. In this case the reported interval ranges from 0 µM to 5.21µM. This seems asymmetrical because the EC50 is contrained to be greater than zero. If you removed that constraint, Prism would report a confidence interval ranging from -0.42 µM to 5.21µM, which is symmetrical around the best-fit value of 2.4 µM. This symmetrical interval does a really poor job of quantifying the precision of the IC50. The simple fact is that the uncertainty is asymmetrical, and a symmetrical interval can be very misleading. Even setting aside the negative portion of the confidence interval, it includes all values from zero on up, so doesn't provide any guidance about how low the EC50 might really be. That is why we recommend that you choose asymmetrical intervals routinely.

Do not transform the standard error of the logEC50

We suggest that you always ask Prism to report the asymmetrical confidence intervals, and not to report the standard error of the parameters. The SE of the parameters does a poor job of quantifying precision. It gets particularly confusing when logarithms get involved. When some people see the SE of the logEC50, they are tempted to convert this to the standard error of the EC50 by taking the antilog. This is invalid.

In the example, the SE of the logEC50 is 0.2158. The antilog of 0.2158 equals 1.644. This value is not the SE of the EC50. It cannot be interpreted as a plus/minus error value. It is a times/divided by error value. To get an equivalent to the logEC50 plus or minus one of the logEC50, find the lower limit by dividing the EC50 by 1.644 and the upper limit by multiplying by 1.644. But this is not commonly done, so it would be confusing if you tried to report results this way.

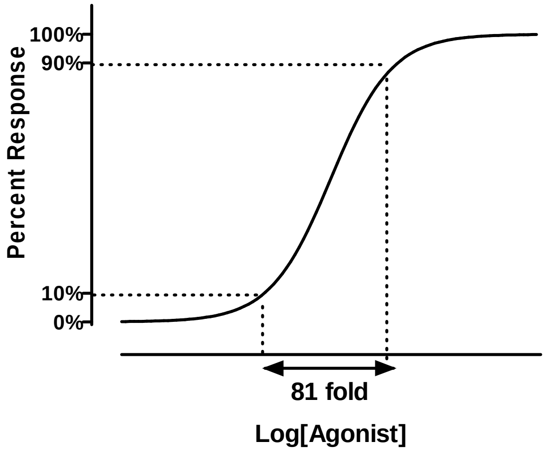
Averaging the EC50 from several experiments

The uncertainty is close to symmetrical when you express the midpoint of a dose-response curve as a logEC50, but is usually far from symmetrical when you express it as the EC50. When pooling several experiments, therefore, it is best to average the logEC50 values, which will give a different result than averaging the EC50 values.

4.5.5.1.6 Hill slope

The usual steepness of a dose-response curve

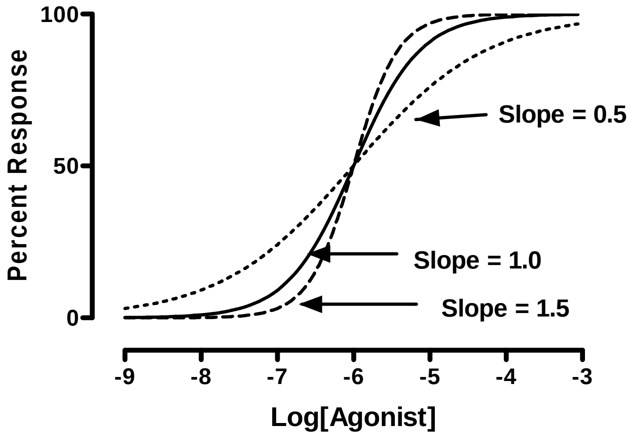
Many dose-response curves follow the shape of a receptor binding curve. As shown below, 81 times more agonist is needed to achieve 90% response than a 10% response.



Since the linkage between agonist binding and response can be very complex, any shape is possible. It seems surprising, therefore, that so many dose-response curves have shapes almost identical to receptor binding curves, even when we know there are multiple steps between binding and measured response. It turns out that no matter how many steps intervene between agonist binding and response, the dose-response curve will have the usual steepness so long as each messenger binds to a single binding site according to the law of mass action.

The slope factor or Hill slope

Some dose-response curves are steeper or shallower than the standard curve. The steepness is quantified by the Hill slope, also called a slope factor. A dose-response curve with a standard slope has a Hill slope of 1.0. A steeper curve has a higher slope factor, and a shallower curve has a lower slope factor.



If you use a single concentration of agonist and varying concentrations of antagonist, the curve goes downhill and the slope factor is negative. The steeper the downhill slope, the more negative the Hill slope.

Standard slope or variable slope?

Because this standard slope is so common, Prism comes with equations with the standard slope built in. The equations that don't have 'variable slope' in their name assume the standard slope (1.0 for stimulation, -1.0 for inhibition).

Deciding whether to fit a model with a standard slope or a variable slope is not easy.

If you have lots of data points (more than a dozen, perhaps lots more), then you can fit the slope by picking a variable slope equation. If you have fewer data points, and a standard system, it makes sense to choose an equation with a standard slope.

4.5.5.1.7 Choosing a dose-response equation

To choose a dose-response model in Prism, you need to answer four questions:

Are the X values concentrations (or doses) or the logarithm of concentration or dose?

Prism offers two sets of equations for dose-response curves. One set assumes X is the logarithm of dose or concentration. The other set assumes X is the dose or concentration. Be sure to pick an equation that matches the data.

Stimulation or inhibition?

Prism offers one set of dose-response equations for stimulation and another set for inhibition. The inhibitory equations are set up to run downhill. The only difference is that the inhibitory equations fit the IC50 ("I" for inhibition) while the stimulation equations fit the EC50 ("E" for effective).

If the curve goes up hill, choose from the set of stimulation equations. If the curve goes down hill, choose from the set of inhibition equations.

Standard slope or variable slope?

Prism offers equations using a standard slope, which have a Hill slope 454 of 1.00 (for stimulation) or -1.00 (for inhibition), and variable slope (fit by Prism). The equations that don't have 'variable slope' in their name assume the standard slope. The standard slope is expected when measuring binding of a ligand to a receptor where there is no heterogeneity or cooperativity. But it turns out that many other log(dose) vs. response curves have the same standard slope.

If your data has more than a few concentrations that lead to a response between say 10 and 90%, then you can ask Prism to fit the slope. If your data provide only one or two concentrations that have a response between 10% and 90%, then your data don't really provide information to define the slope and you'll probably need to choose a model with a fixed slope.

The choice is not straightforward, and there are many situations where it is not clear which approach is better.

Normalized or not?

If your data have been normalized so the curve runs from Y= 0 to Y=100, you may wish to choose a normalized model. These models don't fit the bottom and top plateaus, but rather force the bottom plateau to equal 0 and the top plateau to equal 100. Only choose a 'normalized response' equation when you have determined the values that define 0 and 100 very precisely. Just because the data have been normalized doesn't mean to have to constrain the curve in that way. 458

There are many situations where it is not clear whether it makes sense to use a normalized model or not. It is not always a straightforward decision.

Special dose-response models

Prism has two sets of special models used for special dose-response situations. One set designed to accept X entered as concentration values and one set designed to accept X entered as log(concentration) values:

Dose-Response -- Special (X is Concentration)

Asymmetrical (five parameter) 502

Biphasic dose-response 505

Bell-shaped dose-response 507

Operational model - Depletion 509

Operational model - Partial agonist 512

Gaddum/Schild EC50 shift 515

Allosteric EC50 shift 521

ECanything 524

EC50 shift 519

Absolute IC50 526

Dose-Response -- Special (X is Log[Concentration])

Asymmetrical (five parameter) 529

Biphasic dose-response 532

Bell-shaped dose-response 534

Operational model - Depletion 535

Operational model - Partial agonist 538

Gaddum/Schild EC50 shift 541

Allosteric EC50 shift 547

ECanything 550

EC50 shift 545

Absolute IC50 552

4.5.5.1.8 Pros and cons of normalizing the data

The dose-response model has four parameters: the bottom plateau, the top plateau, the EC50, and the slope factor (which is often constrained to a standard value).

The main goal of fitting the dose-response curve in many situations is to determine the best-fit value of the EC50, which is the concentration that provokes a response halfway between the top and bottom plateaus. If those plateaus are not well defined, the EC50 will be very uncertain. Think of it this way: If you have not defined "100" and "0" very precisely, you also have not defined "50" precisely, and therefore cannot determine the EC50 precisely. If your data doesn't really define the plateaus and you don't have any control data that assesses the top and bottom plateau, then there is no way to determine the EC50 precisely.

Many experiments include controls to measure the maximum and minimum possible response. There are two ways to include these controls in your analysis:

- Constrain the Top and Bottom to values determined from your controls. Then only fit the EC50 and slope.

- Normalize your data so responses run from 0 to 100, using your controls to define 0 and 100 using Prism's Normalize analysis. Then use nonlinear regression to fit a "normalized response" model. These models don't fit the bottom and top plateaus, but rather force the bottom plateau to equal 0 and the top plateau to equal 100.

- Include all the data in the fit. For the control defined by the absence of drug, enter a very low concentration as X, perhaps two orders of magnitude less than the smallest concentration you actual use. For the nonspecific control defined by another drug, enter a large concentration, perhaps two orders of magnitude larger than your largest concentration. Now all the data are in one table, with sensible X values. Use nonlinear regression to fit all four parameters (Bottom, Top, EC50 and slope). The blank and nonspecific data help define the curve, but don't overwhelm the calculations as they do with the first two methods listed above. Weimer and colleagues suggest using this method (1).

Notes:

- It is not necessary to normalize before fitting dose-response data. In many cases, it is better to show the actual data.

- You can only plot several different dose-response curves on one graph using one axis when they are comparable. If the different experiments measured different variables, normalizing puts them into comparable units. This can be useful.

- Whether or not you choose to normalize your data, you still need to choose how to fit the data. Do you want Prism to find best-fit values for the Top and Bottom plateaus? Or do you want those plateaus to be determined by control data? This is an important decision.

- If you normalize your data, you can choose one of the normalized dose-response equations. These constrain the the curve to run from 0% to 100%. This kind of constraint only makes sense, when 0% and 100%

are defined by good control data. If the definitions of 0% and 100% are ambiguous, then so is the definition of "50%", and thus the EC50 is also ambiguous.

- Just because you chose to normalize your data doesn't mean you must constrain the curve to run from 0 to 100%. It can still make sense for Prism fit those two plateaus, so all the data (and not just the controls used to normalize) are used to fit the plateaus. Weimer and colleagues show that this is a good way to analyze data (1).

- If you don't normalize your data, you can use the Constrain tab to fix Top and Bottom to values determined from control experiments. So the decision to constrain Top and Bottom is quite distinct from the decision to normalize your data before fitting.

- It is possible to fix one of those parameters (Top or Bottom) to a constant value but not the other.

- If you normalize, don't also choose to differentially weight the data 62. Once you subtract off the baseline (nonspecific) values, the variance among Y replicates is unlikely to be proportional to Y.

Reference

1. Weimer, M., Jiang, X., Ponta, O., Stanzel, S., Freyberger, A., and Kopp-Schneider, A. (2012). The impact of data transformations on concentration–response modeling. Toxicology Letters 213: 292–298.

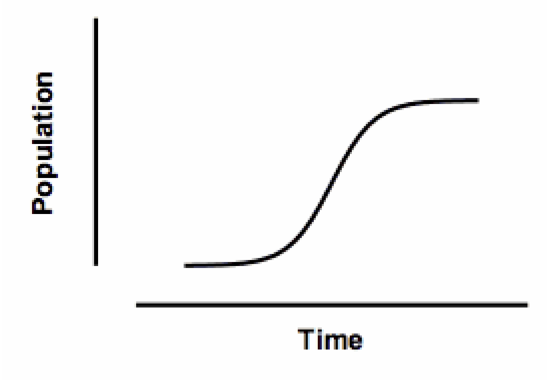
4.5.5.1.9 The term "logistic"

The terms *logistic* has three meanings which have little relationship to each other (1).

Meaning 1: Logistic population growth

The term "logistic" was first invented in the nineteenth century to describe population growth curves. The idea is pretty simple. Population growth is limited, so can't ever exceed some value we'll call Nmax. The rate of change of population at any time t is proportional to the number of individuals alive at that time (Nt). But population growth slows down as it reaches the maximum, so is also proportional to (Nmax - Nt). So the rate of change of population is proportional to Nt(Nmax - Nt).

Integrate that differential equation, and the result is called a logistic equation. It defines a sigmoidal shaped curve that defines the population at any time. The model has three parameters: the starting population, the maximum population, and the time it takes to reach half-maximal. Sometimes it is modified to add a fourth parameter to define the steepness of the curve.

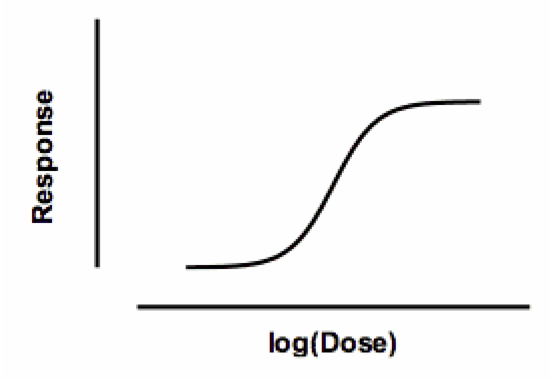


This curve was used by demographers in the past, but actually doesn't do a very good job of describing the growth of human populations. It is still used to model the growth of tumors, and to model the fraction of a population that uses a new product (like a mobile phone).

This model is also used for autocatalytic reactions, where the product of the reaction is also a catalyst for that reaction. With this kind of reaction, the rate of product accumulation is proportional to the concentration of product already produced times the concentration of remaining substrate. This has the same mathematical form as the population growth model. The graph is identical to the one above, except the Y-axis would be the concentration of the product produced by the enzyme reaction (instead of population).

Meaning 2: "Logistic" dose-response curves

Bindslev reviews the history of the many equations used to describe dose response curves(2). A common equation, based on work by Hill, defines a response as a function of the minimum response, the maximum response, the concentration required to evoke a response half-way between the minimum and maximum (the EC50), and a parameter that describes how steep the curve is. In Prism, this equation is called the log(agonist) vs response curve -- Variable slope (four parameters).



Response at any dose is defined by an equation with four parameters: minimum response, maximum response, the log(EC50), and the slope factor (also called the Hill slope). Surprisingly, this equation has the same mathematical form as the logistic equation describing population growth, even though the two uses are entirely different. For this reason, the dose-response equation is sometimes called a "logistic" equation. Since there are four parameters, it is sometimes called the four parameter logistic equation, abbreviated 4PL.

This dose-response model describes many systems, but it is purely empirical. It does not correspond to any molecular model (except, perhaps, when the Hill slope is an integer).

Note the huge differences between the use of a logistic equation to describe population growth, and its use to describe dose-response curves.

- The X axis of the population growth curve curve is time. The X axis of the dose-response curve is the logarithm of dose or concentration.

- The equation for population growth comes from theory. The equation for dose-response relationships is empirical.

- Despite its name, no logarithms are used in the logistic equation for population growth. The X axis of the logistic dose-response curve is the logarithm of dose or concentration.

Meaning 3: Logistic regression

The third use of the word *logistic,* completely unrelated to the other two uses, is logistic regression. This is a multiple regression method used when the outcome (dependent variable) has two possible outcomes. When the model only has one predictor variable, it's called "Simple logistic regression 31", while models that contain multiple predictor variables are referred to as "Multiple logistic regression 41". The point of the equation is to predict the probability, P, of an outcome from a set of independent variables. To create a model that always predicts a value for P that is between 0 and 1 the logistic regression model actually predicts the natural logarithm of the odds. The function that computes the natural logarithm of the odds from a fraction is called the *logit* function (pronounced with a long O and a soft G), so regression used to predict the logit of a probability from multiple independent variables is called *logistic regression*.

References

1. J.S. Cramer. The origins and development of the logit model. Chapter 9 of Logit models from economics and other fields, Cambridge University Press, 2003

2. N. Bindslev, Hill in Hell (pdf), Chapter 10 of Drug-Acceptor Interactions, ISBN: 978-91-977071-0-7

4.5.5.1.10 50% of what? Relative vs absolute IC50

IC50 can be ambiguous

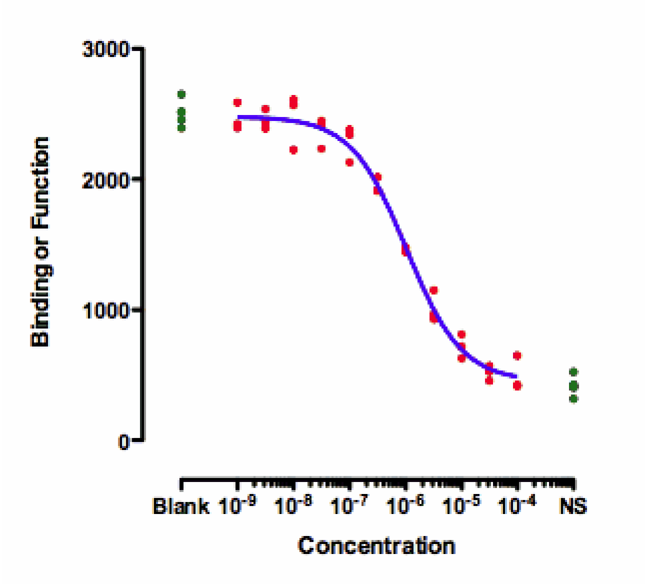
50% of what? There can be more than one answer in some situations.

The concepts of IC50 is fundamental to pharmacology. The IC50 is the concentration of an inhibitor where the response (or binding) is reduced by half. Seems simple enough. But when you actually go to fit data to determine these values, there are several complexities and ambiguities.

The rest of this article is about IC50 (I for inhibition, for downward sloping dose-response curves). All the ideas can be applied to stimulatory curves and EC50 (E for effective) as well. Just stand on your head when you view the figures!.

The ideal situation

This figure shows an ideal situation:



The green symbols show measurements made with controls. The ones on the left (Blank) have no inhibitor, so define "100%". The ones on the right are in the presence of a maximal concentration of a standard inhibitor, so define "0%". The data of the experimental dose-response curve (red dots) extend all the way between the two control values.

When fitting this curve, you need to decide how to fit the top plateau of the curve. You have three choices:

- Fit the data only, ignoring the Blank control values.

- Average the Blank control values, and set the parameter Top to be a constant value equal to the mean of the blanks.

- Enter the blank values as if they were part of the dose-response curve. Simply enter a low dose, perhaps 10-10 or 10-11. You can't enter zero, because zero is not defined on a log scale.

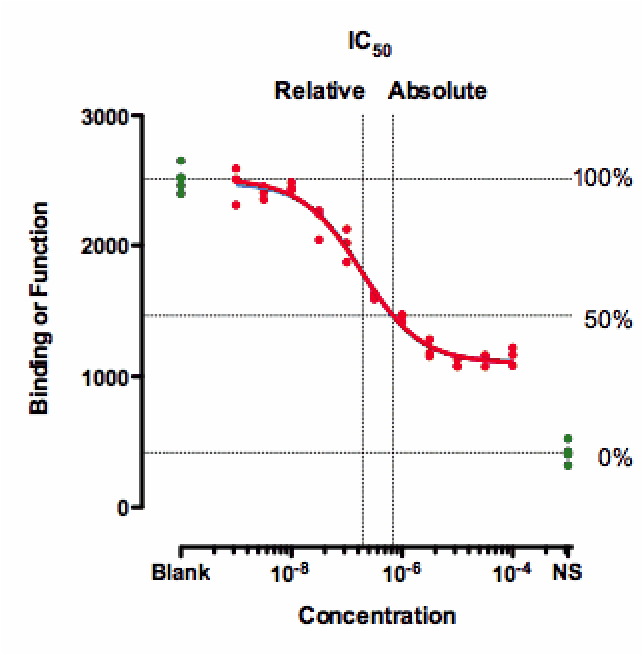
The results will be very similar with any of these methods, because the data form a complete dose-response curve with a clear top plateau that is indistinguishable from the blank. I prefer the third method, as it analyzes all the data, but that is not a strong preference.

Similarly, there are three ways to deal with the bottom plateau: Fit the data only, set Bottom to be a constant equal to the average of the NS controls, and put the NS controls into the fit as if they were a very high concentration of inhibitor.

That is the ideal situation. There is no ambiguity about what IC50 means.

A potentially ambiguous situation

The graph below shows an unusual situation where the inhibition curve plateaus well above the control values (NS) defined by a high concentration of a standard drug. This leads to alternative definitions of IC50.



Clearly, a single value cannot summarize such a curve. You'd need at least two values, one to quantify the middle of the curve (the drug's potency) and one to quantify how low it gets (the drug's maximum effect).

The graph above shows two definitions of the IC50.

The **relative IC50** is by far the most common definition, and the adjective *relative* is usually omitted. It is the concentration required to bring the curve down to point half way between the top and bottom plateaus of the curve. The NS values are totally ignored with this definition of IC50. This definition is the one upon which classical pharmacological analysis of agonist and antagonist interactions is based. With appropriate consideration of the biological system and concentrations of interacting ligands, estimated Kd values can often be derived from the IC50 value defined this way (not so for the "so-called absolute IC50" mentioned below).

The concentration that provokes a response halfway between the Blank and the NS value is sometimes called the **absolute IC50**, The horizontal dotted lines show how 100% and 0% are defined, which then defines 50%. This term is not entirely standard. Since this value does not quantify the potency of a drug, the authors of the International Union of Pharmacology Committee on Receptor Nomenclature (1) think that the concept of absolute IC50 (and that term) is not useful (R. Neubig, personal communication). I agree.

The concept (but not the term "absolute IC50") is used to quantify drugs that slow cell growth. The abbreviation GI50 is used for what we call here the absolute IC50. It is also used by the Environmental Protection Agency (EPA) in evaluating endocrine disrupters (Appendix A). That document uses the term IC50 to refer to the absolute IC50, and the term EC50 to refer to the relative IC50. It doesn't use the terms *relative* and *absolute*.

The absolute IC50 requires that the inhibitor brings the binding or response down to less than 50%. If the inhibitor only brings the response down to (say) 60%, then the curve never crosses the Y=50% line, so the absolute IC50 is undefined.

If you really want to use the absolute IC50, the next page has instructions for fitting a curve to find it 467.

Reference

1. R. R. Neubig et al. International Union of Pharmacology Committee on Receptor Nomenclature and Drug Classification. XXXVIII. Update on terms and symbols in quantitative pharmacology. Pharmacol Rev (2003) vol. 55 (4) pp. 597-606

[Download the Prism file used to create all the graphs in this article.]

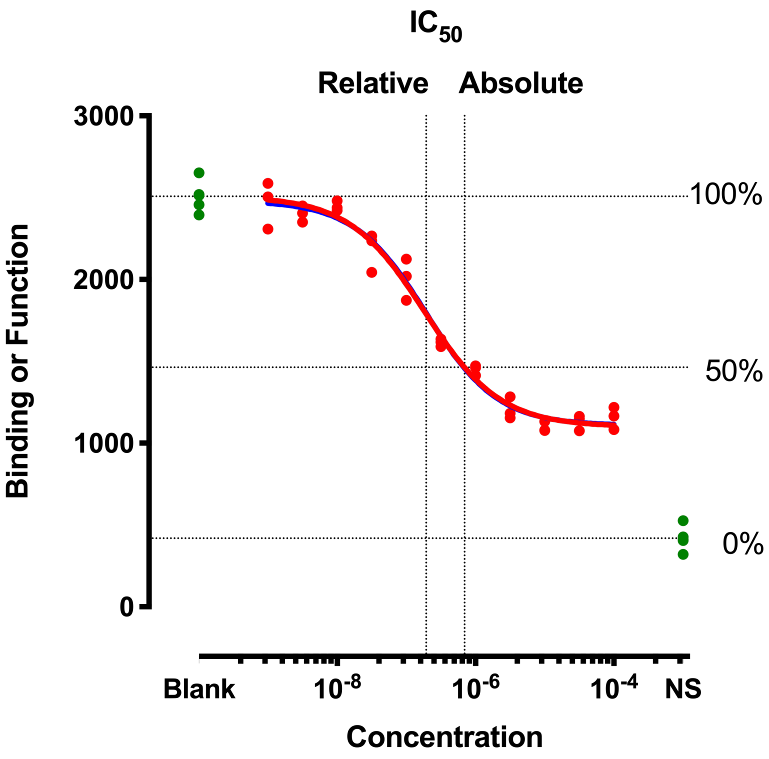
4.5.5.1.11 Fitting the absolute IC50

Fitting a dose-response curve to find the absolute IC50

The concept of an absolute IC50 is not standard, and many find it not to be useful. But if you do, it is not hard to fit a curve to determine it. This EPA document gives the needed equation (which I have generalized a bit, so does not require that the data already be normalized).

```
Fifty=(Top+Baseline)/2
Y= Bottom + (Top-Bottom)/(1+10^((LogIC50-X)*HillSlope + log((Top-Bottom)/(Fift
```

Note the distinction between the parameter *Bottom* and *Baseline*. *Bottom* is the Y value of the bottom plateau of the curve itself. *Baseline* is the Y value that defines 0% -- maximal inhibition by a standard drug. You'll definitely want to constrain *Baseline* to be a constant value based on controls. You may also want to constrain *Top*.



Download the Prism file that fits that equation to make the graph shown above. When fitting data to that equation, don't forget to constrain Baseline and Top to appropriate values determined by controls. Additionally, this file contains another graph where the data are already normalized to run from 0 to 100%. These data are fit to a simpler equation where Baseline is set to equal zero, and Top is set to equal 100. These are hard wired into the equation, so you don't have to remember to constrain those two parameters to constant values.

An alternative approach for normalized data

Here is an alternative approach you can use if your data are normalized.

1. Make sure that your data are normalized to some controls. That means the response at the left end (low concentrations) is near 100%, and the response at higher concentrations is above 0%. If the response plateaus at 0%, then an absolute and relative IC50 are the same, and you can just fit the usual dose-response curve to find the IC50.

2. At the bottom of the data table, add a new row of data. Enter 50 into each Y column. Leave X blank for this row.

3. Use nonlinear regression to fit the data to the log(inhibitor) vs. response (variable slope) curve.

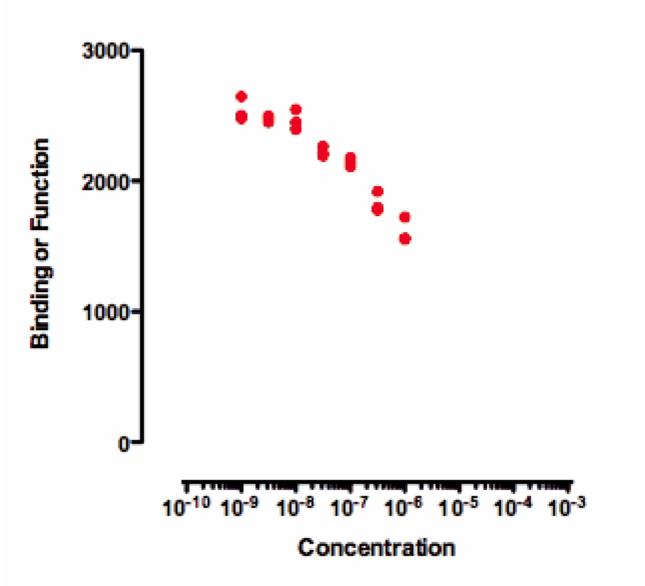
4. On the first (fit) tab of the nonlinear regression dialog, check the option: " Interpolate unknowns from standard curve." Of course, these data are not a standard curve, and there are unknowns. But this option asks Prism to interpolate the X value of the curve when Y=50.

5. On the Constrain tab, consider constraining Top to have a constant value of 100. If you have used good controls to normalize your data, then you know the top plateau of the curve has to be Y=100, so should tell Prism to use that constraint.

6. The IC50 reported as part of the main results table will be the relative IC50.

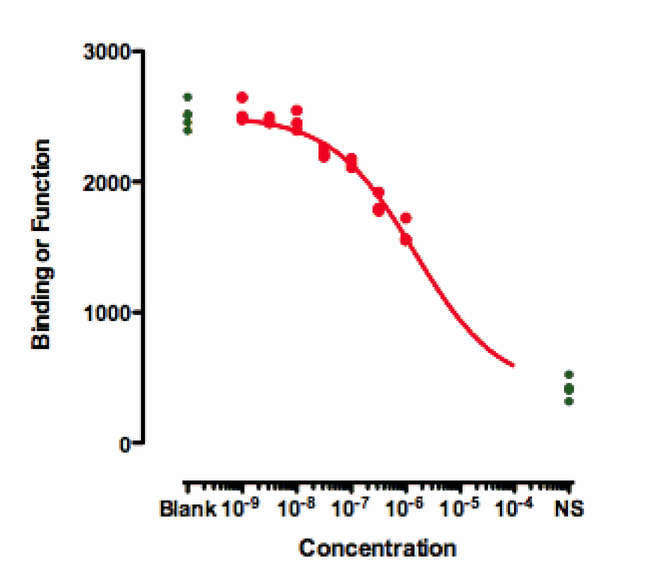
7. Look on the additional results page of interpolated results to see the value of X when Y=50. This is the absolute IC50.

4.5.5.1.12 Incomplete dose-respone curves



Any attempt to determine an IC50 by fitting a curve to the data in the graph above will be useless. Prism might, or might not, be able to fit a dose-response curve to the data. But if the curve fits, the value of the IC50 is likely to be meaningless and have a very wide confidence interval. The data simply don't really define a top plateau (which would define 100) and certainly doesn't provide even a hint of a bottom plateau (which would define 0). If data haven't defined 100 or 0, then 50 is undefined too, as is the IC50.

If you also have control values that define 100 and 0, then the curve can be easily fit. The curve below was created by fitting a dose response curve, but constraining the Top plateau to be a constant value equal to the mean of the Blanks values, and the Bottom plateau equal to the mean of the NS values. Note that the blank and NS values are shown in green.



The value of the IC50 fit this way only makes sense if you assume that higher concentrations of the inhibitor would eventually inhibit down to the NS values. That is an assumption that can't be tested with the data at hand.

An alternative approach would be to enter the blank values with a small X value as if it is response provoked by a super tiny concentration, and enter the nonspecific data with a large X value. Then fit all four parameters without constraint. This prevents the blank and NS data from having too much weight. Weimer and colleagues suggest this is a better way to analyze the data than using the Blank and NS data to constrain the curve (1).

The distinction between relative and absolute IC50 463 doesn't really apply to these data. Because the data don't define a bottom plateau, the IC50 can only be defined relative to the NS control values.

Reference

1. Weimer, M., Jiang, X., Ponta, O., Stanzel, S., Freyberger, A., and Kopp-Schneider, A. (2012). The impact of data transformations on concentration–response modeling. Toxicology Letters 213: 292–298.

4.5.5.1.13 Troubleshooting fits of dose-response curves

First consult the analysis checklists

If your goal is to interpolate from the standard curve, look at the analysis checklist designed for that purpose 428.

If your goal is to fit a model to determine parameter values, review the general analysis checklist for fitting a model 422.

Questions to ask about dose-response curves

- Are the X values concentrations (or doses) or the logarithm of concentrations (or doses)? Prism offers equations for each. You can also use the Transform analysis to make a new results table where the X values are logarithms. Note that stretching an axis to a log scale is not at all the same thing. Prism's nonlinear regression fits the values it is given. Using the Format Axis dialog to stretch the X axis to a logarithmic scale does not change the values the curve fitter sees.

- Is the top plateau defined by the data? If not, can it be constrained or shared? If the top plateau is not defined either by the data or by a constraint, then any fit of a dose-response curve is unlikely to be useful.

- Is the bottom plateau defined by the data? If not, can it be constrained or shared? If the bottom plateau is not defined either by the data or by a constraint, then any fit of a dose-response curve is unlikely to be useful.

- Were the data normalized 458? To what values? If the data were normalized, consider constraining the top and bottom plateaus to be 0 and 1 (or 100) in the Constraint tab?

- Is the Hill slope 454 fixed? A Hill slope of 1.0 or -1.0 is commonly seen in many systems, but not all.

- Four parameter assumes symmetry. The usual equations to fit dose response curves have four parameters (top, bottom, EC50, Hill Slope), and define symmetrical curves. You can choose an equation that adds a fifth parameter to fit asymmetrical curves 529.

- Are you fitting the relative IC50 but expecting an absolute IC50 463, and so are surprised by the results.

4.5.5.2 Dose-response - Stimulation

4.5.5.2.1 Equation: log(agonist) vs. response

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 479 when X values are concentrations or doses.

Introduction

Many log(dose) vs. response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the EC50 of the agonist - the concentration that provokes a response half way between the basal (Bottom) response and the maximal (Top) response.

This model assumes that the dose response curve has a **standard slope**, equal to a Hill slope (or slope factor) of 1.0. This is the slope expected when a ligand binds to a receptor following the law of mass action, and is the slope expected of a dose-response curve when the second messenger created by receptor stimulation binds to its receptor by the law of mass action. If you don't have many data points, consider using the standard slope model. If you have lots of data points, pick the variable slope model to determine the Hill slope from the data.

This equation is sometimes called a three parameter dose-response curve. If you also fit the Hill slope, then it is a four parameter equation.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

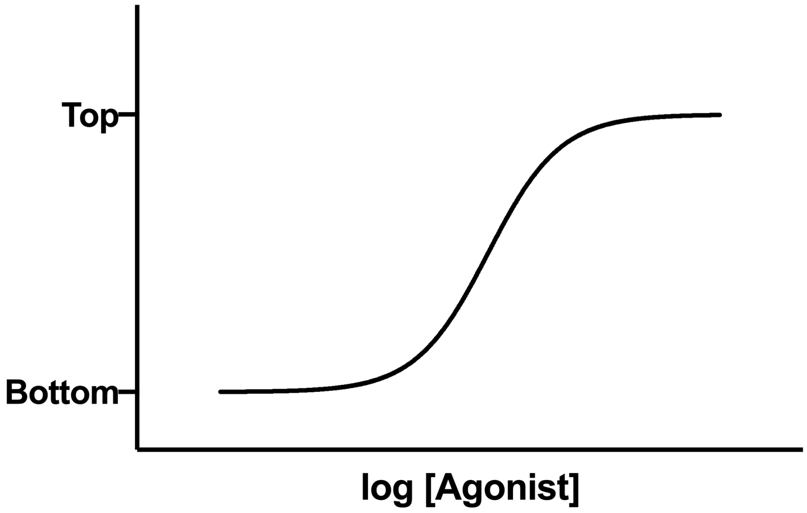
If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 447.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Stimulation" and then choose the equation *log(Agonist) vs. response.*

If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Model

```
Y=Bottom + (Top-Bottom)/(1+10^(LogEC50-X))
```



Interpret the parameters

EC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the EC50 may give a response nowhere near "50". Prism reports both the EC50 and its log.

Top and **Bottom** are plateaus in the units of the Y axis.

4.5.5.2.2 Equation: log(agonist) vs. response -- Variable slope

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 481 when X values are concentrations or doses.

Introduction

Many log(dose) response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the EC50 of the agonist - the concentration that provokes a response half way between the basal (Bottom) response and the maximal (Top) response.

Many dose-response curves have a standard slope of 1.0. This model does not assume a standard slope but rather fits the Hill Slope from the data, and so is called a **Variable slope** model. This is preferable when you have plenty of data points. It is also called a four-parameter dose-response curve, or four-parameter logistic curve, abbreviated 4PL.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 447.

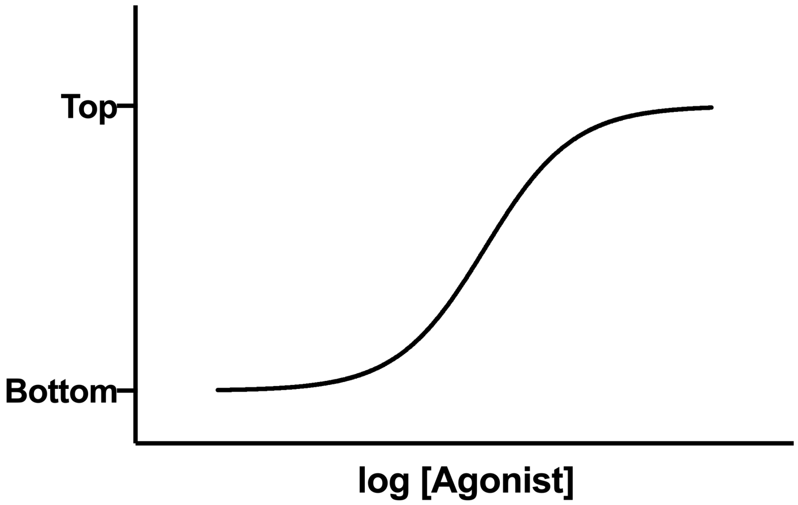
From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Stimulation" and then choose the equation "*log(Agonist) vs. response -- Variable slope*".

Consider constraining the parameter HillSlope to its standard values of 1.0. This is especially useful if you don't have many data points, and therefore cannot fit the slope very well.

If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Model

```
Y=Bottom + (Top-Bottom)/(1+10^((LogEC50-X)*HillSlope))
```



Interpret the parameters

EC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the EC50 may give a response nowhere near "50". Prism reports both the EC50 and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of 1.0 is standard, and you should consider constraining the Hill Slope to a constant value of 1.0. A Hill slope greater than 1.0 is steeper, and a Hill slope less than 1.0 is shallower.

Top and **Bottom** are plateaus in the units of the Y axis.

4.5.5.2.3 Equation: log(agonist) vs. normalized response

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 483 when X values are concentrations or doses.

Introduction

Many log(dose) vs. response curves follow the familiar symmetrical sigmoidal shape.

If you have good control data, it can make sense to **normalize** the response to run between 0% and 100%. This model assumes that the data have been normalized, so forces the curve to run from 0% to 100%. The goal is to determine the EC50 of the agonist - the concentration that provokes a response equal to 50%.

It only makes sense to fit a normalized model when you are sure you have defined 0% and 100% quite accurately. If your data define a complete sigmoidal curve, it is best to fit the entire curve and let Prism fit the Top and Bottom plateaus 473. If your data don't form a full sigmoidal curve, but you can define the bottom and top by solid control data, then fitting to a normalized model is preferable.

This model assumes that the dose response curve has a **standard slope**, equal to a Hill slope (or slope factor) of 1.0. This is the slope expected when a ligand binds to a receptor following the law of mass action, and is the slope expected of a dose-response curve when the second (and third...) messengers created by receptor stimulation binds to its receptor by the law of mass action. If you don't have many data points, consider using the standard slope model. If you have lots of data points, pick the variable slope model to determine the Hill slope from the data.

Step by step

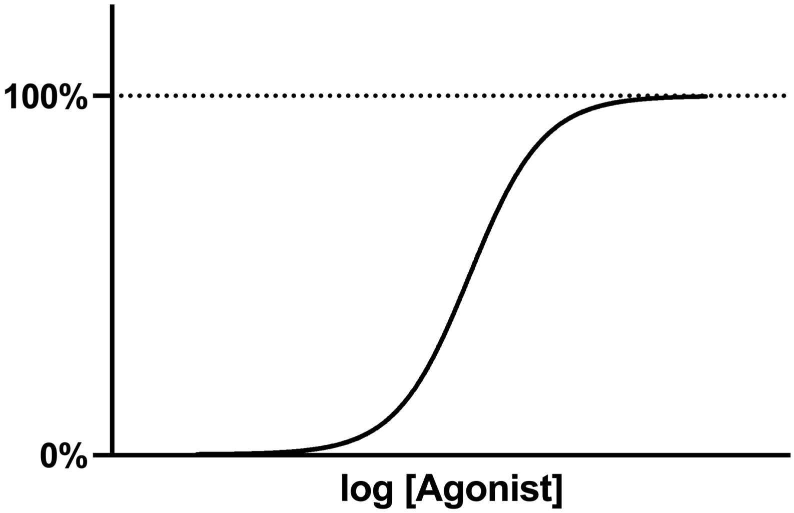
Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 447.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Stimulation" and then choose the equation "*log(Agonist) vs. normalized response*".

Model

```
Y=100/(1+10^(LogEC50-X))
```



Interpret the parameter

EC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the EC50 may give a response nowhere near "50". Prism reports both the EC50 and its log.

4.5.5.2.4 Equation: log(agonist) vs. normalized response -- Variable slope

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 485 when X values are concentrations or doses.

Introduction

Many log(dose) response curves follow the familiar symmetrical sigmoidal shape.

If you have good control data, it can make sense to normalize the response to run between 0% and 100%. This model assumes that the data have been **normalized**, so forces the curve to run from 0% to 100%. The goal is to determine the EC50 of the agonist - the concentration that provokes a response equal to 50%.

It only makes sense to fit a normalized model when you are sure you have defined 0% and 100% quite accurately. If your data define a complete sigmoidal curve, it is best to fit the entire curve and let Prism fit the Top and Bottom plateaus 473. If your data don't form a full sigmoidal curve, but you can define the bottom and top by solid control data, then fitting to a normalized model is preferable.

Many dose-response curves have a standard slope of 1.0. This model does not assume a standard slope but rather fits the Hill Slope from the data, and so is called a **Variable slope** model. This is preferable when you have plenty of data points.

Step by step

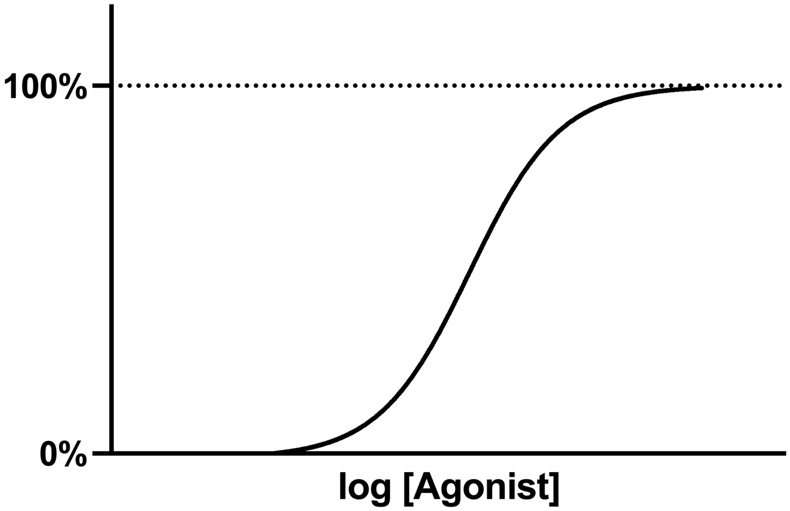
Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 447.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Stimulation" and then choose the equation "*log(Agonist) vs. normalized response -- Variable slope*".

Model

```
Y=100/(1+10^((LogEC50-X)*HillSlope))
```



Interpret the parameters

EC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the EC50 may give a response nowhere near "50". Prism reports both the EC50 and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of 1.0 is standard, and you should consider constraining the Hill Slope to a constant value of 1.0.

4.5.5.2.5 Equation: [Agonist] vs. response

This equation is used when X values are concentrations. Use a related equation 472 when X values are logarithms of concentrations or doses.

Introduction

Many log(dose) vs. response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the EC50 of the agonist - the concentration that provokes a response half way between the basal (Bottom) response and the maximal (Top) response.

This model assumes that the dose response curve has a **standard slope**, equal to a Hill slope (or slope factor) of 1.0. This is the slope expected when a ligand binds to a receptor following the law of mass action, and is the slope expected of a dose-response curve when the second messenger created by receptor stimulation binds to its receptor by the law of mass action. If you don't have many data points, consider using the standard slope model. If you have lots of data points, pick the variable slope model to determine the Hill slope from the data.

This equation is sometimes called a three parameter dose-response curve. If you also fit the Hill slope, then it is a four parameter equation.

Step by step

Create an XY data table. Enter the concentration of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Stimulation" and then choose the equation *[Agonist] vs. response.*

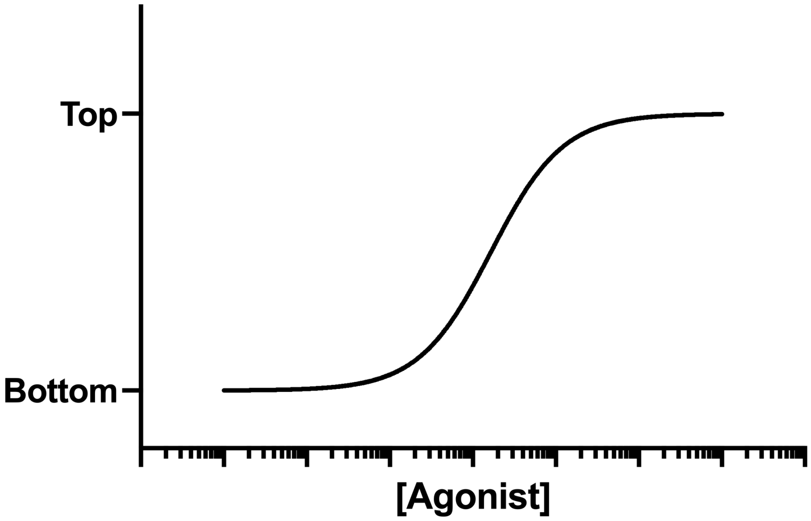
If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Since the uncertainty of the EC50 is very asymmetric, be sure to choose to compute the confidence intervals using the likelihood ratio asymmetric method 347.

Double click on the X axis of the graph, and choose (at the upper left of the Format Graph dialog) to stretch the axis to a logarithmic scale.

Model

```
Y=Bottom + X*(Top-Bottom)/(EC50 + X)
```



Interpret the parameters

EC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the EC50 may give a response nowhere near "50". Prism reports both the EC50 and its logartithm

Top and **Bottom** are plateaus in the units of the Y axis.

4.5.5.2.6 Equation: [Agonist] vs. response -- Variable slope

This equation is used when X values are concentrations. Use a related equation 473 when X values are logarithms of concentrations or doses.

Introduction

Many log(dose) response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the EC50 of the agonist - the concentration that provokes a response half way between the basal (Bottom) response and the maximal (Top) response.

Many dose-response curves have a standard slope of 1.0. This model does not assume a standard slope but rather fits the Hill Slope from the data, and so is called a **Variable slope** model. This is preferable when you have plenty of data points. It is also called a four-parameter dose-response curve, or four-parameter logistic curve, abbreviated 4PL.

Step by step

Create an XY data table. Enter the concentrations of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Stimulation" and then choose the equation "[*Agonist] vs. response -- Variable slope*".

Consider constraining the parameter HillSlope to its standard values of 1.0. This is especially useful if you don't have many data points, and therefore cannot fit the slope very well.

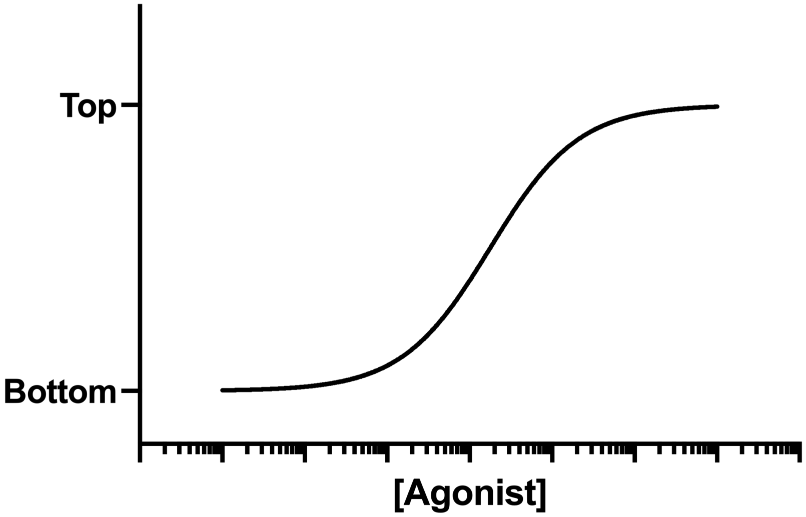
If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Since the uncertainty of the EC50 is very asymmetric, be sure to choose to compute the confidence intervals using the likelihood ratio asymmetric method 347.

Double click on the X axis of the graph, and choose (at the upper left of the Format Graph dialog) to stretch the axis to a logarithmic scale.

Model

```
Y=Bottom + (X^Hillslope)*(Top-Bottom)/(X^HillSlope + EC50^HillSlope)
```



Interpret the parameters

EC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the EC50 may give a response nowhere near "50". Prism reports both the EC50 and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of 1.0 is standard, and you should consider constraining the Hill Slope to a constant value of 1.0. A Hill slope greater than 1.0 is steeper, and a Hill slope less than 1.0 is shallower.

Top and **Bottom** are plateaus in the units of the Y axis.

4.5.5.2.7 Equation: [Agonist] vs. normalized response

This equation is used when X values are concentrations. Use a related equation 475 when X values are logarithms of concentrations or doses.

Introduction

Many log(dose) vs. response curves follow the familiar symmetrical sigmoidal shape.

If you have good control data, it can make sense to **normalize** the response to run between 0% and 100%. This model assumes that the data have been normalized, so forces the curve to run from 0% to 100%. The goal is to determine the EC50 of the agonist - the concentration that provokes a response equal to 50%.

It only makes sense to fit a normalized model when you are sure you have defined 0% and 100% quite accurately. If your data define a complete sigmoidal curve, it is best to fit the entire curve and let Prism fit the Top and Bottom plateaus 473. If your data don't form a full sigmoidal curve, but you can define the bottom and top by solid control data, then fitting to a normalized model is preferable.

This model assumes that the dose response curve has a **standard slope**, equal to a Hill slope (or slope factor) of 1.0. This is the slope expected when a ligand binds to a receptor following the law of mass action, and is the slope expected of a dose-response curve when the second (and third...) messengers created by receptor stimulation binds to its receptor by the law of mass action. If you don't have many data points, consider using the standard slope model. If you have lots of data points, pick the variable slope model to determine the Hill slope from the data.

Step by step

Create an XY data table. Enter the concentrations of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

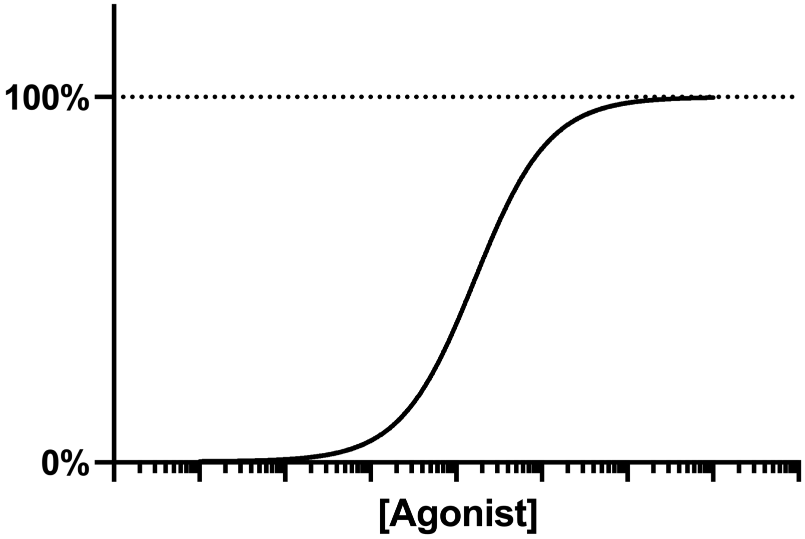
From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Stimulation" and then choose the equation "*[Agonist] vs. normalized response*".

Since the uncertainty of the EC50 is very asymmetric, be sure to choose to compute the confidence intervals using the likelihood ratio asymmetric method 347.

Double click on the X axis of the graph, and choose (at the upper left of the Format Graph dialog) to stretch the axis to a logarithmic scale.

Model

```
Y=100*X/(EC50+X)
```



Interpret the parameter

EC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the EC50 may give a response nowhere near "50". Prism reports both the EC50 and its log.

4.5.5.2.8 Equation: [Agonist] vs. normalized response -- Variable slope

This equation is used when X values are concentrations. Use a related equation 477 when X values are logarithms of concentrations or doses.

Introduction

Many log(dose) response curves follow the familiar symmetrical sigmoidal shape.

If you have good control data, it can make sense to normalize the response to run between 0% and 100%. This model assumes that the data have been **normalized**, so forces the curve to run from 0% to 100%. The goal is to determine the EC50 of the agonist - the concentration that provokes a response equal to 50%.

It only makes sense to fit a normalized model when you are sure you have defined 0% and 100% quite accurately. If your data define a complete sigmoidal curve, it is best to fit the entire curve and let Prism fit the Top and Bottom plateaus 473. If your data don't form a full sigmoidal curve, but you can define the bottom and top by solid control data, then fitting to a normalized model is preferable.

Many dose-response curves have a standard slope of 1.0. This model does not assume a standard slope but rather fits the Hill Slope from the data, and so is called a **Variable slope** model. This is preferable when you have plenty of data points.

Step by step

Create an XY data table. Enter the concentrations of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

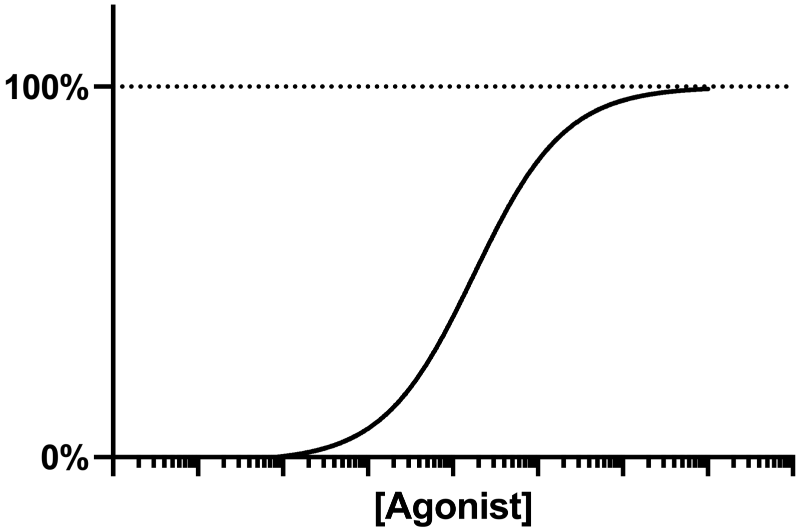
From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Stimulation" and then choose the equation "*[Agonist] vs. normalized response -- Variable slope*".

Since the uncertainty of the EC50 is very asymmetric, be sure to choose to compute the confidence intervals using the likelihood ratio asymmetric method 347.

Double click on the X axis of the graph, and choose (at the upper left of the Format Graph dialog) to stretch the axis to a logarithmic scale.

Model

```
Y=100*(X^HillSlope)/(EC50^HillSlope + (X^HillSlope))
```



Interpret the parameters

EC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the EC50 may give a response nowhere near "50". Prism reports both the EC50 and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of 1.0 is standard, and you should consider constraining the Hill Slope to a constant value of 1.0.

4.5.5.3 Dose-response - Inhibition

4.5.5.3.1 Equation: log(inhibitor) vs. response

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 494 when X values are concentrations or doses.

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the IC50 of the inhibitor - the concentration that provokes a response half way between the maximal (Top) response and the maximally inhibited (Bottom) response.

This model assumes that the dose response curves has a **standard slope**, equal to a Hill slope (or slope factor) of -1.0. This is the slope expected when a ligand binds to a receptor following the law of mass action, and is the slope expected of a dose-response curve when the second messenger created by receptor stimulation binds to its receptor by the law of mass action. If you don't have many data points, consider using the standard slope model. If you have lots of data points, pick the variable slope model to determine the Hill slope from the data.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

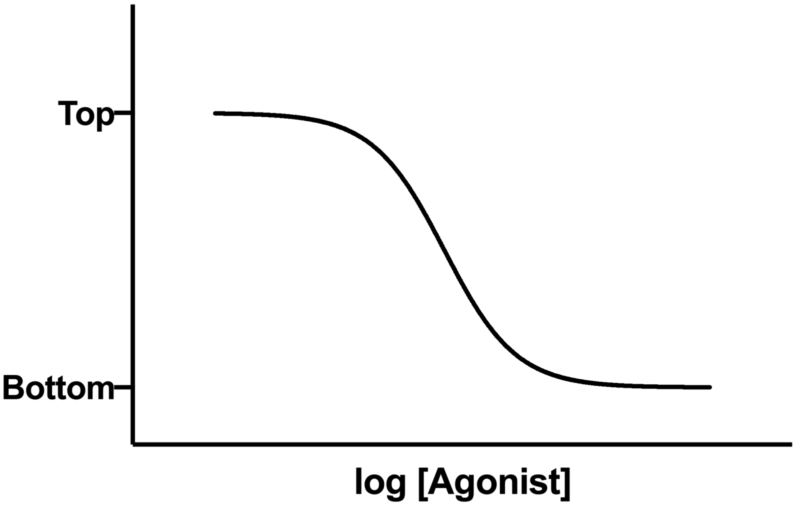
If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 447.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Inhibition" and then choose the equation "*log(inhibitor) vs. response*".

If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Model

```
Y=Bottom + (Top-Bottom)/(1+10^(X-LogIC50))
```



Interpret the parameters

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may give a response nowhere near "50". Prism reports both the IC50 and its log.

Top and **Bottom** are plateaus in the units of the Y axis.

4.5.5.3.2 Equation: log(inhibitor) vs. response -- Variable slope

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 496 when X values are concentrations or doses.

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the IC50 of the inhibitor - the concentration that provokes a response half way between the maximal (Top) response and the maximally inhibited (Bottom) response.

Many inhibitory dose-response curves have a standard slope of -1.0. This model does not assume a standard slope but rather fits the Hill Slope from the data, and so is called a **Variable slope** model. This is preferable when you have plenty of data points. It is also called a four-parameter dose-response curve, or four-parameter logistic curve, abbreviated 4PL.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

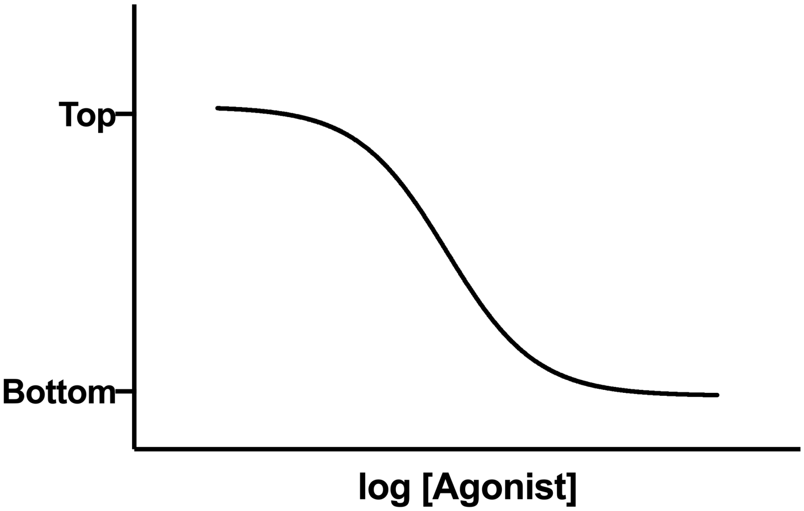
If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 447.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Inhibition" and then choose the equation "*log(inhibitor) vs. response -- Variable slope*".

If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Model

```
Y=Bottom + (Top-Bottom)/(1+10^((LogIC50-X)*HillSlope))
```



Interpret the parameters

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may give a response nowhere near "50". Prism reports both the IC50 and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of -1.0 is standard, and you should consider constraining the Hill Slope to a constant value of -1.0. A Hill slope more negative than -1 (say -2) is steeper.

Top and **Bottom** are plateaus in the units of the Y axis.

4.5.5.3.3 Equation: log(inhibitor) vs. normalized response

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 498 when X values are concentrations or doses.

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape.

If you have good control data, it can make sense to **normalize** the response to run between 0% and 100%. This model assumes that the data have been normalized, so forces the curve to run from 100% down to 0%. The goal is to determine the IC50 of the inhibitor - the concentration that provokes a response equal to 50%.

It only makes sense to fit a normalized model when you are sure you have defined 0% and 100% quite accurately. If your data define a complete sigmoidal curve, it is best to fit the entire curve and let Prism fit the Top and Bottom plateaus 487. If your data don't form a full sigmoidal curve, but you can define the bottom and top by solid control data, then fitting to a normalized model is preferable.

This model assumes that the dose response curve has a **standard slope**, equal to a Hill slope (or slope factor) of -1.0. This is the slope expected when a ligand binds to a receptor following the law of mass action, and is the slope expected of a dose-response curve when the second messenger created by receptor stimulation binds to its receptor by the law of mass action. If you don't have many data points, consider using the standard slope model. If you have lots of data points, pick the variable slope model to determine the Hill slope from the data.

Step by step

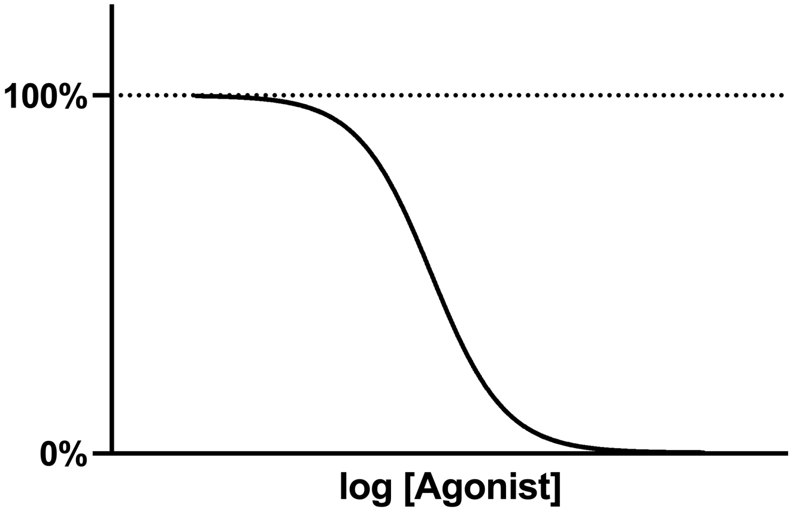
Create an XY data table. Enter the logarithm of the concentration of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 447.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Inhibition" and then choose the equation "*log(inhibitor) vs. normalized response*".

Model

```
Y=100/(1+10^(X-LogIC50))
```



Interpret the parameter

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may give a response nowhere near "50". Prism reports both the IC50 and its log.

4.5.5.3.4 Equation: log(inhibitor) vs. normalized response -- Variable slope

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 500 when X values are concentrations or doses.

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape.

If you have good control data, it can make sense to **normalize** the response to run between 0% and 100%. This model assumes that the data have been normalized, so forces the curve to run from 100% down to 0%. The goal is to determine the IC50 of the inhibitor - the concentration that provokes a response equal to 50%.

It only makes sense to fit a normalized model when you are sure you have defined 0% and 100% quite accurately. If your data define a complete sigmoidal curve, it is best to fit the entire curve and let Prism fit the Top and Bottom plateaus 487. If your data don't form a full sigmoidal curve, but you can define the bottom and top by solid control data, then fitting to a normalized model is preferable.

Many inhibitory dose-response curves have a standard slope of -1.0. This model does not assume a standard slope but rather fits the Hill Slope from the data, and so is called a **Variable slope** model. This is preferable when you have plenty of data points.

Step by step

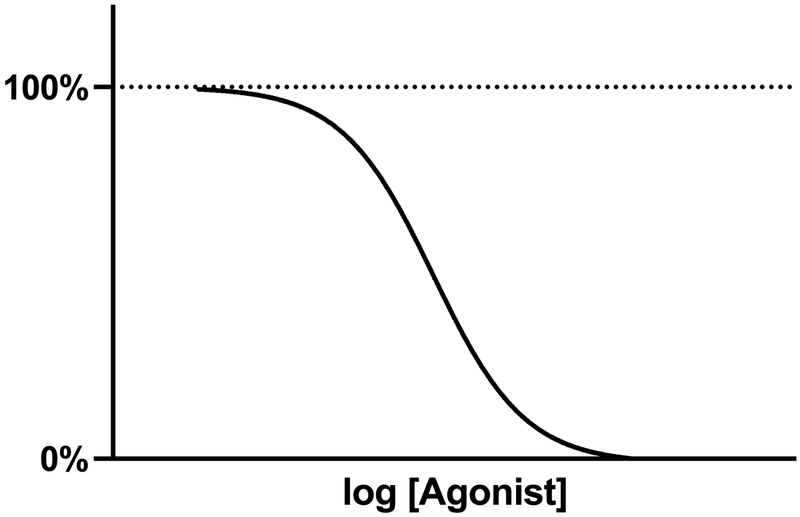
Create an XY data table. Enter the logarithm of the concentration of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 447.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Inhibition" and then choose the equation "*log(inhibitor) vs. normalized response -- variable slope*".

Model

```
Y=100/(1+10^((LogIC50-X)*HillSlope)))
```



Interpret the parameters

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may give a response nowhere near "50". Prism reports both the IC50 and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of -1.0 is standard, and you should consider constraining the Hill Slope to a constant value of -1.0. A Hill slope more negative than -1 (say -2) is steeper.

4.5.5.3.5 Equation: [Inhibitor] vs. response

This equation is used when X values are concentrations or doses. Use a related equation 487 when X values are logarithms of concentrations or doses.

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the IC50 of the inhibitor - the concentration that provokes a response half way between the maximal (Top) response and the maximally inhibited (Bottom) response.

This model assumes that the dose response curves has a **standard slope**, equal to a Hill slope (or slope factor) of -1.0. This is the slope expected when a ligand binds to a receptor following the law of mass action, and is the slope expected of a dose-response curve when the second messenger created by receptor stimulation binds to its receptor by the law of mass action. If you don't have many data points, consider using the standard slope model. If you have lots of data points, pick the variable slope model to determine the Hill slope from the data.

Step by step

Create an XY data table. Enter the concentrations of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Inhibition" and then choose the equation "*[Inhibitor] vs. response*".

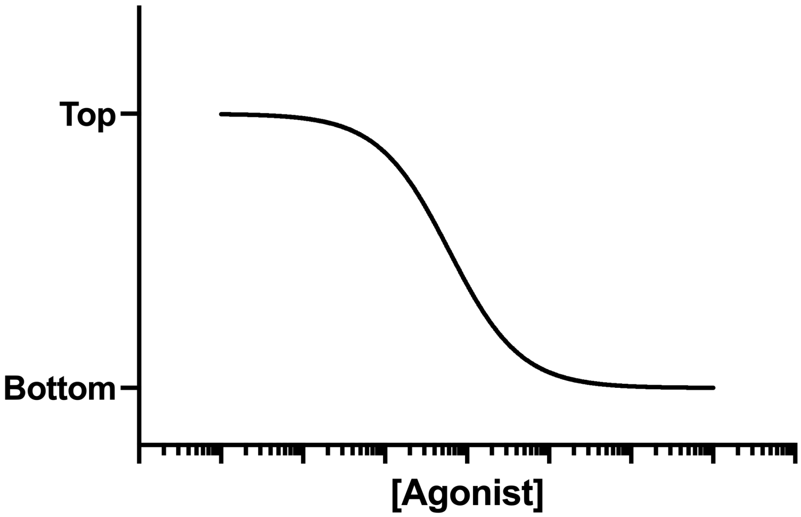
If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Since the uncertainty of the EC50 is very asymmetric, be sure to choose to compute the confidence intervals using the likelihood ratio asymmetric method 347.

Double click on the X axis of the graph, and choose (at the upper left of the Format Graph dialog) to stretch the axis to a logarithmic scale.

Model

```
Y=Bottom + (Top-Bottom)/(1+(X/IC50))
```



Interpret the parameters

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may give a response nowhere near "50". Prism reports both the IC50 and its log.

Top and **Bottom** are plateaus in the units of the Y axis.

4.5.5.3.6 Equation: [Inhibitor] vs. response -- Variable slope

This equation is used when X values are concentrations or doses. Use a related equation 489 when X values are logarithms of concentrations or doses.

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the IC50 of the inhibitor - the concentration that provokes a response half way between the maximal (Top) response and the maximally inhibited (Bottom) response.

Many inhibitory dose-response curves have a standard slope of -1.0. This model does not assume a standard slope but rather fits the Hill Slope from the data, and so is called a **Variable slope** model. This is preferable when you have plenty of data points. It is also called a four-parameter dose-response curve, or four-parameter logistic curve, abbreviated 4PL.

Step by step

Create an XY data table. Enter the concentrations of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Inhibition" and then choose the equation "*[Inhibitor] vs. response -- Variable slope*".

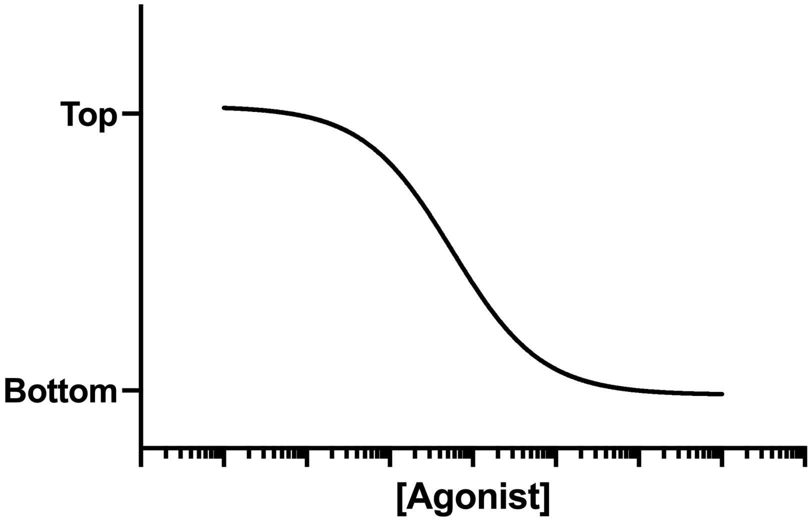
If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Since the uncertainty of the EC50 is very asymmetric, be sure to choose to compute the confidence intervals using the likelihood ratio asymmetric method 347.

Double click on the X axis of the graph, and choose (at the upper left of the Format Graph dialog) to stretch the axis to a logarithmic scale.

Model

```
Y=Bottom + (Top-Bottom)/(1+(IC50/X)^HillSlope)
```



Interpret the parameters

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may give a response nowhere near "50". Prism reports both the IC50 and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of -1.0 is standard, and you should consider constraining the Hill Slope to a constant value of -1.0. A Hill slope more negative than -1 (say -2) is steeper.

Top and **Bottom** are plateaus in the units of the Y axis.

4.5.5.3.7 Equation: [Inhibitor] vs. normalized response

This equation is used when X values are concentrations or doses. Use a related equation 491 when X values are logarithms of concentrations or doses.

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape.

If you have good control data, it can make sense to **normalize** the response to run between 0% and 100%. This model assumes that the data have been normalized, so forces the curve to run from 100% down to 0%. The goal is to determine the IC50 of the inhibitor - the concentration that provokes a response equal to 50%.

It only makes sense to fit a normalized model when you are sure you have defined 0% and 100% quite accurately. If your data define a complete sigmoidal curve, it is best to fit the entire curve and let Prism fit the Top and Bottom plateaus 487. If your data don't form a full sigmoidal curve, but you can define the bottom and top by solid control data, then fitting to a normalized model is preferable.

This model assumes that the dose response curve has a **standard slope**, equal to a Hill slope (or slope factor) of -1.0. This is the slope expected when a ligand binds to a receptor following the law of mass action, and is the slope expected of a dose-response curve when the second messenger created by receptor stimulation binds to its receptor by the law of mass action. If you don't have many data points, consider using the standard slope model. If you have lots of data points, pick the variable slope model to determine the Hill slope from the data.

Step by step

Create an XY data table. Enter the concentrations of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

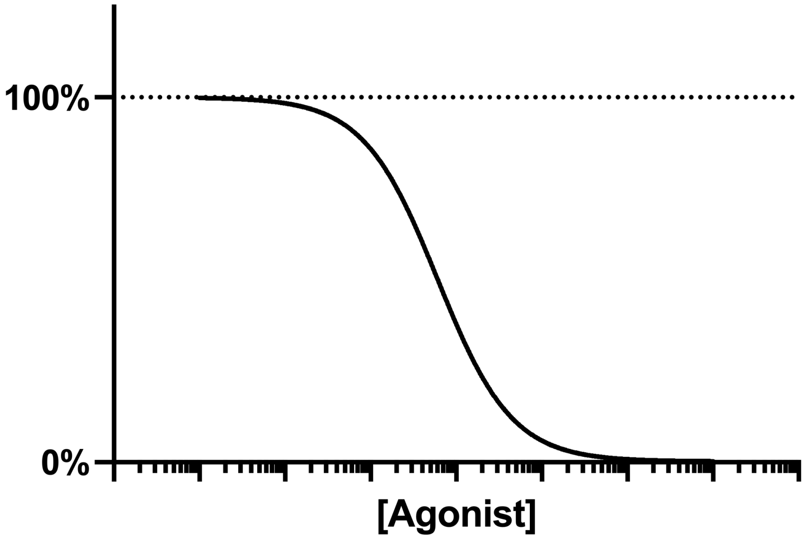
From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Inhibition" and then choose the equation "[I*nhibitor] vs. normalized response*"*.*

Since the uncertainty of the EC50 is very asymmetric, be sure to choose to compute the confidence intervals using the likelihood ratio asymmetric method 347.

Double click on the X axis of the graph, and choose (at the upper left of the Format Graph dialog) to stretch the axis to a logarithmic scale.

Model

```
Y=100/(1+X/IC50)
```



Interpret the parameter

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may give a response nowhere near "50". Prism reports both the IC50 and its log.

4.5.5.3.8 Equation: [Inhibitor] vs. normalized response -- Variable slope

This equation is used when X values are concentrations or doses. Use a related equation 492 when X values are logarithms of concentrations or doses.

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape.

If you have good control data, it can make sense to **normalize** the response to run between 0% and 100%. This model assumes that the data have been normalized, so forces the curve to run from 100% down to 0%. The goal is to determine the IC50 of the inhibitor - the concentration that provokes a response equal to 50%.

It only makes sense to fit a normalized model when you are sure you have defined 0% and 100% quite accurately. If your data define a complete sigmoidal curve, it is best to fit the entire curve and let Prism fit the Top and Bottom plateaus 487. If your data don't form a full sigmoidal curve, but you can define the bottom and top by solid control data, then fitting to a normalized model is preferable.

Many inhibitory dose-response curves have a standard slope of -1.0. This model does not assume a standard slope but rather fits the Hill Slope from the data, and so is called a **Variable slope** model. This is preferable when you have plenty of data points.

Step by step

Create an XY data table. Enter the concentrations of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

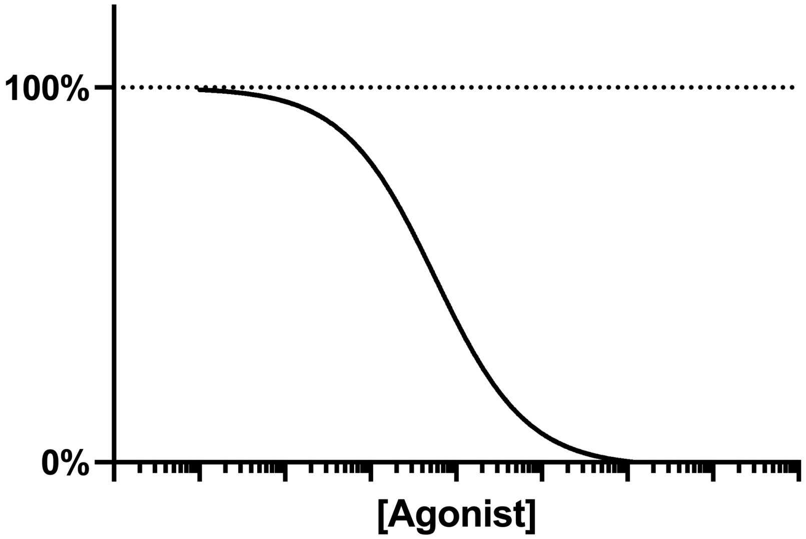
From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Inhibition" and then choose the equation "*log(inhibitor) vs. normalized response -- variable slope*".

Since the uncertainty of the EC50 is very asymmetric, be sure to choose to compute the confidence intervals using the likelihood ratio asymmetric method 347.

Double click on the X axis of the graph, and choose (at the upper left of the Format Graph dialog) to stretch the axis to a logarithmic scale.

Model

```
Y=100/(1+(IC50/X)^HillSlope)
```



Interpret the parameters

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may give a response nowhere near "50". Prism reports both the IC50 and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of -1.0 is standard, and you should consider constraining the Hill Slope to a constant value of -1.0. A Hill slope more negative than -1 (say -2) is steeper.

4.5.5.4 Dose-response -- Special (X is Concentration)

.

4.5.5.4.1 Equation: Asymmetrical (five parameter)

This equation is used when X values are concentrations or doses. Use a related equation 529 when X values are logarithms of concentrations or doses.

Introduction

The standard dose-response curve is sometimes called the **four-parameter logistic equation**. It fits four parameters: the bottom and top plateaus of the curve, the EC50 (or IC50), and the slope factor (Hill slope). This curve is symmetrical around its midpoint. To extend the model to handle curves that are not symmetrical, the Richards equation adds an additional parameter, S, which quantifies the asymmetry. This equation is sometimes referred to as a **five-parameter logistic equation**, abbreviated 5PL.

Step by step

Create an XY data table. Enter the concentration of the agonist into X. Enter response into Y in any convenient units.

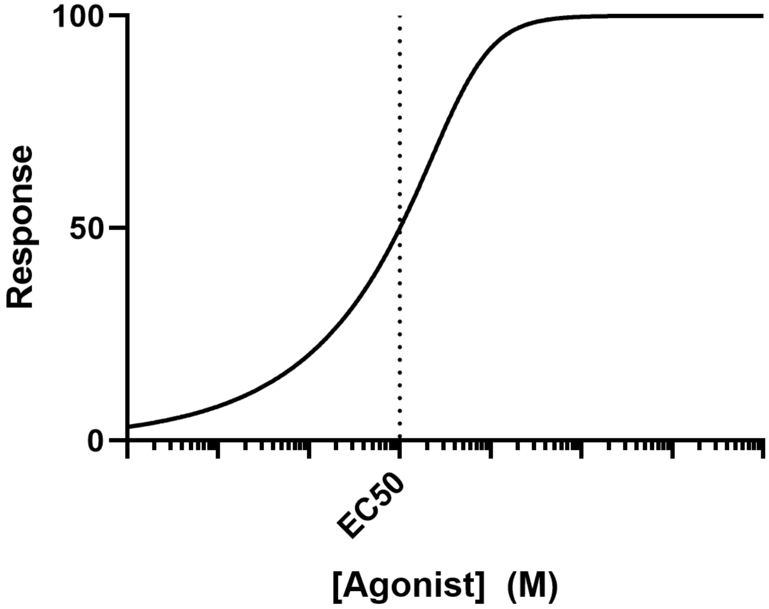
From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is concentration. Then choose *Asymmetrical (five parameter), X is concentration.*

Consider constraining the Hill Slope to a constant value of 1.0 (stimulation) or -1 (inhibition).

Also consider whether Bottom or Top should be fixed to constant values, or shared between data sets.

Model

```
Denominator = (1+(2^(1/S)-1)*((EC50/X)^HillSlope))^S
Numerator = Top - Bottom
Y = Bottom + (Numerator/Denominator)
```



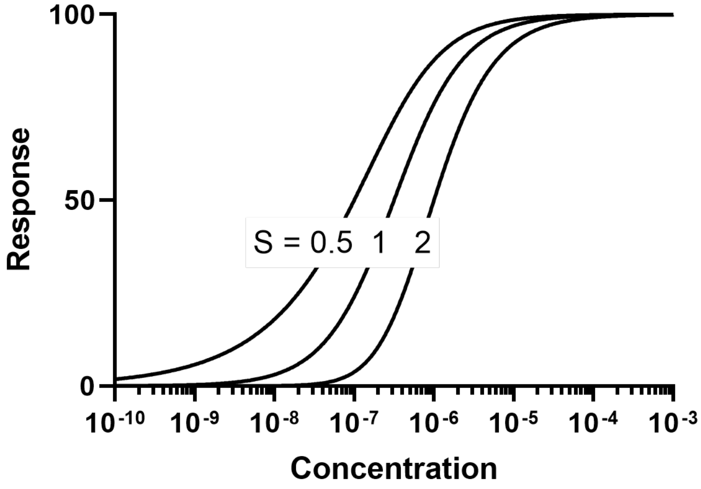
Interpret the parameters

Bottom and **Top** are the plateaus at the left and right ends of the curve, in the same units as Y.

EC50 is the concentrations that give half-maximal effects, in the same units as X.

HillSlope is the unitless slope factor or Hill slope. Consider constraining it to equal 1.0 (stimulation) or -1 (inhibition).

S is the unitless symmetry parameter. If S=1, the curve is symmetrical and identical to the standard dose-response equation. If S is distinct than 1.0, then the curve is asymmetric as shown below.



Notes

- If your goal is to obtain meaningful best-fit parameters, then you'll need lots of high quality data. It is very hard to fit both slope and asymmetry with tight confidence intervals. If your goal is just to interpolate unknowns from a standard curve, the width of the confidence intervals of the parameters doesn't really matter. What you want is a curve that follows the data, and in some cases an asymmetrical five parameter model does so better than a four parameter model.

- It can be tricky to get a good fit to the 5PL. See this discussion.

- Other formulations of asymmetrical dose-response curves have been developed. For example, Ricketts and Head developed a model for use in baroreflex studies.

- Bindslev has written a lengthy on-line text, Drug-Acceptor Interactions. Chapter 10, Hill in Hell discusses many models of dose-response curves, including asymmetrical ones.

- Liao and Liu have done simulations that show the advantage of fitting the EC50 rather Xb.

- Gottschalk and Dunn review the properties of the 5Pl.

- The equation built-in to Prism is only one of several ways to express a five parameter concentration-response curve.

References

Bindslev, Drug-Acceptor Interactions. Chapter 10, Hill in Hell

Cumberland, W.N., Fong, Y., Yu, X., Defawe, O., Frahm, N., and De Rosa, S. (2014). Nonlinear Calibration Model Choice between the Four and Five-Parameter Logistic Models. Journal of Biopharmaceutical Statistics 25: 972–983.

Giraldo, J., Vivas, N. M., Vila, E. & Badia, A. Assessing the (a)symmetry of concentration-effect curves: empirical versus mechanistic models. Pharmacol Ther 95, 21–45 (2002).

Gottschalk, P. G. & Dunn, J. R. The five-parameter logistic: a characterization and comparison with the four-parameter logistic. Anal Biochem 343, 54–65 (2005).

Liao, J.J.Z, Liu, R., Re-parameterization of five-parameter logistic function, Journal of Chemometrics, 23:248-253 (2009)

Ricketts, J. H. and Head, G.A. A five-parameter logistic equation for investigating asymmetry of curvature in baroreflex studies. Am. J. Physiology, 277: R441-54 (1999)

4.5.5.4.2 Equation: Biphasic dose-response

This equation is used when X values are concentrations or doses. Use a related equation 532 when X values are logarithms of concentrations or doses.

Introduction

A common deviation from the standard monotonic sigmoid shape is the biphasic sigmoid shape.

Step by step

Create an XY data table. Enter the concentration of the agonist into X. Enter response into Y in any convenient units.

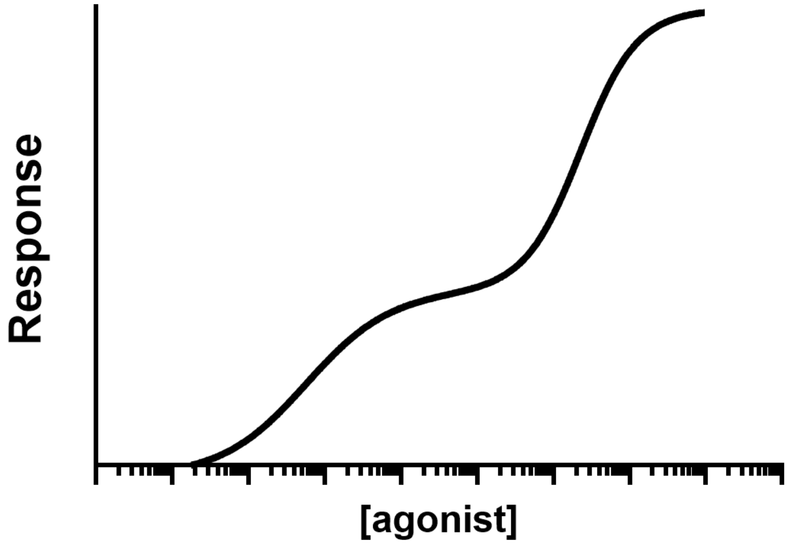
From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is concentration. Then choose *Biphasic dose-response, X is concentration.*

Consider constraining nH1 and nH2 to constant values of 1.0 (stimulation) or -1 (inhibition).

Also consider whether Bottom or Top should be fixed to constant values, or shared between data sets.

Model

```
Span=Top-Bottom
Section1=Span*Frac/(1+(EC50_1/X)^nH1)
Section2=Span* (1-Frac)/(1+(EC50_2/X)^nH2)
Y=Bottom + Section1 +Section2
```



Interpret the parameters

Bottom and **Top** are the plateaus at the left and right ends of the curve, in the same units as Y.

EC50_1 and **EC50_2** are the concentrations that give half-maximal stimulatory and inhibitory effects in the same units as X.

nH1 and **nH2** are the unitless slope factors or Hill slopes. Consider constraining these to equal 1.0 (stimulation) and -1 (inhibition).

Frac is the proportion of maximal response due to the more potent phase.

4.5.5.4.3 Equation: Bell-shaped dose-response

This equation is used when X values are concentrations or doses. Use a related equation 534 when X values are logarithms of concentrations or doses.

Introduction

Some drugs may cause an inhibitory response at low concentrations, and a stimulatory response at high concentrations, or vice-versa. The net result is a bell-shaped dose-response curve.

The model explained here is the sum of two dose-response curves, one that stimulates and one that inhibits. you will need lots of data to determine all the parameters without ambiguity, so this model will rarely be useful for data analysis. But it might be useful as a way to draw a smooth curve through the data

Step by step

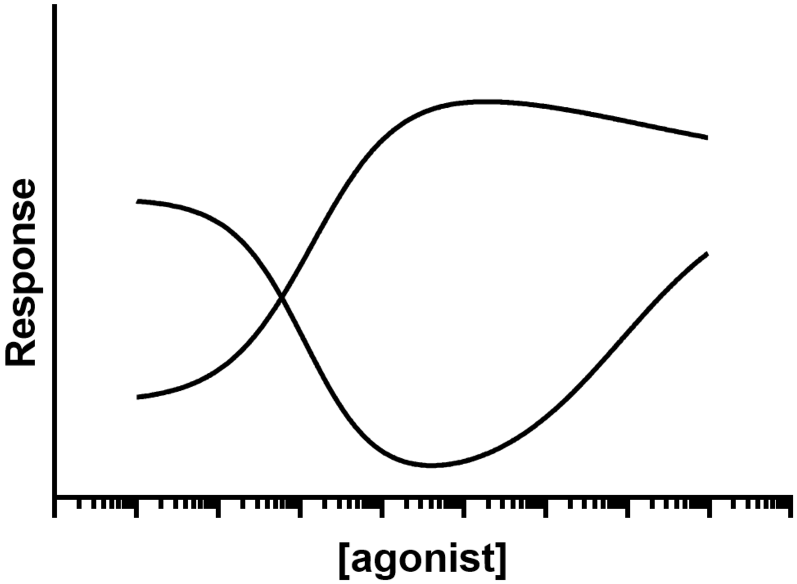
Create an XY data table. Enter the concentration of the agonist into X. Enter response into Y in any convenient units.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is concentration. Then choose *Bell-shaped dose-response, X is concentration.*

Consider constraining nH1 and nH2 to constant values of 1.0 (stimulation) and -1 (inhibition).

Model

```
Span1=Plateau1-Dip
Span2=Plateau2-Dip
Section1=Span1/(1+(EC50_1/X)^nH1)
Section2=Span2/(1+(X/EC50_2)^nH2)
Y=Dip+Section1+Section2
```



Interpret the parameters

Plateau1 and **Plateau2** are the plateaus at the left and right ends of the curve, in the same units as Y.

Dip is the plateau level in the middle of the curve, in same units as Y. Note that when the curve goes up first (then down), this parameter named *Dip* would better be named *Peak*.

EC50_1 and **EC50_2** are the concentrations that give half-maximal stimulatory and inhibitory effects in the same units as X.

nH1 and **nH2** are the unitless slope factors or Hill slopes. Consider constraining these to equal 1.0 (stimulation) and -1 (inhibition).

4.5.5.4.4 Equation: Operational model - Depletion

This equation is used when X values are concentrations or doses. Use a related equation 535 when X values are logarithms of concentrations or doses.

Introduction to the operational model

The EC50 is determined by two properties of the agonist:

- How well it binds to the receptor, quantified by the **affinity** of the drug for binding to its receptor.
- How well it causes a response once bound. This property is known as the agonist's **efficacy**. Since efficacy depends on both agonist and tissue, a single drug acting on a single kind of receptor can have different efficacies, and thus different EC50 values, in different tissues.

A single dose-response experiment cannot determine affinity and efficacy. A drug that binds with high affinity but has low efficacy will produce exactly the same dose-response curve as a drug with low affinity and high efficacy.

To untangle affinity from efficacy, globally fit a dose-response curve of a full agonist and a second dose-response curve determined after treating the cells or tissue with an alkylating agent (or some other irreversible treatment) that reduces the number of accessible receptors. With fewer receptors, the dose response curve is shifted down and usually to the right.

The operational model assumes that the affinity of the drug for the receptors is not altered by reducing the number of available receptors. It also assumes that the maximum possible response in the tissue remains unchanged (the treatment was specific for the receptors you are studying). Accepting these assumptions, fitting the operational model globally will determine the affinity of the agonist for the receptors.

Step by step

Create an XY data table. Enter the concentration of the agonist ligand into X. Enter response into Y in any convenient units. Enter data with a full agonist and no receptor depletion into column A. Enter data collected after receptor depletion into column B. Repeat, if you have data with different levels of receptor depletion for column C, D, E, ... You don't have to know the degree to which the receptors are depleted, and don't have to enter any values in the column titles (although they are useful as labels).

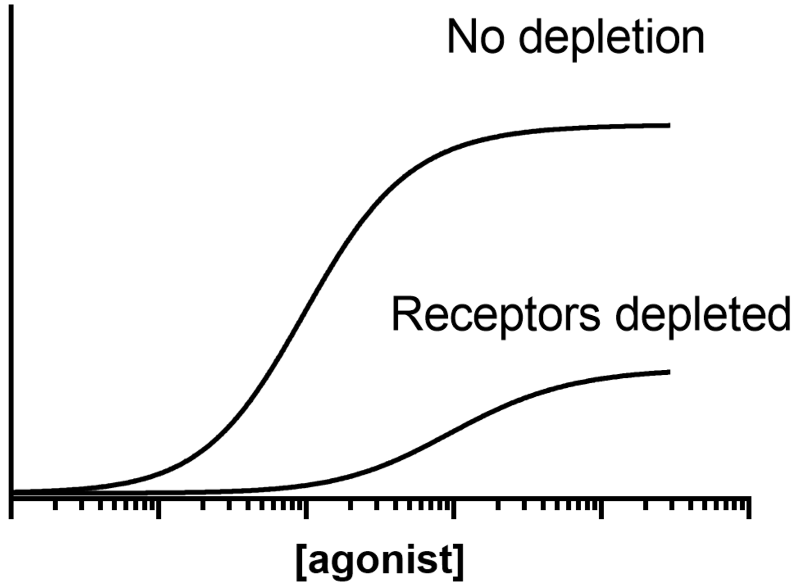
From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is concentration. Then choose *Operational Model - Depletion, X is concentration.*

If you have subtracted off any basal response, consider constraining the parameter Basal to a constant value of zero.

Also consider constraining the transducer slope n to a constant value of 1.0. When set to 1.0, all dose-response curves are constrained to have Hill slopes of 1.0, which is observed commonly.

Model

```
operate= ((Ka+X)/(Tau*X))^n
Y=Basal + (Effectmax-Basal)/(1+operate)
```



Interpret the parameters

Effectmax is the maximum possible system response, in units of the Y axis. It is the top plateau of the dose-response curve obtained with a full agonist without receptor depletion. If your agonist isn't a full agonist, the EffectMax might be higher than the top plateau of the no depletion curve.

Basal is the response in absence of agonist, in same units as Y. If you have subtracted off any basal response, constrain basal to a constant value of zero.

KA is the agonist-receptor dissociation constant, in same units as X (usually molar). It measures the affinity of the full agonist for the receptors, which is the main goal of this kind of experiment. Prism reports both KA and its logarithm. It is not the same as the EC50.

tau is the transducer constant, a practical measure of efficacy. It is the inverse of the fraction of receptors that must be occupied by agonist to obtain the half-maximal response. If tau equals 10, that means that occupation of only 10% of the receptors leads to a half-maximal response. If tau equals 1.0, that means that it requires occupation of all the receptors to give a half-maximal response. This would happen in a tissue where the receptors had been substantially depleted. Because tau is a property of both the tissue and receptor system, it is not a direct measure of intrinsic efficacy, which is commonly defined as a property belonging only to an agonist-receptor pair, irrespective of the assay system in which it is measured. Prism reports both tau and its logarithm for each data set.

n is the Unitless transducer slope. It is similar to, but not identical to, the Hill slope. In most cases, n is constrained to a constant value of 1.0, in which case all the dose-response curves will have Hill slopes of 1.0. If n does not equal 1.0, the Hill Slope does not equal either 1.0 or n.

Notes

Since Tau measures efficacy, Prism fits a different value of tau for each data set. Receptor depletion reduce the value of tau. The other parameters are fit globally, to find one best-fit value for all the data sets.

Reference

Black and Leff (Proc. R. Soc. Lond. B, 220: 141-162, 1983

4.5.5.4.5 Equation: Operational model - Partial agonist

This equation is used when X values are concentrations or doses. Use a related equation 538 when X values are logarithms of concentrations or doses.

Introduction to the operational model

The EC50, fit by standard dose-response models, is determined by two properties of the agonist:

- How well it binds to the receptor, quantified by the **affinity** of the drug for binding to its receptor.

- How well it causes a response once bound. This property is known as the agonist's **efficacy**. Since efficacy depends on both agonist and tissue, a single drug acting on a single kind of receptor can have different efficacies, and thus different EC50 values, in different tissues.

A single dose-response experiment cannot determine affinity and efficacy. A drug that binds tightly with high affinity but has low efficacy, will produce exactly the same dose-response curve as a drug with low affinity and high efficacy.

To determine the affinity of a partial agonist, use the operational model to globally fit the dose-response curves of both a full agonist and the partial agonist. The data from the full-agonist determines the maximum possible effect. Knowing that, the fitting can determine the affinity of the partial agonist.

Step by step

Create an XY data table. Enter the concentration of the agonist ligand into X. Enter response into Y in any convenient units. Enter data with the full agonist into column A. Enter data collected with a partial agonist into column B. Repeat, if you have data with different partial agonists, for column C, D, E, ..., each with a different amount of depletion.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is concentration. Then choose *Operational Model - Partial agonist, X is concentration.*

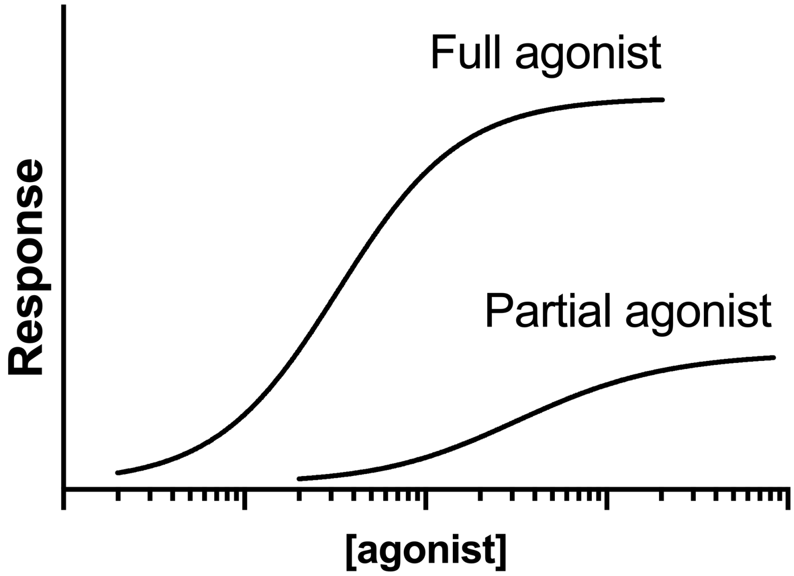
If you have subtracted off any basal response, consider constraining the parameter Basal to a constant value of zero.

Also consider constraining the transducer slope n to a constant value of 1.0. When set to 1.0, all dose-response curves are constrained to have Hill slopes of 1.0, which is observed commonly. If n is not 1.0, the Hill slopes will not be 1.0, but the Hill slopes will not equal exactly n.

Model

```
operate= ((Ka+X)/(Tau*X))^n
<A> Y = Basal + (Effectmax-Basal)/(1+(EC50/X)^n)
<~A> Y = Basal + (Effectmax-Basal)/(1+operate)
```

The second line is preceded with <A> which means it only applies to the first data set. It fits a variable slope dose-response curve. The third line is preceded with <~A> which means it applies to all data set except the first. It fits the operational model to determine the affinity (KA) of the partial agonist.



Interpret the parameters

Effectmax is the maximum possible system response, in units of the Y axis. It is the top plateau of the full agonist's dose-response curve.

Basal is the response in absence of agonist, in same units as Y. If you have subtracted off any basal response, constrain basal to a constant value of zero.

n is the Unitless transducer slope. It is similar to, but not identical to, the Hill slope. In most cases, n is constrained to a constant value of 1.0, in which case all the dose-response curves will have Hill slopes of 1.0. If n does not equal 1.0, the Hill Slope does not equal either 1.0 or n.

KA is the equilibrium dissociation constant of the partial agonist(s), in same units as X (usually molar). It measures the affinity of the partial agonist for the receptors, which is the main goal of this kind of experiment. Prism reports both KA and its logarithm. It is not the same as the EC50. By definition, it cannot be computed for the first data set (the full agonist).

tau is the transducer constant, a practical measure of efficacy. It is the inverse of the fraction of receptors that must be occupied by agonist to obtain the half-maximal response. If t equals 10, that means that occupation of only 10% of the receptors leads to a half-maximal response. If t equals 1.0, that means that it requires occupation of all the receptors to give a half-maximal response. This would happen with a partial agonist. Prism reports both tau and its logarithm, and fits tau individually for each data set. By definition, it cannot be computed for the first data set (the full agonist).

Reference

Black and Leff (Proc. R. Soc. Lond. B, 220: 141-162, 1983

4.5.5.4.6 Equation: Gaddum/Schild EC50 shift

This equation is used when X values are concentrations or doses. Use a related equation 541 when X values are logarithms of concentrations or doses.

Introduction

A competitive inhibitor competes for agonist binding to a receptor, and shifts the dose-response curve to the right without changing the maximum response. By fitting all the curves globally, you can determine the affinity of the competitive inhibitor.

Step by step

1. Create an XY data table with subcolumns to match your experimental design.

2. Enter the concentration of the agonist ligand into X. For example, if a concentration is 1nM, enter 1e-9. Don't enter the logarithm of the concentration (i.e. -9).

3. Enter response into Y in any convenient units. Enter data with no inhibitor into column A. Enter data collected with a constant concentration of inhibitor into column B. Repeat, if you have data, for column C, D, E, ..., each with a different concentration of inhibitor.

4. Enter the inhibitor concentration in molar as the column titles of the data table. If the concentration is 1nM enter '1e-9' or '0.00000001' into the column titles. Don't enter '-9'. Don't forget to enter '0' as the column title for data set A, since these are the control data with no inhibitor. These values entered into the column title are used in the

analyis; they are more than just labels, so need to be entered in the correct format.

5. From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special. Then choose *Gaddum/Schild EC50 shift.*

6. Consider constraining the parameters HillSlope and SchildSlope to their standard values of 1.0. This is especially useful if you don't have many data points, and therefore cannot fit these parameters.

Notes on units

- This equation was written so the X values are entered as the concentration of the agonist, while the column titles are entered as concentration of antagonist. This is simply the way this equation was written. It would be easy enough to clone the equation and write it with different conventions if you wanted to. Prism also offers a related equation 541 in which the X values are entered as the logarithm of the concentration of the agonist.

- The EC50 is reported in the same units you used to enter X values. If you entered X as molar concentration values, then the EC50 will be in molar. If you entered X values as a dose in milligrams, then the EC50 will be in milligrams.

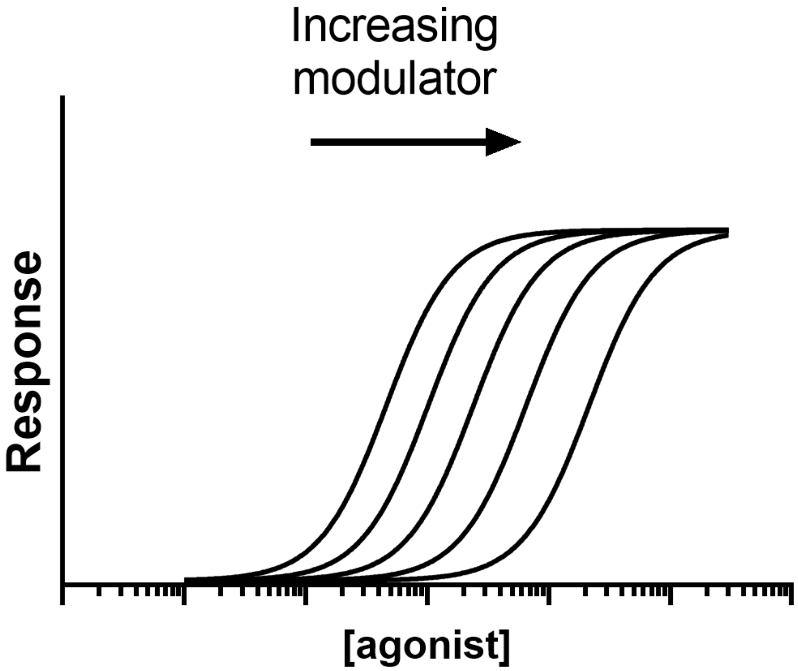
- The pA2 is the negative logarithm of the concentration of antagonist needed to shift the curve by a factor of 2. That concentration is in the same units you used to enter the antagonist concentration into the column titles. If you entered concentration in nM into the column titles, the pA2 is the negative logarithm of nM. If you entered concentration as a dose in mg/kg, then the pA2 is the negative logarithm of mg/kg.

- If you want to transform the pA2 to a concentration, first flip the sign (multiply by -1) and then take the antilog (ten to that power). If you also do so for the confidence limits, note that flipping the sign also changes the order of the limits, so the "lower" confidence limit is actually the upper limit, and the "upper" limit is actually the lower limit.

Model

```
Antag=1+(B/(10^(-1*pA2)))^SchildSlope
```

```
Y=Bottom + (Top-Bottom)/(1+(EC50*Antag/X)^HillSlope)
```



Interpret the parameters

EC50 is the concentration of agonist that gives half maximal response in the absence of inhibitor. Prism reports both the EC50 and its log. The logEC50 is in the same units you used to enter the X values. The EC50 is in the antilog of those units.

pA2 is the negative logarithm of the concentration of antagonist needed to shift the dose response curve by a factor of 2. If the HillSlope and SchildSlope are fixed to 1.0, it is the pKb, the negative log of the equilibrium dissociation constant (Molar) of inhibitors binding to the receptors. The units are the negative log of whatever units you used to enter concentrations as column titles on the data table.

HillSlope describes the steepness of the family of curves. A HillSlope of 1.0 is standard, and you should consider constraining the Hill Slope to a constant value of 1.0.

SchildSlope quantifies how well the shifts correspond to the prediction of competitive interaction. If the competitor is competitive, the SchildSlope will equal 1.0. You should consider constraining SchildSlope to a constant value of 1.0. antagonist term, [B], is now raised to the power S, where S denotes the Schild slope factor. If the shift to the right is greater than predicted by competitive interactions, S will be greater than 1. If the rightward shift is less than predicted by competitive interaction, then S will be less than 1.

Top and **Bottom** are plateaus in the units of the Y axis.

Notes:

- All six of these are shared among all the data sets, so you will only see one best fit value for each parameter, not one for each data set.
- Prism does not report the EC50 for each curve, only the EC50 for the first curve (the one without any antagonist).
- The variable B in the model is defined to be a data set constant whose value comes from the column titles. The results page will show these values of B. Make sure they are the concentration of antagonist used in each data set (and not the log of concentrations).

When is the Schild model valid?

Colquhoun (1) has shown that the Schild model is valid whenever you can make these assumptions:

- The antagonist, B, is a true antagonist that, alone, does not change the conformation of the receptor.
- Binding of agonist, A, and antagonist, B, is mutually exclusive at every binding site.
- B has the same affinity for every binding site.
- The observed response is the same if the occupancy of each site by A is the same, regardless of how many sites are occupied by B.
- Measurements are made at equilibrium.

1. Colquhoun, D.. *Why the Schild method is better than Schild realised*. Trends Pharmacol Sci (2007) vol. 28 (12) pp. 608-14

4.5.5.4.7 Equation: EC50 shift

This equation is used when X values are concentrations or doses. Use a related equation 545 when X values are logarithms of concentrations or doses.

Introduction

A competitive inhibitor competes for agonist binding to a receptor, and shifts the dose-response curve to the right without changing the maximum response. This model fits two dose response curves and determines the fold shift.

Step by step

1. Create an XY data table.

2. Enter the concentration of the agonist ligand into the X column.

3. Enter response into Y in any convenient units. Enter data with no inhibitor into column A. Enter data collected with a constant concentration of inhibitor into column B. You can label columns A and B with any column title you like. The column title does not get used as part of the analysis.

4. From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is concentration. Then choose *EC50 shift, X is concentration.*

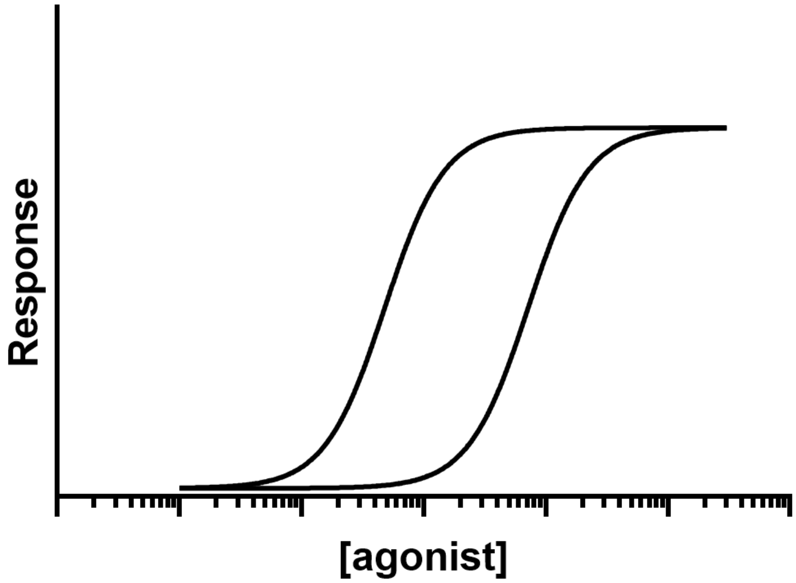
5. If you have subtracted off any baseline signal, constrain the parameter Bottom to a constant value of zero.

Model

```
<A>EC=EC50Control
```

```
<~A>EC=EC50Control + EC50Ratio

Y=Bottom + (Top-Bottom)/(1+(EC/X)^HillSlope)
```



EC50Control is the concentration of agonist that gives half maximal response in the absence of modulator.

Top and **Bottom** are plateaus in the units of the Y axis (shared).

EC50Ratio is the ratio of EC50 in presence of inhibitor divided by EC50 of agonist alone.

HillSlope is the slope factor (shared)

Notes

- If you have several concentrations of antagonist, use a different model that will directly fit the Schild model (using either X as concentration of agonist[515] or X as the logarithm of the concentration of agonist[541]) and determine the pA2.
- If you want to test the null hypothesis that the EC50 ratio is 1.0 (no shift), that is easy to do. Go to the Compare tab, choose to test whether a selected parameter differs from a hypothetical value,

choose the parameter EC50Ratio, and enter the hypothetical value 1.0.

- This model shares Top, Bottom and HillSlope. So it assumes that both models are exactly the same except (perhaps) for the EC50. You can go to the Constrain tab and lift this constraint for any or all of these three parameters, if you think it makes sense not to share them.

4.5.5.4.8 Equation: Allosteric EC50 shift

This equation is used when X values are concentrations or doses. Use a related equation 547 when X values are logarithms of concentrations or doses.

Introduction

An allosteric modulator can reduce or enhance agonist binding. This model fits entire dose-response curves determined in the absence and presence of a modulator. The goal is to learn the affinity of the modulator for binding to its site, and also determine the value of alpha, the ternary complex constant that quantifies the degree to which binding of the modulator alters the affinity of the radioligand for the receptor site.

Step by step

1. Create an XY data table.

2. Enter the concentration of the agonist ligand into X. If the concentration is 1nM, enter 1e-9.

3. Enter response into Y in any convenient units. Enter data with no modulator into column A. Enter data collected with a constant concentration of modulator into column B. Repeat, if you have data, for column C, D, E, ..., each with a different concentration of modulator.

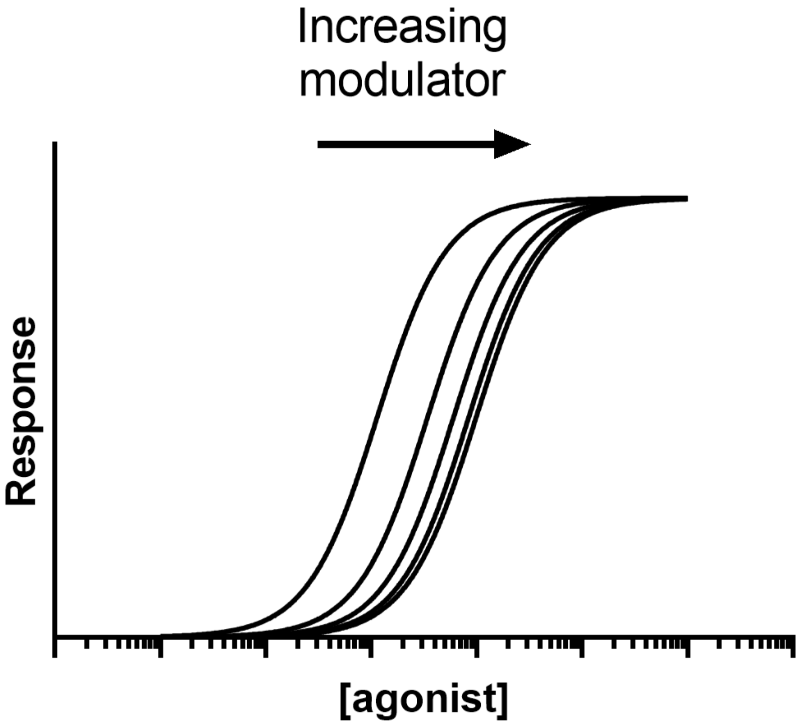
4. Enter the modulator concentration (in molar so 1nM is entered as '1e-9') into the column titles. Don't forget to enter '0' as the column title for data set A. These column titles are not just labels. The values you enter become part of the analysis.

5. From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is concentration. Then choose *Allosteric EC50 shift, X is concentration.*

You do not need to constrain any parameters to constant values

Model

```
Antag=(1+B/KB)/(1+alpha*B/KB)
Y=Bottom+(Top-Bottom)/(1+(EC50*Antag/X)^HillSlope)
```



logEC50 and **EC50** are the logarithm of the concentration of agonist and concentration of the agonist - respectively - that gives half maximal response in the absence of modulator.

KB is the equilibrium dissociation constant (Molar) of modulator binding to its allosteric site. It is in the same molar units used to enter the modulator concentration into column titles on the data table.

Alpha is the ternary complex constant. When Alpha=1.0, the modulator won't alter binding. If Alpha is less than 1.0, then the modulator reduces ligand binding. If Alpha is greater than 1.0, then the modulator increases binding. In the example shown about, Alpha equals 0.01 so the modulator greatly decreases binding.

Top and **Bottom** are plateaus in the units of the Y axis.

Notes

- This model is designed to analyze data when the modulator works via an allosteric site. Since the agonist and modulator are acting via different sites, it is incorrect to refer to the modulator as a competitor.

- The model is written to fit alpha itself, rather than the logarithm of alpha. However, alpha is asymmetrical: All values from 0 to 1 mean that the modulator decreases binding, while all values from 1 to infinity mean that the modulator enhances binding. On a log scale, its values are more symmetrical, so the confidence interval computed on a log scale (as Prism does) are more accurate. Prism reports both alpha and log(alpha).

- This model assumes that the allosteric modulator is present in excess, so the concentration you added is very close to its free concentration. This model won't work when the concentration of allosteric modulator is limiting (as it is when G proteins alter agonist binding to many receptors). No explicit model can handle this situation. You need to define the model with an implicit equation (Y on both sides of the equals sign) and Prism cannot handle such equations.

- Note two important points about the progression of curves from left to right with increasing concentrations of the allosteric modulator. First, note that the maximum response doesn't change. Second, note that the effect of the modulator to right-shift the dose-response curve reaches a maximum as the modulator saturates its binding site.

Reference

A. Christopoulos and T. Kenakin, Pharmacol Rev, 54: 323-374, 2002

4.5.5.4.9 Equation: ECanything

This equation is used when X values are concentrations or doses. Use a related equation 550 when X values are logarithms of concentrations or doses.

Introduction

Many log(dose) response curves follow the familiar symmetrical sigmoidal shape. The usual goal is to determine the EC50 of the agonist - the concentration that provokes a response half way between the basal (Bottom) response and the maximal (Top) response. But you can determine any spot along the curve, say a EC80 or EC90.

Many dose-response curves have a standard slope of 1.0. This model does not assume a standard slope but rather fits the Hill Slope from the data. Hence the name **Variable slope** model. This is preferable when you have plenty of data points.

Step by step

Create an XY data table. Enter the concentration of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is concentration. Then choose "*[Agonist] vs. response -- Find ECanything*".

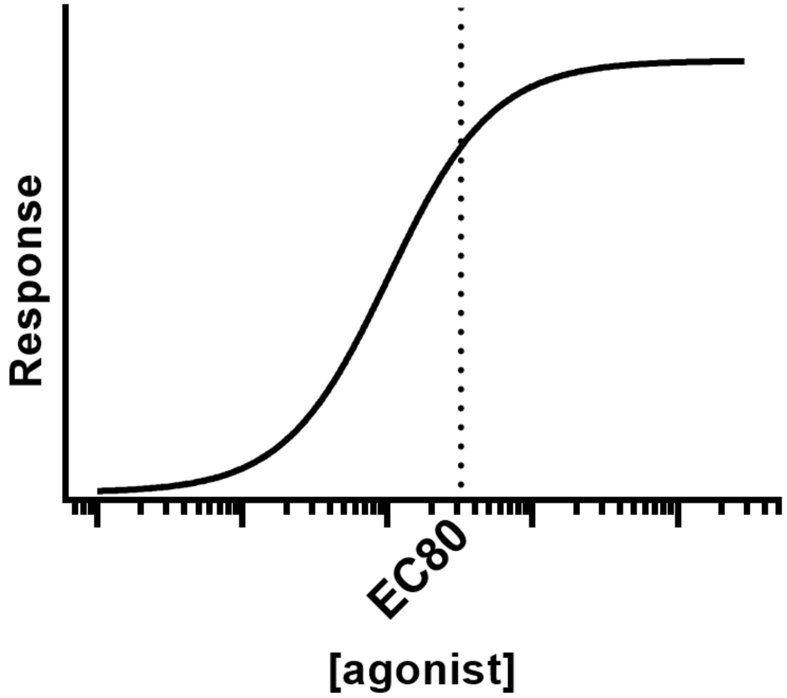
You must constrain the parameter F to have a constant value between 0 and 100. Set F to 80 if you want to fit the EC80. If you constrain F to equal 50, then this equation is the same as a variable slope dose-response curve.

Consider constraining the parameter HillSlope to its standard value of 1.0 of -1. This is especially useful if you don't have many data points, and therefore cannot fit the slope very well.

If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Model

```
EC50=ECF / (F/(100-F))^(1/HillSlope)
Y=Bottom + (Top-Bottom)/(1+(EC50/X)^HillSlope)
```



Interpret the parameters

ECF is the concentration of agonist that gives a response F percent of the way between Bottom and Top. Prism reports both the ECF and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of 1.0 is standard, and you should consider constraining the Hill Slope to a constant value of 1.0. A Hill slope greater than 1.0 is steeper, and a Hill slope less than 1.0 is shallower.

Top and **Bottom** are plateaus in the units of the Y axis.

Adapting this equation

ICanything

This equation can also fit inhibitory data where the curve goes downhill rather than uphill. The best-fit value of the Hill Slope will be negative in this case. The result will always be reported as ECF. Let's say you set F=80. Then the ECF for inhibitory data would be the concentration (X value) required to bring the curve down to 80%. If you want the concentration that brings the curve down by 80%, to 20%, then you'd need to set F equal to 20. If you want the result to say "ICF" rather than "ECF", clone the equation and then edit to make that change

X is log concentration rather than concentration

Prism has a related equation 550 built-into its equation library that will allow you to fit the ECF (or ICF) using values where X represents the logarithm of the concentration of the agonist instead of the concentration of the agonist. Both equations will report the ECF as well as the log(ECF).

4.5.5.4.10 Equation: Absolute IC50

This equation is used when X values are concentrations or doses. Use a related equation 552 when X values are logarithms of concentrations or doses.

Introduction

Many dose-response curves follow a familiar sigmoidal shape. These curves are often defined by four parameters: Top, Bottom, Hill slope, and the EC50 (or IC50). The Top and Bottom parameters describe the values at which the curve reach a plateau - coming infinitely close, but never quite reaching these values. The Hill slope describes the slope of the sigmoidal curve between these two plateaus. The EC50 (or IC50) refer to a concentration of agonist (or antagonist) required to increase (or reduce) the measured response to half - or 50% - of its maximal value.

However, there are a number of ways that this concept of "half" can be interpreted. Are you interested in the concentration that results in a measured response halfway between the maximal (Top) and minimal (Bottom) values of the curve? Or do you want the concentration that results in a measured response halfway between the maximal (Top) value of the curve and some defined "baseline" control value?

When considering inhibitory data (measured response decreases with increasing concentration of antagonist), these two values are referred to as the **Relative IC50** and **Absolute IC50**, respectively.

Relative IC50

This is by far the most common definition, and is usually what people mean by "the IC50" of a response. It is the concentration required to bring the curve down to a point halfway between the "Top" and "Bottom" plateaus of the curve. Unless the data are normalized to run from 0 to 100, this is **not** the value that corresponds to 50% on the Y axis.

Absolute IC50

The concentration that provokes a response halfway between the blank (no antagonist, maximum measured response) and some positive control (to represent a fully inhibited response) is sometimes called the Absolute IC50. In many cases, the positive control used may be a different, standard drug known to elicit a maximal inhibitory effect on the measured response. Often, the response measured for this standard positive control is lower than the "Bottom" of the dose-response curve. As a result, the Absolute IC50 and Relative IC50 *will not be the same*.

In some cases, an Absolute IC50 simply cannot be calculated. For example, the antagonist being tested may only capable of reducing a measured response to 60% of its maximal (blank) value, but no further. In this case, the Absolute IC50 would not be defined (no increase in the concentration of antagonist would be capable of lowering the measured response to 50% of the maximal response). Be sure to know which IC50 you want to report, and what your values mean.

Read more about Absolute IC50 here.

Step by step

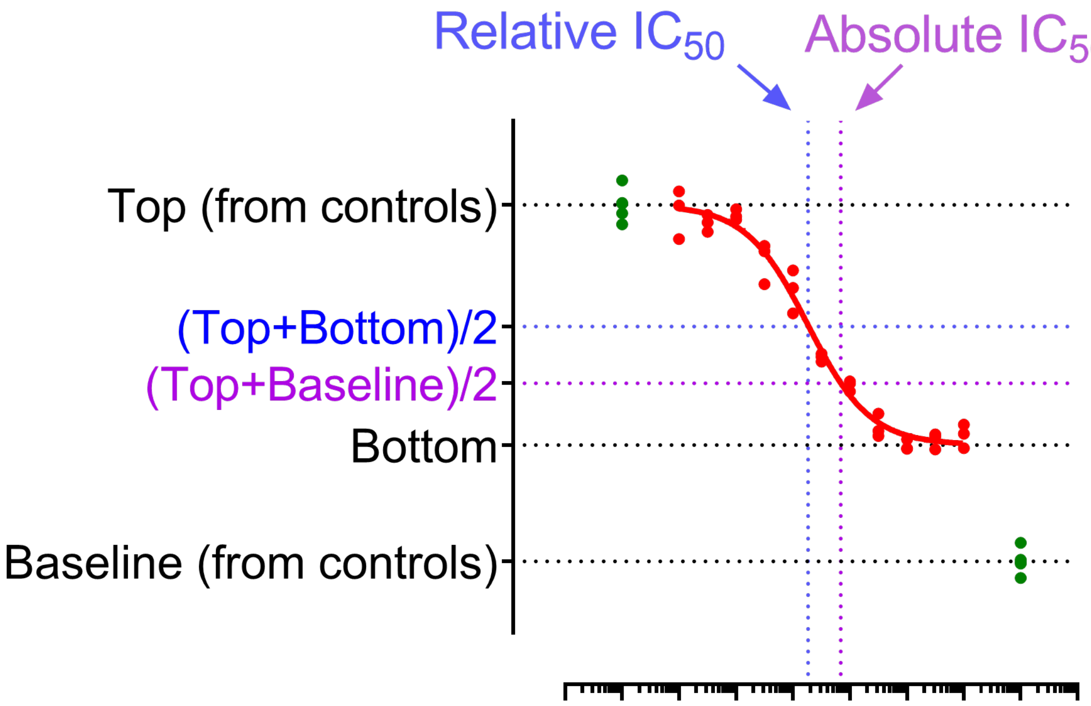
Create an XY data table. Enter the concentration of the agonist into X. Enter response into Y in any convenient units.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is concentration. Then choose "*Absolute IC50, X is concentration*".

You must enter a value for the parameter "Baseline" on the Constrain tab of the analysis parameters dialog. Note the important distinction between "Baseline" and "Bottom" in this model. "Bottom" is the Y value of the bottom plateau of the curve itself. "Baseline" is the Y value that defines 0% - maximal inhibition by a standard drug. While you *must* constrain the baseline, you may want to consider constraining the "Top" parameter as well.

Model

```
Fifty=(Top+Baseline)/2
Y= Bottom + (Top-Bottom)/(1+((Top-Bottom)/(Fifty-Bottom)-1)*(AbsoluteIC50/X)^H
```



Interpret the parameters

Top and **Bottom** are plateaus in the units of the Y axis.

HillSlope describes the steepness of the family of curves. A HillSlope of 1.0 is standard, and you should consider constraining the Hill Slope to a constant value of 1.0. A Hill slope greater than 1.0 is steeper, and a Hill slope less than 1.0 is shallower.

Baseline is the measured response of a "standard" drug or control resulting in a maximally inhibited response. This value must be entered by the user in the constrain tab of the analysis parameters dialog.

AbsoluteIC50 is the concentration of antagonist that gives a response half way between Top and Baseline. This is not the same as the response half way between Top and Bottom (this is the relative IC50, and is not reported by this analysis). Prism also reports the **LogAbsoluteIC50**.

4.5.5.5 Dose-response -- Special (X is Log[Concentration])

.

4.5.5.5.1 Equation: Asymmetrical (five parameter)

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 502 when X values are concentrations or doses.

Introduction

The standard dose-response curve is sometimes called the **four-parameter logistic equation**. It fits four parameters: the bottom and top plateaus of the curve, the EC50 (or IC50), and the slope factor (Hill slope). This curve is symmetrical around its midpoint. To extend the model to handle curves that are not symmetrical, the Richards equation adds an additional parameter, S, which quantifies the asymmetry. This equation is sometimes referred to as a **five-parameter logistic equation**, abbreviated 5PL.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units.

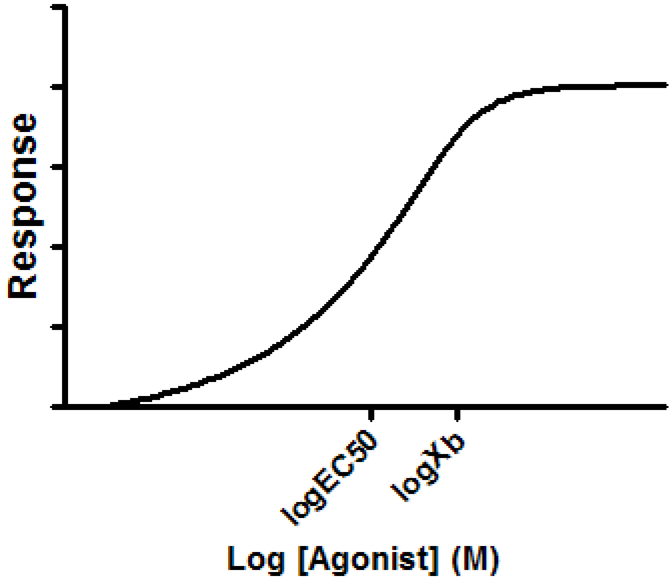
From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is log(concentration). Then choose *Asymmetrical (five parameter), X is log(concentration).*

Consider constraining the Hill Slope to a constant value of 1.0 (stimulation) or -1 (inhibition).

Also consider whether Bottom or Top should be fixed to constant values, or shared between data sets.

Model

```
LogXb = LogEC50 + (1/HillSlope)*Log((2^(1/S))-1)
Numerator = Top - Bottom
Denominator = (1+10^((LogXb-X)*HillSlope))^S
Y = Bottom + (Numerator/Denominator)
```



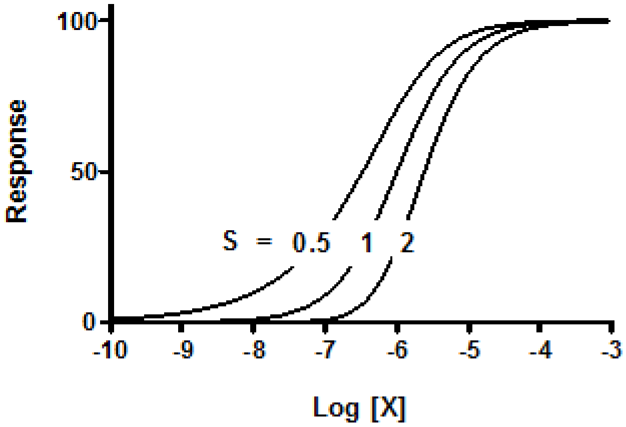
Interpret the parameters

Bottom and **Top** are the plateaus at the left and right ends of the curve, in the same units as Y.

LogEC50 is the concentrations that give half-maximal effects, in the same units as X. Note that the logEC50 is not the same as logXb.

HillSlope is the unitless slope factor or Hill slope. Consider constraining it to equal 1.0 (stimulation) or -1 (inhibition).

S is the unitless symmetry parameter. If S=1, the curve is symmetrical and identical to the standard dose-response equation. If S is distinct than 1.0, then the curve is asymmetric as shown below.



Notes

- If your goal is to obtain meaningful best-fit parameters, then you'll need lots of high quality data. It is very hard to fit both slope and asymmetry with tight confidence intervals. If your goal is just to interpolate unknowns from a standard curve, the width of the confidence intervals of the parameters doesn't really matter. What you want is a curve that follows the data, and in some cases an asymmetrical five parameter model does so better than a four parameter model.

- It can be tricky to get a good fit to the 5PL. See this discussion.

- Other formulations of asymmetrical dose-response curves have been developed. For example, Ricketts and Head developed a model for use in baroreflex studies.

- Bindslev has written a lengthy on-line text, Drug-Acceptor Interactions. Chapter 10, Hill in Hell discusses many models of dose-response curves, including asymmetrical ones.

- Liao and Liu have done simulations that show the advantage of fitting the EC50 rather Xb.

- Gottschalk and Dunn review the properties of the 5Pl.

- The equation built-in to Prism is only one of several ways to express a five parameter concentration-response curve.

References

Bindslev, Drug-Acceptor Interactions. Chapter 10, Hill in Hell

Cumberland, W.N., Fong, Y., Yu, X., Defawe, O., Frahm, N., and De Rosa, S. (2014). Nonlinear Calibration Model Choice between the Four and Five-Parameter Logistic Models. Journal of Biopharmaceutical Statistics 25: 972–983.

Giraldo, J., Vivas, N. M., Vila, E. & Badia, A. Assessing the (a)symmetry of concentration-effect curves: empirical versus mechanistic models. Pharmacol Ther 95, 21–45 (2002).

Gottschalk, P. G. & Dunn, J. R. The five-parameter logistic: a characterization and comparison with the four-parameter logistic. Anal Biochem 343, 54–65 (2005).

Liao, J.J.Z, Liu, R., Re-parameterization of five-parameter logistic function, Journal of Chemometrics, 23:248-253 (2009)

Ricketts, J. H. and Head, G.A. A five-parameter logistic equation for investigating asymmetry of curvature in baroreflex studies. Am. J. Physiology, 277: R441-54 (1999)

4.5.5.5.2 Equation: Biphasic dose-response

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 505 when X values are concentrations or doses.

Introduction

A common deviation from the standard monotonic sigmoid shape is the biphasic sigmoid shape.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units.

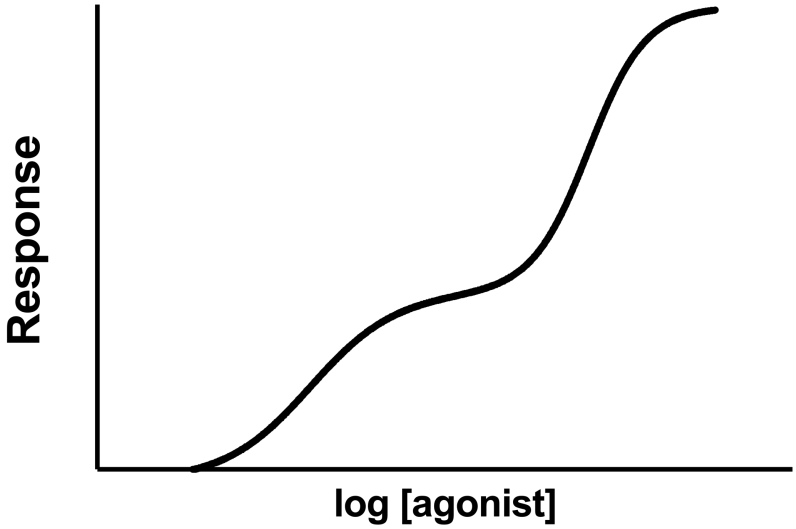
From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is log(concentration). Then choose *Biphasic dose-response, X is log(concentration).*

Consider constraining nH1 and nH2 to constant values of 1.0 (stimulation) or -1 (inhibition).

Also consider whether Bottom or Top should be fixed to constant values, or shared between data sets.

Model

```
Span=Top-Bottom
Section1=Span*Frac/(1+10^((LogEC50_1-X)*nH1))
Section2=Span* (1-Frac)/(1+10^((LogEC50_2-X)*nH2))
Y=Bottom + Section1 +Section2
```



Interpret the parameters

Bottom and **Top** are the plateaus at the left and right ends of the curve, in the same units as Y.

LogEC50_1 and **LogEC50_2** are the concentrations that give half-maximal stimulatory and inhibitory effects in the same units as X.

nH1 and **nH2** are the unitless slope factors or Hill slopes. Consider constraining these to equal 1.0 (stimulation) and -1 (inhibition).

Frac is the proportion of maximal response due to the more potent phase.

4.5.5.5.3 Equation: Bell-shaped dose-response

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 507 when X values are concentrations or doses.

Introduction

Some drugs may cause an inhibitory response at low concentrations, and a stimulatory response at high concentrations, or vice-versa. The net result is a bell-shaped dose-response curve.

The model explained here is the sum of two dose-response curves, one that stimulates and one that inhibits. you will need lots of data to determine all the parameters without ambiguity, so this model will rarely be useful for data analysis. But it might be useful as a way to draw a smooth curve through the data

Step by step

Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units.

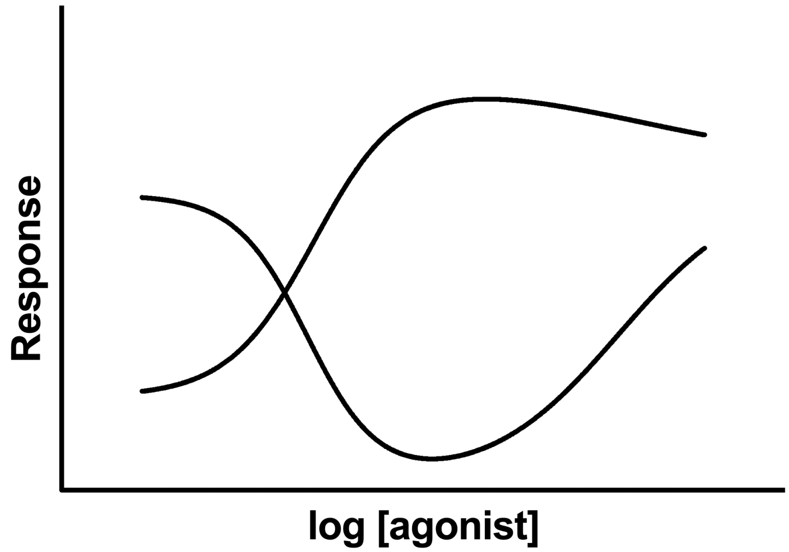
From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is log(concentration). Then choose *Bell-shaped dose-response, X is log(concentration).*

Consider constraining nH1 and nH2 to constant values of 1.0 (stimulation) and -1 (inhibition).

Model

```
Span1=Plateau1-Dip
Span2=Plateau2-Dip
Section1=Span1/(1+10^((LogEC50_1-X)*nH1))
```

```
Section2=Span2/(1+10^((X-LogEC50_2)*nH2))
Y=Dip+Section1+Section2
```



Interpret the parameters

Plateau1 and **Plateau2** are the plateaus at the left and right ends of the curve, in the same units as Y.

Dip is the plateau level in the middle of the curve, in same units as Y. Note that when the curve goes up first (then down), this parameter named *Dip* would better be named *Peak*.

LogEC50_1 and **LogEC50_2** are the concentrations that give half-maximal stimulatory and inhibitory effects in the same units as X.

nH1 and **nH2** are the unitless slope factors or Hill slopes. Consider constraining these to equal 1.0 (stimulation) and -1 (inhibition).

4.5.5.5.4 Equation: Operational model - Depletion

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 509 when X values are concentrations or doses.

Introduction to the operational model

The EC50 is determined by two properties of the agonist:

- How well it binds to the receptor, quantified by the **affinity** of the drug for binding to its receptor.

- How well it causes a response once bound. This property is known as the agonist's **efficacy**. Since efficacy depends on both agonist and tissue, a single drug acting on a single kind of receptor can have different efficacies, and thus different EC50 values, in different tissues.

A single dose-response experiment cannot determine affinity and efficacy. A drug that binds with high affinity but has low efficacy will produce exactly the same dose-response curve as a drug with low affinity and high efficacy.

To untangle affinity from efficacy, globally fit a dose-response curve of a full agonist and a second dose-response curve determined after treating the cells or tissue with an alkylating agent (or some other irreversible treatment) that reduces the number of accessible receptors. With fewer receptors, the dose response curve is shifted down and usually to the right.

The operational model assumes that the affinity of the drug for the receptors is not altered by reducing the number of available receptors. It also assumes that the maximum possible response in the tissue remains unchanged (the treatment was specific for the receptors you are studying). Accepting these assumptions, fitting the operational model globally will determine the affinity of the agonist for the receptors.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the agonist ligand into X. Enter response into Y in any convenient units. Enter data with a full agonist and no receptor depletion into column A. Enter data collected after receptor depletion into column B. Repeat, if you have data with different levels of receptor depletion for column C, D, E, ... You don't have to know the degree to which the receptors are depleted, and don't have to enter any values in the column titles (although they are useful as labels).

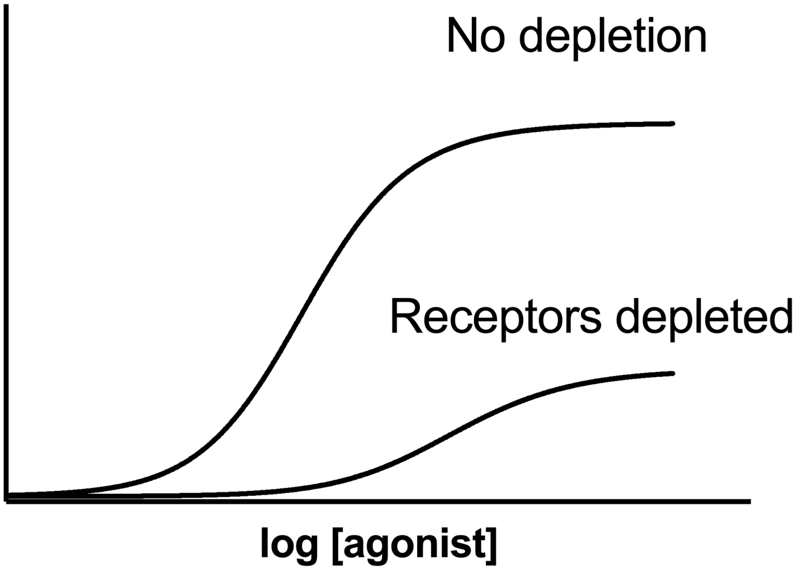
From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is log(concentration). Then choose *Operational Model - Depletion, X is log(concentration).*

If you have subtracted off any basal response, consider constraining the parameter Basal to a constant value of zero.

Also consider constraining the transducer slope n to a constant value of 1.0. When set to 1.0, all dose-response curves are constrained to have Hill slopes of 1.0, which is observed commonly.

Model

```
operate= (((10^logKA)+(10^X))/(10^(logtau+X)))^n
Y=Basal + (Effectmax-Basal)/(1+operate)
```



Interpret the parameters

Effectmax is the maximum possible system response, in units of the Y axis. It is the top plateau of the dose-response curve obtained with a full agonist without receptor depletion. If your agonist isn't a full agonist, the EffectMax might be higher than the top plateau of the no depletion curve.

Basal is the response in absence of agonist, in same units as Y. If you have subtracted off any basal response, constrain basal to a constant value of zero.

KA is the agonist-receptor dissociation constant, in same units as X (usually molar). It measures the affinity of the full agonist for the receptors, which is the main goal of this kind of experiment. Prism reports both KA and its logarithm. It is not the same as the EC50.

tau is the transducer constant, a practical measure of efficacy. It is the inverse of the fraction of receptors that must be occupied by agonist to obtain the half-maximal response. If tau equals 10, that means that occupation of only 10% of the receptors leads to a half-maximal response. If tau equals 1.0, that means that it requires occupation of all the receptors to give a half-maximal response. This would happen in a tissue where the receptors had been substantially depleted. Because tau is a property of both the tissue and receptor system, it is not a direct measure of intrinsic efficacy, which is commonly defined as a property belonging only to an agonist-receptor pair, irrespective of the assay system in which it is measured. Prism reports both tau and its logarithm for each data set.

n is the Unitless transducer slope. It is similar to, but not identical to, the Hill slope. In most cases, n is constrained to a constant value of 1.0, in which case all the dose-response curves will have Hill slopes of 1.0. If n does not equal 1.0, the Hill Slope does not equal either 1.0 or n.

Notes

Since Tau measures efficacy, Prism fits a different value of tau for each data set. Receptor depletion reduce the value of tau. The other parameters are fit globally, to find one best-fit value for all the data sets.

Reference

Black and Leff (Proc. R. Soc. Lond. B, 220: 141-162, 1983

4.5.5.5.5 Equation: Operational model - Partial agonist

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 512 when X values are concentrations or doses.

Introduction to the operational model

The EC50, fit by standard dose-response models, is determined by two properties of the agonist:

- How well it binds to the receptor, quantified by the **affinity** of the drug for binding to its receptor.

- How well it causes a response once bound. This property is known as the agonist’s **efficacy**. Since efficacy depends on both agonist and tissue, a single drug acting on a single kind of receptor can have different efficacies, and thus different EC50 values, in different tissues.

A single dose-response experiment cannot determine affinity and efficacy. A drug that binds tightly with high affinity but has low efficacy, will produce exactly the same dose-response curve as a drug with low affinity and high efficacy.

To determine the affinity of a partial agonist, use the operational model to globally fit the dose-response curves of both a full agonist and the partial agonist. The data from the full-agonist determines the maximum possible effect. Knowing that, the fitting can determine the affinity of the partial agonist.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the agonist ligand into X. Enter response into Y in any convenient units. Enter data with the full agonist into column A. Enter data collected with a partial agonist into column B. Repeat, if you have data with different partial agonists, for column C, D, E, ..., each with a different amount of depletion.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is log(concentration). Then choose *Operational Model - Partial agonist, X is log(concentration).*

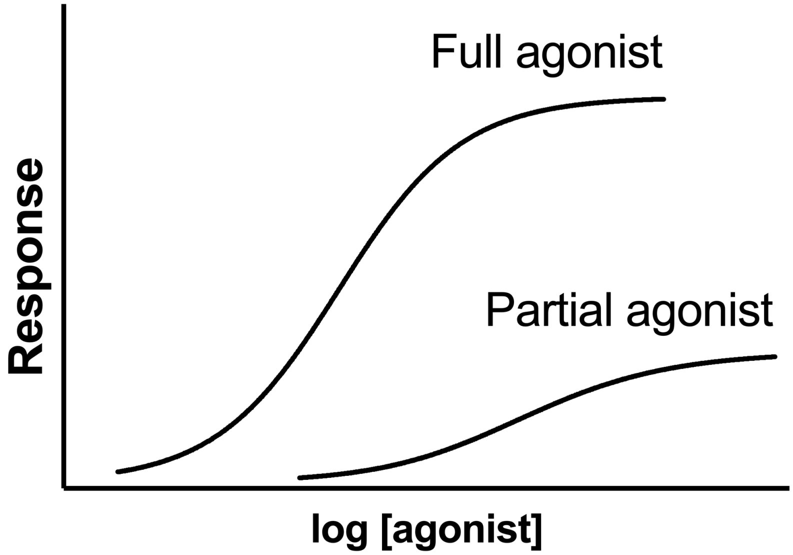
If you have subtracted off any basal response, consider constraining the parameter Basal to a constant value of zero.

Also consider constraining the transducer slope n to a constant value of 1.0. When set to 1.0, all dose-response curves are constrained to have Hill slopes of 1.0, which is observed commonly. If n is not 1.0, the Hill slopes will not be 1.0, but the Hill slopes will not equal exactly n.

Model

```
operate= (((10^logKA)+(10^X))/(10^(logtau+X)))^n
<A> Y = Basal + (Effectmax-Basal)/(1+10^((LogEC50-X)*n))
<~A> Y = Basal + (Effectmax-Basal)/(1+operate)
```

The second line is preceded with <A> which means it only applies to the first data set. It fits a variable slope dose-response curve. The third line is preceded with <~A> which means it applies to all data set except the first. It fits the operational model to determine the affinity (KA) of the partial agonist.



Interpret the parameters

Effectmax is the maximum possible system response, in units of the Y axis. It is the top plateau of the full agonist's dose-response curve.

Basal is the response in absence of agonist, in same units as Y. If you have subtracted off any basal response, constrain basal to a constant value of zero.

n is the Unitless transducer slope. It is similar to, but not identical to, the Hill slope. In most cases, n is constrained to a constant value of 1.0, in which case all the dose-response curves will have Hill slopes of 1.0. If n does not equal 1.0, the Hill Slope does not equal either 1.0 or n.

KA is the equilibrium dissociation constant of the partial agonist(s), in same units as X (usually molar). It measures the affinity of the partial agonist for the receptors, which is the main goal of this kind of experiment. Prism reports both KA and its logarithm. It is not the same as the EC50. By definition, it cannot be computed for the first data set (the full agonist).

tau is the transducer constant, a practical measure of efficacy. It is the inverse of the fraction of receptors that must be occupied by agonist to obtain the half-maximal response. If t equals 10, that means that occupation of only 10% of the receptors leads to a half-maximal response. If t equals 1.0, that means that it requires occupation of all the receptors to give a half-maximal response. This would happen with a partial agonist. Prism reports both tau and its logarithm, and fits tau individually for each data set. By definition, it cannot be computed for the first data set (the full agonist).

Reference

Black and Leff (Proc. R. Soc. Lond. B, 220: 141-162, 1983

4.5.5.5.6 Equation: Gaddum/Schild EC50 shift

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 515 when X values are concentrations or doses.

Introduction

A competitive inhibitor competes for agonist binding to a receptor, and shifts the dose-response curve to the right without changing the maximum response. By fitting all the curves globally, you can determine the affinity of the competitive inhibitor.

Step by step

1. Create an XY data table with subcolumns to match your experimental design.

2. Enter the logarithm of the concentration of the agonist ligand into X. For example, if a concentration is 1nM, enter -9. Don't enter 1e-9.

3. Enter response into Y in any convenient units. Enter data with no inhibitor into column A. Enter data collected with a constant concentration of inhibitor into column B. Repeat, if you have data, for column C, D, E, ..., each with a different concentration of inhibitor.

4. Enter the inhibitor concentration in molar as the column titles of the data table. If the concentration is 1nM enter '1e-9' or '0.00000001' into the column titles. Don't enter '-9'. Don't forget to enter '0' as the column title for data set A, since these are the control data with no inhibitor. These values entered into the column title are used in the analyis; they are more than just labels, so need to be entered in the correct format.

5. From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is log(concentration). Then choose *Gaddum/Schild EC50 shift, X is log(concentration).*

6. Consider constraining the parameters HillSlope and SchildSlope to their standard values of 1.0. This is especially useful if you don't have many data points, and therefore cannot fit these parameters.

Notes on units

- This equation was written so the X values are entered as logarithm of concentration of the agonist, while the column titles are entered as concentration of antagonist. This is simply the way this equation was written. It would be easy enough to clone the equation and write it with different conventions if you wanted to. Prism also offers a related equation 515 in which the X values are entered as the concentration of the agonist.

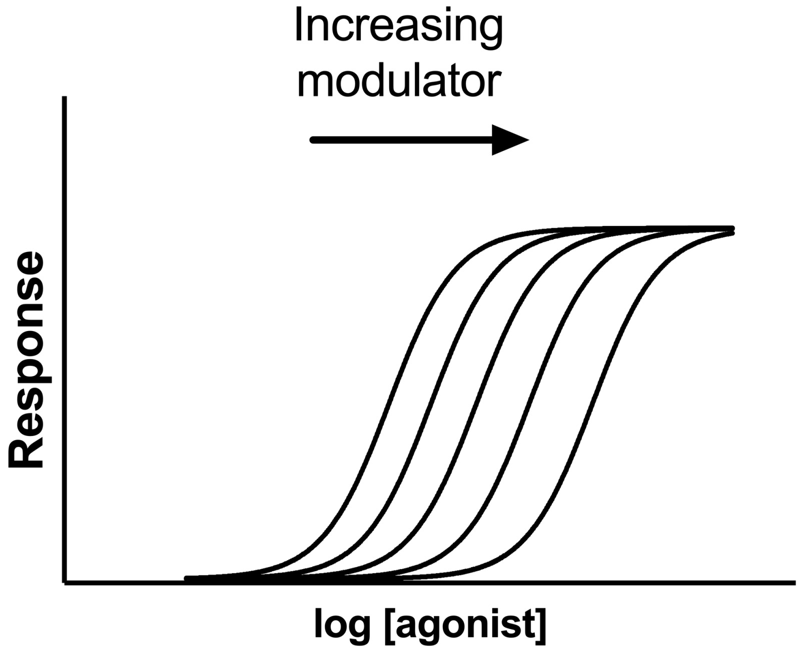
- The EC50 is reported in the antilog of the units you used to enter X values. If you entered X as log(molar), then the EC50 will be in molar. If you entered X values as log(dose in milligrams), then the EC50 will be in milligrams.

- The pA2 is the negative logarithm of the concentration of antagonist needed to shift the curve by a factor of 2. That concentration is in the same units you used to enter the antagonist concentration into the column titles. If you entered concentration in nM into the column titles, the pA2 is the negative logarithm of nM. If you entered concentration as a dose in mg/kg, then the pA2 is the negative logarithm of mg/kg.

- If you want to transform the pA2 to a concentration, first flip the sign (multiply by -1) and then take the antilog (ten to that power). If you also do so for the confidence limits, note that flipping the sign also changes the order of the limits, so the "lower" confidence limit is actually the upper limit, and the "upper" limit is actually the lower limit.

Model

```
EC50=10^LogEC50
Antag=1+(B/(10^(-1*pA2)))^SchildSlope
LogEC=Log(EC50*Antag)
Y=Bottom + (Top-Bottom)/(1+10^((LogEC-X)*HillSlope))
```



Interpret the parameters

EC50 is the concentration of agonist that gives half maximal response in the absence of inhibitor. Prism reports both the EC50 and its log. The logEC50 is in the same units you used to enter the X values. The EC50 is in the antilog of those units.

pA2 is the negative logarithm of the concentration of antagonist needed to shift the dose response curve by a factor of 2. If the HillSlope and

SchildSlope are fixed to 1.0, it is the pKb, the negative log of the equilibrium dissociation constant (Molar) of inhibitors binding to the receptors. The units are the negative log of whatever units you used to enter concentrations as column titles on the data table.

HillSlope describes the steepness of the family of curves. A HillSlope of 1.0 is standard, and you should consider constraining the Hill Slope to a constant value of 1.0.

SchildSlope quantifies how well the shifts correspond to the prediction of competitive interaction. If the competitor is competitive, the SchildSlope will equal 1.0. You should consider constraining SchildSlope to a constant value of 1.0. antagonist term, [B], is now raised to the power S, where S denotes the Schild slope factor. If the shift to the right is greater than predicted by competitive interactions, S will be greater than 1. If the rightward shift is less than predicted by competitive interaction, then S will be less than 1.

Top and **Bottom** are plateaus in the units of the Y axis.

Notes:

- All six of these are shared among all the data sets, so you will only see one best fit value for each parameter, not one for each data set.

- Prism does not report the EC50 for each curve, only the EC50 for the first curve (the one without any antagonist).

- The variable B in the model is defined to be a data set constant whose value comes from the column titles. The results page will show these values of B. Make sure they are the concentration of antagonist used in each data set (and not the log of concentrations).

When is the Schild model valid?

Colquhoun (1) has shown that the Schild model is valid whenever you can make these assumptions:

- The antagonist, B, is a true antagonist that, alone, does not change the conformation of the receptor.

- Binding of agonist, A, and antagonist, B, is mutually exclusive at every binding site.
- B has the same affinity for every binding site.
- The observed response is the same if the occupancy of each site by A is the same, regardless of how many sites are occupied by B.
- Measurements are made at equilibrium.

1. Colquhoun, D.. *Why the Schild method is better than Schild realised*. Trends Pharmacol Sci (2007) vol. 28 (12) pp. 608-14

4.5.5.5.7 Equation: EC50 shift

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 519 when X values are concentrations or doses.

Introduction

A competitive inhibitor competes for agonist binding to a receptor, and shifts the dose-response curve to the right without changing the maximum response. This model fits two dose response curves and determines the fold shift.

Step by step

1. Create an XY data table.
2. Enter the logarithm of the concentration of the agonist ligand into the X column.
3. Enter response into Y in any convenient units. Enter data with no inhibitor into column A. Enter data collected with a constant concentration of inhibitor into column B. You can label columns A and

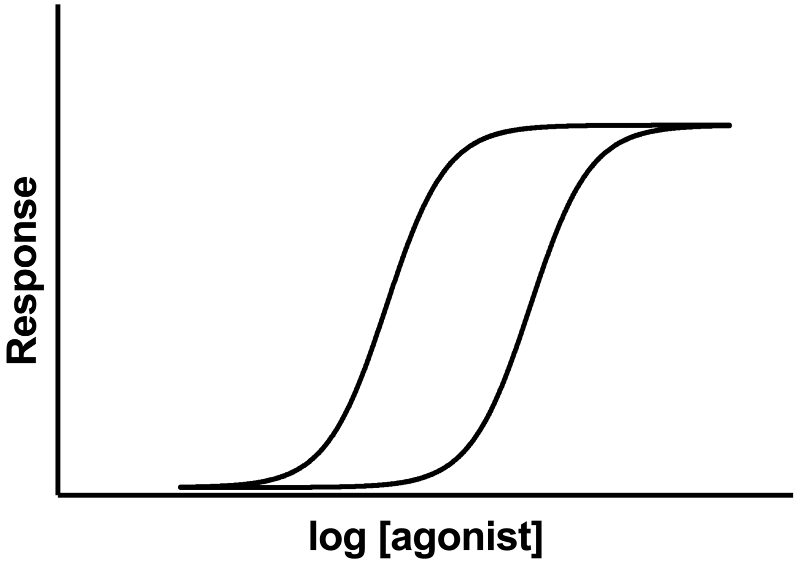
B with any column title you like. The column title does not get used as part of the analysis.

4. From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is log(concentration). Then choose *EC50 shift, X is log(concentration).*

5. If you have subtracted off any baseline signal, constrain the parameter Bottom to a constant value of zero.

Model

```
<A>LogEC=LogEC50Control
<~A>LogEC=LogEC50Control + log(EC50Ratio)
Y=Bottom + (Top-Bottom)/(1+10^((LogEC-X)*HillSlope))
```



EC50Control is the concentration of agonist that gives half maximal response in the absence of modulator.

Top and **Bottom** are plateaus in the units of the Y axis (shared).

EC50Ratio is the ratio of EC50 in presence of inhibitor divided by EC50 of agonist alone.

HillSlope is the slope factor (shared)

Notes

- If you have several concentrations of antagonist, use a different model that will directly fit the Schild model (using either X as concentration of agonist 515 or X as the logarithm of the concentration of agonist 541) and determine the pA2.

- If you want to test the null hypothesis that the EC50 ratio is 1.0 (no shift), that is easy to do. Go to the Compare tab, choose to test whether a selected parameter differs from a hypothetical value, choose the parameter EC50Ratio, and enter the hypothetical value 1.0.

- This model shares Top, Bottom and HillSlope. So it assumes that both models are exactly the same except (perhaps) for the EC50. You can go to the Constrain tab and lift this constraint for any or all of these three parameters, if you think it makes sense not to share them.

4.5.5.5.8 Equation: Allosteric EC50 shift

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 521 when X values are concentrations or doses.

Introduction

An allosteric modulator can reduce or enhance agonist binding. This model fits entire dose-response curves determined in the absence and presence of a modulator. The goal is to learn the affinity of the modulator for binding to its site, and also determine the value of alpha, the ternary complex constant that quantifies the degree to which binding of the modulator alters the affinity of the radioligand for the receptor site.

Step by step

1. Create an XY data table.

2. Enter the logarithm of the concentration of the agonist ligand into X. If the concentration is 1nM, enter -9.

3. Enter response into Y in any convenient units. Enter data with no modulator into column A. Enter data collected with a constant concentration of modulator into column B. Repeat, if you have data, for column C, D, E, ..., each with a different concentration of modulator.

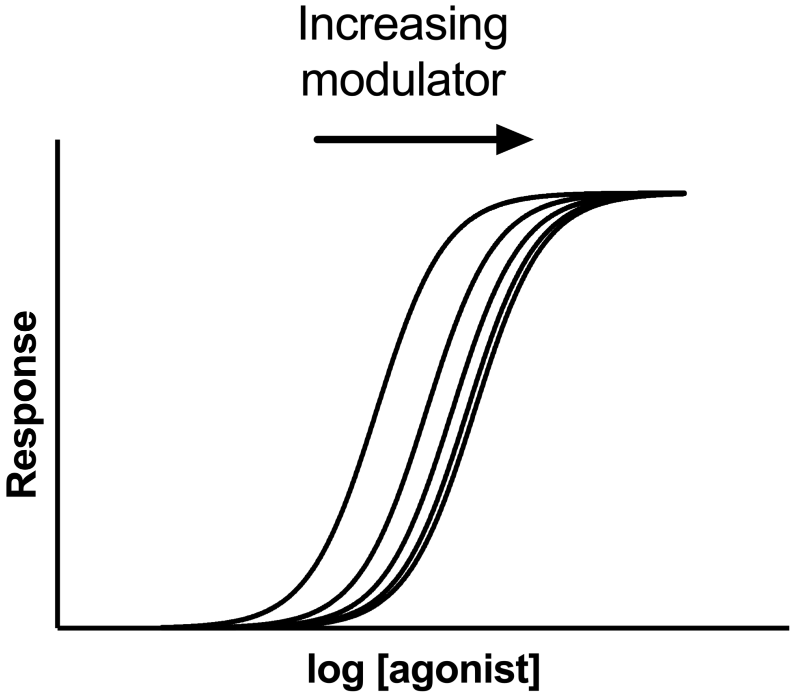
4. Enter the modulator concentration (in molar so 1nM is entered as '1e-9') into the column titles. Don't forget to enter '0' as the column title for data set A. These column titles are not just labels. The values you enter become part of the analysis.

5. From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is log(concentration). Then choose *Allosteric EC50 shift, X is log(concentration).*

You do not need to constrain any parameters to constant values

Model

```
EC50=10^LogEC50
KB=10^LogKB
alpha=10^Logalpha
Antag=(1+B/KB)/(1+alpha*B/KB)
LogEC=Log(EC50*Antag)
Y=Bottom+(Top-Bottom)/(1+10^((LogEC-X)*HillSlope))
```



logEC50 is the logarithm of the concentration of agonist that gives half maximal response in the absence of modulator.

logKb is the logarithm of the equilibrium dissociation constant (Molar) of modulator binding to its allosteric site. It is in the same molar units used to enter the modulator concentration into column titles on the data table.

log Alpha is the logarithm of the ternary complex constant. When alpha=1.0, the modulator won't alter binding. If alpha is less than 1.0, then the modulator reduces ligand binding. If alpha is greater than 1.0, then the modulator increases binding. In the example shown about, alpha equals 0.01 so the modulator greatly decreases binding.

Top and **Bottom** are plateaus in the units of the Y axis.

Notes

- This model is designed to analyze data when the modulator works via an allosteric site. Since the agonist and modulator are acting via different sites, it is incorrect to refer to the modulator as a competitor.

- The model is written to fit the logarithm of alpha, rather than alpha itself. This is because alpha is asymmetrical: All values from 0 to 1 mean that the modulator decreases binding, while all values from 1 to infinity mean that the modulator enhances binding. On a log scale, its values are more symmetrical, so the confidence interval computed on a log scale (as Prism does) are more accurate. Prism reports both alpha and log(alpha).

- This model assumes that the allosteric modulator is present in excess, so the concentration you added is very close to its free concentration. This model won't work when the concentration of allosteric modulator is limiting (as it is when G proteins alter agonist binding to many receptors). No explicit model can handle this situation. You need to define the model with an implicit equation (Y on both sides of the equals sign) and Prism cannot handle such equations.

- Note two important points about the progression of curves from left to right with increasing concentrations of the allosteric modulator. First, note that the maximum response doesn't change. Second, note that the effect of the modulator to right-shift the dose-response curve reaches a maximum as the modulator saturates its binding site.

Reference

A. Christopoulos and T. Kenakin, Pharmacol Rev, 54: 323-374, 2002

4.5.5.5.9 Equation: ECanything

This equation is used when X values are logarithms of doses or concentrations. Use a related equation 524 when X values are concentrations or doses.

Introduction

Many log(dose) response curves follow the familiar symmetrical sigmoidal shape. The usual goal is to determine the EC50 of the agonist - the concentration that provokes a response half way between the basal (Bottom) response and the maximal (Top) response. But you can determine any spot along the curve, say a EC80 or EC90.

Many dose-response curves have a standard slope of 1.0. This model does not assume a standard slope but rather fits the Hill Slope from the data. Hence the name **Variable slope** model. This is preferable when you have plenty of data points.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is log(concentration). Then choose "*log(Agonist) vs. response -- Find ECanything*"*.*

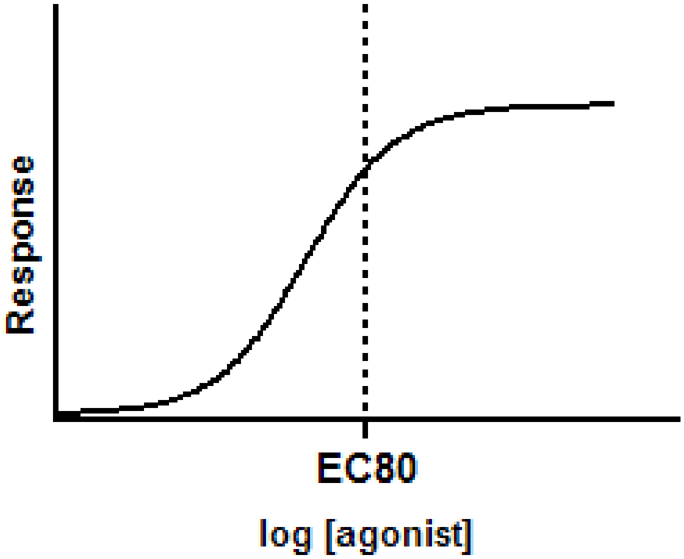
You must constrain the parameter F to have a constant value between 0 and 100. Set F to 80 if you want to fit the EC80. If you constrain F to equal 50, then this equation is the same as a variable slope dose-response curve.

Consider constraining the parameter HillSlope to its standard value of 1.0 of -1. This is especially useful if you don't have many data points, and therefore cannot fit the slope very well.

If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Model

```
logEC50=logECF - (1/HillSlope)*log(F/(100-F))
Y=Bottom + (Top-Bottom)/(1+10^((LogEC50-X)*HillSlope))
```



Interpret the parameters

ECf is the concentration of agonist that gives a response F percent of the way between Bottom and Top. Prism reports both the ECF and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of 1.0 is standard, and you should consider constraining the Hill Slope to a constant value of 1.0. A Hill slope greater than 1.0 is steeper, and a Hill slope less than 1.0 is shallower.

Top and **Bottom** are plateaus in the units of the Y axis.

Adapting this equation

ICanything

This equation can also fit inhibitory data where the curve goes downhill rather than uphill. The best-fit value of the Hill Slope will be negative in this case.The result will always be reported as ECf. Let's say you set F=80. Then the ECf for inhibitory data would be the concentration (X value) required to bring the curve down to 80%. If you want the concentration that brings the curve down by 80%, to 20%, then you'd need to set F equal to 20. If you want the result to say "ICF" rather than "ECF", clone the equation and then edit to make that change

X is concentration rather than log concentration

Prism has a related equation 524 built-into its equation library that will allow you to fit the ECF (or ICF) using values where X represents the logarithm of the concentration of the agonist instead of the concentration of the agonist. Both equations will report the ECF as well as the log(ECF).

4.5.5.5.10 Equation: Absolute IC50

This equation is used when X values are logarithms of concentrations or doses. Use a related equation 526 when X values are concentrations or doses.

Introduction

Many dose-response curves follow a familiar sigmoidal shape. These curves are often defined by four parameters: Top, Bottom, Hill slope, and the EC50 (or IC50). The Top and Bottom parameters describe the values at which the curve reach a plateau - coming infinitely close, but never quite reaching these values. The Hill slope describes the slope of the sigmoidal curve between these two plateaus. The EC50 (or IC50) refer to a concentration of agonist (or antagonist) required to increase (or reduce) the measured response to half - or 50% - of its maximal value.

However, there are a number of ways that this concept of "half" can be interpreted. Are you interested in the concentration that results in a measured response halfway between the maximal (Top) and minimal (Bottom) values of the curve? Or do you want the concentration that results in a measured response halfway between the maximal (Top) value of the curve and some defined "baseline" control value?

When considering inhibitory data (measured response decreases with increasing concentration of antagonist), these two values are referred to as the **Relative IC50** and **Absolute IC50**, respectively.

Relative IC50

This is by far the most common definition, and is usually what people mean by "the IC50" of a response. It is the concentration required to bring the curve down to a point halfway between the "Top" and "Bottom" plateaus of the curve. Unless the data are normalized to run from 0 to 100, this is **not** the value that corresponds to 50% on the Y axis.

Absolute IC50

The concentration that provokes a response halfway between the blank (no antagonist, maximum measured response) and some positive control (to represent a fully inhibited response) is sometimes called the Absolute IC50. In many cases, the positive control used may be a different, standard drug known to elicit a maximal inhibitory effect on the measured response. Often, the response measured for this standard positive control is lower than the "Bottom" of the dose-response curve. As a result, the Absolute IC50 and Relative IC50 *will not be the same*.

In some cases, an Absolute IC50 simply cannot be calculated. For example, the antagonist being tested may only capable of reducing a measured response to 60% of its maximal (blank) value, but no further. In this case, the Absolute IC50 would not be defined (no increase in the concentration of antagonist would be capable of lowering the measured response to 50% of the maximal response). Be sure to know which IC50 you want to report, and what your values mean.

Read more about Absolute IC50 here.

Step by step

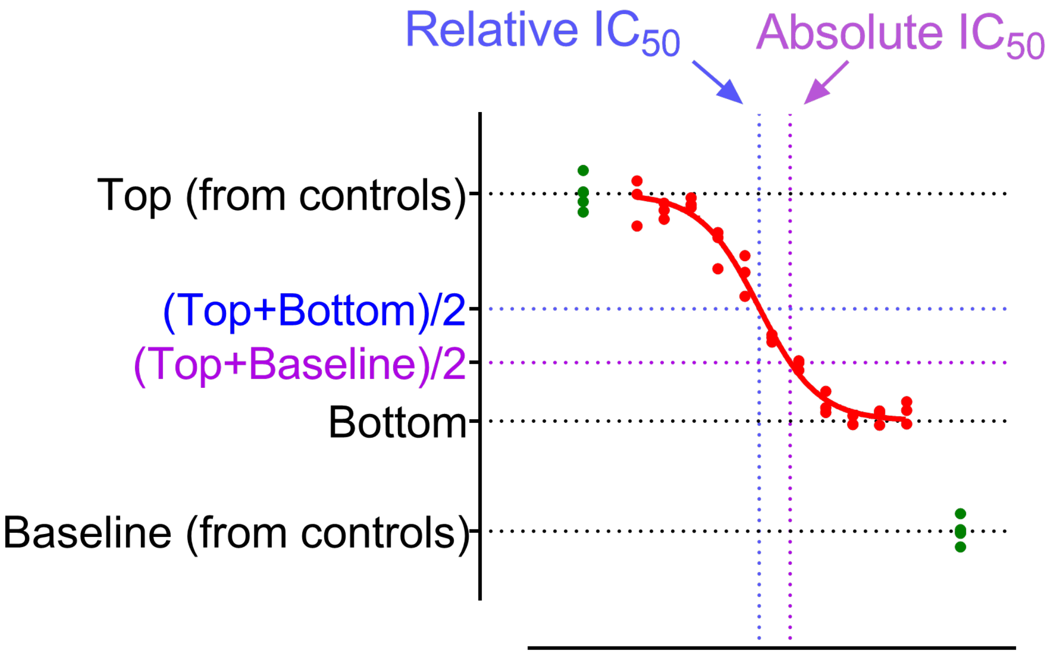
Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special, X is log(concentration). Then choose "*Absolute IC50, X is log(concentration)*"*.*

You must enter a value for the parameter "Baseline" on the Constrain tab of the analysis parameters dialog. Note the important distinction between "Baseline" and "Bottom" in this model. "Bottom" is the Y value of the bottom plateau of the curve itself. "Baseline" is the Y value that defines 0% - maximal inhibition by a standard drug. While you *must* constrain the baseline, you may want to consider constraining the "Top" parameter as well.

Model

```
Fifty=(Top+Baseline)/2
Y= Bottom + (Top-Bottom)/(1+10^((LogAbsoluteIC50-X)*HillSlope + log((Top-Botto
```



Interpret the parameters

Top and **Bottom** are plateaus in the units of the Y axis.

HillSlope describes the steepness of the family of curves. A HillSlope of 1.0 is standard, and you should consider constraining the Hill Slope to a constant value of 1.0. A Hill slope greater than 1.0 is steeper, and a Hill slope less than 1.0 is shallower.

Baseline is the measured response of a "standard" drug or control resulting in a maximally inhibited response. This value must be entered by the user in the constrain tab of the analysis parameters dialog.

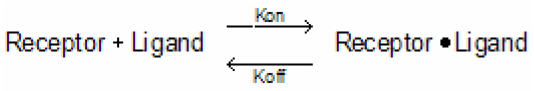
LogAbsoluteIC50 is the log concentration of antagonist that gives a response half way between Top and Baseline. This is not the same as the response half way between Top and Bottom (this is the relative logIC50, and is not reported by this analysis). Prism also reports the **AbsoluteIC50**.

4.5.5.6 Receptor binding - Key concepts

4.5.5.6.1 Law of mass action

What is the law of mass action?

Analysis of radioligand binding experiments is based on a simple model, called the law of mass action. This model assumes that binding is reversible.



Binding occurs when ligand and receptor collide due to diffusion, and when the collision has the correct orientation and enough energy. The rate of association is:

Number of binding events per unit of time =[Ligand]·[Receptor]·kon.

Once binding has occurred, the ligand and receptor remain bound together for a random amount of time. The probability of dissociation is the same at every instant of time. The receptor doesn't "know" how long it has been bound to the ligand. The rate of dissociation is:

Number of dissociation events per unit time = [ligand×receptor]×koff.

After dissociation, the ligand and receptor are the same as at they were before binding. If either the ligand or receptor is chemically modified, then the binding does not follow the law of mass action.

Equilibrium is reached when the rate at which new ligand×receptor complexes are formed equals the rate at which the ligand×receptor complexes dissociate. At equilibrium:

$$[\mathrm{Ligand}]\cdot[\mathrm{Receptor}]\cdot k_{on} = [\mathrm{Ligand}\cdot\mathrm{Receptor}]\cdot k_{off}$$

Meaning of Kd

Rearrange that equation to define the equilibrium dissociation constant Kd.

$$\frac{[\mathrm{Ligand}]\cdot[\mathrm{Receptor}]}{[\mathrm{Ligand}\cdot\mathrm{Receptor}]} = \frac{k_{off}}{k_{on}} = K_d$$

The Kd has a meaning that is easy to understand. Set [Ligand] equal to Kd in the equation above. The Kd terms cancel out, and you will see that [Receptor]/ [Ligand×Receptor]=1, so [Receptor] equals [Ligand×Receptor]. Since all the receptors are either free or bound to ligand, this means that half the receptors are free and half are bound to ligand. In other words, when the concentration of ligand equals the Kd, half the receptors will be occupied at equilibrium. If the receptors have a high affinity for the ligand, the Kd will be low, as it will take a low concentration of ligand to bind half the receptors.

The term "dissociation constant"

Don't mix up Kd, the equilibrium dissociation constant, with koff, the dissociation rate constant. They are not the same, and aren't even expressed in the same units.

Variable	Name	Units
kon	Association rate constant or on-rate constant	M^{-1} min^{-1}

koff	Dissociation rate constant or off-rate constant	min^{-1}
Kd	Equilibrium dissociation constant	M

Fractional occupancy

The law of mass action predicts the fractional receptor occupancy at equilibrium as a function of ligand concentration. Fractional occupancy is the fraction of all receptors that are bound to ligand.

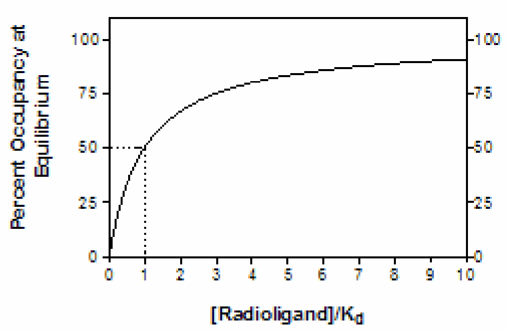
$$\text{Fractional occupancy} = \frac{[\text{Ligand} \cdot \text{Receptor}]}{[\text{Total Receptor}]} = \frac{[\text{Ligand} \cdot \text{Receptor}]}{[\text{Receptor}] + [\text{Ligand} \cdot \text{Receptor}]}$$

This equation is not useful, because you don't know the concentration of unoccupied receptor, [Receptor]. A bit of algebra creates a useful equation.

$$\text{Fractional occupancy} = \frac{[\text{Ligand}]}{[\text{Ligand}] + K_d}$$

This equation assumes equilibrium. To make sense of it, think about a few different values for [Ligand].

[Ligand]	**Fractional Occupancy**
0	0%
1.Kd	50%
4.Kd	80%
9.Kd	90%
99.Kd	99%



Note that when [Ligand]=Kd, fractional occupancy is 50%.

Assumptions

Although termed a "law", the law of mass action is simply a model that can be used to explain some experimental data. Because it is so simple, the model is not useful in all situations. The model assumes:

- All receptors are equally accessible to ligands.
- Receptors are either free or bound to ligand. It doesn't allow for more than one affinity state, or states of partial binding.
- Binding does not alter the ligand or receptor.
- Binding is reversible.

Despite its simplicity, the law of mass action has proven to be very useful in describing many aspects of receptor pharmacology and physiology.

4.5.5.6.2 Nonspecific binding

In addition to binding to receptors of interest, radioligands also bind to other sites. Binding to the receptor of interest is called specific binding, while binding to the other sites is called nonspecific binding. This means that nonspecific binding can represent several phenomena:

- In most cases, the bulk of nonspecific binding represents some sort of interaction of the ligand with membranes. The molecular details are unclear, but nonspecific binding depends on the charge and hydrophobicity of a ligand – but not its exact structure.
- Nonspecific binding can also be binding to receptors, transporters, or other proteins not of interest to the investigator. For example binding of the adrenoceptor agonist, epinephrine, to serotonin receptors or metabolic enzymes can be considered “nonspecific”.
- Nonspecific binding can also be binding to the filters used to separate bound from free ligand.

Nonspecific binding is usually (but not necessarily) proportional to the concentration of radioligand (within the range it is used). Add twice as much radioligand, and you will see twice as much nonspecific binding.

Nonspecific binding is detected by measuring radioligand binding in the presence of a saturating concentration of an unlabeled drug that binds to the receptors. Under those conditions, virtually all the receptors are occupied by the unlabeled drug so the radioligand can only bind to nonspecific sites. Subtract the nonspecific binding at a particular concentration of radioligand from the total binding at that concentration to calculate the specific radioligand binding to receptors.

Which unlabeled drug should you use for determining nonspecific binding? The obvious answer is to use the same compound as the radioligand, but in its unlabeled form. In many cases this is necessary, as no other drug is known to bind to the receptors. But most investigators avoid using the same compound as the hot and cold ligand and prefer to define nonspecific binding with a drug that is chemically distinct from the radioligand but which binds to the same receptor.

What concentration of unlabeled drug should you use? You want to use enough to block virtually all the specific radioligand binding, but not so much that you cause more general physical changes to the membrane that might alter binding. If you are studying a well-characterized receptor, a useful rule-of-thumb is to use the unlabeled compound at a concentration equal to 100 times its Kd for the receptors, or 100 times the highest concentration of radioligand, whichever is higher.

Ideally, you should get the same results defining nonspecific binding with a range of concentrations of several drugs, and you should test this when possible. In many assay systems, nonspecific binding is only 10-20% of the total radioligand binding. If the nonspecific binding makes up more than half of the total binding, you will find it hard to get quality data. If your system has a lot of nonspecific binding, try different kinds of filters, a larger volume of washing buffer, warmer washing buffer, or a different radioligand.

4.5.5.6.3 Ligand depletion

In many experimental situations, you can assume that a very small fraction of the ligand binds to receptors (or to nonspecific sites). In these situations, you can also assume that the free concentration of ligand is approximately equal to the concentration you added. This assumption vastly simplifies the analysis of binding experiments, and the standard analysis methods depend on this assumption.

In other situations, a large fraction of the ligand binds to the receptors (or binds nonspecifically). This means that the concentration of ligand free in the solution does not equal the concentration you added. The discrepancy is not the same in all tubes or at all times. The free ligand concentration is depleted by binding.

Many investigators use this rule of thumb: If less than 10% of the ligand binds, don't worry about ligand depletion; if more than 10% of the ligand binds, you have three choices:

- Change the experimental conditions. Increase the reaction volume without changing the amount of tissue. The problem with this approach is that it requires more radioligand, which is usually very expensive.
- Measure the free concentration of ligand in every tube. This is possible if you use centrifugation or equilibrium dialysis, but is quite difficult if you use vacuum filtration to remove free radioligand.
- Use analysis techniques that adjust for the difference between the concentration of added ligand and the concentration of free ligand. Prism includes such models for analyzing saturation 566 and competition 592 data. These special analyses only work with radioactive ligands, so the assessment of added ligand and bound ligand are in the same counts-per-minute units. These methods don't work with fluorescent ligands.

4.5.5.6.4 The radioactivity web calculator

GraphPad Software provides a free radioactivity calculator on graphpad.com. Use it to perform seven common calculations.

Calculation	Description
Isotope decay	Calculates radioactive decay during a specified number of days. Select one of the common isotopes, or enter the half-life of another isotope.
Conc. of stock	Enter mCi/ml and Ci/mmole, which should be on the label. If you are using a molecule labeled with

Calculation	Description
	125I, the specific activity equals 2200 Ci/mmole if each molecule is labeled with one iodine. Also enter the percent of the original isotope remaining (calculated above). The calculations assume that the decay product is not biologically active, so the concentration of stock that is biologically active decreases over time.
Dilution of stock	Enter the concentration in your stock solution, after accounting for decay. Also enter the concentration and volume you want. The result is the volume of stock you need to use.
Specific activity (cpm/fmol)	Enter the specific radioactivity as Ci/mmol which should be on the label. If you are using a molecule labeled with 125I, the specific activity equals 2200 Ci/mmol if each molecule is labeled with one iodine. Also enter the counter efficiency - the fraction of radioactive disintegrations that are detected. The efficiency depends on the isotope and instrumentation. With low energy isotopes such as tritium, the efficiency also depends on the experimental details such as the choice of scintillation fluid, the amount of water in the sample, and the presence of any colored substances in the sample.
Cpm to fmol/mg	Enter the specific radioactivity as cpm/fmol, the number of cpm counted, and the protein content of the sample in mg. The result is the number of binding sites in fmol/mg protein.
Cpm to sites/cell	Enter the specific radioactivity as cpm/fmol, the number of cpm counted, and the cell count. The result is the number of binding sites per cell.
Cpm to nM	Enter the specific radioactivity as cpm/fmol, the number of cpm counted, and the volume

Calculation	Description
	counted. The result is the concentration of radioligand in nM.

4.5.5.7 Receptor binding - Saturation binding

4.5.5.7.1 Key concepts: Saturation binding

What is saturation binding?

In a saturation binding experiment, you vary the concentration of radioligand and measure binding at equilibrium. The goal is to determine the Kd (ligand concentration that binds to half the receptor sites at equilibrium) and Bmax (maximum number of binding sites).

Total, nonspecific and specific binding

The ligand binds not only to receptors sites, but also to nonspecific sites. There are three approaches to dealing with nonspecific binding.

- Subtract off the nonspecific, and analyze only the specific binding 568.
- Analyze the total binding only 563, inferring the amount of nonspecific binding from the shape of the total binding curve.
- Globally fit total and nonspecific binding together 564.

We recommend the third approach (global fitting of total and nonspecific). The problem with fitting specific binding is that you have to make some assumptions in order to subtract nonspecific from total, and the resulting values that you fit aren't really data. When possible, we suggest that you fit the data you actually collect, and avoid creating derived data sets (specific binding, in this case).

Fitting total binding only requires less data, so saves experimental time and money. But most people feel unconformable defining nonspecific binding purely from the shape of a binding curve, without experimentally measuring nonspecific binding. One advantage of fitting total binding only is that equations have been derived for fitting such data, even when a substantial fraction of the ligand binds, resulting in ligand depletion 566 (free concentration substantially less than the added concentration).

One-vs. two sites

Prism offers models for fitting one or two sites. You can use choices in the Compare tab to compare the two fits. When comparing the fits to the one- and two-site models, use common sense as well as statistics. Don't accept a two site model, if one of the sites is only a tiny fraction of the total, or if its Kd is outside the range of radioligand concentrations you used in the experiment.

4.5.5.7.2 Equation: One site -- Total binding

Introduction

You don't have to measure nonspecific binding directly. Instead, you can determine Bmax and Kd by fitting only total binding by assuming that the amount of nonspecific binding is proportional to the concentration of radioligand.

Step by step

Create an XY data table. Enter radioligand concentration into X, and total binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

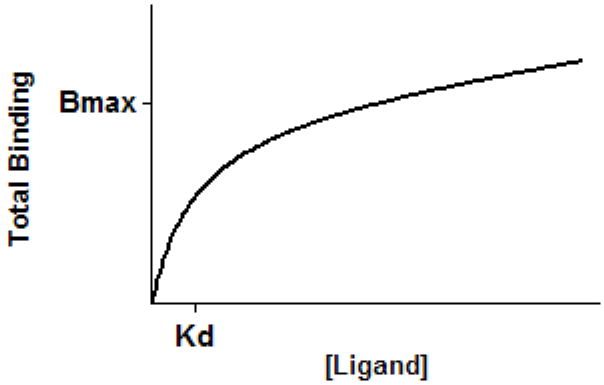
Use any convenient units for X and Y. The Kd will be reported in the same units as X, and the Bmax will be reported in the same units as Y.

From the table of total binding, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *One site -- Total.*

Consider constraining the parameter Background to a constant value of zero. This is the measured 'binding' when there is no radioligand binding added, so represents the counter background, if there is any.

Model

```
Y=Bmax*X/(Kd+X) + NS*X + Background
```



Interpret the parameters

Bmax is the maximum specific binding in the same units as Y.

Kd is the equilibrium dissociation constant, in the same units as X. It is the radioligand concentration needed to achieve a half-maximum binding at equilibrium.

NS is the slope of nonspecific binding in Y units divided by X units.

Background is the amount of nonspecific binding with no added radioligand. This represents counter background. If your counter automatically subtracts off the background signal, you can constrain Background to a constant value of zero.

Notes

This analysis assumes that only a small fraction of radioligand binds, which means that the concentration you added is virtually identical to the free concentration. If you can't make this assumption, use an alternative analysis 566.

4.5.5.7.3 Equation: One site -- Fit total and nonspecific binding

Introduction

In a saturation binding experiment, you vary the concentration of radioligand and measure binding. The goal is to determine the Kd (ligand concentration that binds to half the receptor sites at equilibrium) and Bmax (maximum number of binding sites).

The ligand binds not only to receptors sites, but also to nonspecific sites. There are three approaches to dealing with nonspecific binding.

- Subtract off the nonspecific, and analyze only the specific binding 568.
- Analyze the total binding only, inferring the amount of nonspecific binding from the shape of the total binding curve. Learn more 563.
- Globally analyze the total and nonspecific binding at one time. This is the best approach, and the details are explained below.

Step by step

Create an XY data table. Enter radioligand concentration into X, total binding into Y, and nonspecific binding into column B.

Use any convenient units for X. The Kd will be reported in those same concentration units. Use the same units for total and nonspecific binding. The Bmax will be reported in those same units.

Alternatively choose the sample data set: *Binding - Saturation binding to total and nonspecific.*

From the data table, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *One site -- Total and nonspecific binding.*

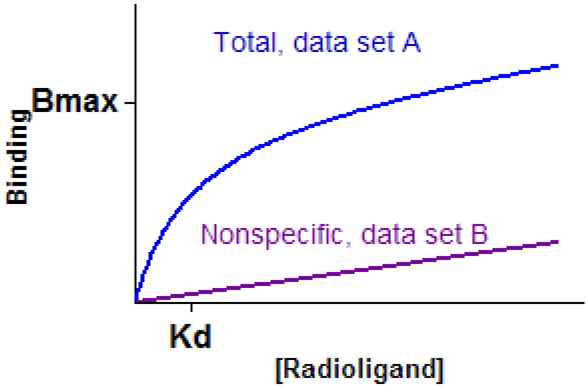
Consider constraining the parameter Background to a constant value of zero. This is the measured 'binding' when there is no radioligand binding added, so represents the counter background, if there is any.

Model

```
specific=Bmax*X/(X+Kd)
nonspecific=NS*X + Background
<A>Y=specific+nonspecific
<B>Y=nonspecific
```

The <A> and <B> syntax means that the third line is only used for data set A (total binding) while the fourth line is used only for data set B (nonspecific).

The parameters NS and Background are shared 70 between the two data sets.



Interpret the parameters

Bmax is the maximum specific binding in the same units as Y. It is the specific binding extrapolated to very high concentrations of radioligand, so it value is is almost always higher than any specific binding measured in your experiment.

Kd is the equilibrium dissociation constant, in the same units as X. It is the radioligand concentration needed to achieve a half-maximum binding at equilibrium.

NS is the slope of nonspecific binding in Y units divided by X units.

Background is the amount of nonspecific binding with no added radioligand. This represents counter background. If your counter automatically subtracts off the background signal, you can constrain Background to a constant value of zero.

4.5.5.7.4 Equation: One site -- Total, accounting for ligand depletion

Introduction

You don't have to measure nonspecific binding directly. Instead, you can determine Bmax and Kd by fitting only total binding by assuming that the amount of nonspecific binding is proportional to the concentration of radioligand.

If only a small fraction of radioligand binds, you can use a simpler model 563.

This equation allows for a substantial fraction of the added ligand to bind. This only works with radioactive ligands, so the assessment of added ligand and bound ligand are in the same counts-per-minute units. This method doesn't work with fluorescent ligands.

Step by step

Create an XY data table. Enter radioligand concentration into X, and total binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

Enter both X and Y in CPM units. This is essential for the analysis to work.

From the table of total binding, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *One site -- Total, accounting for ligand depletion.*

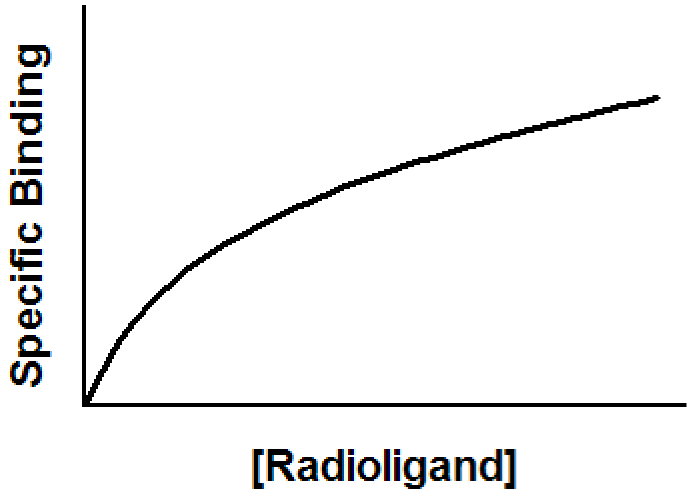
You must constrain two parameters to constant values based on your experimental design:

- SpAct is the specific radioactivity in cpm/fmol

- Vol is the reaction volume in ml

Model

```
KdCPM=KdnM * Vol * 1000 * SpecAct
; (nm/L * mL * 0.001 L/ml * 1000000 fmol/nmol * cpm/fmol)
a=-1-NS
b=KdCPM + NS*KdCPM + X + 2*X*NS + Bmax
c=-1*X*(NS*KdCPM + X*NS+Bmax)
Y=(-b+sqrt(b*b-4*a*c) )/(2*a) ;Y is in cpm
```



Interpret the parameters

Bmax is the maximum specific binding in cpm.

KdnM is the equilibrium dissociation constant in nM.

NS is the slope of nonspecific binding in Y units divided by X units.

Notes

This analysis accounts for the fact that a large fraction of the added radioligand binds to the receptors. If you are able to assume that only a small fraction of radioligand binds, which means that the concentration you added is virtually identical to the free concentration, use an alternative analysis 563.

Reference

This equation came from S. Swillens (Molecular Pharmacology, 47: 1197-1203, 1995)

4.5.5.7.5 Equation: One site -- Specific binding

Introduction

In a saturation binding experiment, you vary the concentration of radioligand and measure binding. The goal is to determine the Kd (ligand concentration that binds to half the receptor sites at equilibrium) and Bmax (maximum number of binding sites).

The ligand binds not only to receptors sites, but also to nonspecific sites. There are three approaches to dealing with nonspecific binding.

- Subtract off the nonspecific, and analyze only the specific binding. Read on for this approach.

- Analyze the total binding only, inferring the amount of nonspecific binding from the shape of the total binding curve. Learn more 563.

- Globally analyze the total and nonspecific binding at one time. Learn more. 564

Step by step

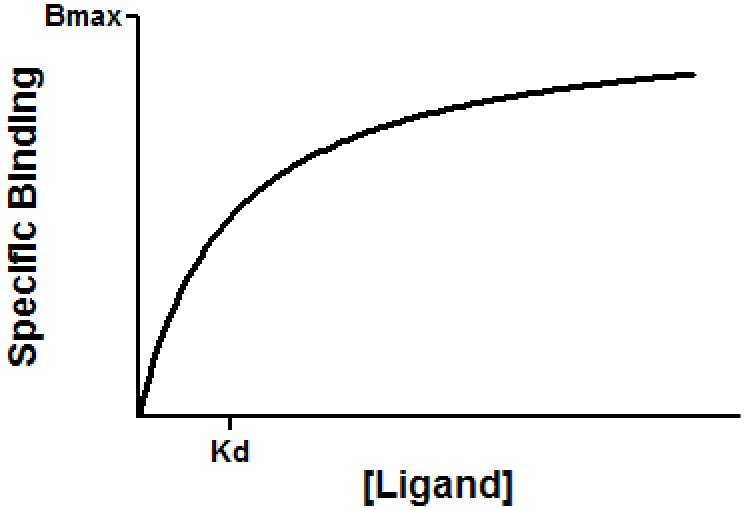
Create an XY data table. Enter radioligand concentration into X, and specific binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

An alternative approach would be to enter total binding into column A, and nonspecific into column B. Then use the Remove Baseline analysis to subtract column B from column A, creating a new results table with the specific binding.

From the table of specific binding, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *One site specific binding.*

Model

```
Y = Bmax*X/(Kd + X)
```



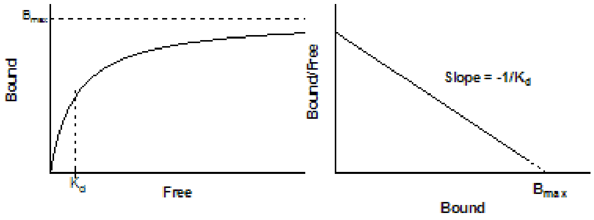
Interpret the parameters

Bmax is the maximum specific binding in the same units as Y. It is the specific binding extrapolated to very high concentrations of radioligand, and so its value is almost always higher than any specific binding measured in your experiment.

Kd is the equilibrium dissociation constant, in the same units as X. It is the radioligand concentration needed to achieve a half-maximum binding at equilibrium.

Create a Scatchard plot

Before nonlinear regression was available, investigators had to transform curved data into straight lines, so they could analyze with linear regression. One way to do this is with a Scatchard plot, which plots specific binding vs. the ratio of specific binding to the concentration of free radioligand.



If you create a Scatcahrd plot, use it only to display your data. The human retina and visual cortex evolved to detect edges (straight lines), not rectangular hyperbolas, and so it can help to display data this way. Scatchard plots are often shown as insets to the saturation binding curves. They are especially useful when you want to show a change in Bmax or Kd.

Don't use the slope and intercept of a linear regression line to determine values for Bmax and Kd. If you do this, you won't get the most accurate values for Bmax and Kd. The problem is that the transformation distorts the experimental error, so the data on the Scatchard plot do not obey the assumptions of linear regression. Use nonlinear regression to obtain the most accurate values of Kd and Bmax.

To create a Scatchard plot from your specific binding data, use Prism's Transform analysis, and choose the Scatchard transform from the panel of biochemistry and pharmacology transforms.

To create a Scatchard line corresponding to the nonlinear regression fit, follow these steps:

1. Create a new XY data table, with no subcolumns.

2. Into row 1 enter X=0, Y=Bmax/Kd (previously determined by nonlinear regression).You need to do the calculation manually, and enter a number.

3. Into row 2 enter X=Bmax and Y=0. Again enter the number into the X column, not the text 'Bmax'.

4. Note the name of this data table. Perhaps rename it to something appropriate.

5. Go to the Scatchard graph.

6. Drag the new table from the navigator and drop onto the graph.

7. Double-click on one of the new symbols for that data set to bring up the Format Graph dialog.

8. Choose to plot no symbols, but to connect with a line.

Notes

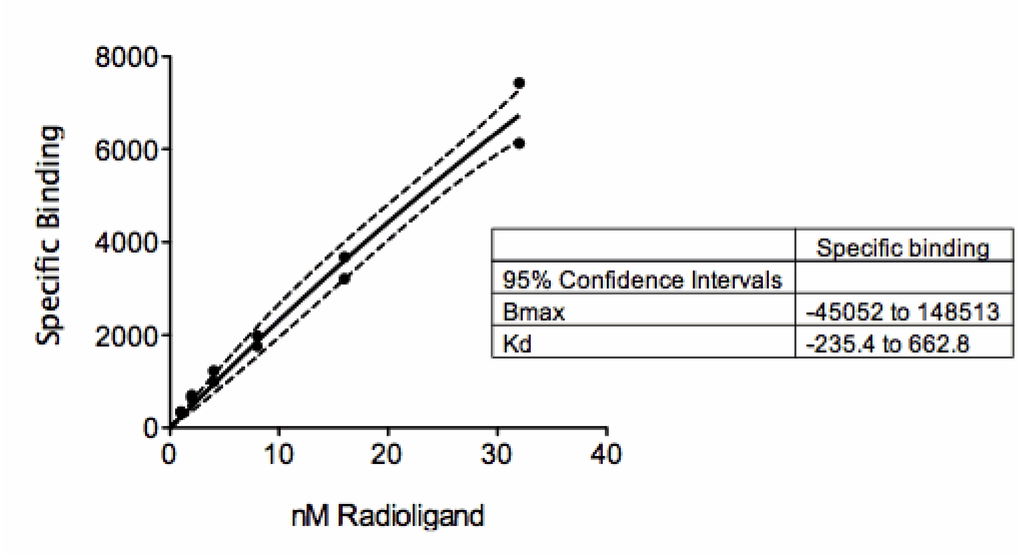
- This is not the best way to determine Bmax and Kd. It is better to globally fit total and nonspecific binding 564, without subtracting to compute specific binding.

- When making a Scatchard plot, you have to choose what units you want to use for the Y-axis. Some investigators express both free ligand and specific binding in cpm so the ratio bound/free is a unitless fraction. While this is easy to interpret (it is the fraction of radioligand bound to receptors), an alternative is to express specific binding in sites/cell or fmol/mg protein, and to express the free radioligand concentration in nM. While this makes the Y-axis hard to interpret visually, it provides correct units for the slope (which equals -1/Kd).

- This equation is equivalent to the Langmuir isotherm for the absorption of a gas onto a surface.

4.5.5.7.6 Binding potential

The problem: Binding isotherms without going to high enough concentrations

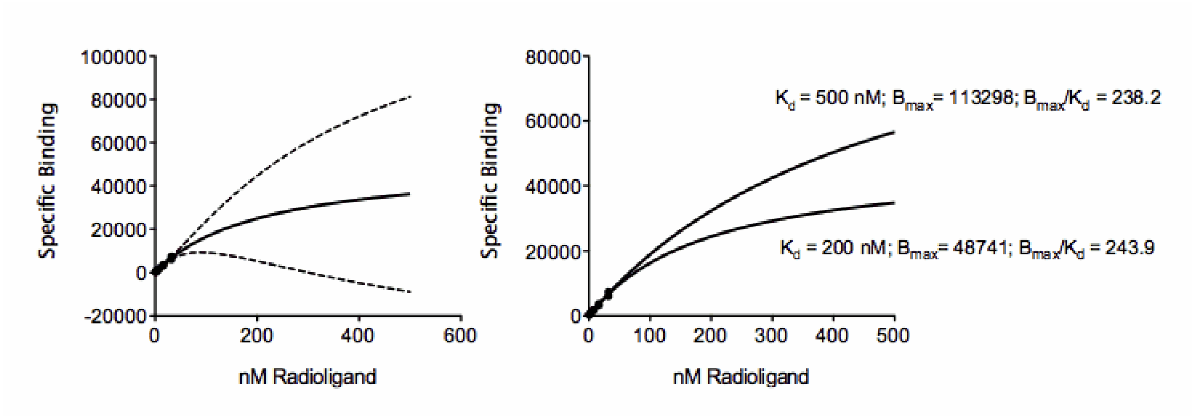


	Specific binding
95% Confidence Intervals	
Bmax	-45052 to 148513
Kd	-235.4 to 662.8

The figure above shows a saturation binding equation fit to specific binding data. But all the concentrations are relatively low (compared to the Kd of binding) so the data almost form a straight line. The saturation binding model fits the data fine (solid curve), with a narrow 95%

confidence band around the curve (dashed lines). But look at the confidence interval for Bmax and Kd (shown in the box)! They are super wide, even descending into negative (impossible) values. With these data, the curve fit the data fine, but neither parameter in the model (Bmax and Kd) were fit with reasonable confidence intervals. This is no surprise. The data don’t show even a hint of plateauing, so the data simply don’t define the Bmax and Kd at all.

The problem is that the data are consistent with a system with a huge number of low affinity receptors (high Bmax; high Kd) or a smaller number of high affinity receptors (low Bmax; low Kd). The figure below shows two ways the visualize this. The graph on the left shows the same fit as the one above, but with the best-fit curve and its confidence bands extended out to larger concentrations. While the confidence band is tight near the points, is it super wide as it goes beyond concentrations with data. The graph on the right show two different fits, with the Kd constrained to equal 200 nM or 500 nM. The curves are very different, yet both go near the data.



Another way to look at the problem with these data is that the Bmax and Kd are correlated. Prism can report the covariance matrix as part of its nonlinear regression results. With only two parameters, there is only one value in that "matrix". The covariance between Kd and Bmax (which can range from 0.0 to 1.0) is 0.9993. Prism can also report the dependency of each parameter (which also can range from 0.0 to 1.0). With only two parameters, both have the same dependency, which is 0.9986 for this example. This is not quite high enough for Prism to declare the results *ambiguous*, but the threshold for that designation (dependency > 0.9999) is arbitrary.

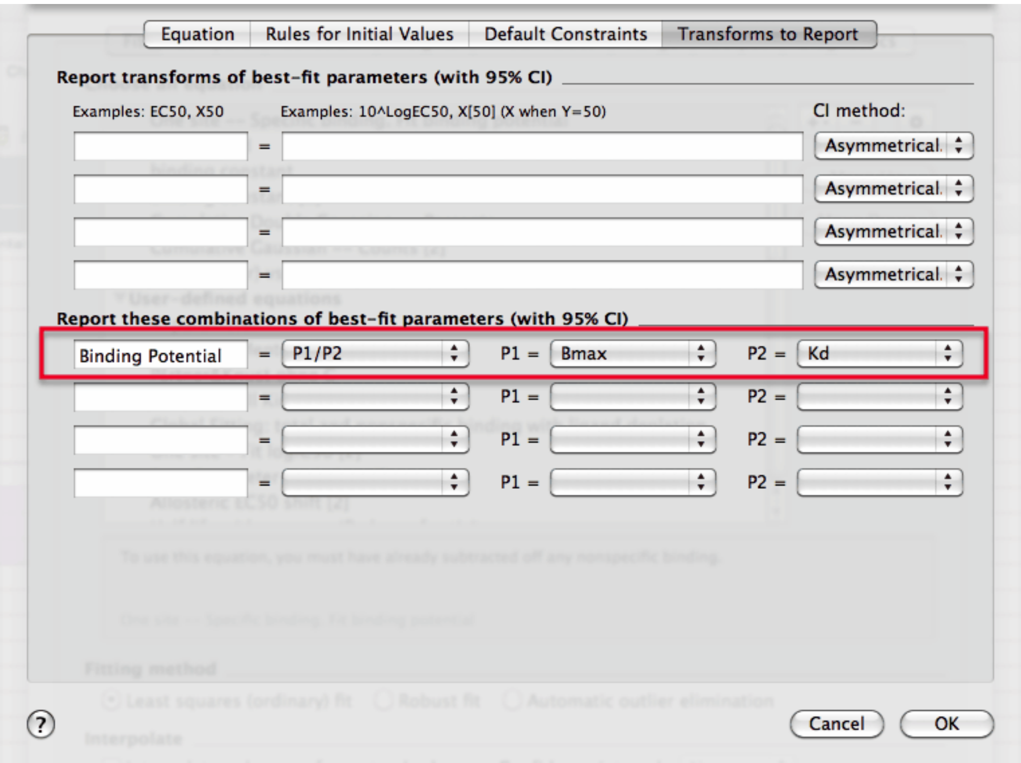
It is tempting to give up at this point and say that nothing can be determined without more data at higher concentrations. But in some systems, especially those using PET scanning to detect receptors, data like this are typical. It is impossible to use higher concentrations of ligand.

Fitting the binding potential

What can be learned from these data? The data are nearly linear. What does the slope of that line tell us? The slope, or rather the initial slope, is termed the *binding potential* (1). Don't be mislead by the term potential. It has nothing to do with potential energy and thermodynamics. Rather it is a measure of how much binding will occur with a small amount of ligand.

If you start with the specific binding equation, take its derivative with respect to X (which is concentration), and set X equal to zero, you'll derive the initial slope of a specific binding curve. It turns out that this slope, the binding potential, equals Bmax/Kd. Bmax is measured in the same units as the Y values in the data. Kd is measured in the same units as the X values. So the binding potential has units equal to the Y units divided by the X units.

Prism can fit a specific binding curve, and also report the ratio of Bmax/Kd with its confidence interval. To do this either create a new user-defined equation or (easier) clone the built in one. Then go to the last tab of the dialog that defines the equation. "Transforms to Report". Choose to report the ratio of two parameters, and define the numerator and denominator.



Report transforms of best-fit parameters (with 95% CI)

Examples: EC50, X50 Examples: 10^LogEC50, X[50] (X when Y=50) CI method:

= Asymmetrical

= Asymmetrical

= Asymmetrical

= Asymmetrical

Report these combinations of best-fit parameters (with 95% CI)

Binding Potential = P1/P2 P1 = Bmax P2 = Kd

= P1 = P2 =

= P1 = P2 =

= P1 = P2 =

In this example, Prism finds that the Binding Potential is 242.1, with a confidence interval ranging from 183.4 to 300.7. That confidence interval is reasonably narrow, so the result is quite useful.

Reference

1. Innis et al. Consensus nomenclature for in vivo imaging of reversibly binding radioligands. Journal of Cerebral Blood Flow & Metabolism (2007) vol. 27 (9) pp. 1533-1539

Download the Prism file for this example.

4.5.5.7.7 Equation: Two sites -- Specific binding only

Introduction

In a saturation binding experiment, you vary the concentration of radioligand and measure binding. The goal is to determine the Kd (ligand concentration that binds to half the receptor sites at equilibrium) and Bmax (maximum number of binding sites) of both kinds of receptors.

The ligand binds not only to receptors sites, but also to nonspecific sites. There are three approaches to dealing with nonspecific binding.

- Subtract off the nonspecific, and analyze only the specific binding. Read on for this approach.
- Analyze the total binding only, inferring the amount of nonspecific binding from the shape of the total binding curve. This approach doesn't work well when there are two classes of receptors.
- Globally analyze the total and nonspecific binding at one time. Learn more. 578

Step by step

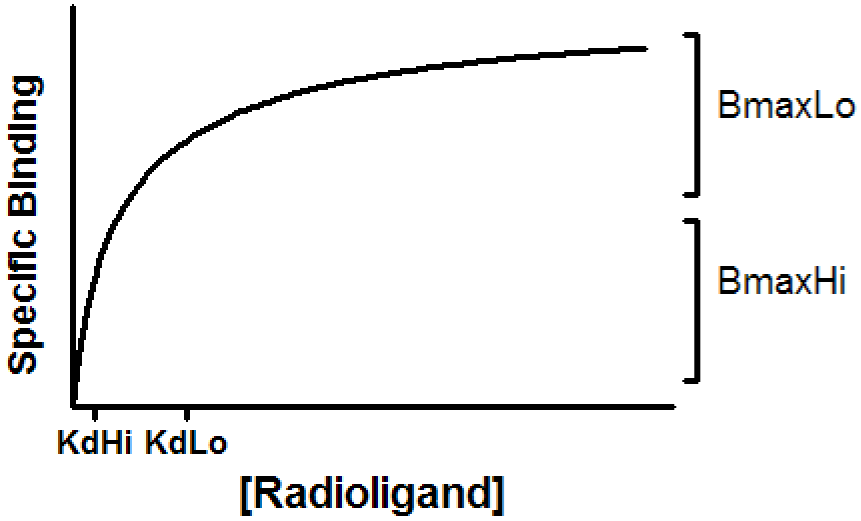
Create an XY data table. Enter radioligand concentration into X, and specific binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

An alternative approach would be to enter total binding into column A, and nonspecific into column B. Then use the Remove Baseline analysis to subtract column B from column A, creating a new results table with the specific binding.

From the table of specific binding, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *Two sites -- Specific binding.*

Model

```
Site1=BmaxHi*X/(KdHi+X)
Site2=BmaxLo*X/(KdLo+X)
Y=Site1 + Site2
```



Interpret the parameters

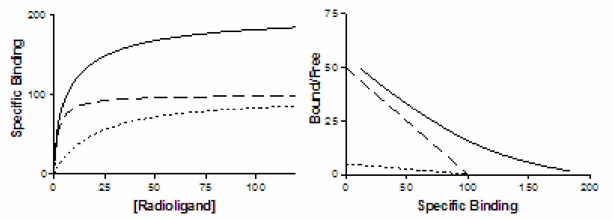
BmaxHi and **BmaxLo** are the maximum specific bindings to the two sites in the same units as Y.

KdHi and **KdLo** are the equilibrium dissociation constants, in the same units as X. It is the radioligand concentration needed to achieve a half-maximum binding at equilibrium

Scatchard plots of two site binding

The left panel below shows binding of a radioligand to two independent binding sites present in equal concentrations, but with a tenfold difference in Kd . The two individual curves are shown as dotted and dashed curves. When you do the experiment, you can't observe the individual components, but observe the sum, which is shown as a solid curve. Note that this curve is not obviously biphasic.

The right panel shows the same data plotted on a Scatchard plot. The binding to each receptor is shown as a straight line (dotted, or dashed). The total binding, plotted on a Scatchard plot, is curved. Note that the two lines that represent binding to each type of receptor are NOT the asymptotes of the curve.



To plot the two straight lines that correspond to the nonlinear regression fit, create a new data table that defines the two lines as shown below, using Bmax and Kd values determined by nonlinear regression.

X	A	B
0	Bmax1/Kd1	
Bmax1	0	
0		Bmax2/Kd2
Bmax2	0	

Go to the graph of the Scatchard transformed data and drag the new table to that graph. Use the Format Graph dialog to plot the two data sets from the table using connecting lines but no symbols.

4.5.5.7.8 Equation: Two sites -- Fit total and nonspecific binding

Introduction

In a saturation binding experiment, you vary the concentration of radioligand and measure binding. The goal is to determine the Kd (ligand concentration that binds to half the receptor sites at equilibrium) and Bmax (maximum number of binding sites).

The ligand binds not only to receptors sites, but also to nonspecific sites. There are three approaches to dealing with nonspecific binding.

- Subtract off the nonspecific, and analyze only the specific binding 575.
- Analyze the total binding only, inferring the amount of nonspecific binding from the shape of the total binding curve. This approach doesn't work well when the ligand binds to two sites
- Globally analyze the total and nonspecific binding at one time. This is the best approach, and the details are explained below.

Step by step

Create an XY data table. Enter radioligand concentration into X, total binding into Y, and nonspecific binding into column B.

Use any convenient units for X. The Kd will be reported in those same concentration units. Use the same units for total and nonspecific binding. The Bmax will be reported in those same units.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *Two sites -- Total and nonspecific binding.*

The parameter Background is the measured 'binding' when there is no radioligand binding added, so represents the counter background, if there is any. Consider constraining it to a constant value of zero.

Model

```
Specific1=BmaxHi*X/(X+KdHi)
Specific2=BmaxLo*X/(X+KdLo)
Nonspecific=NS*X + Background
<A>Y=Specific1 + Specific2 + Nonspecific
<B>Y=Nonspecific
```

The <A> and <B> syntax means that the fourth line is only used for data set A (total binding) while the fifth line is used only for data set B (nonspecific).

The parameters NS and Background are shared between the two data sets.

Interpret the parameters

BmaxHi and **BmaxLo** are the maximum specific bindings to the two sites in the same units as Y.

KdHi and **KdLo** are the equilibrium dissociation constants, in the same units as X. It is the radioligand concentration needed to achieve a half-maximum binding at equilibrium.

NS is the slope of nonspecific binding in Y units divided by X units.

Background is the amount of nonspecific binding with no added radioligand. This represents counter background. If your counter automatically subtracts off the background signal, you can constrain Background to a constant value of zero.

4.5.5.7.9 Equation: One site with allosteric modulator

Introduction

An allosteric modulator can reduce radioligand binding. This model fits experiments, where entire radioligand binding curves are measured in the absence and presence of modulator. The goal is to learn the affinity of the modulator for binding to its site, and also determine the value of alpha, the ternary complex constant that quantifies the degree to which binding of the modulator alters the affinity of the radioligand for the receptor site.

Step by step

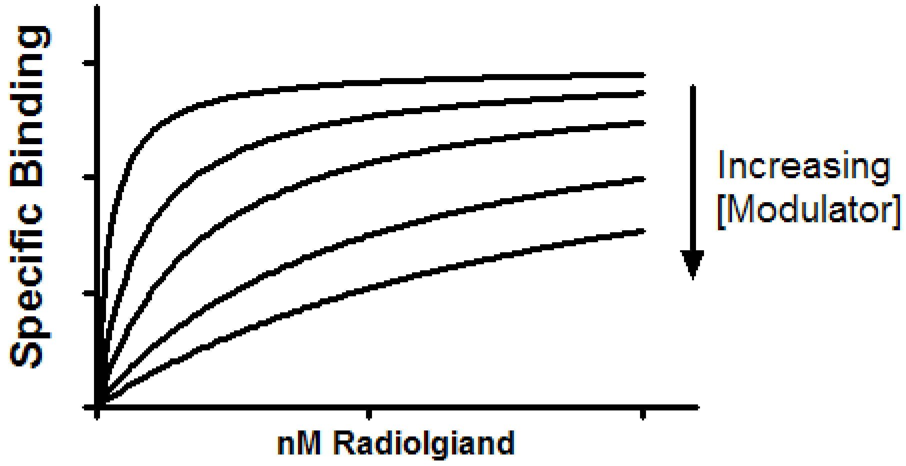
Create an XY data table. Enter the concentration of the labeled ligand into X, using any convenient units (maybe nM). Enter specific binding into Y in any convenient units. Enter data with no modulator into column A. Enter data collected with a constant concentration of modulator into column B. Repeat, if you have data, for column C, D, E, ..., each with a different concentration of modulator. Enter the modulator concentration (in nanomolar so 1nM is entered as '1') into the column titles. Don't forget to enter '0' as the column title for data set A.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *Allosteric modulator shift.*

You don't need to constrain any parameters to constant values.

Model

```
Hot=X
Alpha=10^logalpha
KB=10^logKB
KApp=KDHot*((1+Allo/KB)/(1+alpha*Allo/KB))
Y=Bmax*Hot/(Hot+KApp)
```



Interpret the parameters

Kb is the equilibrium dissociation constant (Molar) of modulator binding to its allosteric site. It is in the same molar units used to enter the modulator concentration into column titles on the data table.

KdHot is the equilibrium dissociation constant of the radioligand. It is expressed in the same units used to enter X values, nM in the example.

Alpha is the ternary complex constant. When alpha=1.0, the modulator won't alter binding. If alpha is less than 1.0, then the modulator reduces ligand binding. If alpha is greater than 1.0, then the modulator increases binding. In the example shown about, alpha equals 0.01 so the modulator greatly decreases binding.

Notes

- This model is designed to analyze data when the unlabeled compound works via an allosteric site. Since the labeled and unlabeled ligands are acting via different sites, it is inappropriate (and incorrect) to refer to the modulator as a competitor.

- The model is written to fit the logarithm of alpha, rather than alpha itself. This is because alpha is asymmetrically (all values from 0 to 1 mean that the modulator decreases binding, while all values from 1 to infinity mean that the modulator enhances binding. On a log scale, its values are more symmetrical, so the confidence interval computed on a log scale (as Prism does) are more accurate. Prism reports both alpha and log(alpha).

- This model assumes that the allosteric modulator is present in excess, so the concentration you added is very close to its free concentration. This model won't work when the concentration of allosteric modulator is limiting (as it is when G proteins alter agonist binding to many receptors). No explicit model can handle this situation. You need to define the model with an implicit equation (Y on both sides of the equals sign) and Prism cannot handle such equations.

Reference

A. Christopoulos and T. Kenakin, Pharmacol Rev, 54: 323-374, 2002

4.5.5.7.10 Equation: Specific binding with Hill slope

Introduction

In a saturation binding experiment, you vary the concentration of radioligand and measure binding. The goal is to determine the Kd (ligand concentration that binds to half the receptor sites at equilibrium) and Bmax (maximum number of binding sites).

This equation assumes you have subtracted off the nonspecific, and are only analyzing specific binding.

This equation fits a Hill slope. If you assume the Hill slope is 1.0 (for mass action binding of a monomer to one site) use a simpler equation 568 .

Step by step

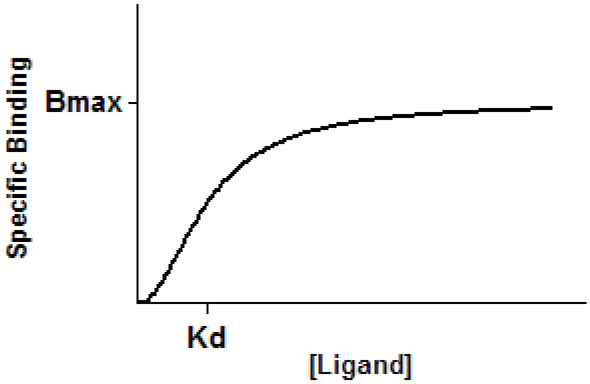
Create an XY data table. Enter radioligand concentration into X, and specific binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

An alternative approach would be to enter total binding into column A, and nonspecific into column B. Then use the Remove Baseline analysis to subtract column B from column A, creating a new results table with the specific binding.

From the table of specific binding, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *Specific binding with Hill Slope.*

Model

```
Y=Bmax*X^h/(Kd^h + X^h)
```



Note that the X axis is concentration, not log(concentration).

Interpret the parameters

Bmax is the maximum specific binding in the same units as Y. It is the specific binding extrapolated to very high concentrations of radioligand, and so its value is almost always. higher than any specific binding measured in your experiment.

Kd is the radioligand concentration needed to achieve a half-maximum binding at equilibrium, expressed in the same units as X. If h=1.0, this is the equilibrium dissociation constant. If h is not equal to 1.0, then the molecular interpretation of the Kd depends on why h is not 1.0.

h is the Hill slope. It equals 1.0 when a monomer binds with no cooperativity to one site. When it is greater than 1.0, you see a sigmoidal look to the graph as shown above. This happens when the receptor or ligand has multiple binding sites with positive cooperativity. The Hill slope is less than zero when there are multiple binding sites with different affinities for ligand or when there is negative cooperativity.

Note

Before version 7, this equation was called "One site -- specific binding with Hill slope". We removed "one site" from the name because if the Hill slope is not 1.0, this equation doesn't really describe binding to one site.

4.5.5.8 Receptor binding - Competitive binding

4.5.5.8.1 Key concepts: Competitive binding

What is competitive binding?

In a competitive binding experiment, you use a single concentration of labeled (hot) ligand and vary the concentration of unlabeled (cold) drugs, and measure binding at equilibrium.

Comparing one- and two-site models

Prism offers models for fitting one or two sites. You can use choices in the Compare tab to compare the two fits. When comparing the fits to the one- and two-site models, use common sense as well as statistics. Don't accept a two site model, if one of the sites is only a tiny fraction of the total, or if its Ki is outside the range of competitor concentrations you used in the experiment.

Ligand depletion

If a large fraction of the added radioligand binds to the receptors, the ligand is depleted so the concentration you added is greater than the free concentration. You need to fit these data to a model that accounts for ligand depletion 592.

Homologous binding

An **homologous** binding experiment is one where the labeled and unlabeled ligands have identical affinities for the receptors. Generally this is because the two are chemically identical. Receptor number and affinity are determined by analyzing the competition of varying concentrations of unlabeled ligand for one (or better, two) concentrations of labeled ligand. Prism offers a special model for fitting homologous competition experiments 594.

Allosteric modulators

Allosteric modulators can alter radioligand binding, even though they bind to different sites. Since the hot and cold ligands bind to different sites, the term 'competition' is not apt, but we include this model here because the experimental design is the same as used for competitive binding. Prism can fit binding inhibition 596 (or augmentation) by an allosteric modulator based on the ternary complex model. Note that this model assumes the allosteric modulator is present in excess, so is not depleted by binding to the receptors.

4.5.5.8.2 Equation: One site - Fit Ki

Introduction

You can determine the equilibrium dissociation constant of an unlabelled ligand by measuring its competition for radioligand binding.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the unlabeled compound into X, and binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc. Use subcolumns to enter replicates.

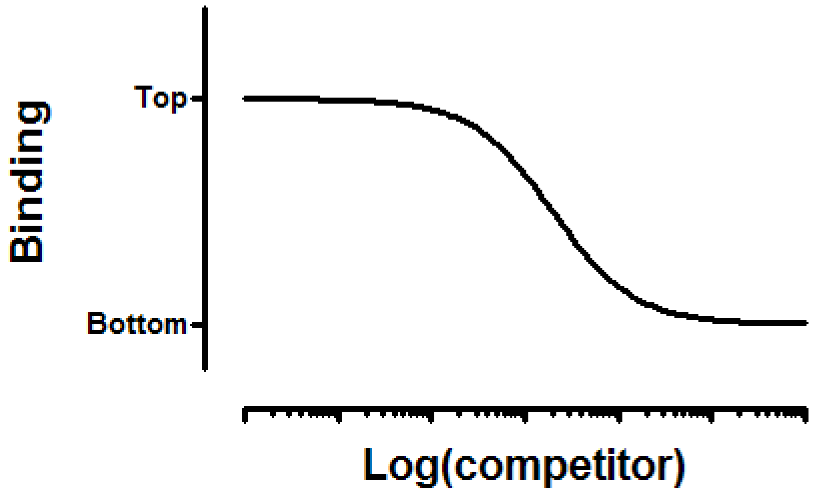
From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *One site - Fit Ki.*

You must constrain two parameters to constant values based on your experimental design:

- RadioligandNM is the concentration of labeled ligand in nM. A single concentration of radioligand is used for the entire experiment.
- HotKdNM is the equilibrium dissociation constant of the labeled ligand in nM.

Model

```
logEC50=log(10^logKi*(1+RadioligandNM/HotKdNM))
Y=Bottom + (Top-Bottom)/(1+10^(X-LogEC50))
```



Interpret the parameters

Top and **Bottom** are plateaus in the units of Y axis.

logKI is the log of the molar equilibrium dissociation constant of unlabeled ligand.

Ki is the equilibrium dissociation constant in Molar.

Notes

This model fits the Ki of the unlabelled ligand directly. It does not report the EC50, so you do not need to apply the Cheng and Prusoff correction(1). Instead you enter the concentration of radioligand and its Kd as constants, and Prism directly fits the Ki of your cold compound. If you want to know the IC50, fit a log(dose)-response curve.

If you want to know the IC50 (which is not very informative in this situation) fit the data using an alternative equation 587.

The analysis assumes that you have one site, and that the binding is reversible and at equilibrium.

1. Cheng, Y. and Prusoff, W. H. Relationship between the inhibition constant (K1) and the concentration of inhibitor which causes 50 per cent inhibition (I50) of an enzymatic reaction. Biochem Pharmacol, 22: 3099-3108, 1973.

4.5.5.8.3 Equation: One site - Fit logIC50

Introduction

You can determine the equilibrium dissociation constant of an unlabelled ligand by measuring its competition for radioligand binding.

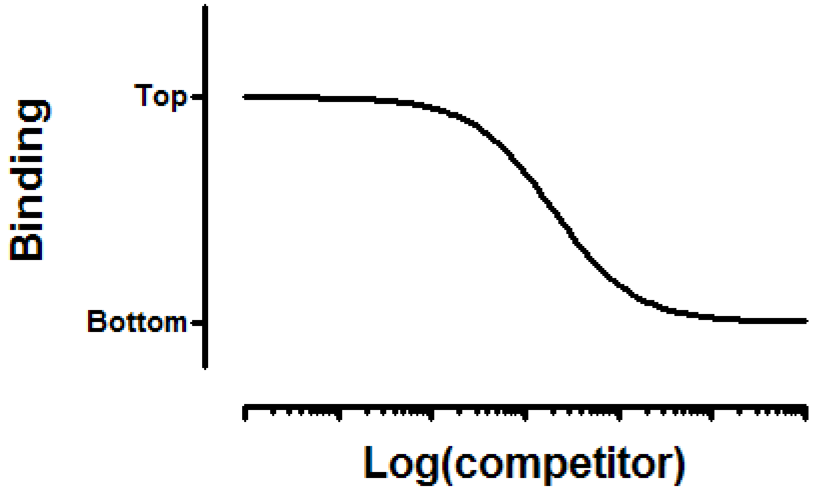
Step by step

Create an XY data table. Enter the logarithm of the concentration of the unlabeled compound into X, and binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc. Use subcolumns to enter replicates.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *One site - Fit logEC50.*

Model

```
Y=Bottom + (Top-Bottom)/(1+10^(X-LogIC50))
```



Interpret the parameters

Top and **Bottom** are plateaus in the units of Y axis.

logIC50 is the log of the concentration of competitor that results in binding half-way between Bottom and Top

Notes

This model is the same as an inhibitory dose-response curve. It fits the logIC50, which is not the same as the Ki of the unlabelled ligand for binding. The Ki depends on the IC50, the concentration of radioligand, and its Kd for binding. You can fit the Ki directly using a different equation 585.

The analysis assumes that you have one site, and that the binding is reversible and at equilibrium.

4.5.5.8.4 Equation: Two sites - Fit Ki

Introduction

You can determine the equilibrium dissociation constant of an unlabelled ligand by measuring its competition for radioligand binding.

Step by step

Create an XY data table. Enter the logarithm of the Molar concentration of the unlabeled compound into X, and binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc. Use subcolumns to enter replicates.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *Two sites - Fit Ki.*

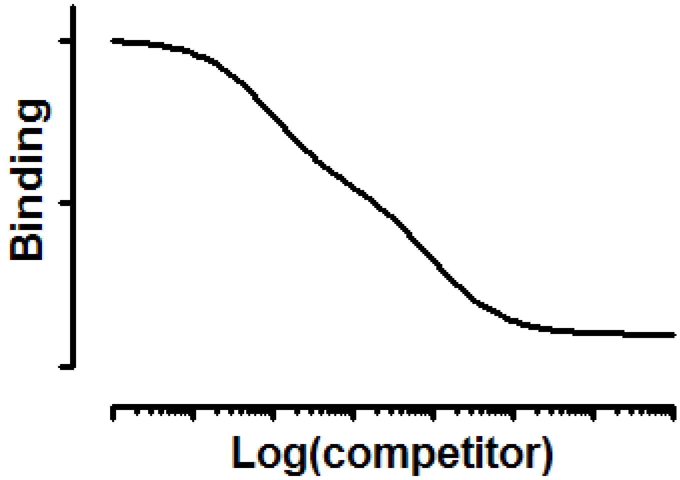
You must constrain three parameters to constant values based on your experimental design:

- **HotNM** is the concentration of labeled ligand in nM. A single concentration of radioligand is used for the entire experiment.

- **HotKdNMHi** is the equilibrium dissociation constant of the labeled ligand for the high-affinity site in nM.
- **HotKdNMLo** is the equilibrium dissociation constant of the labeled ligand for the low-affinity site in nM.

Model

```
logEC50Lo=log(10^logKiLo*(1+HotNM/HotKdNMLo))
logEC50Hi=log(10^logKiHi*(1+HotNM/HotKdNMHi))
Span=Top - Bottom
Part1=FractionHi*Span/(1+10^(X-LogEC50Hi) )
Part2=(1-FractionHi)*Span/(1+10^(X-LogEC50Lo) )
Y=Bottom + Part1 + Part2
```



Interpret the parameters

Top and **Bottom** are plateaus in the units of Y axis.

FractionHi is the fraction of all the sites that have high affinity for the competitor.

logKi_Hi and **logKi_Lo** are the logarithms of the two molar Ki values.

Notes

This model fits the two log(Ki) values of the unlabelled ligand directly. It does not report the IC50s, so you do not need to apply the Cheng and Prusoff correction(1). Instead you enter the concentration of radioligand and its Kd as constants, and Prism directly fits the Ki of your cold compound. If you want to fit the two IC50 values instead of the Ki values, use a different equation 590.

The analysis assumes that you know the affinity of both sites for the labeled ligand. In many cases, the radioligand has the same affinity for both sites. In that case, simply enter that value twice. If the two sites have different affinities for the labeled ligand, enter both values (determined from other experiments). Watch out for the labels. The constant KdHi is the Kd of the hot ligand for the receptors with the high affinity for the unlabeled ligand, and KdLo is the Kd of the hot ligand for the receptors with lower affinity for the unlabeled ligand. So KdHi may be larger or smaller than KdLo.

This analysis assumes that the binding is reversible and at equilibrium. It also assumes that the labeled and unlabeled ligands compete for the same binding sites.

1. Cheng, Y. and Prusoff, W. H. Relationship between the inhibition constant (K1) and the concentration of inhibitor which causes 50 per cent inhibition (I50) of an enzymatic reaction. Biochem Pharmacol, 22: 3099-3108, 1973.

4.5.5.8.5 Equation: Two sites - Fit logIC50

Introduction

You can determine the equilibrium dissociation constant of an unlabelled ligand by measuring its competition for radioligand binding. This model assumes that there are two classes of sites with different affinities for the radioligand and competitor.

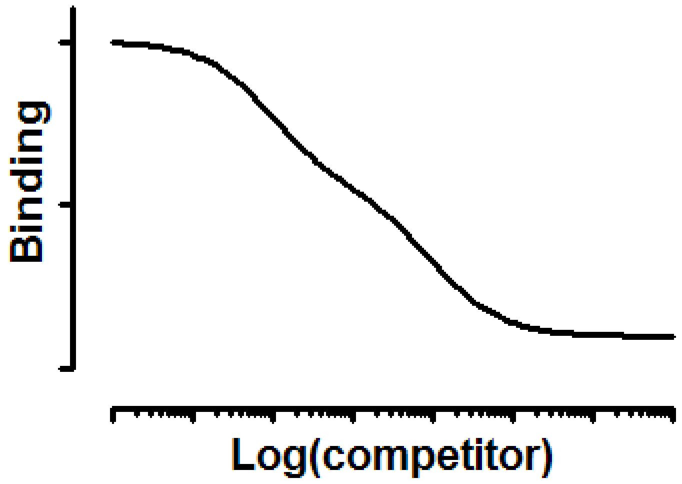
Step by step

Create an XY data table. Enter the logarithm of the molar concentration of the unlabeled compound into X, and binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc. Use subcolumns to enter replicates.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *Two sites - Fit logIC50.*

Model

```
logEC50Lo=log(10^logKiLo*(1+HotNM/HotKdNMLo))
logEC50Hi=log(10^logKiHi*(1+HotNM/HotKdNMHi))
Span=Top - Bottom
Part1=FractionHi*Span/(1+10^(X-LogEC50Hi) )
Part2=(1-FractionHi)*Span/(1+10^(X-LogEC50Lo) )
Y=Bottom + Part1 + Part2
```



Interpret the parameters

Top and **Bottom** are plateaus in the units of Y axis.

FractionHi is the fraction of all the sites that have high affinity for the competitor.

logIC50Hi and **logIC50Lo** are the logarithms of the two IC50 values.

Notes

This model fits the two IC50 values of the unlabelled ligand. It does not report the two Ki values. The Ki values depend on the IC50s, the concentration of radioligand, and its Kd for binding. You can fit the Ki values directly using a different equation 588.

The analysis assumes that you know the affinity of both sites for the labeled ligand. In many cases, the radioligand has the same affinity for both sites. In that case, simply enter that value twice. If the two sites have different affinities for the labeled ligand, enter both values (determined from other experiments). Watch out for the labels. The constant HotKdnmHi is the Kd of the hot ligand for the receptors with the high affinity for the unlabeled ligand, and HotKdnmLo is the Kd of the hot ligand for the receptors with lower affinity for the unlabeled ligand. So HotKdnmHi may be larger or smaller than HotKdnmLo.

This analysis assumes that the binding is reversible and at equilibrium. It also assumes that the labeled and unlabeled ligands compete for the same binding sites.

4.5.5.8.6 Equation: One site - Ligand depletion

Introduction

This model for competitive binding is useful when a large fraction of the added radioligand binds to the receptors, so the concentration you added is greater than the free concentration.

These equation allows for a substantial fraction of the added ligand to bind. This only works with radioactive ligands, so the assessment of added ligand and bound ligand are in the same counts-per-minute units. This method doesn't work with fluorescent ligands.

Step by step

Create an XY data table. Enter the logarithm of the molar concentration of the unlabeled compound into X, and binding into Y. The Y values must be in cpm.

If you have several experimental conditions, place the first into column A, the second into column B, etc. Use subcolumns to enter replicates.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *One site -- Heterologous with depletion.*

You must constrain four parameters to constant values based on your experimental design:

- **Hot** is the amount of labeled ligand in cpm. A single concentration of radioligand is used for the entire experiment.

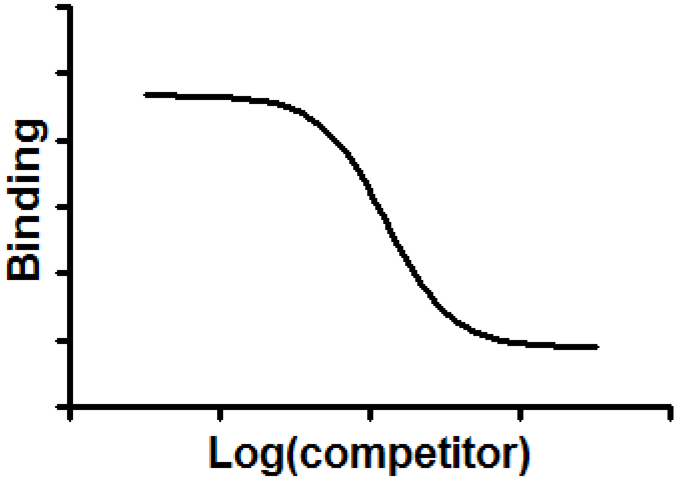
- **KdNM** is the equilibrium dissociation constant of the labeled ligand in nM.

- **SpAct** is the specific radioactivity in cpm/fmol.

- **Vol** is the reaction volume in ml.

Model

```
KdCPM=KdnM*SpAct*vol*1000
; nmol/L *(cpm/fmol * ml * .001L/ml * 1000000fmol/nmol) = cpm
R=NS+1
S=[1+10^(X-LogKi)]*KdCPM+Hot
a=-1*R
b=R*S+NS*Hot + Bmax
c= -1*Hot*(S*NS + Bmax)
Y= (-1*b + sqrt(b*b-4*a*c))/(2*a)
```



Interpret the parameters

logKI is the log of the molar equilibrium dissociation constant of unlabeled ligand.

Ki is the equilibrium dissociation constant in Molar.

Bmax is the maximum binding of ligand to receptors in cpm. This represents binding to all the receptors so is higher than the top plateau of the curve.

NS is the fraction of the radioligand that binds to nonspecific sites.

Notes

This analysis accounts for the fact that a large fraction of the added radioligand binds to the receptors. If you are able to assume that only a small fraction of radioligand binds, which means that the concentration you added is virtually identical to the free concentration, use an alternative analysis 585.

Reference

This equation came from S. Swillens (Molecular Pharmacology, 47: 1197-1203, 1995)

4.5.5.8.7 Equation: One site - Homologous

Introduction

An **homologous** binding experiment is one where the labeled and unlabeled ligands have identical affinities for the receptors. Generally this is because the two are chemically identical. Receptor number and affinity are determined by analyzing the competition of varying concentrations of unlabeled ligand for one (or better, two) concentrations of labeled ligand.

Step by step

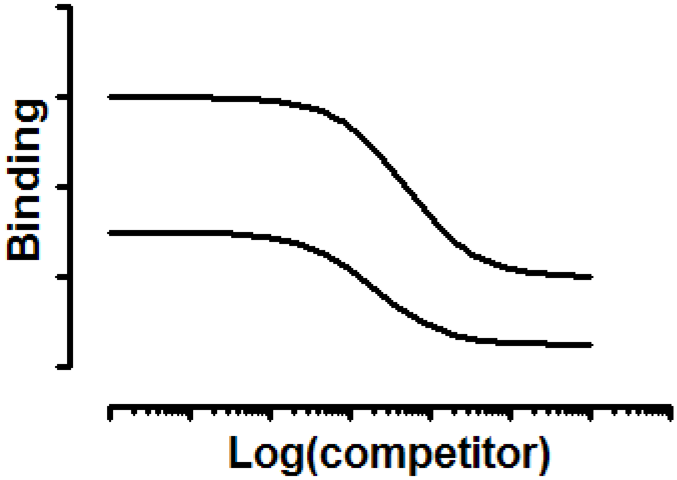
Create an XY data table. Enter the logarithm of the concentration of the unlabeled compound into X, and binding into Y. Enter the concentration of labeled ligand (in nM) as the column title. You will get better results if you use two different concentrations of labeled ligand.

If you have several experimental conditions, place the first into column A, the second into column B, etc. Use subcolumns to enter replicates.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *One site - Homologous.*

Model

```
ColdNM=10^(x+9) ;Cold concentration in nM
KdNM=10^(logKD+9) ;Kd in nM
Y=(Bmax*HotnM)/(HotnM + ColdNM + KdNM) + Bottom
```



Interpret the parameters

logKd Log of the equilibrium dissociation constant. Since the analysis is based on the assumption that labeled and unlabeled ligand bind with identical affinities, the logKd applies to both forms of the ligand.

Bmax is the maximum binding of ligand to receptors in cpm. This represents binding to all the receptors so is higher than the top plateau of the curve.

NS is a measure of nonspecific binding. It is the bottom plateau in units of the Y axis divided by the concentration of the hot ligand in nM. In other words, it is the fraction of hot ligand that binds nonspecifically.

Prism fits the curves globally to find one shared value for logKd, Bmax and NS from all sets of data

Notes

This model assumes that the hot and cold ligand binds identically to the receptor, and that you use two concentrations of hot ligand (in column A, and B ..) and vary cold. It assumes that a small fraction of added ligand binds, so the free concentration is close to what you added.

4.5.5.8.8 Equation: Allosteric modulator

Introduction

Fits a curve of "competition" of binding by an allosteric modulator, based on the ternary complex model. Note that this model assumes the allosteric modulator is present in excess, so is not depleted by binding to the receptors. Since it binds to a different site than the radioligand, the term 'competition' is not apt, but we list it here because the experimental design is the same as used for competitive binding.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the unlabeled modulator (in nM) into X, and specific binding into Y in any convenient units.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *Allosteric modulator titration.*

You must constrain two parameters to constant values based on your experimental design:

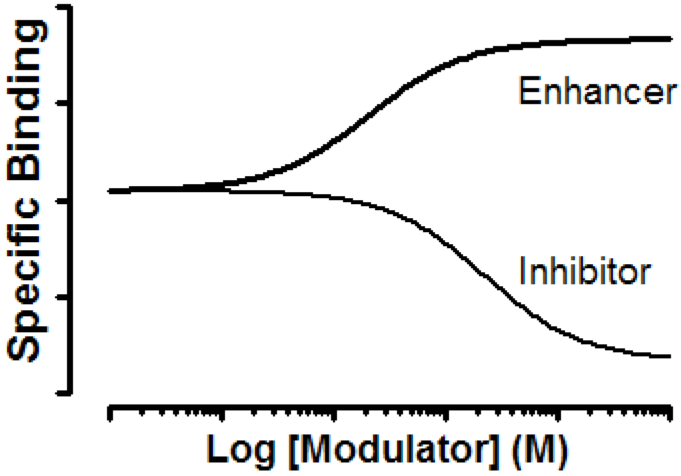
- RadioligandNM is the concentration of labeled ligand in nM. A single concentration of radioligand is used for the entire experiment.

- HotKdNM is the equilibrium dissociation constant of the labeled ligand in nM.

Also consider constraining Y0 (radioligand binding in the absence of modulator) to a constant value.

Model

```
AlloNM=10^(X+9)
KbNM=10^(logKb +9)
alpha=10^logAlpha
KAppNM=HotKDnm*(((1+(AlloNM/KBNM))/(1+alpha*(AlloNM/KBNM))))
HotOccupancy = RadioligandNM/(RadioligandNM + HotKDnm)
Y=(Y0/HotOccupancy)*(RadioligandNM/(RadioligandNM + KAppNM))
```



Interpret the parameters

Kb is the equilibrium dissociation constant (Molar) of modulator binding.

Alpha is the ternary complex constant. When alpha=1.0, the modulator does not alter binding. If alpha is less than 1.0, then the modulator reduces ligand binding. If alpha is greater than 1.0, then the modulator increases binding.

Y0 is the radioligand binding in the absence of modulator. Consider constraining this to a constant value.

Notes

This model is designed to analyze data when the unlabeled compound works via an allosteric site. Since the labeled and unlabeled ligands are acting via different sites, it is inappropriate (and incorrect) to refer to these types of experiments as “competition binding assays”. In some cases, in fact, the allosteric modulator enhances radioligand binding.

The model is written to fit the logarithm of alpha, rather than alpha itself. This is because alpha is asymmetrically (all values from 0 to 1 mean that the modulator decreases binding, while all values from 1 to infinity mean that the modulator enhances binding. On a log scale, its values are more symmetrical, so the confidence interval computed on a log scale (as Prism does) are more accurate.

The Y axis plots specific binding. Even at very high concentrations of inhibitor, the specific binding does not descend to zero. This is the nature of allosteric inhibition. If alpha is very high, then the binding is inhibited almost to zero. If alpha is not so high, then the maximum inhibition is more modest. For example, if alpha=3, the maximum inhibition is down to 33%.

This model assumes that the allosteric modulator is present in excess, so the concentration you added is very close to its free concentration. This model does not apply when the concentration of allosteric modulator is limiting (as it is when G proteins alter agonist binding to many receptors). No explicit model can handle this situation. You need to define the model with an implicit equation (Y on both sides of the equals sign) and Prism cannot handle such equations.

Reference

A. Christopoulos and T. Kenakin, Pharmacol Rev, 54: 323-374, 2002

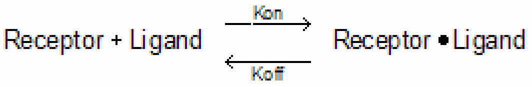
4.5.5.9 Receptor binding - Kinetics

Kinetics refers to changes over time. Kinetic binding experiments are used to determine the association and dissociation rate constants.

4.5.5.9.1 Key concepts: Kinetics of binding

Law of mass action

Kinetic binding experiments are used to determine the association and dissociation rate constants. All the analyses assume that binding follows the law of mass action:



At any given time, the rate at which receptor-ligand complexes form is proportional to the radioligand concentration and the number of receptors still unoccupied. The rate of dissociation is proportional to the concentration of receptor-ligand complexes.

Rate of dissociation

A dissociation binding experiment measures the “off rate” for radioligand dissociating from the receptor. Initially ligand and receptor are allowed to bind, perhaps to equilibrium. At that point, you need to block further binding of radioligand to receptor (by adding an unlabeled drug or by dilution) so you can measure the rate of dissociation, which follows a one-phase exponential decay with a rate constant equal to the rate of radioligand dissociation.

Rate of association

In an association experiment, you add radioligand and measure specific binding at various times thereafter.

Binding increases over time until it plateaus. This plateau is not the same as the Bmax. The plateau in an association experiment depends on the concentration of radioligand used, while the Bmax is extrapolated to an infinite concentration of radioligand.

The rate at which binding increases is determined by three factors (as well as experimental conditions such as pH and temperature):

- The association rate constant, kon or k+1. This is what you are trying to determine.

- The concentration of radioligand. If you use more radioligand, the system equilibrates faster.

- The dissociation rate constant, koff or k-1. Some people are surprised to see that the observed rate of association depends in part on the dissociation rate constant. During the incubation, radioligand both binds to and dissociates from receptors. The system reaches equilibrium when the two rates are equal. The observed rate of association measures how long it takes to reach equilibrium. If the radioligand dissociates quickly from the receptor, equilibrium will be reached faster (but with less binding).

It is not possible, therefore, to fit a simple association experiment to determine the association rate constant unless you constrain the value of the dissociation rate constant 602, measure association kinetics with two or more concentrations of radioligand 603, or determine the rate of association and dissociation in one experiment 605.

4.5.5.9.2 Equation: Dissociation kinetics

Introduction

A dissociation binding experiment measures the “off rate” for radioligand dissociating from the receptor. Initially ligand and receptor are allowed to bind, perhaps to equilibrium. At that point, you need to block further binding of radioligand to receptor so you can measure the rate of dissociation. There are several ways to do this:

- If the tissue is attached to a surface, you can remove the buffer containing radioligand and replace with fresh buffer without radioligand.

- Spin the suspension and resuspend in fresh buffer.

- Add a very high concentration of an unlabeled ligand. If this concentration is high enough, it will instantly bind to nearly all the unoccupied receptors and thus block binding of the radioligand.

- Dilute the incubation by a large factor, at least 100 fold dilution. This will reduce the concentration of radioligand by that factor. At such a low concentration, new binding of radioligand will be negligible. This method is only practical when you use a fairly low concentration of radioligand so its concentration after dilution is far below its Kd for binding.

You then measure binding at various times after that to determine how rapidly the radioligand falls off the receptors.

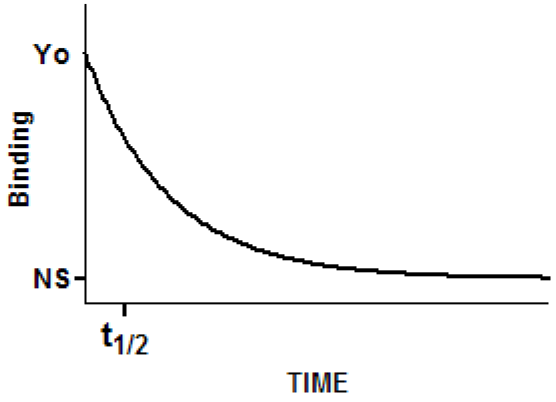
Step by step

Create an XY data table. Enter time into X, and total binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

From the table of specific binding, click Analyze, choose nonlinear regression, choose the panel of Kinetics Binding equations, and choose *Dissociation - One phase exponential decay .*

Model

```
Y=(Y0-NS)*exp(-K*X) + NS
```



Interpret the parameters

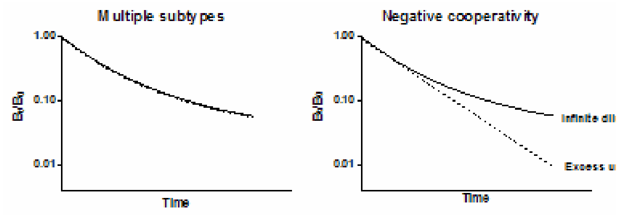
Y0 is the binding at time zero, in the units of the Y axis.

NS is the binding (nonspecific) at infinite times, in the units of the Y axis.

K is the rate constant in inverse units of the X axis. The half-life equals the ln(2) divided by K.

Checking for cooperativity

If the law of mass action applies, binding of a ligand to one binding site does not alter the affinity of another binding site. This also means that dissociation of a ligand from one site should not change the dissociation of ligand from other sites. To test this assumption, compare the dissociation rate after initiating dissociation by infinite dilution with the dissociation rate when initiated by addition of a large concentration of unlabeled drug. If the radioligand is bound to multiple noninteracting binding sites, the dissociation will be identical in both experimental protocols as shown in the left figure below. Note that the Y axis is shown using a log scale. If there were a single binding site, you would expect the dissociation data to appear linear on this graph. With two binding sites, the graph is curved even on a log axis.



The right figure shows ideal dissociation data when radioligand is bound to interacting binding sites with negative cooperativity. The data are different depending on how dissociation was initiated. If dissociation is initiated by infinite dilution, the dissociation rate will change over time. The dissociation of some radioligand will leave the remaining ligand bound more tightly. When dissociation is initiated by addition of cold drug, all the receptors are always occupied by ligand (some hot, some cold) and dissociation occurs at its maximal unchanging rate.

4.5.5.9.3 Equation: Association kinetics (one ligand concentration)

Introduction

When you measure the association rate of a radioligand, the rate at which the binding equilibrates depends not only on the association rate constant and the amount of ligand you used, but also on its dissociation rate constant. (Why? 598) The only way to fit the association rate constant by analyzing association data from one concentration of radioligand, is to constrain the dissociation rate constant to a value you determined in a different experiment.

Alternative methods of determining an association rate constant are to globally fit data obtained with multiple radioligand concentrations 603, or to analyze an experiment that measuresboth association and dissociation 605 rate sequentially.

Step by step

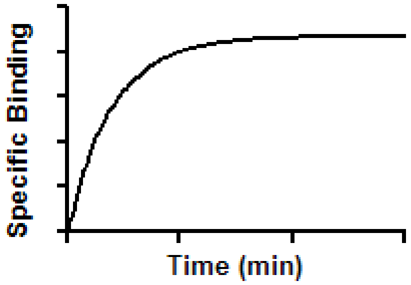
Create an XY data table. Enter time in minutes into X, and specific binding into Y.

From the table of specific binding, click Analyze, choose nonlinear regression, choose the panel of Kinetics Binding equations, and choose *Association kinetics - One conc. of hot.*

You must constrain Hotnm ([radioligand] in nM) and Koff (dissociation rate constant, in inverse minutes) to constant values.

Model

```
Kd=Koff/Kon
L=Hotnm*1e-9
Kob=Kon*L+Koff
Occupancy=L/(L+Kd)
Ymax=Occupancy*Bmax
Y=Ymax*(1 - exp(-1*kob*X))
```



Interpret the parameters

Kon is the association rate constant, in units of M^{-1} min^{-1}

Kd is the equilibrium dissociation constant, in Molar, computed as Koff/Kon

Bmax is the maximal binding at equilibrium, extrapolated to maximal radioligand concentration, in the units used to enter the Y values.

Notes

4.5.5.9.4 Equation: Association kinetics (two ligand concentrations)

Introduction

You cannot determine k_{on} from an association binding measured at a single concentration of radioligand unless you can constrain k 602 $_{off}$ 602 to a constant value based on other experiments 600. The observed association rate depends on the association rate constant, the amount of ligand you used, and its dissociation rate constant. With one concentration of radioligand, the results are ambiguous. .

If you perform an association kinetic experiments with multiple radioligand concentration, you can globally fit the data to the association kinetic model to derive a single best-fit estimate for k_{on} and one for k_{off}.

Shown below is an example of an association kinetic experiment conducted using two concentrations of radioligand. All other conditions (temperature, pH, etc.) were the same for both runs, of course. Times were entered into the X column, specific binding for one concentration of radioligand were entered into the first (A) Y column, and binding for the other concentration were entered into column B.

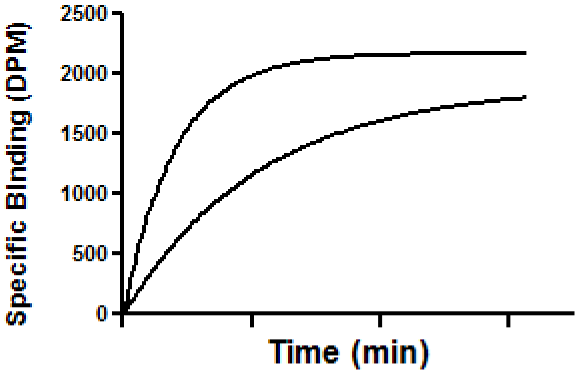
Step by step

Create an XY data table. Enter time in minutes into X, and total binding into Y. Enter binding at one concentration of radioligand into the column A, binding at another concentration into column B, etc. Enter the concentrations, in nM, into the column titles.

From the table of specific binding, click Analyze, choose nonlinear regression, choose the panel of Kinetics Binding equations, and choose *Association - Two or more conc. of hot.*

Model

```
Kd=Koff/Kon
L=Hotnm*1e-9
Kob=Kon*L+Koff
Occupancy=L/(L+Kd)
Ymax=Occupancy*Bmax
Y=Ymax*(1 - exp(-1*kob*X))
```



Interpret the parameters

Koff is the dissociation rate constant in min^{-1}.

Kon is the association rate constant, in units of M^{-1} min^{-1}.

Kd is the equilibrium dissociation constant, in Molar, computed as Koff/Kon

Bmax is the maximal binding at equilibrium, at maximal radioligand concentration, in the units you used to enter Y.

Notes

According to the law of mass action, the ratio of koff to kon is the Kd of receptor binding:

$$K_d = \frac{k_{off}}{k_{on}}$$

Compare the Kd calculated this way (from kinetic experiments) with the Kd determined from a saturation binding curve. If binding follows the law of mass action, the two Kd values should be indistinguishable.

4.5.5.9.5 Equation: Association then dissociation

Introduction

You cannot determine the association rate constant by simply observing the association of a single concentration of radioligand. The rate at which a ligand reaches equilibrium is determined not only by the association rate constant and the ligand concentration, but also by the dissociation constant.

One way to determine the association rate constant is to globally fit data obtained with two different concentrations of radioligand 602. An alternative approach, explained here is to measure association and dissociation in one experiment.

Add a radioligand and measure total binding at multiple time points, then at Time0 initiate dissociation (by adding an antagonist or by massive dilution) and then measure dissociation at various times.

Step by step

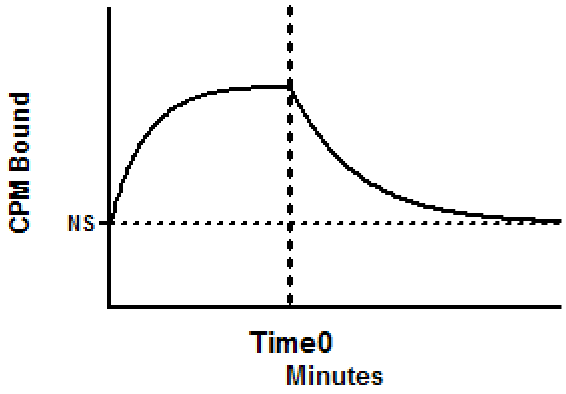
Create an XY data table. Enter time in minutes into X, and total binding into Y.

From the table of specific binding, click Analyze, choose nonlinear regression, choose the panel of Kinetics Binding equations, and choose *Association then dissociation.*

Constrain HotNM ([radioigand in nM] and Time0 (time at which dissociation was initiated) to constant values. If you entered specific binding into the Y column, also constrain NS to a constant value of zero.

Model

```
Radioligand=HotNM*1e-9
Kob=[Radioligand]*Kon+Koff
Kd=Koff/Kon
Eq=Bmax*radioligand/(radioligand + Kd)
Association=Eq*(1-exp(-1*Kob*X))
YatTime0 = Eq*(1-exp(-1*Kob*Time0))
Dissociation= YatTime0*exp(-1*Koff*(X-Time0))
Y=IF(X<Time0, Association, Dissociation) + NS
```



Interpret the parameters

Koff is the dissociation constant in min^{-1}.

Kon is the association constant in inverse minutes multiplied by inverse concentration.

KD is computed from Koff/Kon. Expressed in Molar units.

Bmax is the maximum binding at equilibrium with maximum concentration of radioligand, in units of Y axis. Unless you used a very high concentration of radioligand, Bmax will have a wide confidence interval as the experiment is not designed to determine Bmax.

NS is the nonspecific binding, in units of the Y axis. It is the Y value at time 0, and also the Y value at very late times after all the ligand has dissociated.

Notes

4.5.5.9.6 Equation: Kinetics of competitive binding

Introduction

Kinetics experiments can determine the dissociation and association rate constants (off-rate and on-rate) of an unlabeled compound. Add labeled and unlabeled ligand together and measure the binding of the labeled ligand over time. Fit to the appropriate model described below, constraining the rate constants of the labeled ligand to constant values determined from other experiments, and fit the rate constants of the unlabeled compound.

Using only a single concentration of labeled and radioligand, it is very hard to determine the rate constants with any reasonable precision. But measure the kinetics at two (or more) concentrations of the unlabeled ligand, and the results are much more precise.

Step by step

Create an XY data table. Enter time in minutes into X, and specific binding in cpm into Y. Enter the binding for one concentration of of the unlabeled ligand in column A, and another concentration in column B, etc.. Enter the concentrations, in nM, into the column titles.

From the table of specific binding, click Analyze, choose nonlinear regression, choose the panel of Kinetics Binding equations, and choose *Kinetics of competitive binding.*

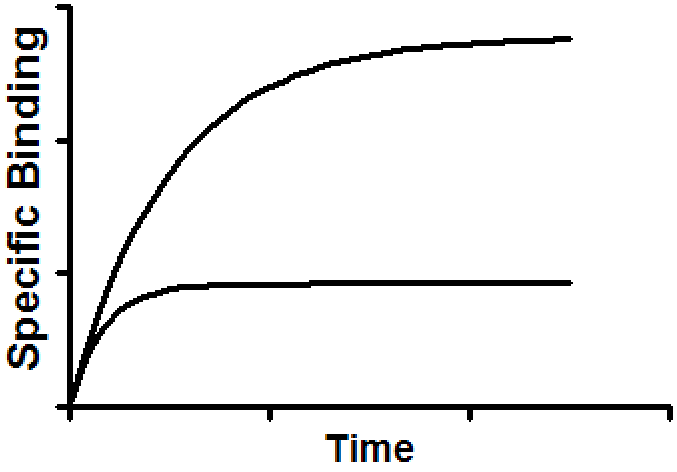
Constrain k1 and k2 to constant values determined from kinetic binding experiments. k1 is the association rate constant of the hot ligand in M^{-1} min^{-1} and k2 is its dissociation rate constant in units of min^{-1}.

Also constrain L to be a constant value equal to the concentration of labeled ligand in nM.

I is constrained to be a column constant whose value comes from the column titles.

Model

```
KA = K1*L*1E-9 + k2
KB = K3*I*1e-9 + K4
S=SQRT((KA-KB)^2+4*K1*K3*L*I*1e-18)
KF = 0.5 * (Ka + KB + S)
KS = 0.5 * (KA + KB - S)
DIFF=KF - KS
Q=Bmax*K1*L*1e-9/DIFF
Y=Q*(k4*DIFF/(KF*KS)+((K4-Kf)/KF)*exp(-KF*X)-((K4-KS)/KS)*exp(-KS*X))
```



Interpret the parameters

k3 is the association rate constant of unlabeled ligand in M^{-1} min^{-1}.

k4 is the dissociation rate constant of unlabeled ligand in min^{-1}.

Bmax is the total number of receptors. Either leave as a variable or set to a constant you know from other experiments. The Bmax is the maximum binding at equilibrium with a very high concentration of radioligand. It is usually much larger than any binding seen in the experiment.

All three parameters are constrained to be shared, so Prism fits one value of each of the three parameters for all the data sets.

Notes

- This equation does not account for ligand depletion. It assumes that only a small fraction of radioligand binds to receptors, so that the free concentration of radioligand is very close to the added concentration.
- This method will only give reliable results if you have plenty of data points at early time points.
- The ratio K4/K3 is the equilibrium dissociation constant of the cold ligand in Molar. You should compare this value (determined via kinetics) with the same value determined by equilibrium competition.

Extensions to this model

Pharmechanics has adapted this equation to extend the analysis of the kinetics of competitive binding in two ways:

- To work with different orders of addition of receptor, hot compound and cold compound. The kinetics of competitive binding equation was designed for the most commonly-used assay format, in which receptor is exposed to both ligands simultaneously, i.e. receptor is added last. There are occasions when the ordering of reagent addition is different, for example preincubation of receptor with unlabeled compound, or with labeled ligand
- To work with rapidly-dissociating compounds. The kinetics of competitive binding equation was developed for unlabeled ligands that dissociate sufficiently slowly that the fit provides reasonable estimates of the kinetic parameters. When the unlabeled ligand dissociates rapidly, an ambiguous fit often results because the ligand equilibrates so rapidly that the kinetic assumption no longer applies.In this scenario, the data can be better fit by an equation that assumes the unlabeled ligand rapidly equilibrates with the free receptors.

Learn more and download a Prism file here.

Reference

This method was described by Motulsky and Mahan in Molecular Pharmacology 25:1-9, 1984.

4.5.5.10 Enzyme kinetics -- Key concepts

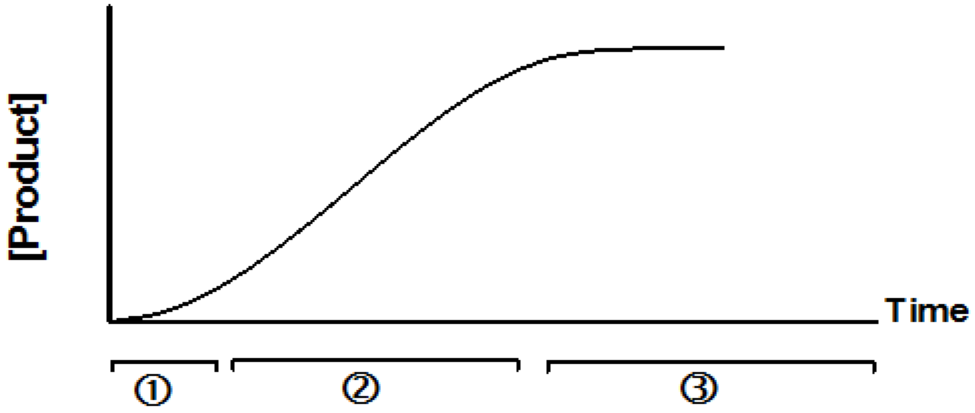
4.5.5.10.1 Key concepts: Terminology

What is an enzyme?

Living systems depend on chemical reactions which, on their own, would occur at extremely slow rates. Enzymes are catalysts that reduce the needed activation energy so these reactions proceed at rates that are useful to the cell. The study of enzyme kinetics can help us understand the function and regulation of enzymes.

Enzyme progress curves

In most cases, an enzyme converts one chemical (the substrate) into another (the product). A graph of product concentration vs. time follows three phases marked on the graph below.



1. At very early time points (usually less than a second), the rate of product accumulation increases over time. Special techniques, not available in Prism, are needed to study the early kinetics of enzyme action. The graph above exaggerates this first phase.

2. For an extended period of time, the product concentration increases linearly with time. All the analyses built-in to Prism use data collected during this second phase.

3. At later times, the substrate is depleted, so the curve starts to level off. Eventually the concentration of product reaches a plateau.

It is very difficult to fit a curve to these kind of data. The model simply cannot be reduced to an equation that expresses product concentration as a function of time. To fit these kind of data (called an enzyme progress curve) you need to use a program that can fit data to a model defined by differential equations or by an implicit equation. For more details, see RG Duggleby, Analysis of Enzyme Reaction Progress Curves by Nonlinear Regression, Methods in Enzymology, 249: 61-60, 1995.

Rather than fit the enzyme progress curve, most analyses of enzyme kinetics (including all those built-in to Prism) measure product at a single time point. Analyses assume that the time point you chose is on the linear (second) phase of product accumulation and ignore the nonlinear first phase (which is usually very short). Therefore, if you divide the amount of product produced by the time the reaction was allowed to proceed, you compute the amount of product formed per unit time, which is the enzyme velocity.

Terminology

The terminology can be confusing. Note these confusing points:

- As mentioned above, almost all studies of enzyme "**kinetics**" are done by collecting data at a single time point. The X axis is substrate (or inhibitor) concentration, not time.

- The second phase shown in the graph above is often called the "**initial rate**", a phrase that makes sense only if you ignore the short transient phase that precedes it.

- That second phase is also called "**steady state**", because the concentration of enzyme-substrate complex doesn't change during that phase. However, the concentration of product accumulates, so the system is not truly at steady state until, much later, the concentration of product truly doesn't change over time.

4.5.5.10.2 Key concepts: Assumptions

Standard analyses of enzyme kinetics (the only kind discussed here) assume:

- The production of product is linear with time during the time interval used.
- The concentration of substrate vastly exceeds the concentration of enzyme. This means that the free concentration of substrate is very close to the concentration you added, and that substrate concentration is essentially constant throughout the assay.
- A single enzyme forms the product.
- There is negligible spontaneous creation of product without enzyme.

With the exception of the built-in allosteric model, Prism's built-in equations also assume:

- No cooperativity. Binding of substrate to one enzyme binding site doesn't influence the affinity or activity of an adjacent site.
- Neither substrate nor product acts as an allosteric modulator to alter the enzyme velocity.

4.5.5.11 Enzyme kinetics - Velocity as a function of substrate

4.5.5.11.1 Key concepts: Substrate vs. velocity

Fitting a substrate-velocity curve

The simplest experiment in enzyme kinetics is to vary the substrate concentration and measure enzyme velocity (also called enzyme activity).

The standard way to fit these data is to fit the Michaelis-Menten model 613 to determine the Vmax (maximum enzyme velocity) and its Km (the concentration of substrate needed to get half-maximal velocity.

The Vmax equals the product of the concentration of active enzyme sites times the turnover rate, kcat. This is the number of substrate molecules each enzyme site can convert to product per unit time. If you know the concentration of enzyme, you can fit the curve to determine kcat and Km 616. The curve will be identical to the Michaelis-Menten fit.

If the enzyme has cooperative subunits, the graph of enzyme velocity as a function of substrate concentration will appear sigmoidal. Prism offers one empirical equation for fitting sigmoidal substrate-velocity curves 619.

4.5.5.11.2 Equation: Michaelis-Menten model

Introduction

The most common kind of enzyme kinetics experiment is to vary the concentration of substrate and measure enzyme velocity. The goal is to determine the enzyme's Km (substrate concentration that yield a half-maximal velocity) and Vmax (maximum velocity). If your goal is to determine the turnover number kcat, rather than the Vmax, use an alternative version 616 of the equation.

Step by step

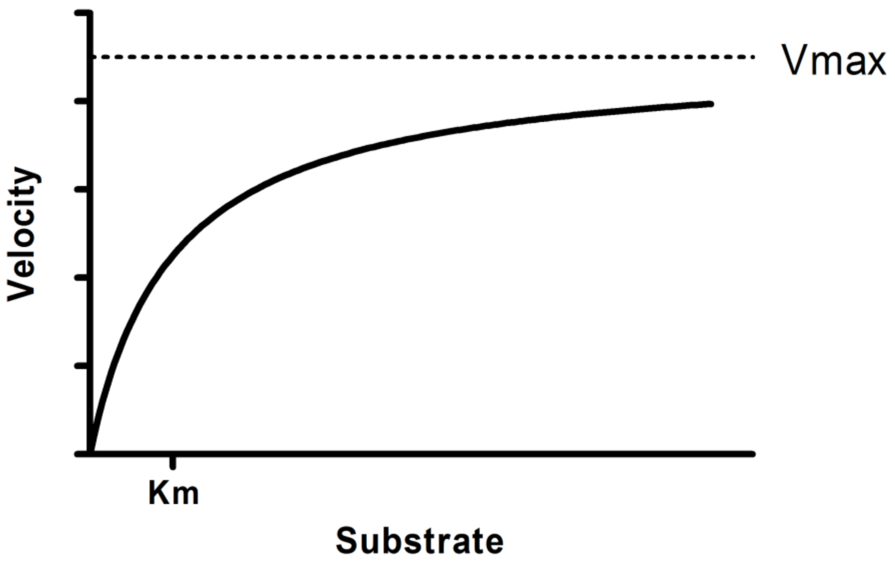
Create an XY data table. Enter substrate concentration into X, and enzyme velocity into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

You can also choose Prism's sample data: Enzyme kinetics -- Michaelis-Menten.

After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Michaelis-Menten enzyme kinetics*.

Model

```
Y = Vmax*X/(Km + X)
```



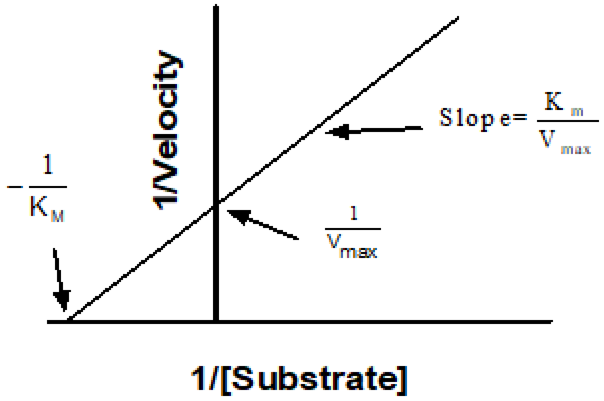
Interpret the parameters

Vmax is the maximum enzyme velocity in the same units as Y. It is the velocity of the enzyme extrapolated to very high concentrations of substrate, so its value is almost always higher than any velocity measured in your experiment.

Km is the Michaelis-Menten constant, in the same units as X. It is the substrate concentration needed to achieve a half-maximum enzyme velocity.

Create a Lineweaver-Burk plot

Before nonlinear regression was available, investigators had to transform curved data into straight lines, so they could analyze with linear regression. One way to do this is with a Lineweaver-Burk plot, which plots the reciprocal of substrate concentration vs. the reciprocal of enzyme velocity.



If you create a Lineweaver-Burk plot, use it only to display your data. Don't use the slope and intercept of a linear regression line to determine values for Vmax and Km. If you do this, you won't get the most accurate values for Vmax and Km. The problem is that the transformations (reciprocals) distort the experimental error, so the double-reciprocal plot does not obey the assumptions of linear regression. Use nonlinear regression to obtain the most accurate values of Km and Vmax.

To create a Lineweaver-Burk plot with Prism, use the Transform analysis, then choose the panel of biochemistry and pharmacology transforms.

To create a Lineweaver-Burke line corresponding to the nonlinear regression fit, follow these steps:

1. Create a new XY data table, with no subcolumns.

2. Into row 1 enter X=-1/KM, Y =0 (previously determined by nonlinear regression).

3. Into row 2 enter X=1/Smin (Smin is the smallest value of [substrate] you want to include on the graph) and Y=(1/Vmax)(1.0 + KM/Smin).

4. Note the name of this data table. Perhaps rename it to something appropriate.

5. Go to the Lineweaver-Burke graph.

6. Drag the new table from the navigator and drop onto the graph.

7. Double-click on one of the new symbols for that data set to bring up the Format Graph dialog.

8. Choose to plot no symbols, but to connect with a line.

Notes

- See the list of assumptions 610 of all analyses of enzyme kinetics.

- This equation fits exactly the same curve as the equation that fits the turnover number Kcat 616 rather than the Vmax. The product of Kcat times Et (the concentration of enzyme sites) equals the Vmax, so if you know Et, Prism can fit kcat.

- This equation is a special case of the equation for allosteric enzymes 619. That allosteric model adds an additional parameter: the Hill slope h. When h equals 1.0, the two models are identical.

- Note that Km is not a binding constant that measures the strength of binding between the enzyme and substrate. Its value takes into account the affinity of substrate for enzyme, and also the rate at which the substrate bound to the enzyme is converted to product.

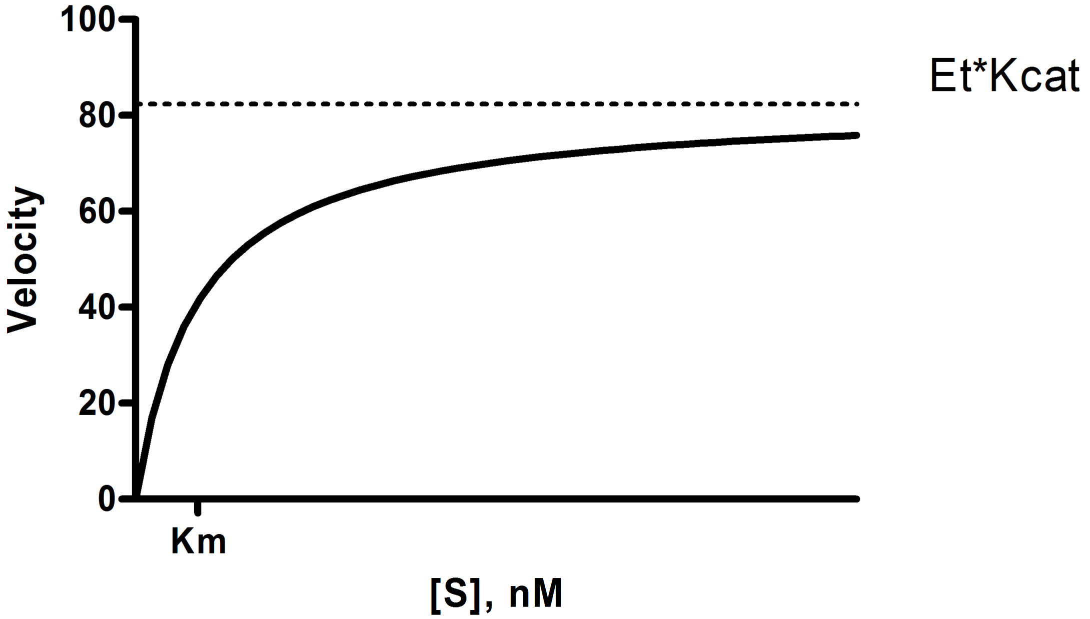
4.5.5.11.3 Equation: Determine kcat

Introduction

Kcat is the turnover number -- the number of substrate molecule each enzyme site converts to product per unit time. If you know the concentration of enzyme sites, you can fit Kcat instead of Vmax when analyzing a substrate vs. velocity curve.

The model

```
Y = Et*kcat*X/(Km + X)
```



X is the substrate concentration.

Y is enzyme velocity.

kcat is the turnover number, the number of times each enzyme site converts substrate to product per unit time. This is expressed in the inverse of the time units of the Y axis. For example, if Y is in micromoles of substrate per minute, then kcat is the number of molecules of substrate produced per catalytic site per minute.

Km is the Michaelis-Menten constant, in the same units as X. It is the substrate concentration needed to achieve a half-maximum enzyme velocity.

Et is the concentration of enzyme catalytic sites. If the enzyme has multiple subunits, note that Et is the concentration of catalytic sites, which can be larger than the concentration of enzyme molecules. The Y values you enter are enzyme velocity entered in units of concentration per time. Et must be entered in those same concentration units (while the time units are defined by kcat).

Vmax is the maximum enzyme velocity in the same units as Y. It is not directly shown in the model above. It is the velocity of the enzyme extrapolated to very high concentrations of substrate, so is almost always higher than any velocity measured in your experiment. It is computed by multiplying Et times kcat.

Relationship to the Michaelis-Menten model

The curve shown above is identical to the curve defined by the Michaelis-Menten model. You'll get identical curves when you fit either model to your data, and identical values for Km.

The Michaelis-Menten model finds the Vmax, which is the maximum enzyme velocity extrapolated out to very high concentrations of substrate. It is expressed in the same units you used to enter your Y values (enzyme activity). Usually it is straightforward to express this (or convert to) moles/minute/mg of protein. The Vmax is determined by how many enzyme sites are present (Et) and the rate at which the enzyme can convert substrate to product (kcat).

If you know the concentration of enzyme sites you've added to the assay (Et) then you can fit the catalytic constant Kcat using the model above.

When calculating Kcat, the concentration units cancel out, so Kcat is expressed in units of inverse time. It is the turnover number -- the number of substrate molecule each enzyme site converts to product per unit time.

Fitting the model with Prism

1. Create an XY data table. Enter substrate concentration into X, and enzyme velocity into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc. You can also choose Prism's sample data: Enzyme kinetics -- Michaelis-Menten.

2. After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Kcat*.

3. You must constrain **Et** to a constant value, based on other experiments. To constrain the value of Et, go to the Constrain tab of the nonlinear regression dialog, make sure that the drop down next to Et is set to "Constant equal to" and enter the value. For the sample data, enter 100 as the value of Et.

   If you don't know the value of Et, you cannot fit the kcat, but instead should fit the Vmax 613. It is not possible for Prism to fit both the kcat and Et, as the two parameters are intertwined, and a substrate-velocity curve gives no information about their individual values.

4. With the sample data, Prism reports that Km= 5.886 with a 95% confidence interval ranging from 3.933 to 7.839. The best fit value of kcat is 13.53 with a 95% confidence interval ranging from 11.97 to 15.09.

Notes

- See the list of assumptions 610 of all analyses of enzyme kinetics.

- This equation fits exactly the same curve as the equation that fits Vmax 613, rather than the turnover number Kcat. The product of Kcat times Et (the concentration of enzyme sites) equals the Vmax.

- This equation is related to the equation for allosteric enzymes. That allosteric model adds an additional parameter: the Hill slope h. When h equals 1.0, the two models are identical.

4.5.5.11.4 Equation: Allosteric sigmoidal

Introduction

If the enzyme has cooperative subunits, the graph of enzyme velocity as a function of substrate concentration will appear sigmoidal. Prism offers one empirical equation for fitting sigmoidal substrate-velocity curves. Read advanced books on enzyme kinetics for alternative methods based on molecular models of allosteric action.

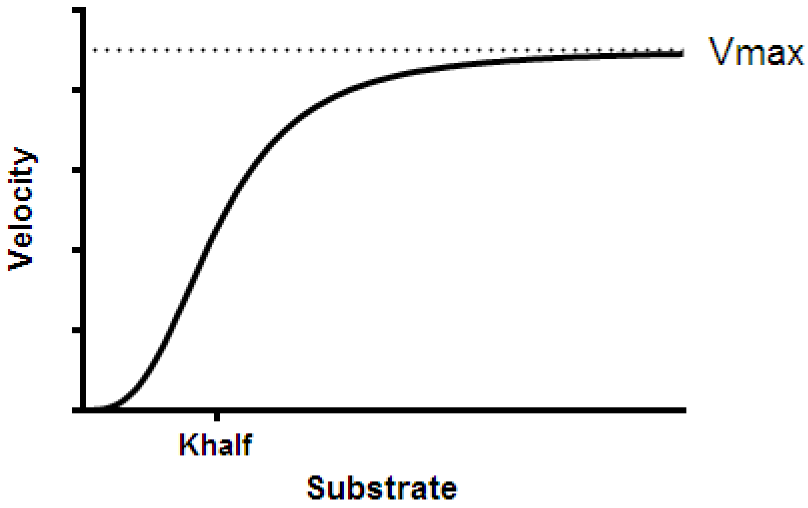
How to enter data

Create an XY data table. Enter substrate concentration into X, and enzyme velocity into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Allosteric sigmoidal enzyme kinetics*.

The model

```
Y=Vmax*X^h/(Khalf^h + X^h)
```



Interpret the parameters

Vmax is the maximum enzyme velocity in the same units as Y. It is the velocity of the enzyme extrapolated to very high concentrations of substrate, and therefore is almost always higher than any velocity measured in your experiment.

Khalf is the concentration of substrate that produces a half-maximal enzyme velocity. It is the EC50.

h is the Hill slope. When h=1, this equation is identical to the standard Michaelis-Menten equation 613. When it is greater than 1.0, the curve is sigmoidal due to positive cooperativity. The variable h does not always equal the number of interacting binding sites, but its value can not exceed the number of interacting sites. Think of h as an empirical measure of the steepness of the curve and the presence of cooperativity.

Kprime is related to the Km. It is computed as Khalf^h, and is expressed in the same units as X.

An alternative form of the equation

An alternative version of the equation (equation 5.47 in the book by Dr. Copeland referenced below) fits Kprime.

```
Y=Vmax*X^h/(Kprime + X^h)
```

Note that Kprime in this equation equals Khalf^h in the equation built in to Prism. The two models generate exactly the same curve, and simply have alternative methods for reporting the parameters. Prism reports both Kprime and Khalf.

Reference

Equation 5.47, in RA Copeland, *Enzymes,* 2nd edition, Wiley, 2000. In this reference, Copeland shows how to fit Kprime. Via personal communication, he extended the model to also fit the Khalf.

4.5.5.12 Enzyme kinetics -- Inhibition

Many drugs work by inhibiting enzyme activity, either by preventing the substrate from binding to the enzyme, or by stabilizing the enzyme-substrate complex so as to slow formation of product.To distinguish between the models of enzyme inhibition and determine the Ki of the inhibitor, measure substrate-velocity curves in the presence of several concentrations of inhibitor (including one curve with no inhibitor).

4.5.5.12.1 Key concepts: Enzyme inhibition

Many drugs work by inhibiting enzyme activity, either by preventing the substrate from binding to the enzyme, or by stabilizing the enzyme-substrate complex so as to slow formation of product.To distinguish between the models of enzyme inhibition and determine the Ki of the inhibitor, measure substrate-velocity curves in the presence of several concentrations of inhibitor (including one curve with no inhibitor).

Prism can fit your data to six models of enzyme inhibition:

- A competitive 622 inhibitor reversibly binds to the same site as the substrate, so its inhibition can be entirely overcome by using a very high concentration of substrate. The maximum velocity of the enzyme doesn't change (if you give it enough substrate), but it takes more substrate to get to half maximal activity. The substrate-velocity curve is shifted to the left but not down.

- A noncompetitive 624 inhibitor binds with equal affinity to the enzyme, and the enzyme-substrate complex. The inhibition is not surmountable by increasing substrate concentration. The substrate-velocity curve is shifted down but neither to the right or left.

- An uncompetitive [627] inhibitor reversibly binds to the enzyme-substrate complex, but not to the enzyme itself. This reduces both the effective Vmax and the effective Km. The substrate-velocity curve is shifted down and to the left.
- The mixed [629] model is a general model that includes competitive, noncompetitive and uncompetitive models as special cases. The model has one more parameter than the others, and the extra parameter (alpha) determines the degree to which the binding of inhibitor changes the affinity of the enzyme for substrate.
- Substrate inhibition [630]. In some cases, the substrate of an enzyme also inhibits the enzyme by binding to a second site on the enzyme.
- Tight inhibition [632]. This model accounts for tight binding, so it does not assume that the free concentration of inhibitor equals the total concentration.

Reference

RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Second edition, Wiley 2013. IBSN: 978-1-118-48813-3.

4.5.5.12.2 Equation: Competitive inhibition

Introduction

A competitive inhibitor reversibly binds to the same site as the substrate, so its inhibition can be entirely overcome by using a very high concentration of substrate. The Vmax doesn't change, and the effective Km increases. You can determine the Ki of a competitive inhibitor by measuring substrate-velocity curves in the presence of several concentrations of inhibitor.

Step by step

Create an XY data table. Enter substrate concentration into the X column, and enzyme activity into the Y columns. Each data set (Y column) represents data collected in the presence of a different concentration of inhibitor, starting at zero. Enter these concentrations into the column titles. Be sure to enter concentrations, not logarithms of concentration.

Alternatively, choose the competitive enzyme inhibition sample data set.

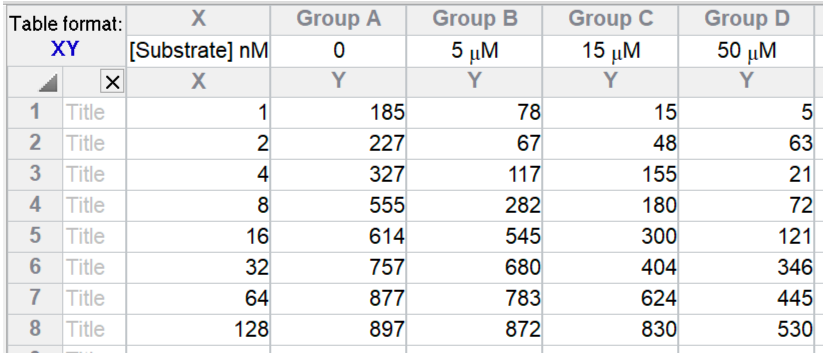


Table format:		X	Group A	Group B	Group C	Group D
XY		[Substrate] nM	0	5 µM	15 µM	50 µM
	☒	X	Y	Y	Y	Y
1	Title	1	185	78	15	5
2	Title	2	227	67	48	63
3	Title	4	327	117	155	21
4	Title	8	555	282	180	72
5	Title	16	614	545	300	121
6	Title	32	757	680	404	346
7	Title	64	877	783	624	445
8	Title	128	897	872	830	530

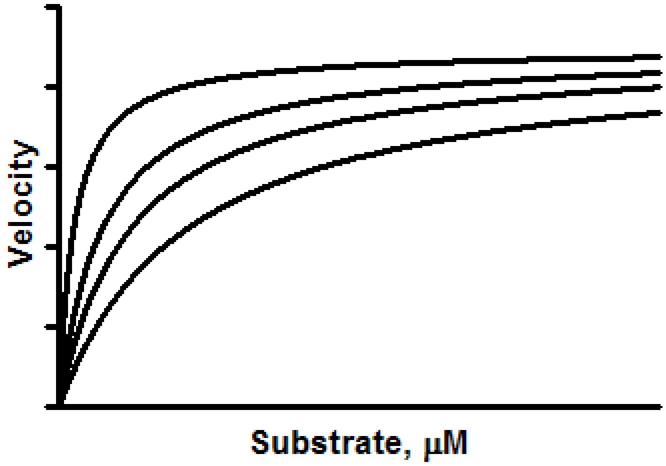
After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Competitive enzyme inhibition*.

The model

```
KmObs=Km*(1+[I]/Ki)
Y=Vmax*X/(KmObs+X)
```

The constant **I** is the concentration of inhibitor, a value you enter into each column title. This is constrained to equal a data set constant.

The parameters **Vmax**, **Km** and **Ki** are shared, so Prism fits one best-fit value for the entire set of data.



Interpreting the parameters

Ki is the inhibition constant, expressed in the same units as I, which you entered into the column titles.

Vmax is the maximum enzyme velocity, *in the absence of inhibitor*, expressed in the same units as Y.

Km is the Michaelis-Menten constant, expressed in the same units as X. It describes the interaction of substrate and enzyme *in the absence of inhibitor*.

If the data don't fit the model well, consider instead fitting to a noncompetitive 624 or uncompetitive 627 model. Or fit to the more general equation for mixed-model inhibition 629.

Reference

Equation 3.1 in: RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. IBSN:0471686964.RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. IBSN:0471686964.

4.5.5.12.3 Equation: Noncompetitive inhibition

Terminology

Copeland suggests not using this equation, which is simplistic. Instead he suggests using the equation we call mixed-model inhibition 629 (but he calls noncompetitive inhibition).

Introduction

A noncompetitive inhibitor reversibly binds to both the enzyme-substrate complex, and the enzyme itself. This means that the effective Vmax decreases with inhibition but the Km does not change. You can determine the Ki of a competitive inhibitor by measuring substrate-velocity curves in the presence of several concentrations of inhibitor.

The term 'noncompetitive' is used inconsistently. It is usually used as defined above, when the inhibitor binds with identical affinity to the free enzyme and the enzyme-substrate complex. Sometimes, however, the term 'noncompetitive' is used more generally, when the two binding affinities differ, which is more often called mixed-model inhibition 629.

Step by step

Create an XY data table. Enter substrate concentration into the X column, and enzyme activity into the Y columns. Each data set (Y column) represents data collected in the presence of a different concentration of inhibitor, starting at zero. Enter these concentrations into the column titles. Be sure to enter concentrations, not logarithms of concentration.

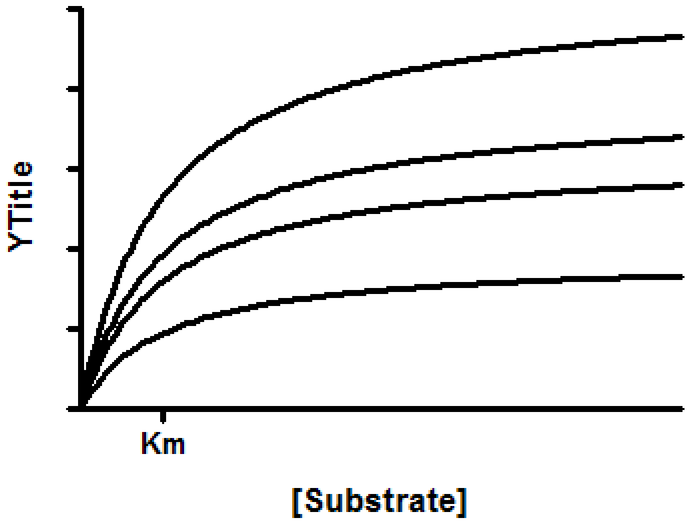
After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Noncompetitive enzyme inhibition*.

The model

```
Vmaxinh=Vmax/(1+I/Ki)
Y=Vmaxinh*X/(Km+X)
```

The constant **I** is the concentration of inhibitor, a value you enter into each column title. This is constrained to be a data set constant.

The parameters **Vmax**, **Km** and **Ki** are shared, so Prism fits one best-fit value for the entire set of data.



Interpreting the parameters

Vmax is the maximum enzyme velocity in *the absence of inhibitor*, expressed in the same units as Y.

Km is the Michaelis-Menten constant (in the *absence of inhibitor*), expressed in the same units as X. It describes the interaction of substrate and enzyme in the absence of inhibitor.

Ki is the inhibition constant, expressed in the same units as I, which you entered.

If the data don't fit the model well, consider instead fitting to a competitive 622 or uncompetitive 627 model. Or fit to the more general equation for mixed-model inhibition 629.

Reference

Equation 3.4 in: RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Second edition, Wiley 2013. IBSN: 978-1-118-48813-3.

4.5.5.12.4 Equation: Uncompetitive inhibition

Introduction

An uncompetitive inhibitor binds to the enzyme-substrate complex, but not the free enzyme. This reduces both the effective Vmax and the effective Km. The substrate-velocity curve is shifted down and to the left.

You can determine the Ki of a competitive inhibitor by measuring substrate-velocity curves in the presence of several concentrations of inhibitor.

Step by step

Create an XY data table. Enter substrate concentration into the X column, and enzyme activity into the Y columns. Each data set (Y column) represents data collected in the presence of a different concentration of inhibitor, starting at zero. Enter these concentrations into the column titles. Be sure to enter concentrations, not logarithms of concentration.

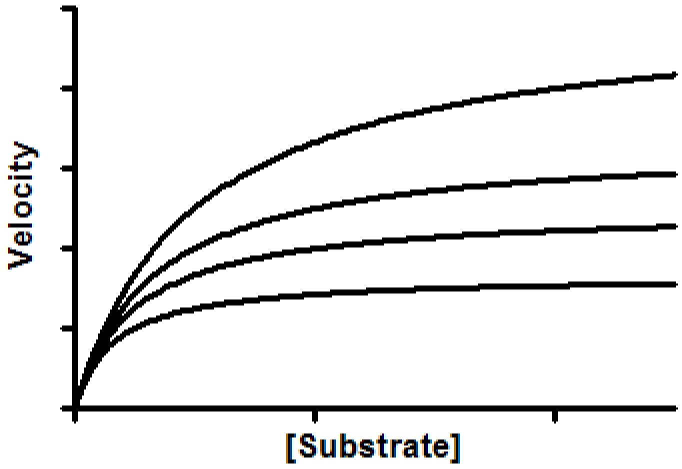
After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Uncompetitive enzyme inhibition*.

The model

```
VmaxApp=Vmax/(1+I/AlphaKi)
KmApp=Km/(1+I/AlphaKi)
Y=VmaxApp*X/(Kmapp+X)
```

The constant **I** is the concentration of inhibitor, a value you enter into each column title. This is constrained to equal a data set constant.

The parameters **Vmax**, **Km** and **Ki** are shared, so Prism fits one best-fit value for the entire set of data.



Interpreting the parameters

Vmax is the maximum enzyme velocity *absence of inhibitor*, expressed in the same units as Y.

Km is the Michaelis-Menten constant (*absence of inhibitor*), expressed in the same units as X. It describes the interaction of substrate and enzyme in the absence of inhibitor.

AlphaKi is the inhibition constant, expressed in the same units as I, which you entered into the column titles. It is the product of Ki (which is very high, because uncompetitive inhibitors don't bind to the enzyme) and alpha (which is very low). It is not possible to fit alpha and Ki separately, but only to determine their product. Some books call this product Ki'.

If the data don't fit the model well, consider instead fitting to a competitive 622 or noncompetitive 624 model. Or fit to the more general equation for mixed-model inhibition 629.

Reference

Equation 3.6 in: RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. IBSN:0471686964.RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. IBSN:0471686964.

4.5.5.12.5 Equation: Mixed-model inhibition

Introduction

The mixed model is a general equation that includes competitive 622, uncompetitive 627 and noncompetitive 624 inhibition as special cases. The model has one more parameter than the others, and this parameter tells you about the mechanism of inhibition.

Step by step

Create an XY data table. Enter substrate concentration into the X column, and enzyme activity into the Y columns. Each data set (Y column) represents data collected in the presence of a different concentration of inhibitor, starting at zero. Enter these concentrations into the column titles. Be sure to enter concentrations, not logarithms of concentration.

After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Mixed model enzyme inhibition*.

Model

```
VmaxApp=Vmax/(1+I/(Alpha*Ki))
KmApp=Km*(1+I/Ki)/(1+I/(Alpha*Ki))
Y=VmaxApp*X/(KmApp + X)
```

The parameter **I** is the concentration of inhibitor, a value you enter into each column title. This is constrained to equal a data set constant.

The parameters **Alpha**, **Vmax**, **Km** and **Ki** are shared, so Prism fits one best-fit value for the entire set of data.

Interpreting the parameters

Vmax is the maximum enzyme velocity *absence of inhibitor*, expressed in the same units as Y.

Km is the Michaelis-Menten constant, expressed in the same units as X. It describes the interaction of substrate and enzyme in the *absence of inhibitor.*

Ki is the inhibition constant, expressed in the same units as I, which you entered into the column titles.

Alpha determines mechanism. Its value determines the degree to which the binding of inhibitor changes the affinity of the enzyme for substrate. Its value is always greater than zero.

- When Alpha=1, the inhibitor has equal affinitity for the enzyme and the enzyme-subtrate conmples. This is identical to noncompetitive 624 inhibition.
- When Alpha>1, the inhibitor preferentially binds to the free enzyme.
- When Alpha is very large, binding is almost entirely to the free enzyme, and the mixed-model approaches competitive 622 inhibition.
- When Alpha<1, the inhibitor preferentially binds to the enzyme-substrate complex.
- When Alpha is very small (but greater than zero), the inhibitor binds almost entirely to the enzyme-substrate complex and the mixed model becomes approaches an uncompetitive 627 model.

Terminology

Copeland suggests avoiding the term *mixed model*, because it is easy to get confused and think that refers to inhibition by a mixture or two or more drugs. Instead, he refers to the model on this page as *noncompetitive*. With this way of thinking, the case where alpha = 1.0 is simply a special case of noncompetitive inhibition.

Reference

Equation 3.2 in: RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Second edition, Wiley 2013. ISBN: 978-1-118-48813-3.

4.5.5.12.6 Equation: Substrate inhibition

Introduction

At high concentrations, some substrates also inhibit the enzyme activity. Substrate inhibition occurs with about 20% of all known enzymes. It happens when two molecules of substrate can bind to the enzyme, and thus block activity.

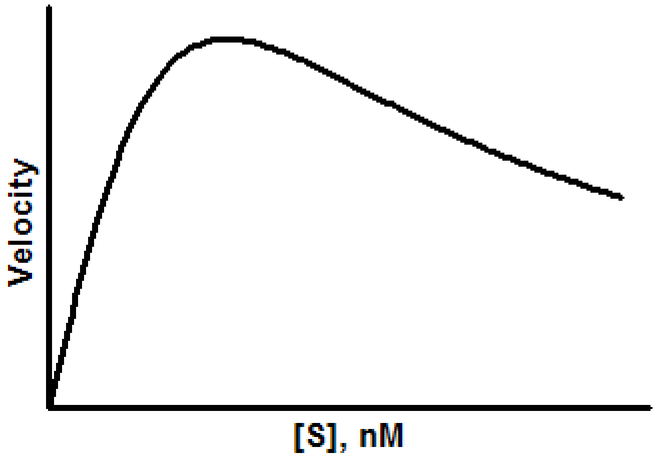
Step by step

Create an XY data table. Enter substrate concentration into the X column, and enzyme activity into the Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Substrate inhibition*.

Model

```
Y=Vmax*X/(Km + X*(1+X/Ki))
```



Parameters

Vmax is the maximum enzyme velocity, *if the substrate didn't also inhibit enzyme activity*, expressed in the same units as Y.

Km is the Michaelis-Menten constant, expressed in the same units as X. It describes the interaction of substrate and enzyme *in the absence of inhibitor*.

Ki is the dissociation constant for substrate binding in such a way that two substrates can bind to an enzyme. It is expressed in the same units as X.

Note: Prism 7 and later use a different (improved) set of rules for initial values than did Prism 6.

Why this model doesn't always converge on a fit

It is easier to understand the problems of fitting this model if it is rearranged:

```
Y=Vmax/(Km/X + 1 + X/Ki)
```

Vmax controls the height of the peak (but doesn't equal the Y value at the peak).

When X values are small, the last term approaches zero, so the early parts of the curve (small X values) are determined by the value of Km.

Conversely, when X values are large, the first term approaches zero, so the later parts of the curve (higher X values) are determined by the value of Ki.

The middle part of the curve is determined by both Ki and Km.

There are two reasons why the fit would result in an "ambiguous" result.

- There are not data points at a wide enough range of X values. There need to be X values less than Km and greater than Ki, ideally lots of X values. If you only have data in the middle of the curve, there simply is no information to separately determine Km and Ki.

- The shape of the curve, while it looks like a substrate inhibition curve, simply isn't quite right. In other words, the data really don't comply with the substrate inhibition model.

Reference

Equation 5.44, in RA Copeland, *Enzymes,* 2nd edition, Wiley, 2000.

4.5.5.12.7 Equation: Tight inhibition (Morrison equation)

Introduction

This equation accounts for tight binding, so it does not assume that the free concentration of inhibitor equals the total concentration.

Step by step

Create an XY data table. Enter inhibitor concentration into the X column (usually in micromolar, but any concentration units is fine), and enzyme activity into the Y columns (any units). If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Morrison Ki*.

Constrain Et, S and Km to constant values

You must constrain three parameters to constant values. To constrain the values, go to the Constrain tab of the nonlinear regression dialog, make sure that the drop down next to Et, S and Km is set to "Constant equal to" and enter the values.

- **Et** is the concentration of enzyme catalytic sites in the same units as the X values. If the enzyme has multiple subunits, note that Et is the concentration of catalytic sites, which can be larger than the concentration of enzyme molecules.

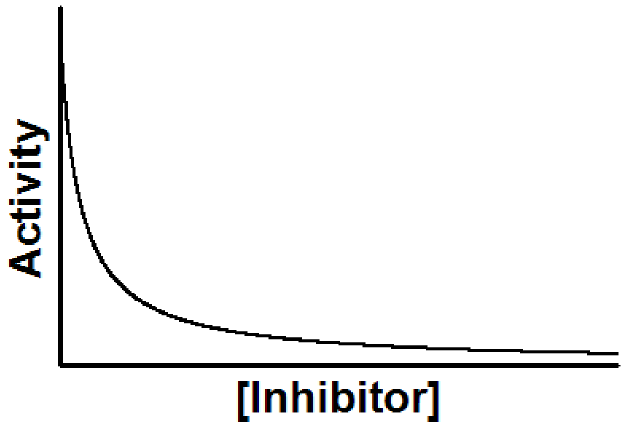
- **S** is the concentration of substrate you chose to use. Use the same units as the X values.

- **Km** is the Michaelis-Menten constant, expressed in the same units as X, determined in an experiment 613 without competitor.

Prism cannot fit any of these parameters from the graph of activity vs inhibitor concentration. You must know S from your experimental design, determine Km and Et in other experiments, and constrain all three to constant values.

Model

```
Q=(Ki*(1+(S/Km)))
Y=Vo*(1-((((Et+X+Q)-(((Et+X+Q)^2)-4*Et*X)^0.5))/(2*Et)))
```



Interpreting parameters

V0 is the enzyme velocity in the *absence of inhibitor*, expressed in the same units as Y. This is not the same as Vmax,which would require a maximal concentration of substrate.

Ki is the inhibition constant, expressed in the same units as X.

The IC50 is not the same as the Ki (Kusmic). Instead, IC50 = Et/2 + Ki

Reference

Equation 9.6, in RA Copeland, *Enzymes,* 2nd edition, Wiley, 2000.

Petr Kuzmic, Why IC50s are bad for you and other surprises

4.5.5.12.8 Comparing models of enzyme inhibition

Comparing the fits of the different enzyme inhibition models requires thinking both about enzyme mechanism and about statistics. Some notes:

- The mixed model 629 generalizes competitive, noncompetitive and uncompetitive. Each of those three models are special cases of the mixed-model.

- Comparing R^2 is an easy way to compare fits, but not a very good method.

- Comparing the adjusted R 418 ^{2} 418 is a better way to compare fits. To do this, you need to ask for reporting of adjusted R^2 in the parameters dialog. But this is not a rigorous way to compare models.

- Prism can compare models using two rigorous approaches, the extra sum-of-squares F test 82 and the AICc approach 84.

- The extra sum-of-squares F test is only valid when one model is a special case of (is nested within) the other model. The mixed model is a general equation that includes competitive 622, uncompetitive 627 and noncompetitive 624 inhibition as special cases, so it is OK to compare the mixed model with any of the other three by F test. The competitive model is a special case of the uncompetitive or noncompetitive modes, so you can compare the competitive model with either uncompetitive or noncompetitive models by F test. The uncompetitive and noncompetitive models are not nested so cannot be compared by F test.

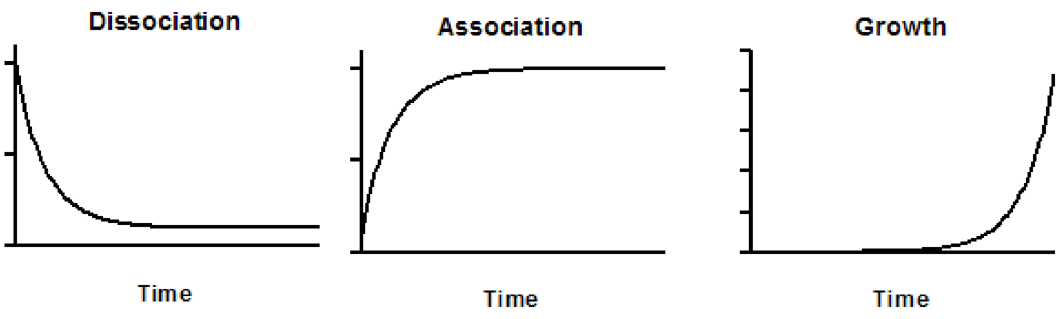
4.5.5.13 Exponential

4.5.5.13.1 Key concepts: Exponential equations

What is exponential?

Processes follow exponential models when the rate at which something is happening depends on the amount that is present.

Exponential dissociation vs. association vs. growth



The dissociation model always heads downhill gradually approaching a plateau.

The association model always heads uphill, and also approaches a plateau.

The growth model goes up and up and up, getting steeper, never reaching a plateau.

Rate constants vs. time constants vs. half-lives

In all the exponential models, one (or more in some cases) parameter describes how rapidly the process occurs. It can be expressed as a rate constant (in units of inverse time) or as a time constant (in units of time), or as a half-life, also in units of time.

The rate constant and time constants are simply reciprocals of each other. Prism always fits the rate constant (k), but computes the time constant (tau) as well and reports the standard error and confidence interval of the time constant just as if the model had been written to fit that constant.

The half-life equals ln(2)/k where ln is the abbreviation for natural logarithm.

4.5.5.13.2 Key concepts: Derivation of exponential decay

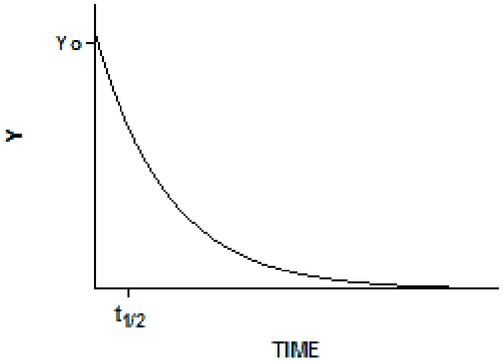
Define X to be time, and Y to be the outcome you are measuring. Three examples:

- Number of ligand-receptor complexes still present -- the ligand has not yet dissociated.
- The number of radioactive atoms that have not yet decayed.
- The concentration of drug in the plasma, not yet metabolized.

At any given time X. The rate of change of Y is proportional to Y. Expressed as a differential equation:

$$\frac{\Delta Y}{\Delta X} = \frac{dY}{dX} = -k \cdot Y$$

Shown as a graph:



When you integrate both sides of the equation, you get the equation for exponential decay:

$Y=Y_0*exp(-k*X)$

The function exp() takes the constant e (2.718...) to the power contained inside the parentheses.

4.5.5.13.3 Equation: One phase decay

Introduction

An exponential decay equation models many chemical and biological processes. It is used whenever the rate at which something happens is proportional to the amount which is left. Here are three examples:

- When ligands dissociate from receptors, the number of molecules that dissociate in any short time interval is proportional to the number that were bound at the beginning of that interval. Equivalently, each individual molecule of ligand bound to a receptor has a certain probability of dissociating from the receptor in any small time interval. That probability does not get higher as the ligand stays on the receptor longer.

- When radioactive isotopes decay, the number of atoms that decay in any short interval is proportional to the number of undecayed atoms that were present at the beginning of the interval. This means that each individual atom has a certain probability of decaying in a small time interval, and that probability is constant. The probability that any particular atom will decay does not change over time. The total decay of the sample decreases with time because there are fewer and fewer undecayed atoms.

- When drugs are metabolized by the liver or excreted by the kidney, the rate of metabolism or excretion is often proportional to the concentration of drug in the blood plasma. Each drug molecule has a certain probability of being metabolized or secreted in a small time interval. As the drug concentration goes down, the rate of its metabolism or excretion goes down as well.

Entering data

Create an XY data table. Enter time into X, and response (binding, concentration ..) into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

You can also choose a sample data set for exponential decay.

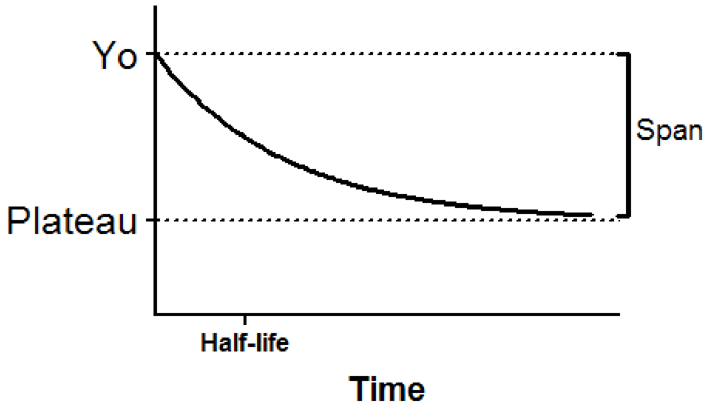
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *One phase decay*.

Consider constraining Plateau to a constant value of zero

If you have subtracted off any background signal, then you know the curve has to plateau at Y=0. In this case, you should constrain the parameter Plateau to be a constant value equal to zero. To do this, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to Plateau to "Constant equal to" and enter the value 0.0.

Model

```
Y=(Y0 - Plateau)*exp(-K*X) + Plateau
```



Y0 is the Y value when X (time) is zero. It is expressed in the same units as Y,

Plateau is the Y value at infinite times, expressed in the same units as Y.

K is the rate constant, expressed in reciprocal of the X axis time units. If X is in minutes, then K is expressed in inverse minutes.

Tau is the time constant, expressed in the same units as the X axis. It is computed as the reciprocal of K.

Half-life is in the time units of the X axis. It is computed as ln(2)/K.

Span is the difference between Y0 and Plateau, expressed in the same units as your Y values.

The derivative of an exponential decay equals -K*Y. So the initial rate equals -K*Y0.

4.5.5.13.4 Equation: Two phase decay

Introduction

An exponential decay equation models many chemical and biological processes. It is used whenever the rate at which something happens is proportional to the amount which is left.

A two-phase model is used when the outcome you measure is the result of the sum of a fast and slow exponential decay. This is also called a double exponential decay.

Entering data

Create an XY data table. Enter time into X, and response (binding, concentration ..) into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Two phase decay*.

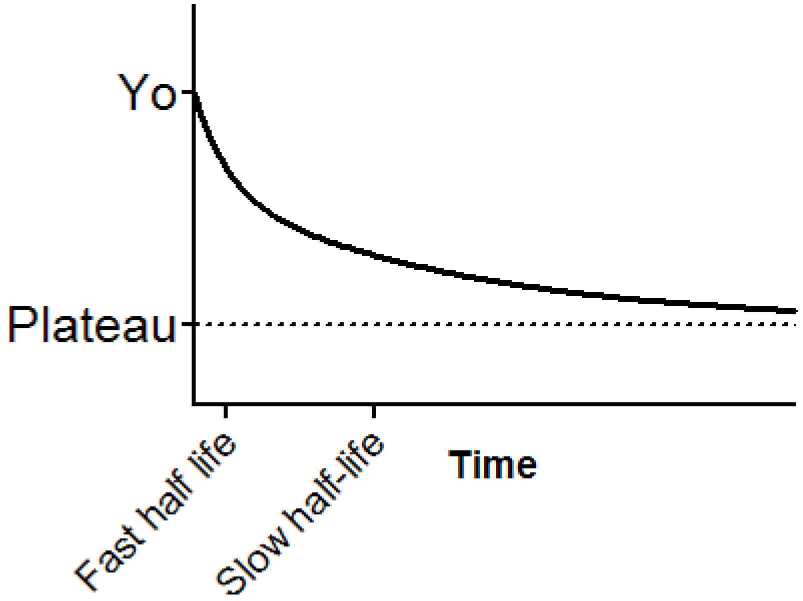
Consider constraining Plateau to a constant value of zero

If you have subtracted off any background signal, then you know the curve has to plateau at Y=0. In this case, you should constrain the parameter Plateau to be a constant value equal to zero. To do this, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to Plateau to "Constant equal to" and enter the value 0.0.

In some situations, you also know that the plateau has to be zero, even without subtracting a background. For example, in pharmacokinetics you know the drug concentration (usually) has to descend to zero at long time points.

Model

```
SpanFast=(Y0-Plateau)*PercentFast*.01
SpanSlow=(Y0-Plateau)*(100-PercentFast)*.01
Y=Plateau + SpanFast*exp(-KFast*X) + SpanSlow*exp(-KSlow*X)
```



Y0 is the Y value when X (time) is zero. It is expressed in the same units as Y,

Plateau is the Y value at infinite times, expressed in the same units as Y.

Kfast and **Kslow** are the two rate constants, expressed in reciprocal of the X axis time units. If X is in minutes, then K is expressed in inverse minutes.

TauFast and **TauSlow** are the two time constants, expressed in the same units as the X axis. They are computed as the reciprocals of the rate constants.

Half-life (fast) and **Half-life (slow)** are in the time units of the X axis. They are computed as ln(2)/K.

PercentFast is the fraction of the span (from Y0 to Plateau) accounted for by the faster of the two components.

Interpretation of the parameters when you constrain Plateau to a constant value of zero

If you have subtracted off any background signal, then you know the curve has to plateau at Y=0. You also know the curve has to descend to Y=0 in some other situations such as pharmacokinetics. In these cases, you should constrain the parameter Plateau to be a constant equal to zero.

With this constraint, it is easier to understand the model. It is the sum of two components, each defined by a rate constant and a starting Y value. The two rate constants are Kfast and Kslow. The starting Y values are defined by these equations:

```
SpanFast = Y0*PercentFast*.01
SpanSlow = Y0*(100-PercentFast)*.01
```

The names SpanFast and SpanSlow are probably better thought of as Y0Fast and Y0Slow.

Note the two phases are both happening at all time points. It is not correct to think that first the fast phase finishes, and only then does the slow phase start.

4.5.5.13.5 Equation: Plateau followed by one phase decay

Introduction

In the standard one-phase decay 637 equation, the decay starts at time 0. This equation is used when you measure a baseline for a while, then do some experimental intervention that starts the decay at some time X0.

Entering data

Create an XY data table. Enter time into X, and response (binding, concentration ..) into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Plateau followed by one phase decay.*

Consider constraining X0 and Plateau to a constant values

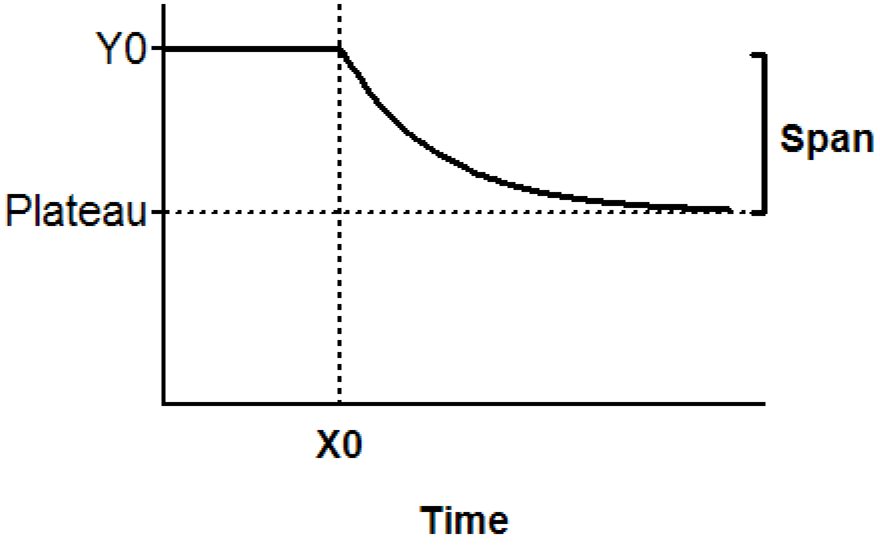
If you know the time at which you initiated the decay, you should constrain X0 to that value.

If you have subtracted off any background signal, then you know the curve has to plateau at Y=0. In this case, you should constrain the parameter Plateau to be a constant value equal to zero.

To constrain parameters to constant values, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to the parameter name to "Constant equal to" and enter the value.

Model

```
Y= IF( X<X0, Y0, Plateau+(Y0-Plateau)*exp(-K*(X-X0)))
```



X0 is the time at which the decay begins. Often you will set that to a constant value based on your experimental design, but otherwise Prism can fit it. It is expressed in the same time units as X.

Y0 is the average Y value up to time X0. It is expressed in the same units as Y,

Plateau is the Y value at infinite times, expressed in the same units as Y.

K is the rate constant, expressed in reciprocal of the X axis time units. If X is in minutes, then K is expressed in inverse minutes.

Tau is the time constant, expressed in the same units as the X axis. It is computed as the reciprocal of K.

Half-life is in the time units of the X axis. It is computed as ln(2)/K.

Span is the difference between Y0 and Plateau, expressed in the same units as your Y values.

4.5.5.13.6 Equation: Three phase decay

Introduction

An exponential decay equation models many chemical and biological processes. It is used whenever the rate at which something happens is proportional to the amount which is left.

A three-phase model is used when the outcome you measure is the result of the sum of a fast, medium and slow exponential decay. You need lots of data with little scatter to adequately fit a three phase model.

Entering data

Create an XY data table. Enter time into X, and response (binding, concentration ..) into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Three phase decay*.

Consider constraining Plateau to a constant value of zero

If you have subtracted off any background signal, then you know the curve has to plateau at Y=0. In this case, you should constrain the parameter Plateau to be a constant value equal to zero. To do this, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to Plateau to "Constant equal to" and enter the value 0.0.

Model

```
YFast=(Y0-Plateau)*PercentFast*.01*exp(-KFast*X)
YSlow=(Y0-Plateau)*PercentSlow*.01*exp(-KSlow*X)
YMedium=(Y0-Plateau)*(100-PercentFast - PercentSlow)*.01*exp(-Kmedium*X)
Y=Plateau + YFast + YMedium +YSlow
```

Y0 is the Y value when X (time) is zero. It is expressed in the same units as Y,

Plateau is the Y value at infinite times, expressed in the same units as Y.

Kfast, Kmedium and **Kslow** are the rate constants, expressed in reciprocal of the X axis time units. If X is in minutes, the rate constants are expressed in inverse minutes.

Half-life (fast, medium and slow) are in the time units of the X axis. they are computed as ln(2) divided by the corresponding rate constant.

PercentFast is the percentage of the span (from Y0 to Plateau) accounted for by the fastest of the three components.

PercentSlow is the percentage of the span (from Y0 to Plateau) accounted for by the slowest of the three components.

4.5.5.13.7 Equation: One phase association

Introduction

This equation describes the pseudo-first order association kinetics of the interaction between a ligand and its receptor, or a substrate and an enzyme. During each time interval a certain fraction of the unoccupied receptors become occupied. But as time advances, fewer receptors are unoccupied so fewer ligand bind and the curve levels off.

Entering data

Create an XY data table. Enter time into X, and response (binding, concentration ..) into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

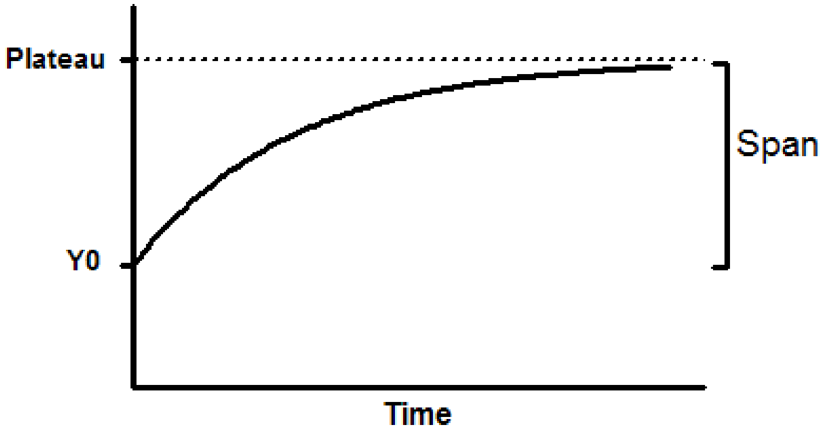
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *One phase association*.

Consider constraining Y0 to a constant value of zero

If you have subtracted off any background signal, then you know that the curve should start at Y=0. In this case, you should constrain the parameter Y0 to be a constant value equal to zero. To do this, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to Y0 to "Constant equal to" and enter the value 0.0. Doing so will force the resulting curve to pass through the origin.

Model

```
Y=Y0 + (Plateau-Y0)*(1-exp(-K*x))
```



Y0 is the Y value when X (time) is zero. It is expressed in the same units as Y,

Plateau is the Y value at infinite times, expressed in the same units as Y.

K is the rate constant, expressed in reciprocal of the X axis time units. If X is in minutes, then K is expressed in inverse minutes.

Tau is the time constant, expressed in the same units as the X axis. It is computed as the reciprocal of K.

Half-time is in the time units of the X axis. It is computed as ln(2)/K.

Span is the difference between Y0 and Plateau, expressed in the same units as your Y values.

4.5.5.13.8 Equation: Plateau followed by one phase association

Introduction

In the standard one-phase association 645 equation, the increase starts at time 0. This alternative equation is used when you measure a baseline for a while, then do some experimental intervention that starts the association at some time X0.

Entering data

Create an XY data table. Enter time into X, and response (binding, concentration ..) into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

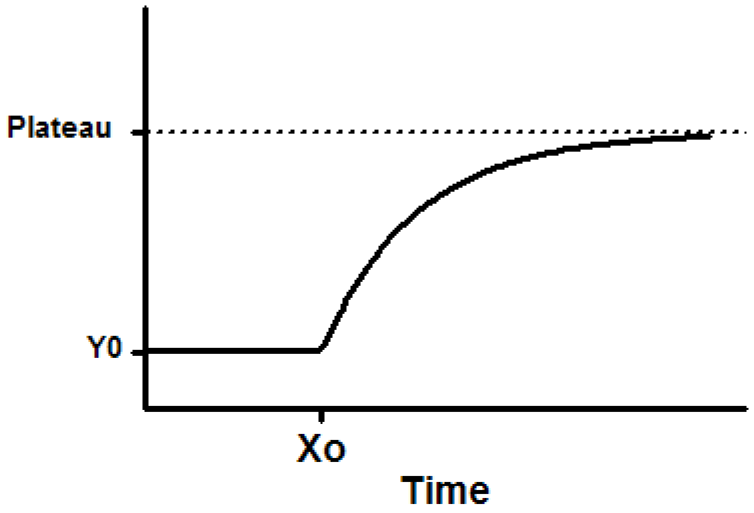
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Plateau followed by one phase association.*

Consider constraining X0 to a constant values

If you know the time at which you initiated the association, you should constrain X0 to that value. To do this, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to X0 to "Constant equal to" and enter the value.

Model

```
Y= IF( X<X0, Y0,Y0 + (Plateau-Y0)*(1 - exp(-K*(X-X0))))
```



X0 is the time at which the association begins. Often you will set that to a constant value based on your experimental design, but otherwise Prism can fit it. It is expressed in the same time units as X.

Y0 is the average Y value up to time X0. It is expressed in the same units as Y,

Plateau is the Y value at infinite times, expressed in the same units as Y.

K is the rate constant, expressed in reciprocal of the X axis time units. If X is in minutes, then K is expressed in inverse minutes.

Tau is the time constant, expressed in the same units as the X axis. It is computed as the reciprocal of K.

Half-life is in the time units of the X axis. It is computed as ln(2)/K.

Span is the difference between Y0 and Plateau, expressed in the same units as your Y values.

4.5.5.13.9 Equation: Two phase association

Introduction

An exponential decay equation models many chemical and biological processes. It is used whenever the rate at which something happens is proportional to the amount which is left.

A two-phase model is used when the outcome you measure is the result of the sum of a fast and slow exponential decay.

Entering data

Create an XY data table. Enter time into X, and response (binding, concentration ..) into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

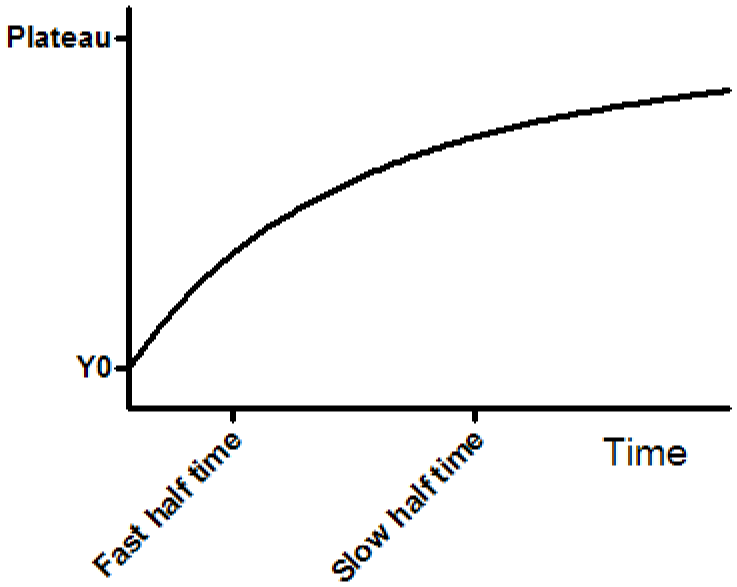
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Two phase association*.

Consider constraining Plateau to a constant value of zero

If you have subtracted off any background signal, then you know the curve has to plateau at Y=0. In this case, you should constrain the parameter Plateau to be a constant value equal to zero. To do this, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to Plateau to "Constant equal to" and enter the value 0.0.

Model

```
SpanFast=(Plateau-Y0)*PercentFast*.01
SpanSlow=(Plateau-Y0)*(100-PercentFast)*.01
Y=Y0+ SpanFast*(1-exp(-KFast*X)) + SpanSlow*(1-exp(-KSlow*X))
```



Y0 is the Y value when X (time) is zero. It is expressed in the same units as Y,

Plateau is the Y value at infinite times, expressed in the same units as Y.

Kfast and **Kslow** are the two rate constant, expressed in reciprocal of the X axis time units. If X is in minutes, then K is expressed in inverse minutes.

TauFast and **TauSlow** are the two time constants, expressed in the same units as the X axis. They are computed as the reciprocals of the rate constants.

Half-time (fast) and **Half-time (slow)** are in the time units of the X axis. They are computed as ln(2)/K.

PercentFast is the fraction of the span (from Y0 to Plateau) accounted for by the faster of the two components.

4.5.5.13.10 Equation: Exponential growth

Introduction

This equation describes the growth with a constant doubling time.

Entering data

Create an XY data table. Enter time into X, and response (cell number ..) into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

Note that Y values must be the actual values. If you transform all the values to logarithms, then it rarely would make sense to fit this equation.

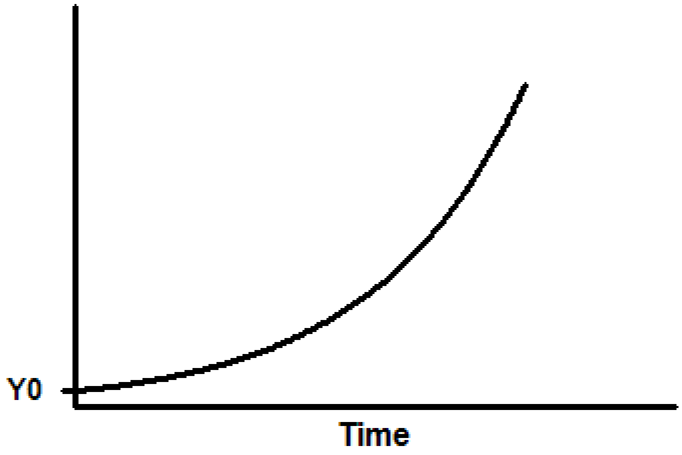
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Exponential growth*.

Consider constraining Y0 to a constant value

The parameter Y0 is the Y value at time zero. In many cases, you will know this value precisely. If so, you should constrain that parameter to be a constant value. To do this, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to Y0 to "Constant equal to" and enter its value.

Model

```
Y=Y0*exp(k*X)
```



Y0 is the Y value when X (time) is zero. It is expressed in the same units as Y,

K is the rate constant, expressed in reciprocal of the X axis time units. If X is in minutes, then K is expressed in inverse minutes.

Tau is the time constant, expressed in the same units as the X axis. It is computed as the reciprocal of K.

Doubling-time is in the time units of the X axis. It is computed as ln(2)/K.

Consider fitting a line (linear regression) to transformed data

When you fit any model with nonlinear regression, you assume that the variation of residuals is Gaussian with the same SD all the way along the curve. With growth data, often the variation goes up as Y goes up. One way to deal with this is by weighting the data. Another approach would be to transform all the Y values to ln(Y) and fit linear regression to the results. The slope in linear regression will have the same meaning as K in the equation above.

4.5.5.14 Lines

4.5.5.14.1 Key concepts: Fitting lines

Choosing nonlinear regression, rather than linear regression to fit a line

Prism offers separate analyses for linear regression and nonlinear regression. But the nonlinear regression analysis can fit a straight-line model. This is useful when you want to take advantage of features in Prism's nonlinear regression analysis that Prism does not offer in its linear regression analysis, such as the ability to compare two models 334, apply weighting 336, or automatically exclude outliers 94. See a longer discussion of the advantages of using the nonlinear regression analysis to fit a straight line 123.

Fitting straight lines on graphs with nonlinear axes

The nonlinear regression analysis fits the data, not the graph. Since Prism lets you choose logarithmic or probability axes, some graphs with data points that form a straight line follow nonlinear relationships. Prism's collection of "Lines" equations includes those that let you fit nonlinear models to graphs that appear linear when the X axis is logarithmic, the Y axis is logarithmic, both axes are logarithmic, or when the Y axis uses a probability scale. In these cases, linear regression will fit a straight line to the data but the graph will appear curved since an axis (or both axes) are not linear. In contrast, nonlinear regression to an appropriate nonlinear model will create a curve that appears straight on these axes.

Segmental linear regression

Segmental regression 656 fits one line to all data points with X less than some value X0, and another line to all points with X greater than X0, ensuring that the two lines intersect at X0.

Segmental linear regression is helpful when X is time, and you did something at time=X0 to change the slope of the line. Perhaps you injected a drug, or rapidly changed the temperature. In these cases, your model really does have two slopes with a sharp transition point.

4.5.5.14.2 Equation: Fitting a straight line with nonlinear regression

Introduction

Linear regression fits a straight line through your data. Nonlinear regression fits any model, which includes a straight line model. Prism offers separate analyses for linear regression and nonlinear regression, so you can choose either one to fit a line.

Prism's nonlinear regression analysis offers more options than its linear regression analysis 22, such as the ability to compare two models 334, apply weighting 336, automatically exclude outliers 94 and perform normality tests 375 on the residuals. See a longer discussion of the advantages of using the nonlinear regression analysis to fit a straight line 123.

Step by step

Create an XY data table. There is one X column, and many Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel equations for lines, and choose *Straight line*.

Model

Y= YIntercept + Slope*X

Interpret the parameters

YIntercept is the Y value where the line intersects the Y axis.

Slope is the slope of the line, expressed in Y units divided by X units.

Special forms of the linear regression equation

Horizontal line

If you constrain the slope to be zero, the line will be horizontal. The only parameter is the Y intercept. Prism has this model built in as "horizontal line". The best-fit value of the Y intercept is the mean of all the Y values. The model is:

```
Y = Mean + 0*X
```

Prism requires that all equations include X. Here X is multiplied by zero, so it is present (as required) but has no effect.

Line through origin

If you constrain the Y intercept to be zero, the line has to go through the origin (X=0, Y=0). Prism has this "Line through origin 655" model built in:

```
Y=Slope*X
```

The only parameter is the slope.

4.5.5.14.3 Equation: Line through origin

Introduction

Prism's linear regression analysis 117 fits a straight line through your data, and lets you force the line to go through the origin. This is useful when you are sure that the line must begin at the origin (X=0 and Y=0).

Prism's nonlinear regression offers the equation Line through origin. It offers more options than its linear regression analysis 22, such as the ability to compare two models 334, apply weighting 336, automatically exclude outliers 94 and perform normality tests 375 on the residuals. See a longer discussion of the advantages of using the nonlinear regression analysis to fit a straight line 123.

When to force the line through the origin

In many scientific situations, it just makes sense that when X=0, Y must also equal 0, so the line should be forced to go through the origin (X=0, Y=0). But even in these situations, it can make sense to fit an ordinary linear regression line that also fits the intercept. The data you are analyzing may be far from the origin, and you may get a better fit through the points (what you care about) when you don't force the line to go through the origin. This can happen when the true model is curved (beginning at the origin), so a line through the data points that is not forced to the origin may fit a whole lot better than line forced to go through the origin.

Prism makes it easy to compare 334 the fit that goes through the origin with one that doesn't.

Step by step

Create an XY data table. There is one X column, and many Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel equations for lines, and choose *Line Through Origin*

Model

Y= Slope*X

Slope is the slope of the line, expressed in Y units divided by X units. It estimates the ratio of Y/X in the entire population.

Weighting

In situations where linear regression through the origin is appropriate, it is common for the variation among replicate Y values increases as X (and Y) increase. Prism provides two weighting choices (in the Weights tab) that are used in this situation. Weight by $1/X^2$ when you think the variance in Y is proportional to the square of X, which means the SD among Y values is proportional to X.Weight by 1/X if you think the variance in Y is proportional to X.

R^2 when fitting a line through the origin

When you constrain a line to go through a point, there are two possible ways to compute R^2:

- Compare the fit of the best-fit line with the fit of a horizontal line at the mean Y value. But that null hypothesis (horizontal line through the Ymean) doesn't obey the constraint that it go through the origin.

- Compare the best-fit line with a horizontal line at Y=0. This obeys the constraint, but often fits the data really badly, pushing up the R^2 value.

When you use nonlinear regression to fit a line through the origin, Prism uses the first definition above. If you use linear regression, and ask Prism to constrain the line to go through the origin, it simply will not report R^2 at all.

Reference

1. J. G. Eisenhauer, Regression through the Origin. *Teaching Statistics* **25**, 76–80 (2003).

4.5.5.14.4 Equation: Segmental linear regression

Introduction

Segmental regression fits one line to all data points with X less than some value X0, and another line to all points with X greater than X0, while ensuring that the two lines intersect at X0.

Segmental linear regression is helpful when X is time, and you did something at time=X0 to change the slope of the line. Perhaps you injected a drug, or rapidly changed the temperature. In these cases, your model really does have two slopes with a sharp transition point.

In other cases, the true model has the slope gradually changing. The data fit a curve, not two straight lines. In this situation, fitting the data with segmental linear regression is not helpful.

Caution

Don't use segmental linear regression to analyze a biphasic Scatchard or Lineweaver-Burk plot. A biphasic Scatchard plot follows a curve, not two intersecting lines. There is no abrupt break point. You should fit the original data to a two-site binding curve instead.

Step by step

Create an XY data table. Enter time into X, and your measurements into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel equations for lines, and choose *Segmental linear regression*.

Model

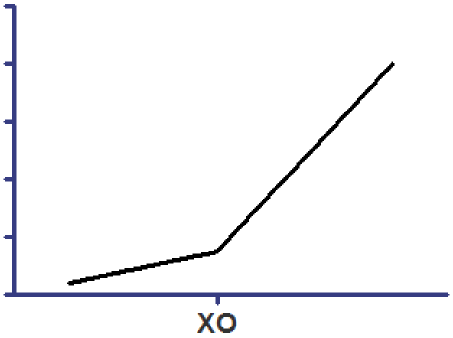
```
Y1 = intercept1 + slope1*X
YatX0 = slope1*X0 + intercept1
Y2 = YatX0 + slope2*(X – X0)
Y = IF(X<X0, Y1, Y2)
```

The first line of the equation defines the first line segment from its intercept and slope.

The second line of the equation computes the Y value of the first regression at the right end of that segment, when X=X0.

The third line of the equation computes the second regression segment. Since we want a continuous line, the Y value at the left end of the second segment must equal the Y value at the right end of the first segment (YatX0). The Y value at any other position along the second segment equals YatX0 plus the increase due to the second regression line. That increase equals the slope of the second segment (slope2) times the distance from X0 to X.

The final line defines Y for all values of X. If X is less than X0 then Y is set equal to Y1. Otherwise Y is set equal to Y2.



Interpret the parameters

Intercept1 is the Y value where the first line segment intersects the Y axis.

Slope1 is the slope of the first line segment, expressed in Y units divided by X units.

Slope2 is the slope of the second line segment, expressed in Y units divided by X units.

X0 is the X value where the two line segments intersect. Often you will want to constrain this to a constant value equal to the time you applied an experimental intervention.

Extending to three segments

Prism does not include an equation for segmental regression with three segments, but you could enter this equation as an user-defined equation:

```
Y1 = intercept1 + slope1*X
  YatX0 = intercept1 + slope1*X0
  Y2 = YatX0 + slope2*(X - X0)
  YatX1 = YatX0 + slope2*(X1-X0)
```

```
Y3 = YatX1 + slope3*(X - X1)
Y = IF(X<X0, Y1, IF(X<X1, Y2, Y3))
```

.

4.5.5.14.5 Equation: Hinge function. Segmental regression lines with gentle connection

Introduction

This hinge function fits one line to all data points with X less than some value X0, and another line to all points with X greater than X0, while ensuring that the two lines intersect at X0. The difference between this and segmental linear regression 656 is that there is a gentle curve connecting the two lines. This hinge function was derived by Andrew Gelman.

Caution

Don't use hinge function to analyze a biphasic Scatchard or Lineweaver-Burk plot. A biphasic Scatchard plot follows a curve, not two intersecting lines. There is no abrupt break point. You should fit the original data to a two-site binding curve instead.

Step by step

Create an XY data table. Enter time into X, and your measurements into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel equations for lines, and choose *Hinge function*.

Consider if you want to constrain X0 (usually you will) or Delta to constant values.

Model

```
Y= Intercept + Slope1*(X – X0) + (Slope2 – Slope1)*Delta*ln(1+exp((X-X0)/De
```

Interpret the parameters

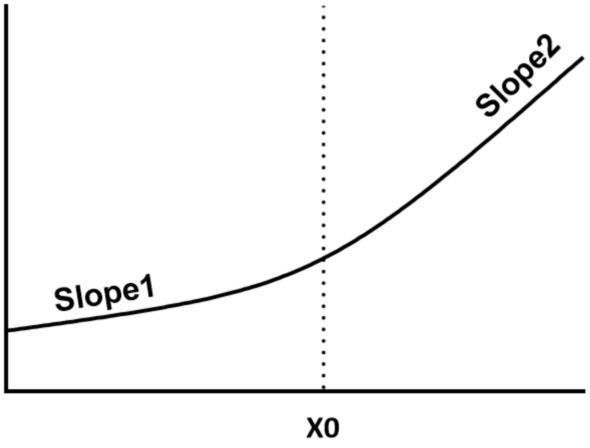
Intercept is the Y value X=0

Slope1 is the slope of the first line segment, expressed in Y units divided by X units.

Slope2 is the slope of the second line segment, expressed in Y units divided by X units.

X0 is the X value where the two line segments would intersect, if there were no curve connecting them. Often you will want to constrain this to a constant value equal to the time you applied an experimental intervention.

Delta defines how sharp the bend is. When delta is close to zero, the result is is very close to two intersecting lines (segmental regression). As Delta is made larger, the curve joining the two lines is more spread out. Delta must always be positive. You may want to set this to a constant and then try various values. Unless there are lots of data near the inflection, it may not work to ask Prism to fit it.



4.5.5.14.6 Equation: Fitting a straight line on a semi-log or log-log graph

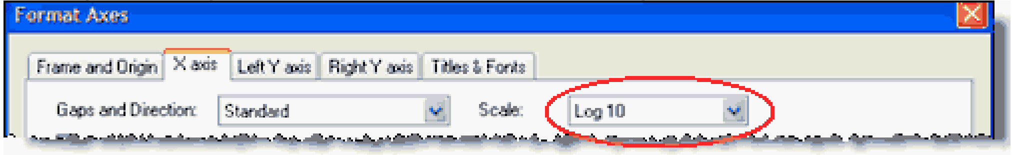
Straight lines on graphs with logarithmic axes

The nonlinear regression analysis fits the data, not the graph. Since Prism lets you choose logarithmic axes, some graphs with data points that form a straight line follow nonlinear relationships. Prism's collection of "Lines" equations includes those that let you fit nonlinear models to graphs that appear linear when the X axis is logarithmic, the Y axis is logarithmic, or both axes are logarithmic. In these cases, linear regression will fit a straight line to the data but the graph will appear curved since an axis (or both axes) are not linear. In contrast, nonlinear regression to an appropriate nonlinear model will create a curve that appears straight on these axes.

Entering and fitting data

1. Create an XY table, and enter your X and Y values.

2. Go to the graph, double click on an axis to bring up the Format Axis dialog. Change one or both axes to a logarithmic scale.



Gaps and Direction: Standard

Scale: Log 10

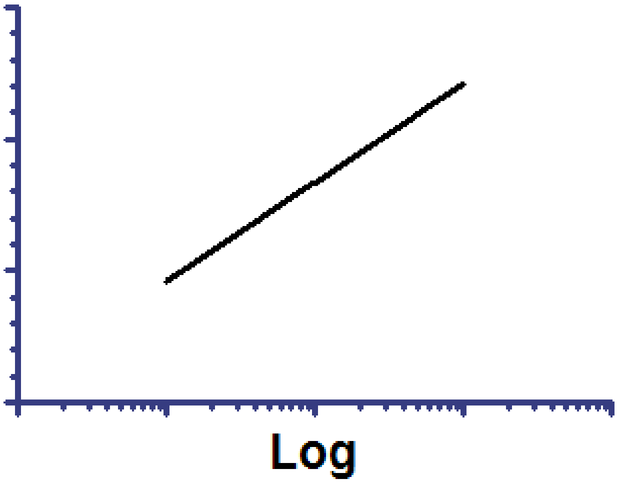
3. Click Analyze, choose Nonlinear regression (not Linear regression) and then choose one of the semi-log or log-log equations from the "Lines" section of equations.

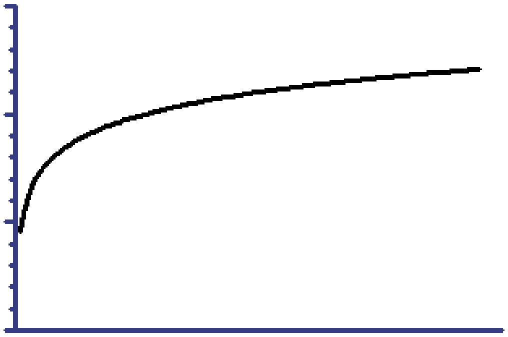
Equations

Semilog line -- X axis is logarithmic, Y axis is linear

Y=Yintercept + Slope*log(X)

On semilog axis **On linear axes**





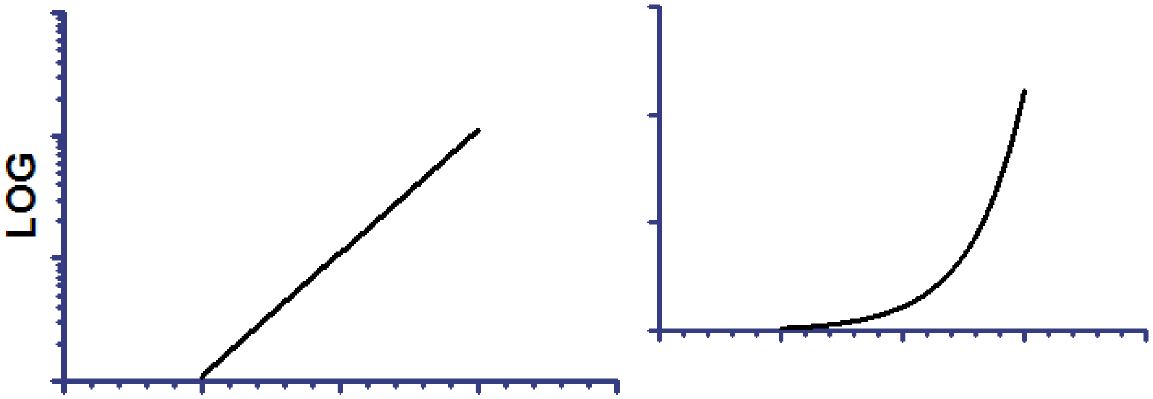
Slope is the change in Y when the log(X) changes by 1.0 (so X changes by a factor of 10).

Yintercept is the Y value when log(X) equals 0.0. So it is the Y value when X equals 1.0.

Semilog line -- X axis is linear, Y axis is logarithmic

Y=10^(Slope*X + Yintercept)

On semilog axis **On linear axes**



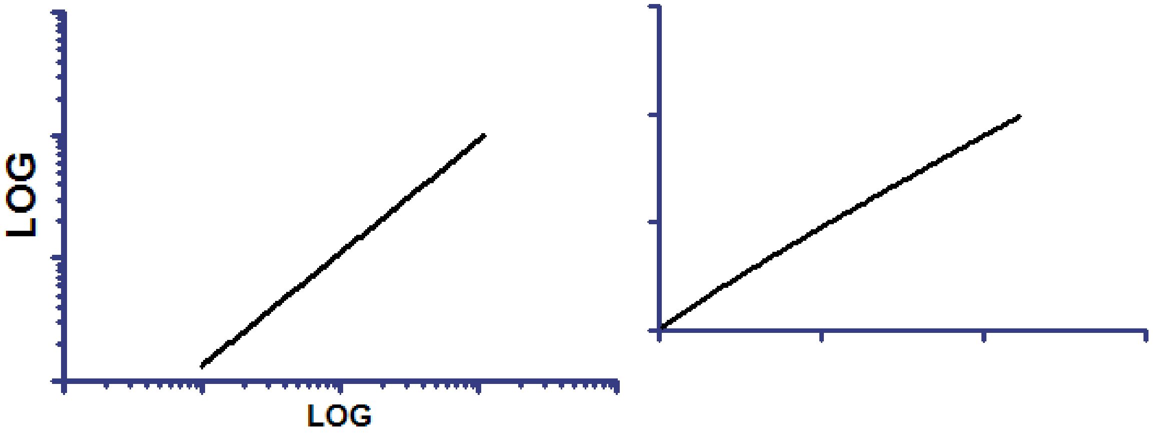
Slope is the change in log(Y) when the X changes by 1.0.

Yintercept is the log(Y) value when X equals 0.0.

Log-log line -- Both X and Y axes are logarithmic

Y = 10^(slope*log(X) + Yintercept)

On log-log axes **On linear axes**



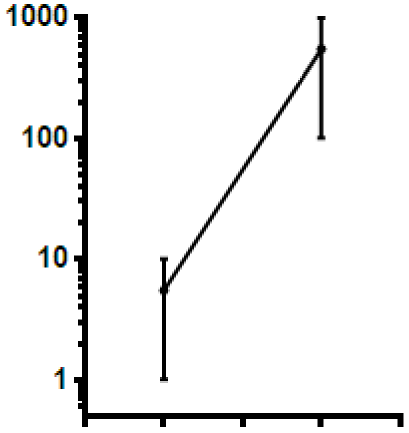
Since both axes are transformed the same way, the graph is linear on both sets of axes. But when you fit the data, the two fits will not be quite identical.

Slope is the change in log(Y) when the log(X) changes by 1.0.

Yintercept is the Y value when log(X) equals 0.0. So it is the Y value when X equals 1.0.

An alternative way to handle these data

The nonlinear regression analysis minimizes the sum of the squares of the difference between the actual Y value and the Y value predicted by the curve. This is not the same as minimizing the sum of squares of the distances (as seen on the graph) between points and curve. In the graph below, the two vertical lines look the same distance but one represents a difference of 9 Y units, and the other a difference of 900.



An alternative approach, that might be better in some circumstances, is to use Prism's transform analysis to transform Y (and maybe also X) to logarithms. Then perform linear regression on the logarithms. The regression results will not be the same as using nonlinear regression on log axes.

4.5.5.14.7 Equation: Fitting a straight line on a graph with a probability axis

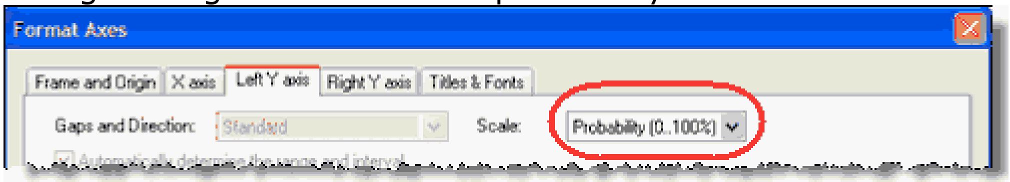
Fitting straight lines on graphs with nonlinear axes

The nonlinear regression analysis fits the data, not the graph. If you choose probability Y axis, graphs with data points that form a straight line follow nonlinear relationships. Prism's collection of "Lines" equations includes those that let you fit nonlinear models to graphs that appear linear when the Y axis is a probability axis. In these cases, linear regression will fit a straight line to the data but the graph will appear curved since an axis (or both axes) are not linear. In contrast, nonlinear regression to an appropriate nonlinear model will create a curve that appears straight on these axes.

Entering and fitting data

1. Create an XY table, and enter your X and Y values.

2. Go to the graph, double click on an axis to bring up the Format Axis dialog. Change the Y axis to a probability scale.



Gaps and Direction: Standard

Scale: Probability (0..100%)

☑ Automatically determine the range and interval

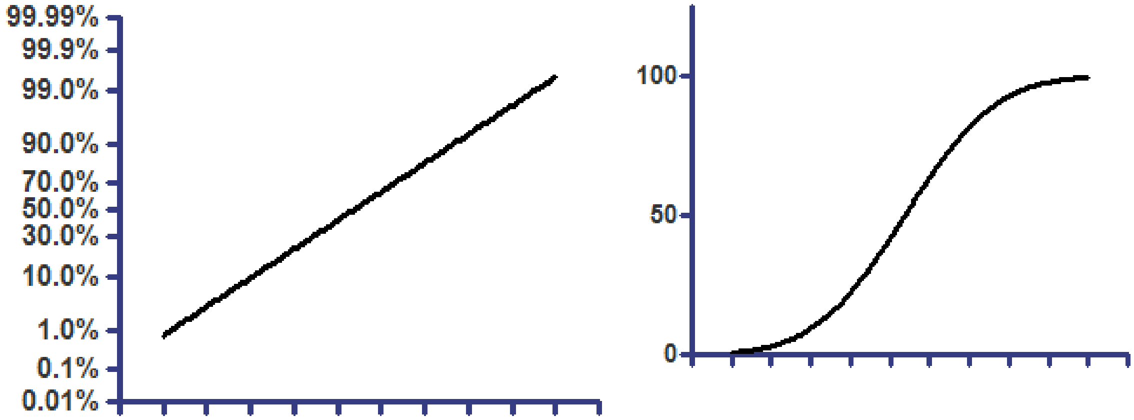
3. Click Analyze, choose Nonlinear regression (not Linear regression) and then choose one of the Cumulative Gaussian distribution equations from the "Lines" section of equations.

Equations

Cumulative Gaussian - Y values are percentages

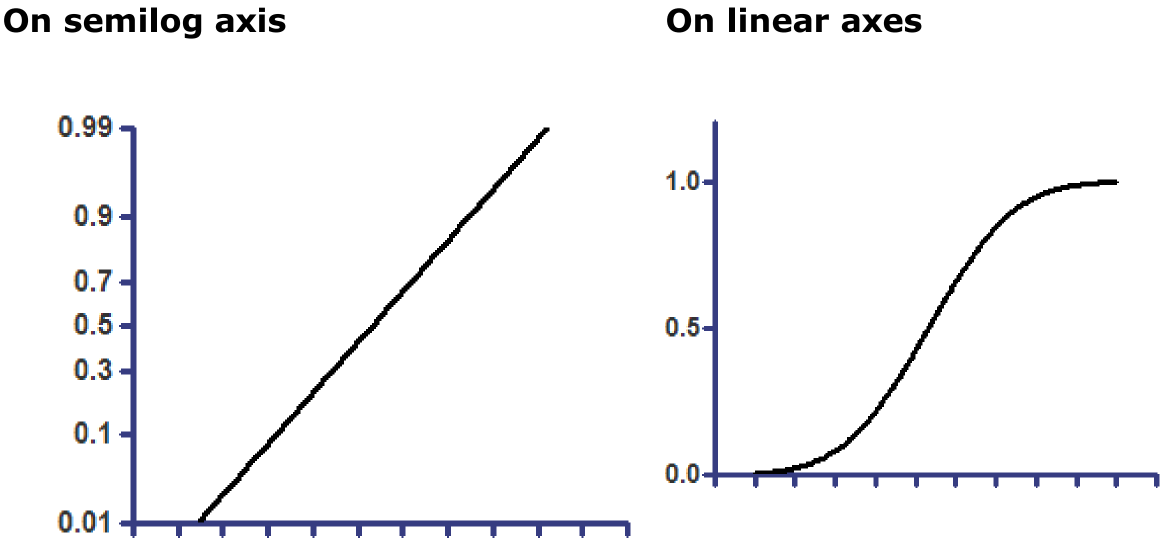
```
Top=100
z=(X-Mean)/SD
Y=Top * zdist(z)
```

On probability axis **On linear axis**



Cumulative Gaussian - Y values are fractions

```
Top=1
z=(X-Mean)/SD
Y=Top * zdist(z)
```


On semilog axis

On linear axes

Parameters

Mean is the average of the original distribution, from which the frequency distribution was created.

SD is the standard deviation of the original distribution.

Both of these parameters are expressed in the same units as the X values plotted on the graph, which is the same as the Y values in the original distribution from which the frequency distribution was generated.

4.5.5.14.8 Equation: Finding the crossing point of two lines

Step by step

Create an XY data table. There is one X column, and two Y columns. If the X values don't match for the two data sets, just leave some Y values blank, staggering the entry down the page.

Click Analyze, choose nonlinear regression, choose the panel equations for lines, and choose *Two intersection lines --fit the crossing point*.

Model

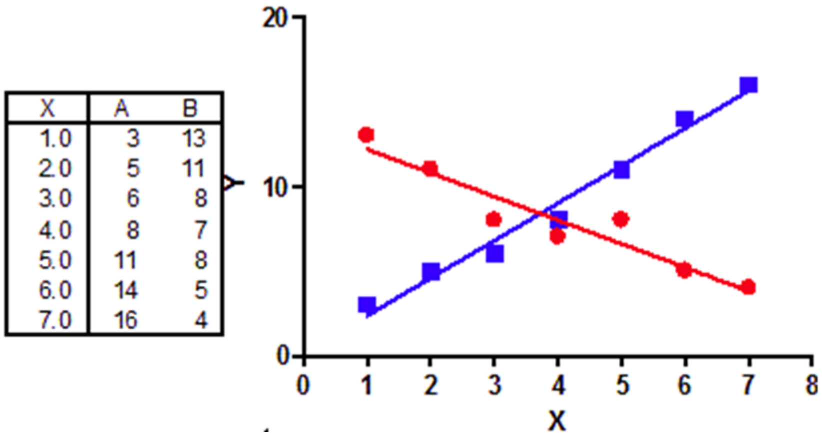
Y= Ycross + (X - Xcross)*Slope

Interpret the parameters

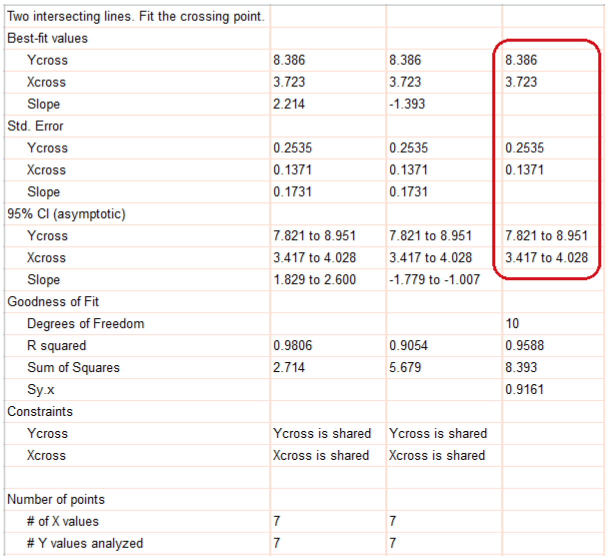
- Slope for the first data set
- Slope for the second data set
- Xcross. The X coordinate of the crossing point (shared for both data sets)
- Ycross. The Y coordinate of the crossing point (shared for both data sets).

That makes sense. You have two lines, and it takes two parameters (usually a slope and intercept) to describe each. Here, the equations were rearranged so the program doesn't fit two separate Y intercepts, but rather fits the X and Y values of the crossing point as well as two slopes.

Here is an example, with the key results circled. Prism file.



X	A	B
1.0	3	13
2.0	5	11
3.0	6	8
4.0	8	7
5.0	11	8
6.0	14	5
7.0	16	4



Two intersecting lines. Fit the crossing point.			
Best-fit values			
Ycross	8.386	8.386	8.386
Xcross	3.723	3.723	3.723
Slope	2.214	-1.393	
Std. Error			
Ycross	0.2535	0.2535	0.2535
Xcross	0.1371	0.1371	0.1371
Slope	0.1731	0.1731	
95% CI (asymptotic)			
Ycross	7.821 to 8.951	7.821 to 8.951	7.821 to 8.951
Xcross	3.417 to 4.028	3.417 to 4.028	3.417 to 4.028
Slope	1.829 to 2.600	-1.779 to -1.007	
Goodness of Fit			
Degrees of Freedom			10
R squared	0.9806	0.9054	0.9588
Sum of Squares	2.714	5.679	8.393
Sy.x			0.9161
Constraints			
Ycross	Ycross is shared	Ycross is shared	
Xcross	Xcross is shared	Xcross is shared	
Number of points			
# of X values	7	7	
# Y values analyzed	7	7	

4.5.5.15 Polynomial

4.5.5.15.1 Key concepts: Polynomial

Usefulness of polynomial models

There are two situations where you might want to choose a polynomial model:

- Your scientific model is described by a polynomial equation. This is rare in biology. Few chemical or pharmacological models are described by polynomial equations.

- You don't have a scientific model, but want to fit a curve to interpolate unknown values. With this goal, you often don't care much about the details of the model. Instead, you care only about finding a model that goes near the data points. Polynomial models often work well.

Which polynomial model?

The order of a polynomial model expresses how many terms it has. Prism offers up to a sixth order equation (and it would be easier to enter higher order equations). The higher order equations have more inflection points.

Choosing the best polynomial model is often a matter of trial and error. If the curve doesn't follow the trend of your data, pick a higher order equation. If it wiggles too much, pick a lower order equation.

How are polynomial models special?

To a mathematician, polynomial models are very special. Strictly speaking, polynomial models are not 'nonlinear'. Even though a graph of X vs. Y is curved (in all but some special cases), the derivative of Y with respect to the parameters is linear.

Because polynomial models are not nonlinear, it is possible (but not with Prism) to fit polynomial models without fussing with initial values. And the fit can be in one step, rather than the iterative approach used for nonlinear models.

Since Prism treats polynomial models the same way it treats nonlinear models, it does require initial values (it chooses 1.0 for each parameter automatically). It doesn't matter what values are used -- polynomial regression cannot encounter false minima 403.

Why you should choose a centered polynomial equation

There are two problems with polynomial fits, often solved by centering:

- When the X values are large, and start well above zero (for example, when X is a calendar year), taking the very large X values to large powers can lead to math errors.
- Even when the X values are not large, the parameters of the model are intertwined, so have high covariance and dependency. This results in large standard errors, wide confidence intervals, and huge confidence or prediction bands.

Both these problems are solved by using a centered polynomial model. The idea of centering is simple. Subtract the mean of all X values from each X value, and use those differences instead of X in the model.

Fitting the centered model leads to exactly the same curve (unless the regular approach led to math errors). Accordingly, the sum-of-squares are R^2 are the same, as are results of model comparisons.However, the parameters have different meanings, so have different best-fit values (except the first parameter which is the same), different standard errors and confidence intervals, smaller covariances and dependencies, and tighter confidence/prediction bands.

Read more about centered polynomial models. 670

4.5.5.15.2 Centered polynomial equations

Included in Prism, are both a set of ordinary polynomial equations and also a set of centered polynomial equations. For example, when you look in the list of polynomials you'll see both 'Second order polynomial' and 'Centered second order polynomial'. We recommend always choosing one of the centered equations instead of an ordinary polynomial equation. This page explains why.

What's wrong with ordinary polynomial models?

The standard polynomial models look like this:

```
Y= B0 + B1*X +B2*X^2
```

More terms are included with the higher order equations.

There are two problems with polynomial fits:

- When the X values are large, and start well above zero (for example, when X is a calendar year), taking the very large X values to large powers can lead to math overflow. Even if the program doesn't report any math error, the results can be inaccurate. Some coefficients will be positive and some negative, so the value of Y depends on subtracting huge numbers from other huge numbers, leading to imprecise results.

- Even when the X values are not large, the parameters of the model are intertwined, so have high covariance and dependency. This results in large standard errors, wide confidence intervals, and very wide confidence or prediction bands. In many cases, this problem is severe enough that Prism reports that the results are 'ambiguous' and so doesn't report confidence intervals for all the parameters and can't graph confidence bands.

What are centered polynomial models?

Both problems go away when the X values are centered. The idea of centering is to subtract the mean X from all X values before fitting the model. This can be done as part of nonlinear regression, using this model:

```
XC = X - Xmean
Y= B0 + B1*XC +B2*XC^2
```

Here XC is the centered X value, equal to the X value minus Xmean, which is the mean of all X values. In other words, XC is the distance of any X value from the mean of all X values. Xmean is constant, and not a parameter that Prism tries to fit. Of course, you can include more terms in the definition of Y to create higher order polynomial equations.

The advantages of centered models

Fitting the centered model leads to exactly the same curve (unless the regular approach led to math errors). Accordingly, the sum-of-squares is the same, as are results of model comparisons.

However, the centered equation has reparameterized the model. The parameters have different meanings, so have different best-fit values (except the first parameter which is the same), different standard errors and confidence intervals, smaller covariances and dependencies, and tighter confidence/prediction bands.

How centered models are implemented in Prism

You can fit data to a built-in centered polynomial equation without knowing how Prism implements the model. If you are curious, read on.

The built-in set of centered polynomial equations, written as shown above, constrain the parameter XMean to equal the mean of X value by constraining it to equal a "Data set constant (= Mean X)".

If you open a file using centered polynomial regression in an version of Prism prior to 5.02 or 5.0b, that constraint will be lost, and centered polynomial regression won't work.

4.5.5.15.3 Equations: Polynomial models

Step by step

Create an XY data table. There is one X column, and many Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel equations for polynomial equations, and choose one.

The "order" of a polynomial equation tells you how many terms are in the equation. Prism offers first to sixth order polynomial equations (and you could enter higher order equations as user-defined equations if you need them). Higher order models wiggle more than do lower order models. Since the equation rarely corresponds to a scientific model, use trial and error. If it isn't close enough to the data, pick a higher order equation. If it wiggles too much, pick a lower order equation.

Polynomial models

Order	Equation
First	Y=B0 + B1*X (straight line)
Second	Y=B0 + B1*X + B2*X^2 (quadratic equation)
Third	Y=B0 + B1*X + B2*X^2 + B3*X^3
Fourth	Y=B0 + B1*X + B2*X^2 + B3*X^3 + B4*X^4
Fifth	Y=B0 + B1*X + B2*X^2 + B3*X^3 + B4*X^4 + B5*X^5
Sixth	Y=B0 + B1*X + B2*X^2 + B3*X^3 + B4*X^4 + B5*X^5 + B6*X^6

There is no general way to interpret the coefficients B0, B1, etc. In most cases, the goal of fitting a polynomial model is to make a curve that looks good, and the parameters really don't matter.

Centered polynomial models

The Centered polynomial models 670 are identical to the ones listed above, with one exception. Wherever X appears above, replace it with (X - XMean), where XMean is the mean of all X values (for rows that have Y values). Using a centered model can avoid computer math problems (overflows), and we recommend that you use them routinely.

4.5.5.16 Gaussian

4.5.5.16.1 Key concepts: Gaussian

What is a frequency distribution?

All the equations in this section fit frequency distributions. The X values represent Y values in the original data set, and the Y values are the frequency or cumulative frequency.

Prism can create a frequency distribution from column data, using an analysis created for that purpose (called, of course, Frequency Distribution).

What is a cumulative frequency distribution?

A frequency distribution plots the number of observations as a function of value. A cumulative frequency distribution plots the *cumulative* number of observations as a function of value. Each Y value is the number of observations in the original data set that have a value less than or equal to the X value.

The Y values can be expressed as the counted number of observations, as fractions or as percentages.

4.5.5.16.2 Equation: Gaussian distribution

Introduction

Data follow a Gaussian distribution when scatter is caused by the sum of many independent and equally weighted factors.

A frequency distribution (histogram) created from Gaussian data will look like a bell-shaped Gaussian distribution.

Step-by-step

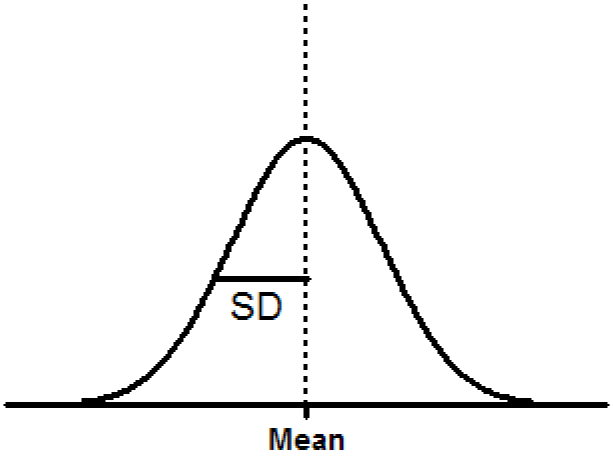
The data you fit must be in the form of a frequency distribution on an XY table. The X values are the bin center and the Y values are the number of observations.

If you start with a column of data, and use Prism to create the frequency distribution, make sure that you set the graph type to "XY graph", with either points or histogram spikes. This ensures that Prism creates an XY results table with the bin centers entered as X values. If you pick a bar graph instead, Prism creates a column results table, creating row labels from the bin centers. This kind of table cannot be fit by nonlinear regression, as it has no X values.

Starting from the frequency distribution table, click Analyze, choose Nonlinear regression from the list of XY analyses, and then choose the "Gaussian" equation from the "Gaussian" family of equations.

Model (Gaussian distribution)

```
Y=Amplitude*exp(-0.5*((X-Mean)/SD)^2)
```



Amplitude is the height of the center of the distribution in Y units.

Mean is the X value at the center of the distribution.

SD is a measure of the width of the distribution, in the same units as X.

The mean and SD will not be identical to the mean and SD computed directly from the raw data. There are two reasons for the discrepancy. The first is that creating the frequency distribution requires a fairly arbitrary decision about bin width, and that will influence the best-fit values of Mean and SD. The second reason is that the nonlinear regression assumes that the residuals (the distances of the points from the curve) follow a Gaussian distribution. This assumption won't be exactly true in a frequency distribution.

Model (sum of two Gaussian distributions)

If your data are a mixture of values sampled from two Gaussian distributions, fit the frequency distribution to a model of the sum of two Gaussian distributions.

```
One=Amplitude1*exp(-0.5*((X-Mean1)/SD1)^2)
Two=Amplitude2*exp(-0.5*((X-Mean2)/SD2)^2)
Y= One + Two
```

Amplitude1 and **Amplitude2** are the heights of the center of the distribution in Y units.

Mean1 and **Mean2** are the X values at the center of the two distributions.

SD1 and **SD2** are measures of the widths of the distributions, in the same units as X.

Prism is not very smart about assigning initial values to the parameters. If you have trouble getting this model to fit, try fussing with the initial parameter values.

Area under a Gaussian distribution

The area under a Gaussian distribution equals Amplitude*SD/0.3989. That constant equals the reciprocal of the square root of two pi.

4.5.5.16.3 Equation: Log Gaussian distribution

Note: Versions of Prism up to 7.00 and 7.0a used a different and nonstandard form of this equation which we called log(Gaussian).

Introduction

Data follow a Gaussian distribution when scatter is caused by the **sum** of many independent and equally weighted factors.

When scatter is caused by the **product** of many independent and equally weighted factors, data follow a lognormal distribution. When plotted on a linear X axis, this is skewed to the right (see below). When plotted on a logarithmic X axis, it looks like a bell-shaped Gaussian distribution.

Step-by-step

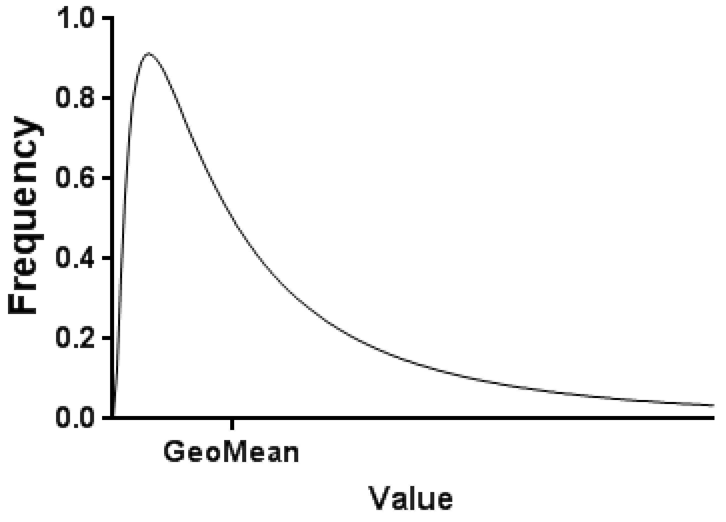
The data must be in the form of a frequency distribution on an XY table. The X values are the bin center and the Y values are the number of observations.

If you start with a column of data, and use Prism to create the frequency distribution, make sure that you set the graph type to "XY graph", with either points or histogram spikes. This ensures that Prism creates an XY results table with the bin centers entered as X values. If you pick a bar graph instead, Prism creates a column results table, creating row labels from the bin centers. This kind of table cannot be fit by nonlinear regression, as it has no X values.

Starting from the frequency distribution table, click Analyze, choose Nonlinear regression from the list of XY analyses, and then choose the "lognormal" equation from the "Gaussian" family of equations.

Model

```
Y=(A/X)*exp(-0.5*(ln(X/GeoMean)/ln(GeoSD))^2)
```



GeoMean is the geometric mean in the units of the data.

GeoSD is the geometric standard deviation factor. It is a unitless ratio.

A is related to the amplitude and area of the distribution.

Amplitude = A / (GeoMean / exp(0.5*ln(GeoSD)^2))

Area = A * sqrt(2 * pi) * ln(GeoSD)

A more standard form of the model (from Wikipedia or MathWorld) is:

```
Y= (1/(X*S*sqrt(2*pi)))*exp(-0.5*(ln(X)-M)^2/(S^2))
```

This standard form of the equation doesn't have the A parameter because the area under the curve, using the equation above, is always 1.0. S in this model equals ln(GeoSD) and M equals ln(GeoMean).

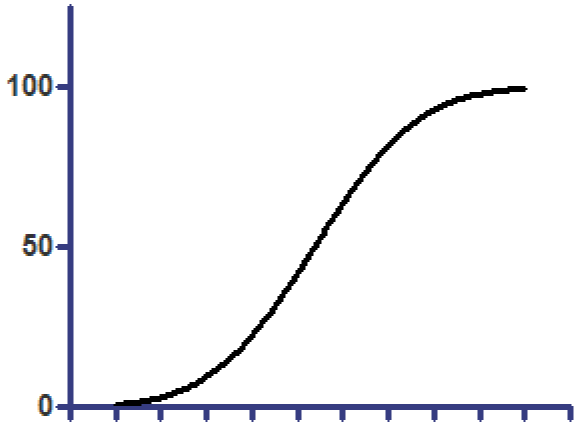
4.5.5.16.4 Equation: Cumulative Gaussian distribution

Introduction

A frequency distribution plots the number of observations as a function of value. A cumulative frequency distribution plots the *cumulative* number of observations as a function of value. Each Y value is the number of observations in the original data set that have a value less than or equal to the X value.

The advantage of creating a cumulative distribution is that you don't have to make any choice regarding bin width.

If your data follow a Gaussian distribution, the cumulative distribution has a sigmoidal shape.



Step-by-step

1. Create an XY table, and enter your X and Y values. The X values correspond to the value in the original data set, and the Y values are the number (or fraction or percent) of values in the original data set that are less than or equal to the Y value.

   Alternatively, enter a stack of values onto a Column data table, and run the frequency distribution analysis choosing to create a cumulative frequency distribution with no bins.

2. From the cumulative frequency distribution, click Analyze, choose Nonlinear regression and then choose one of the Cumulative

Gaussian distribution equations from the "Gaussian" group of equations.

3. If your data are entered as counts (rather than percentages or fractions) constrain N to a constant value equal to the number of observations.

Models

The details of the model depend on whether the Y values are percentages, fractions or counts.

Here is the model if the data are percentages, so the last Y value equals 100.

```
Top=100
z=(X-Mean)/SD
Y=Top * zdist(z)
```

Here is the model if the data are fractions, so the first line of the model defines Top to equal 1.00.

```
Top=1.0
z=(X-Mean)/SD
Y=Top * zdist(z)
```

And finally, here is the model if the data are numbers of observations, so the largest value equals the number of observations (N). In this case, you should constrain N to be a constant value equal to the number of observations.

```
z=(X-Mean)/SD
Y=N * zdist(z)
```

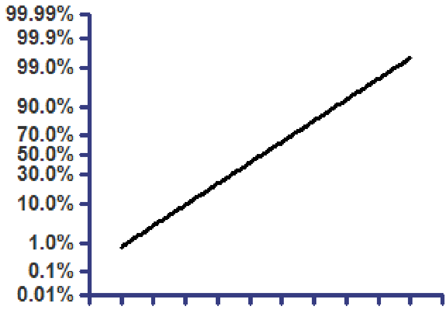
Mean is the average of the original distribution, from which the frequency distribution was created.

SD is the standard deviation of the original distribution.

Both of these parameters are expressed in the same units as the X values plotted on the graph, which is the same as the Y values in the original distribution from which the frequency distribution was generated.

Plotting on a log Y axis

If you choose a Y axis with a probability scale, then the cumulative Gaussian distribution appears as a straight line. For this reason, the cumulative Gaussian models are part of both "Lines" families of equations as well as the "Gaussian" family. The two listings are identical.



4.5.5.16.5 Equation: Lorentzian

Introduction

A Lorentzian distribution is bell shaped, but has much wider tails than does a Gaussian distribution.

Step-by-step

The data must be in the form of a frequency distribution on an XY table. The X values are the bin center and the Y values are the number of observations.

If you start with a column of data, and use Prism to create the frequency distribution, make sure that you set the graph type to "XY graph", with either points or histogram spikes. This ensures that Prism creates an XY results table with the bin centers are entered as X values. If you pick a bar graph instead, Prism creates a column results table, creating row labels from the bin centers. This kind of table cannot be fit by nonlinear regression, as it has no X values.

Starting from the frequency distribution table, click Analyze, choose Nonlinear regression from the list of XY analyses, and then choose the "Lorentzian" equation from the "Gaussian" family of equations.

Model (Lorentzian distribution)

```
Y=Amplitude/(1+((X-Center)/Width)^2)
```

Amplitude is the height of the center of the distribution in Y units.

Center is the X value at the center of the distribution.

Width is a measure of the width of the distribution, in the same units as X. This is not identical to a standard deviation, but has the same general meaning.

Model (sum of two Lorentzian distributions)

```
One=Amplitude1/(1+((X-Center1)/Width1)^2)
Two=Amplitude2/(1+((X-Center2)/Width2)^2)
Y=One + Two
```

Amplitude1 and **Amplitude2** are the heights of the center of the distribution in Y units.

Center1 and **Center2** are the X values at the center of the two distributions.

Width1 and **Width2** are measures of the widths of the distributions, in the same units as X.

Prism is not very smart about assigning initial values to the parameters. If you have trouble getting this model to fit, try fussing with the initial parameter values.

4.5.5.17 Sine waves

4.5.5.17.1 Standard sine wave

Introduction

Sine waves describe many oscillating phenomena.

Step by step

Create an XY data table. There is one X column, and one or many Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel of equations for sine waves, and choose *Standard sine wave*.

If you know the Y value must equal zero at time zero, then constrain PhaseShift to a constant value of zero.

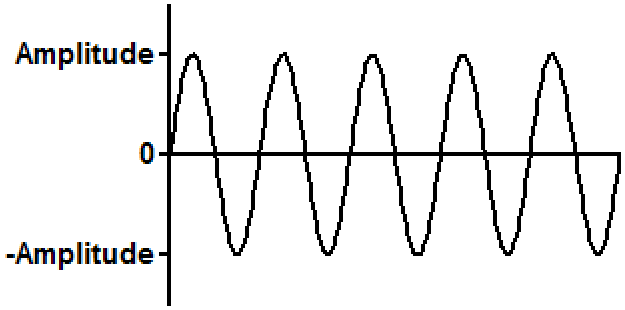
You may need to fuss with the initial values for PhaseShift and Wavelength, as our built-in rules for computing the initial value rarely work.

- For Wavelength. Count the number of peaks between two values X = a and X = b and set the initial value for Wavelength to 1/ (NumberOfPeaks / (b - a)).

- For PhaseShift. If Y=0 at X=0, then PhaseShift is zero. If Y is at its maximum when X=0, then PhaseShift=3.14159. Interpolate in between if necessary.

Model

```
Y= Amplitude*sin((2*pi*X/Wavelength)+PhaseShift)
```



Interpret the parameters

Amplitude is the height of top of the waves, in Y units. If Prism reports a negative Amplitude but you expect a positive Amplitude, simply change the initial value of the PhaseShift by Pi in either direction.

Wavelength is the time it takes for a complete cycle, in units of X

Frequency is the number of cycles per time unit. It is calculated as the reciprocal of wavelength, and is expressed in the inverse of the time units of X.

PhaseShift in radians. A PhaseShift of 0 sets Y equal to 0 at X=0. A PhaseShift of pi sets Y equal to its maximum when X=0. If the best-fit value of PhaseShift surprises you, remember that the sine wave oscillates. You can add or subtract 2*pi from phaseshift to get a different but equivalent phaseshift. Fussing with the initial values can get Prism to fit the one you want.

4.5.5.17.2 Damped sine wave

Introduction

Sine waves describe many oscillating phenomena. Often the peak of each wave decreases or dampens as time goes on.

Step by step

Create an XY data table. There is one X column, and many Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

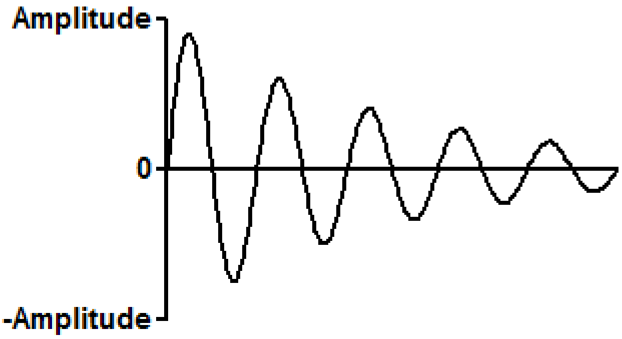
After entering data, click Analyze, choose nonlinear regression, choose the panel of equations for sine waves, and choose *Damped sine wave*.

If you know the Y value is zero at time zero, then constrain PhaseShift to a constant value of zero.

You may need to fuss with the initial values for Phaseshift and Wavelength, as our built-in rules for computing the initial values don't always work very well.

Model

```
Y= Amplitude*exp(-K*X)*sin((2*pi*X/Wavelength)+PhaseShift
```



Interpret the parameters

Amplitude is the height of top of the waves, in Y units.

Wavelength is the time it takes for a complete cycle, in units of X

Frequency is the number of cycles per time unit. It is calculated as the reciprocal of wavelength, and is expressed in the inverse of the time units of X.

PhaseShift in radians. A phaseshift of 0 sets Y equal to 0 at X=0.

K is the decay constant, in the reciprocal of the time units of the X axis.

HalfLlife is the time it takes for the maximum amplitude to decrease by a factor of 2. It is computed as 0.693/K.

4.5.5.17.3 Sinc wave

Introduction

The sinc() function appears frequently in signal and image processing because it is the Fourier transform of a rectangular pulse. It is also called the "sampling" or "sine cardinal" function.

Step by step

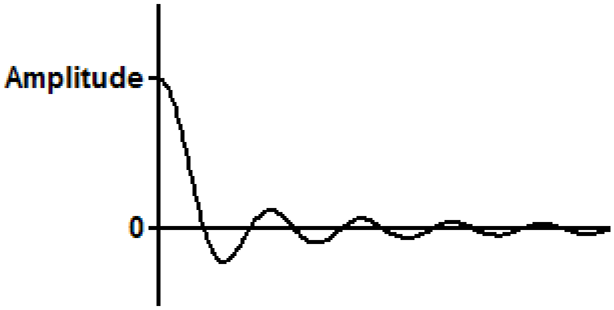
Create an XY data table. There is one X column, and many Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel of equations for sine waves, and choose Sinc() function.

You may need to fuss with the initial values for Phaseshift and Wavelength, as our built-in rules for computing the initial value doesn't always work very well.

Model

```
Y=IF(X=0,Amplitude,Amplitude*sin(2*pi*X/Wavelength)/(2*pi*X/Wavelength) )
```



Interpret the parameters

Amplitude is the height of top of the waves, in Y units.

Wavelength is the time it takes for a complete cycle, in units of X

Frequency is the number of cycles per time unit. It is calculated as the reciprocal of wavelength, and is expressed in the inverse of the time units of X.

4.5.5.17.4 Sine wave with nonzero baseline

Introduction

Sine waves describe many oscillating phenomena. Usually the sine wave oscillates around Y=0 (what we call the standard sine wave). This equation has the wave oscillating around a baseline value that does not have to be zero.

Step by step

Create an XY data table. There is one X column, and can be many Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel of equations for sine waves, and choose *Sine wave with nonzero baseline*.

If you know the Y value is at baseline at time zero, then constrain PhaseShift to a constant value of zero.

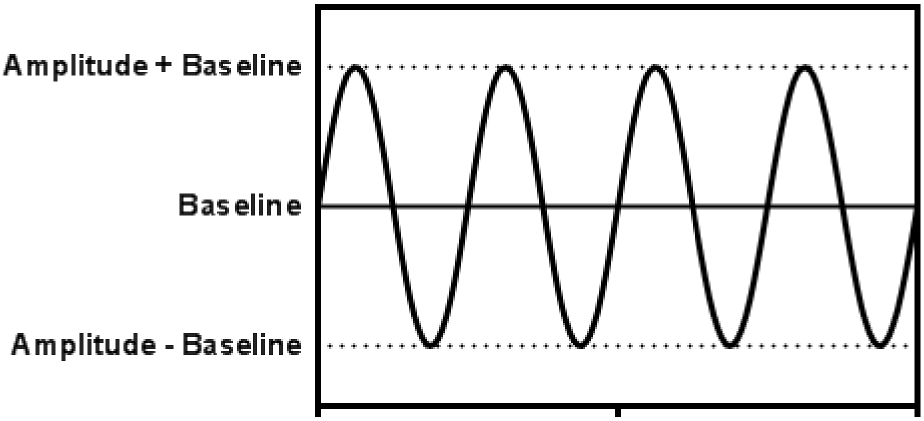
You may need to fuss with the initial values for PhaseShift and Wavelength, as our built-in rules for computing the initial value rarely work.

- For Wavelength. Count the number of peaks between two values X = a and X = b and set the initial value for Wavelength to 1/(NumberOfPeaks / (b - a)).

- For PhaseShift. If Y=Baseline at X=0, then PhaseShift is zero. If Y is at its maximum when X=0, then PhaseShift=3.14159. Interpolate in between if necessary.

Model

```
Y= Amplitude*sin((2*pi*X/Wavelength)+PhaseShift) + Baseline
```



Interpret the parameters

Amplitude is the height of top of the waves from the baseline, in Y units. If Prism reports a negative Amplitude but you expect a positive Amplitude, simply change the initial value of the PhaseShift by Pi in either direction.

Wavelength is the time it takes for a complete cycle, in units of X

Frequency is the number of cycles per time unit. It is calculated as the reciprocal of wavelength, and is expressed in the inverse of the time units of X.

PhaseShift in radians. A phase shift of 0 sets Y equal to Baseline at X=0. A PhaseShift of pi sets Y equal to its maximum when X=0.

Baseline is the Y value around which the curve oscillates, so is in the same units as Y.

4.5.5.18 Growth Equations

.

4.5.5.18.1 Key concepts: Growth equations

What are growth equations used for?

Growth equations are used in many situations: growth of bacterial cultures, growth of organisms, adaptation of technology or ideas among a population, growth of economies...

We provide a few growth equations that might be useful as a starting point. We can't provide much help with these.

Variations

When you read about growth equations, you'll find many variations.

- Different parameterization. This means the other equation has the same number of parameters, and generates the same family of curves, but the parameters have different meanings.

- Different definition of Y. Our equations assume that Y quantifies the value that is growing. Sometimes you'll see equations defined where Y is the logarithm of the ratio of current population value divided by the initial value.

- Differential equations. The equations built in to Prism define Y as a function of X and parameters. You'll sometimes see growth equations defined as the derivative of Y as a function of X.

- t or X? We use *X* to define the independent variable, which for growth equations is time. You'll often see equations using *t* instead.

- More parameters. Some versions of growth equations have more parameters, giving the curve more inflection points.

It is easy to clone our equations 726 and modify 714 as you need.

No accounting for autocorrelation

If you fit a growth model using Prism's nonlinear regression, it works like fitting any other model. The analysis assumes that each point has an ideal Y value predicted by the model with the addition or subtraction of random error. In other words, nonlinear regression assumes that each residual is independent of the others. If one residual is positive, the next one is equally likely to be positive or negative.

With growth studies, sometimes the assumption of independence is clearly violated. If you are following one individual and growth at one time point is a bit higher than predicted by the model, then the growth at the next time point will probably be higher too. The residuals of the two points are not independent. They are sort of cumulative.

Methods exist for fitting models when there are correlated residuals, but Prism doesn't include them.

4.5.5.18.2 Exponential (Malthusian) growth

Introduction

With exponential growth, the population keeps growing forever with a constant doubling time. If you want to use this general model, consider whether you should use the linear (log) version 690 of it instead. When you fit a model, you want the scatter to be about the same at all time points.

This assumption is often more likely to be true of you fit the log of concentration rather than concentration.

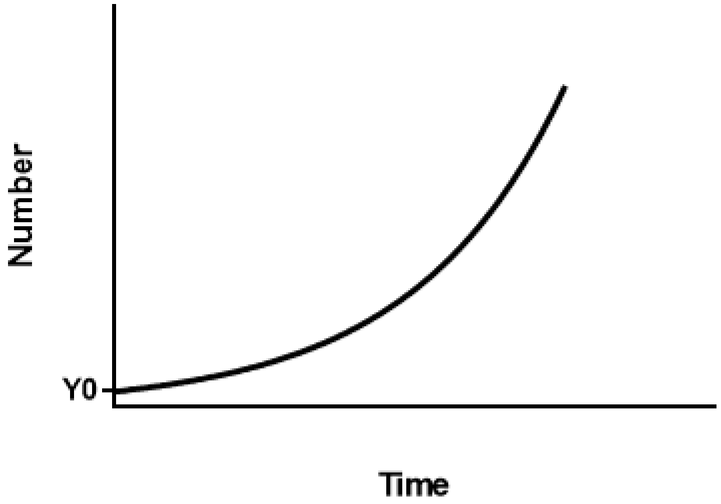
Step by step

Create an XY table. Enter time values into X and population values into Y. After entering data, click Analyze, choose nonlinear regression, choose the panel of growth equations, and choose *Exponential (Malthusian) growth*.

Consider whether you want to constrain Y0 to a fixed value.

Model

```
Y=Y0*exp(k*x)
```



Interpret the parameters

Y0 is the starting population in the same units as Y

k is the rate constant (inverse of the X time units)

DoublingTime is the time need for the population to double. It is calculated as ln(2)/k

4.5.5.18.3 log of exponential growth

Introduction

With exponential growth, the population keeps growing forever with a constant doubling time. If you plot the logarithm of population as Y (instead of the population itself), the graph is linear. When you fit an exponential model, you want the scatter to be about the same at all time points. This assumption is often more likely to be true of you fit the log of concentration rather than concentration.

Step by step

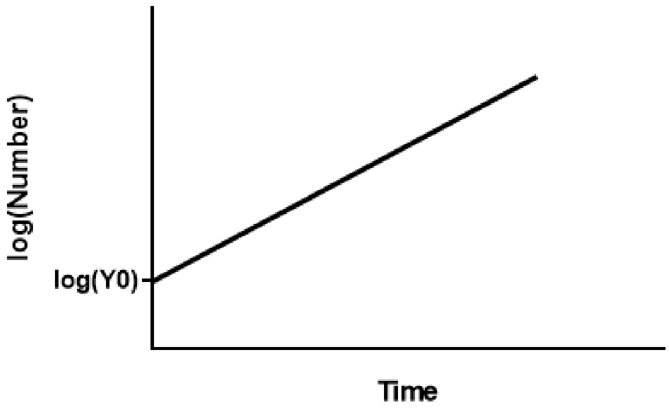
Create an XY table. Enter time values into X and the logarithm of population values into Y. The natural logarithm is use most often for these kinds of data.

After entering data, click Analyze, choose nonlinear regression, choose the panel of growth equations, and choose *Log of exponential growth*.

Consider whether logY0 should be constrained to a fixed values.

Model

```
Y=logY0 + k*X
```



Interpret the parameters

logY0 is the starting population in the same logarithmic units as Y

k is the rate constant (inverse of the X time units).

DoublingTime is the time needed for the population to double. It is calculated as ln(2)/k.

Relationship between this equation and exponential (Malthusian) growth

If the data were transformed using the natural logarithm, then k in this fit is equivalent to k in the exponential (Malthusian) growth equation and the logY0 here is equivalent to the logarithm of Y0 in the exponential equation.

But the two values of k won't usually be identical. The Exponential growth equation is more appropriate when the scatter of data around the curve is the same at all time points. The log of exponential growth equation is more appropriate when the scatter of data around the line is the same at all time points.

4.5.5.18.4 Logistic growth

Introduction

Logistic growth starts off nearly exponential, and then slows as it reaches the maximum possible population.

The logistic model is defined by a linear decrease of the relative growth rate. At any given time, the growth rate is proportional to Y(1-Y/YM), where Y is the current population size and YM is the maximum possible size. As Y approaches the maximum, that second term gets smaller so the growth slows.

Gompertz and logistic models generate curves that are very similar. But when Y is low, the Gompertz model grows more quickly than the logistic model. Conversely, when Y is large, the Gompertz model grows more slowly than the logistic model.

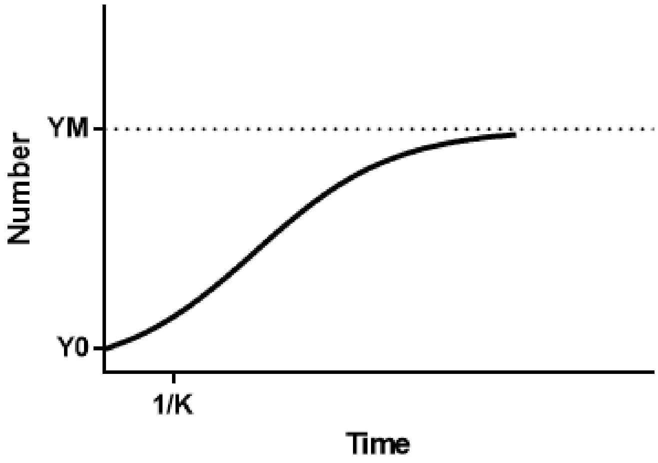
Step by step

Create an XY table. Enter time values into X and population values into Y. After entering data, click Analyze, choose nonlinear regression, choose the panel of growth equations, and choose *Logistic growth*.

Consider whether you want to constrain Y0 and/or Ym to fixed values.

Model

```
Y=YM*Y0/((YM-Y0)*exp(-k*x) +Y0)
```



Interpret the parameters

Y0 is the starting population (same units as Y)

YM is the maximum population (same units as Y)

k is the rate constant (inverse units of X)

1/k is the X coordinate of the first inflection point

4.5.5.18.5 Gompertz growth

Introduction

The defining feature of Gompertz growth is that the growth rate decays exponentially as the population approaches it maximum.

Gompertz and logistic models generate curves that are very similar. But when Y is low, the Gompertz model grows more quickly than the logistic model. Conversely, when Y is large, the Gompertz model grows more slowly than the logistic model.

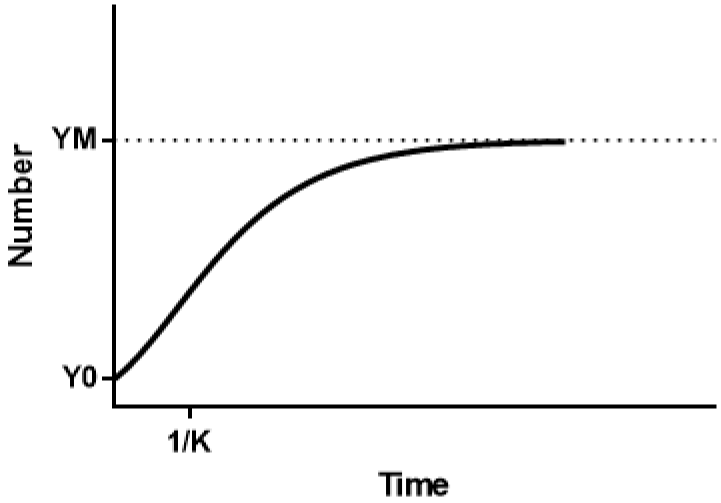
Step by step

Create an XY table. Enter time values into X and population values into Y. After entering data, click Analyze, choose nonlinear regression, choose the panel of growth equations, and choose *Gompertz growth*.

Consider whether you want to constrain Y0 and/or Ym to fixed values.

Model

```
Y = YM*(Y0/YM)^(exp(-K*X))
```



Interpret the parameters

Y0 is the starting population (same units as Y)

YM is the maximum population (same units as Y)

K determines the lag time (inverse of X time units)

1/K is the X value of inflection point

4.5.5.18.6 Exponential plateau

Introduction

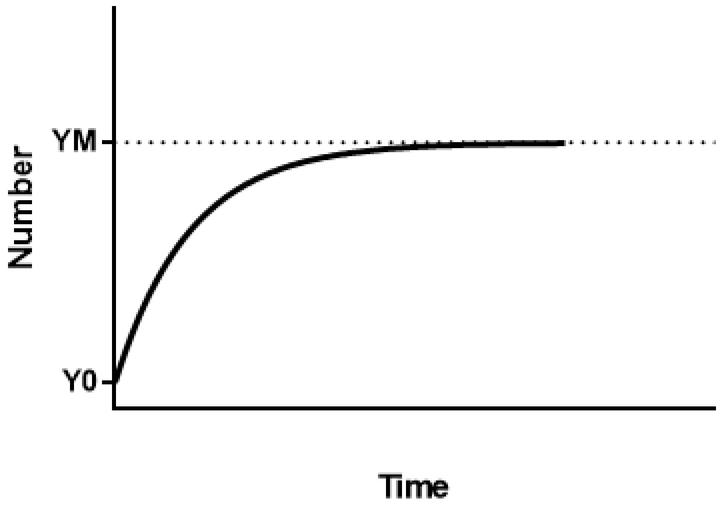
Step by step

Create an XY table. Enter time values into X and population values into Y. After entering data, click Analyze, choose nonlinear regression, choose the panel of growth equations, and choose *Exponential plateau*.

Consider whether you want to constrain Y0 and/or Ym to fixed values.

Model

```
Y=YM -(YM-Y0)*exp(-k*x)
```



Interpret the parameters

Y0 is the starting population (same units as Y)

YM is the maximum population (same units as Y)

K determines is the rate constant (inverse of X time units)

4.5.5.18.7 Beta growth and decline

Introduction

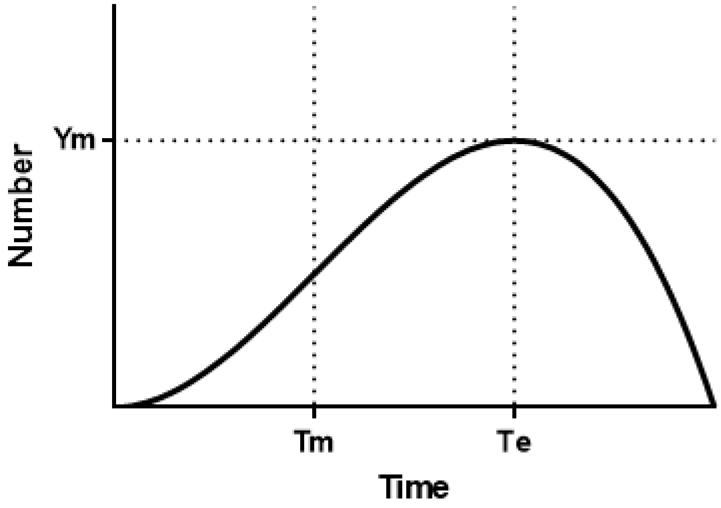
This is one (of many) models that shows a population peaking and then declining. You might find it a useful starting point. But note that the population starts and ends at Y=0, so you may want to add a constant...

Step by step

Create an XY table. Enter time values into X and population values into Y. After entering data, click Analyze, choose nonlinear regression, choose the panel of growth equations, and choose *Beta growth and decline*.

Model

```
Y=Ym*(1+ (Te - X)/(Te - Tm))*(X/Te)^(Te/(Te-Tm))
```



Interpret the parameters

Ym is the population at peak (same units as Y)

Te is the time at which population peaks (same units as X)

Tm is the time of infection point (same units as X)

4.5.5.19 Linear-quadratic model of cell death by radiation

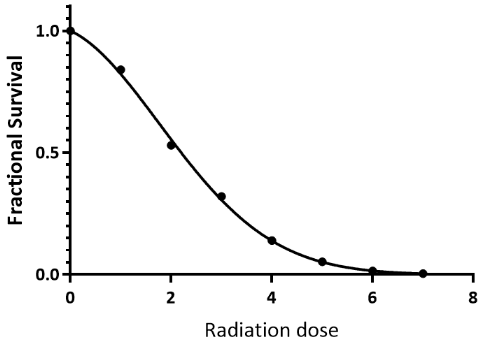
Enter topic text here.

4.5.5.19.1 Key facts: Linear quadratic model

What is the linear-quadratic model?

Mammalian cell death after exposure to radiation after a dose D of radiation often follows the linear-quadratic model (1-3).

Fraction of cells surviving = $e^{-(A*D + B*D^2)}$



The linear-quadratic model was derived by Chadwick and Leenhouts (2). They proposed that the linear component (A*X) represents cell death due to a single lethal hit to the DNA, and that the quadratic (B*X2) component represents cell death that only happens with two hits. However, it is now clear that this mechanism is not correct, and the biological interpretation of the LQ parameters, A and B, is unclear (3).

Even though we don't know the biological basis for the model, it does a reasonable job of describing radiation induced cell death, except perhaps at very high radiation doses. Bodgi et. al. review the linear-quadratic equation and some alternatives (3) and propose a biological model that corresponds to the linear-quadratic model (4).

Sometimes the Y axis is plotted as the logarithm of fraction survival 697, rather than fraction survival itself.

References

Brenner, D. J. (2008). The linear-quadratic model is an appropriate methodology for determining isoeffective doses at large doses per fraction, 18/4: 234–9. Elsevier.

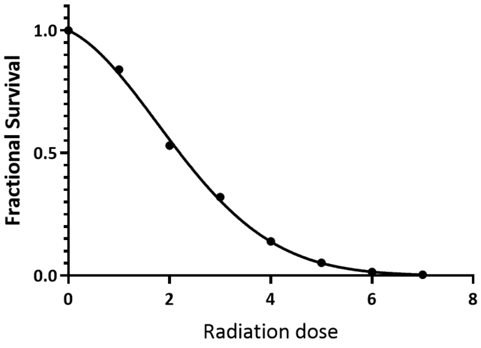
Chadwick, K.H., Leenhouts, H.P., 1973. A molecular theory of cell survival. Phys. Med. Biol. 13, 78–87.

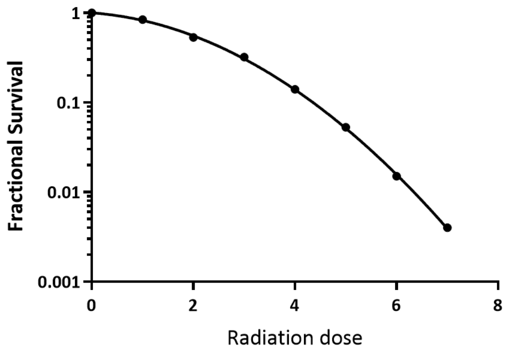
Bodgi, L., Canet, A., Pujo-Menjouet, L., Lesne, A., Victor, J.-M., & Foray, N. (2016). Mathematical models of radiation action on living cells: From the target theory to the modern approaches. A historical and critical review. Journal of Theoretical Biology, 394: 93–101. DOI: 10.1016/j.jtbi.2016.01.018

Bodgi, L., & Foray, N. (2016). The nucleo-shuttling of the ATM protein as a basis for a novel theory of radiation response: resolution of the linear-quadratic model. International Journal of Radiation Biology, 92/3: 117–31. DOI: 10.3109/09553002.2016.1135260

4.5.5.19.2 Graphing the linear-quadratic model

The graph on the left below uses a linear Y axis. The graph on the right uses a logarithmic Y axis. This was done in GraphPad Prism by making the change in the Format Axis dialog. The nonlinear regression "saw" the original data where the Y values were fractions (not logarithm of fractions; that would require a modified model).





Prism does not include variations of the model where the Y values are the logarithms of fractions or percentages of cells dead or alive, but it wouldn't be hard to clone the built-in equations and modify them.

4.5.5.19.3 Linear quadratic: Y is fraction surviving

Introduction

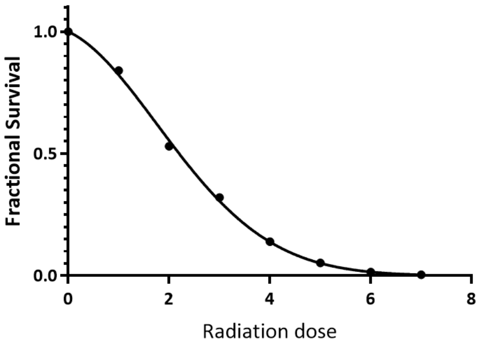
Mammalian cell death after exposure to radiation after a dose D of radiation often follows the linear-quadratic model 696.

Step by step

Create an XY table. Enter radiation doses into X and fraction survival into Y. After entering data, click Analyze, choose nonlinear regression, choose the panel of linear-quadratic equations, and choose Linear quadratic: Y is fraction surviving.

Model

```
Y = exp(-1*(A*X + B*X^2))
```



Interpret the parameters

X: Dose of radiation

Y: Fraction of cells surviving

A: Coefficient for linear killing

B: Coefficient for quadratic killing

4.5.5.19.4 Linear quadratic: Y is percent surviving

Introduction

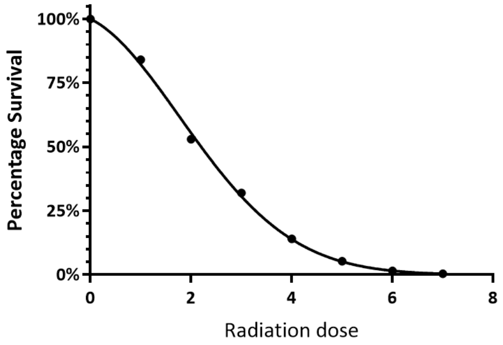
Mammalian cell death after exposure to radiation after a dose D of radiation often follows the linear-quadratic model 696.

Step by step

Create an XY table. Enter radiation doses into X and percent survival into Y. After entering data, click Analyze, choose nonlinear regression, choose the panel of linear-quadratic equations, and choose Linear quadratic: Y is percentage surviving.

Model

```
Y = 100 * exp(-1*(A*X + B*X^2))
```



Interpret the parameters

X: Dose of radiation

Y: Percentage of cells surviving

A: Coefficient for linear killing

B: Coefficient for quadratic killing

4.5.5.19.5 Linear quadratic: Y is number surviving

Introduction

Mammalian cell death after exposure to radiation after a dose D of radiation often follows the linear-quadratic model 696.

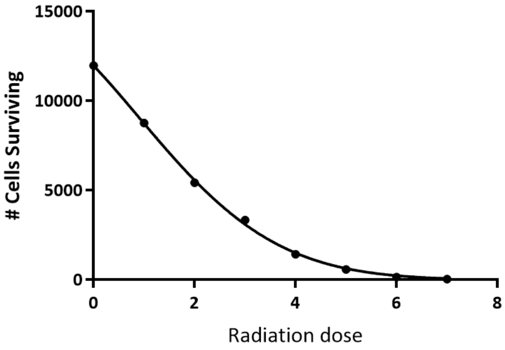
Step by step

Create an XY table. Enter radiation doses into X and number of cells surviving into Y. After entering data, click Analyze, choose nonlinear regression, choose the panel of linear-quadratic equations, and choose Linear quadratic: Y is number of cells surviving.

Consider constraining Y0, the number of cells when X=0, to a constant value.

Model

```
Y = Y0 * exp(-1*(A*X + B*X^2))
```



Interpret the parameters

X: Dose of radiation

Y: Number of cells surviving

A: Coefficient for linear killing

B: Coefficient for quadratic killing

Y0: Number of cells when X=0.

4.5.5.19.6 Linear quadratic: Y is fraction dead

Introduction

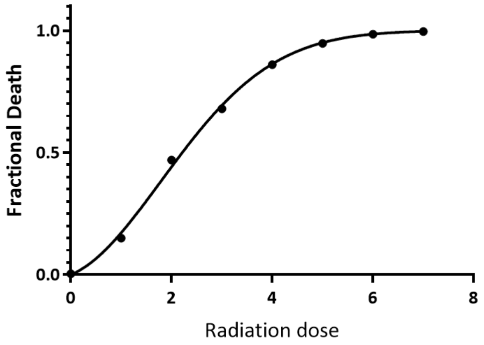
Mammalian cell death after exposure to radiation after a dose D of radiation often follows the linear-quadratic model 696.

Step by step

Create an XY table. Enter radiation doses into X and fraction dead cells into Y. After entering data, click Analyze, choose nonlinear regression, choose the panel of linear-quadratic equations, and choose Linear quadratic: Y is fraction dead.

Model

```
Y = 1.0 - exp(-1*(A*X + B*X^2))
```



Interpret the parameters

X: Dose of radiation

Y: Fraction of cells dead

A: Coefficient for linear killing

B: Coefficient for quadratic killing

4.5.5.19.7 Linear quadratic: Y is percentage dead

Introduction

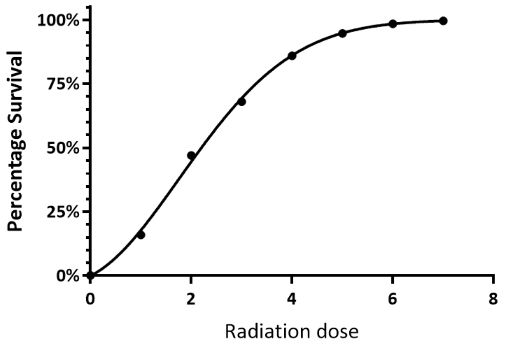
Mammalian cell death after exposure to radiation after a dose D of radiation often follows the linear-quadratic model 696.

Step by step

Create an XY table. Enter radiation doses into X and percent dead cells into Y. After entering data, click Analyze, choose nonlinear regression, choose the panel of linear-quadratic equations, and choose Linear quadratic: Y is percentage dead.

Model

```
Y = 100 * [ 1.0 -exp(-1*(A*X + B*X^2))]
```



Interpret the parameters

X: Dose of radiation

Y: Percentage of cells dead

A: Coefficient for linear killing

B: Coefficient for quadratic killing

4.5.5.19.8 Linear quadratic: Y is number dead

Introduction

Mammalian cell death after exposure to radiation after a dose D of radiation often follows the linear-quadratic model 696.

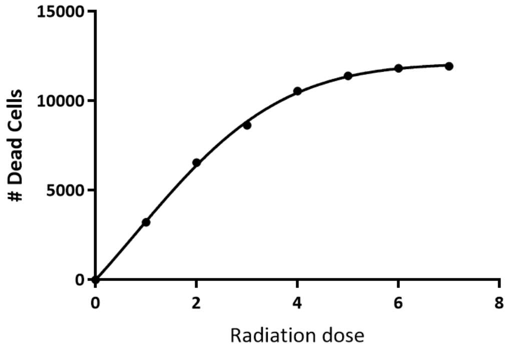
Step by step

Create an XY table. Enter radiation doses into X and number of dead cells into Y. After entering data, click Analyze, choose nonlinear regression, choose the panel of linear-quadratic equations, and choose Linear quadratic: Y is number of cells dead.

Consider constraining Y0, the number of cells alive when X=0, to a constant value.

Model

```
Y = Y0 * [1.0 - exp(-1*(A*X + B*X^2))]
```



Interpret the parameters

X: Dose of radiation

Y: Number of cells surviving

A: Coefficient for linear killing

B: Coefficient for quadratic killing

Y0: Number of cells alive when X=0.

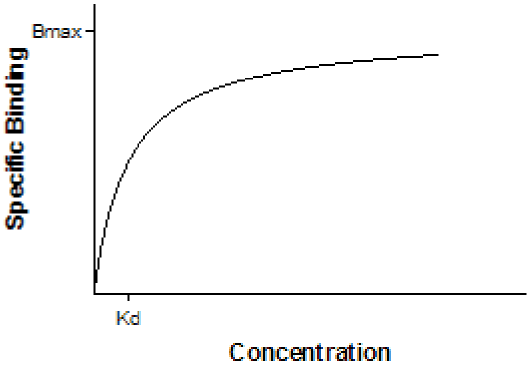
4.5.5.20 Classic equations from old versions of Prism

4.5.5.20.1 Why are thse old equations in Prism?

These equations were the built-in set of equations in Prism versions 1-3. They are present only to maintain compatibility with files made by these versions. All the equations are now present in various forms in the longer list of equations Prism now supports.

4.5.5.20.2 Equation: One site binding (hyperbola)

$$Y = \frac{B_{max} \cdot X}{K_d + X}$$



This curve is known as a *rectangular hyperbola*, *binding isotherm*, or *saturation binding curve*. Y is zero initially, and increases to a maximum plateau value B_{max}.

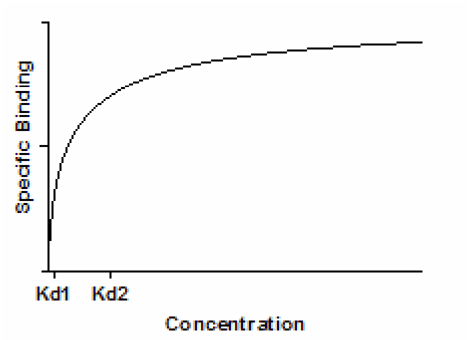
This equation describes the equilibrium binding of a ligand to a receptor as a function of increasing ligand concentration.

- X is the concentration of the ligand.
- Y is the specific binding.
- Bmax is the maximum number of binding sites, expressed in the same units as the Y-axis (usually radioactive counts per minute, sites per cell, or fmol of receptor per mg of tissue).
- Kd is the equilibrium dissociation constant, expressed in the same units as the X-axis (concentration). When the drug concentration equals Kd, half the binding sites are occupied at equilibrium.

This equation also describes the activity of an enzyme as a function of substrate concentration. In this case, the variable labeled Bmax is really Vmax, the maximum enzyme activity, and the variable labeled Kd is really Km, the Michaelis-Menten constant.

4.5.5.20.3 Equation:Two site binding

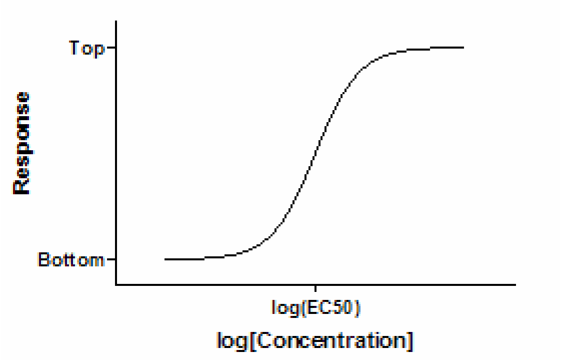
$$Y = \frac{B_{max1} \cdot X}{K_{d1} + X} + \frac{B_{max2} \cdot X}{K_{d2} + X}$$



This equation is an extension of the one site binding curve[704]. It shows the binding of a ligand to two receptors with different affinities (different Kd values). It also describes the enzyme activity as a function of substrate concentration when two isozymes are present. The curve in the example has Kd values that differ by a factor of ten, with equal Bmax values. Even with such a large difference between Kd values, the curve is not obviously biphasic.

4.5.5.20.4 Equation: Sigmoidal dose-response

$$Y = Bottom + \frac{(Top - Bottom)}{1 + 10^{LogEC50 - X}}$$



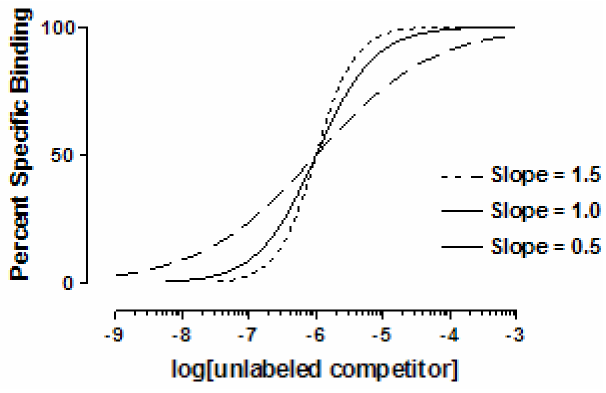
This is a general equation for a dose-response curve. It shows response as a function of the logarithm of concentration. X is the logarithm of agonist concentration and Y is the response. This equation is also called a three-parameter logistic equation.

The variable Bottom is the Y value at the bottom plateau; Top is the Y value at the top plateau, and LogEC50 is the X value when the response is halfway between Bottom and Top. LogEC50 is the logarithm of the EC50 (effective concentration, 50%). With different kinds of variables, this variable is sometimes called ED50 (effective dose, 50%), or IC50 (inhibitory concentration, 50%, used when the curve goes downhill).

This equation assumes a standard slope, where the response goes from 10% to 90% of maximal as X increases over about two log units. The next equation 706 allows for a variable slope.

4.5.5.20.5 Equation: Sigmoidal dose-response (variable slope)

$$Y = \text{Bottom} + \frac{(\text{Top} - \text{Bottom})}{1 + 10^{(\text{LogEC50} - X) \cdot \text{HillSlope}}}$$



This equation extends the previous equation, but allows for a variable slope. This equation is also called a four-parameter logistic equation.

- Bottom is the Y value at the bottom plateau.

- Top is the Y value at the top plateau.

- LogEC50 is the X value when the response is halfway between Bottom and Top. With different kinds of variables, this variable is sometimes called ED50 (effective dose, 50%), or IC50 (inhibitory concentration, 50%, used when the curve goes downhill).

- HillSlope describes the steepness of the curve. This variable is called the Hill slope, the slope factor, or the Hill coefficient. If it is positive, the curve increases as X increases. If it is negative, the curve decreases as X increases. A standard sigmoid dose-response curve (previous equation) has a Hill Slope of 1.0. When HillSlope is less than 1.0, the curve is more shallow. When HillSlope is greater than 1.0, the curve is steeper. The Hill slope has no units.

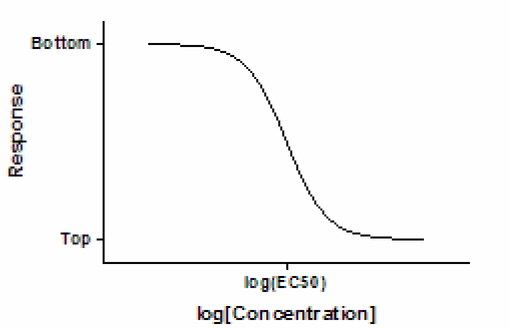
4.5.5.20.6 Equation: One site competition

$$Y = \text{Bottom} + \frac{(\text{Top} - \text{Bottom})}{1 + 10^{X - \text{LogEC50}}}$$

This equation describes the competition of a ligand for receptor binding. It is identical to the sigmoid dose-response curve with HILLSLOPE = -1.0.

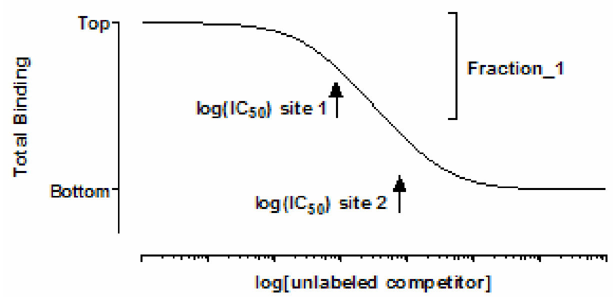
The variable LogEC50 is the concentration of the competitor required to compete for half the specific binding. We use the term EC50 to be consistent with the equations for the other sigmoid curves. The term IC50 is used more frequently ("E" stands for effective; "I" stands for inhibitory).

Usually the Y values are total binding. If you enter specific binding instead, fix BOTTOM to have a constant value of zero. If you enter percent specific binding, also set TOP to be a constant equal to 100.



4.5.5.20.7 Equation: Two site competition

$$Y = \text{Bottom} + (\text{Top} - \text{Bottom})\left[\frac{\text{Fraction\_1}}{1 + 10^{X - \text{LogEC50\_1}}} + \frac{1 - \text{Fraction\_1}}{1 + 10^{X - \text{LogEC50\_2}}}\right]$$

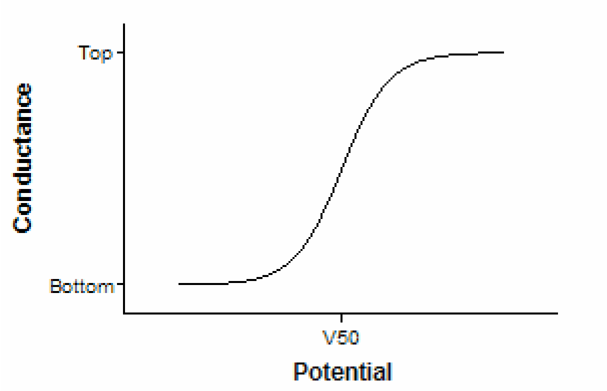


This equation describes the competition of a ligand for two types of receptors. The radioligand has identical affinities for both receptors, but the competitor has a different affinity for each.

Y is binding (total or specific) and X is the logarithm of the concentration of the unlabeled ligand. FRACTION_1 is the fraction of the receptors that have an affinity described by LogEC50_1. The remainder of the receptors have an affinity described by LogEC50_2. If LogEC50_1 is smaller than LogEC50_2, then Fraction_1 is the fraction of high affinity sites. If LogEC50_1 is larger than LogEC50_2, then Fraction_1 is the fraction of low affinity sites.

4.5.5.20.8 Equation: Boltzmann sigmoid

$$Y = \text{Bottom} + \frac{(\text{Top} - \text{Bottom})}{1 + \exp\left(\frac{V50 - X}{\text{Slope}}\right)}$$



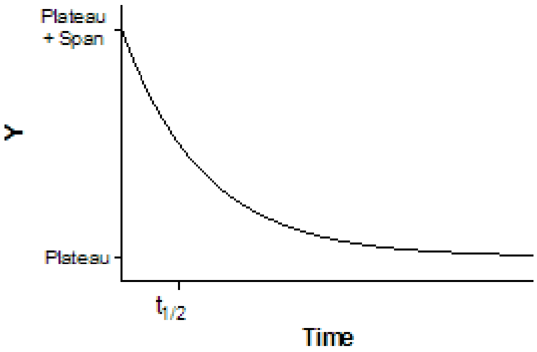
This equation describes voltage dependent activation of ion channels. It describes conductance (Y) as a function of the membrane potential (X). Conductance varies from BOTTOM to TOP. V50 is the potential at which conductance is halfway between BOTTOM and TOP. SLOPE describes the steepness of the curve, with a larger value denoting a shallow curve. Slope is expressed in units of potential, usually mV, and is positive for channels that activate upon depolarization.

Under appropriate experimental conditions, you can use SLOPE to calculate the valence (charge) of the ion moving across the channel. SLOPE equals RT/zF where R is the universal gas constant, T is temperature in °K, F is the Faraday constant, and z is the valence. Since RT/F » -26 mV at 25°C, z = -26/SLOPE.

BOTTOM is commonly made a constant equal to 0.0. If you also make TOP a constant equal to 1.0, then Y can be viewed as the fraction of channels that are activated.

4.5.5.20.9 Equation: One phase exponential decay

$$Y = \text{Span} \cdot e^{-K \cdot X} + \text{Plateau}$$



This equation describes the kinetics such as the decay of radioactive isotopes, the elimination of drugs, and the dissociation of a ligand from a receptor.

X is time.

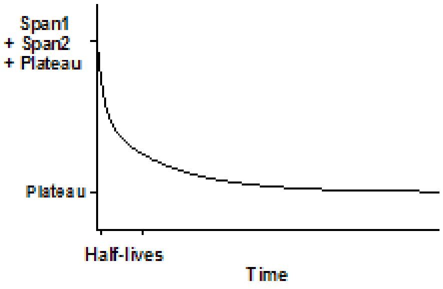
Y may be concentration, binding, or response. Y starts out equal to SPAN+PLATEAU and decreases to PLATEAU with a rate constant K.

The half-life of the decay is 0.6932/K.

SPAN and PLATEAU are expressed in the same units as the Y axis. K is expressed in the inverse of the units used by the X axis. In many circumstances, the plateau equals zero. When fitting data to this equation, consider fixing the plateau to a constant value of zero.

4.5.5.20.10 Equation: Two phase exponential decay

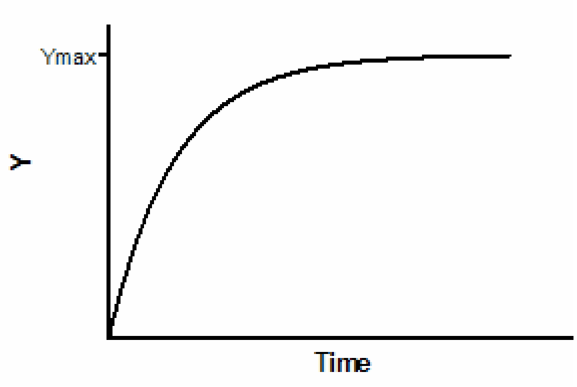
$$Y = \text{Span1} \cdot e^{-K_1 \cdot X} + \text{Span2} \cdot e^{-K_2 \cdot X} + \text{Plateau}$$



This equation describes a two phase exponential decay. Y starts out equal to Span1+Span2+PLATEAU and decays to PLATEAU with fast and slow components. The two half-lives are 0.6932/K1 and 0.6932/K2. In the figure, the two rate constants differ tenfold, but the spans were equal. The curve is not obviously biphasic, and it takes a very practiced eye to see that the curve does not follow a single phase model.

4.5.5.20.11 Equation: One phase exponential association

$$Y = Ymax \cdot \left(1 - e^{-K \cdot X}\right)$$



This equation describes the pseudo-first order association kinetics of the interaction between a ligand and its receptor, or a substrate and an enzyme. Y is either binding or enzyme activity. X is time.

Y starts out equal to zero and increases to a maximum plateau (at equilibrium) equal to YMAX. When X equals 0.6932/K, Y equals 0.5*YMAX.

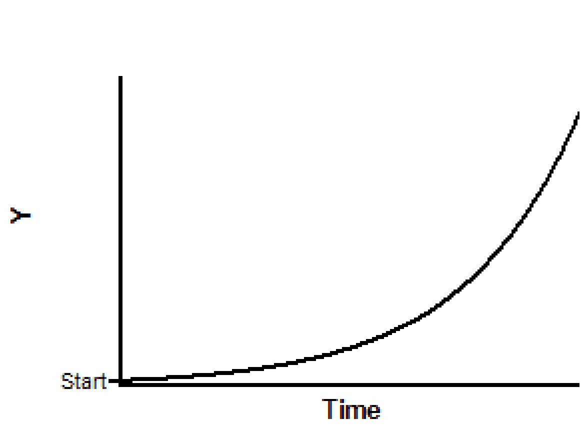
4.5.5.20.12 Equation: Two phase exponential association

$$Y = Ymax_1 \cdot \left(1 - e^{-K_1 \cdot X}\right) + Ymax_2 \cdot \left(1 - e^{-K_2 \cdot X}\right)$$

This is an extension of the exponential association to two phases, corresponding to a radioligand binding to two independent sites.

4.5.5.20.13 Equation: Exponential growth

$$Y = Start \cdot e^{K \cdot X}$$



This describes an exponential growth curve. Y is population size (perhaps cell number) and X is time. At X=0, Y equals START. Y increases geometrically with a doubling time equal to 0.6932/K.

Note: It is difficult to fit data to this equation with nonlinear regression, because a tiny change in the initial values will drastically alter the sum-of-squares. You may need to override the initial values provided by Prism.

4.5.5.20.14 Equation: Power series

This versatile equation has many uses.

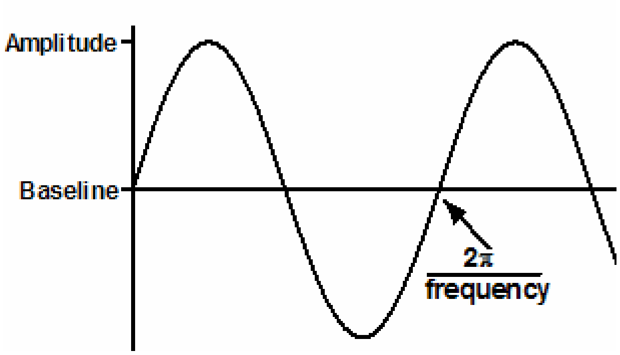
$$Y = A \cdot X^{B} + C \cdot X^{D}$$

Fitting data to a power series model can be difficult. The initial values generated automatically by Prism are not very helpful (all four parameters are set to 1.0). you will probably need to enter better initial values in order to fit this equation to data. The initial values of B and D are important, because small changes in those values can make a huge change in Y.

The equation is not defined, and leads to a floating point error, if X equals zero and B or D are negative numbers or if X is negative and B or D are between 0.0 and 1.0.

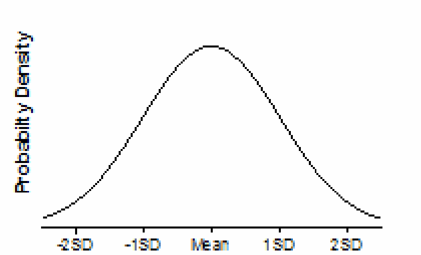
4.5.5.20.15 Equation: Sine wave

$$Y = \text{Baseline} + \text{Amplitude} \cdot \sin(\text{Frequency} \cdot X + \text{Offset})$$



X is in radians. In most cases, you will want to fix BASELINE to a constant value of zero. AMPLITUDE is the maximum height of the curve away from the baseline. FREQUENCY is the number of complete oscillations per 1 X unit.

4.5.5.20.16 Equation: Gaussian distribution



Cumulative probability distribution of a Gaussian bell-shaped distribution with specified mean and SD. The area under the entire curve is AREA. A standard probability distribution is scaled so that AREA equals 1.0. The units of the Y-axis are arbitrary, determined by your choice of AREA.

4.5.6 Entering a user-defined model into Prism

Prism provides lots of built-in models, but you should not hesitate to adapt these or enter additional models. This section shows you how to do so.

- Overview: User-defined equations 714
- How to: Enter a new equation 727
- How to: Clone an equation 726
- How to: Manage your list of equations 756

- Syntax of user-defined equations 730
- Multiline models 732
- Limitations when entering equations 733
- Entering a differential equation 735
- Entering an implicit equation 736
- Available functions for user-defined equations 738
- Fitting different segments of the data to different models 743
- Fitting different models to different data sets 744
- Column constants 746
- Defining equation with two (or more) independent variables 748
- Rules for initial values 721
- Default constraints 720
- Transforms to report 715

4.5.6.1 Overview: User-defined equations

Few people will need to enter user-defined equations into Prism. If you need to do this, keep these points in mind:

- Rather than starting with a blank screen, it is usually much easier to clone an existing equation 726, and then modify it.
- You must define Y as a function of X and some parameters. The equation can't define P as a function of V, or Z as a function of Y. It must define Y as a function of X.
- The syntax Prism uses is pretty standard. But use plenty of parentheses, so there is no doubt about what the equation means.

Many people have made the mistake of entering "A/B+C" expecting Prism to divide A by the sum of B plus C. Instead, Prism first dives A by B, and then adds that quotient to C. Writing the equation with parentheses as "A/(B+C)" avoids any ambiguity.

- Prism lets you use many standard mathematical functions 738 when defining your model. Most of these are standard. But watch out for a few that are implemented differently in different languages. The log() function in Prism is common (base 10) log. If you want to use natural logs, use the ln() function. The sqr() function in Prism squares the value, while the sqrt() function computes the square root.

- Prism lets you write an equation in such a way that different models are fit to different data sets 744.

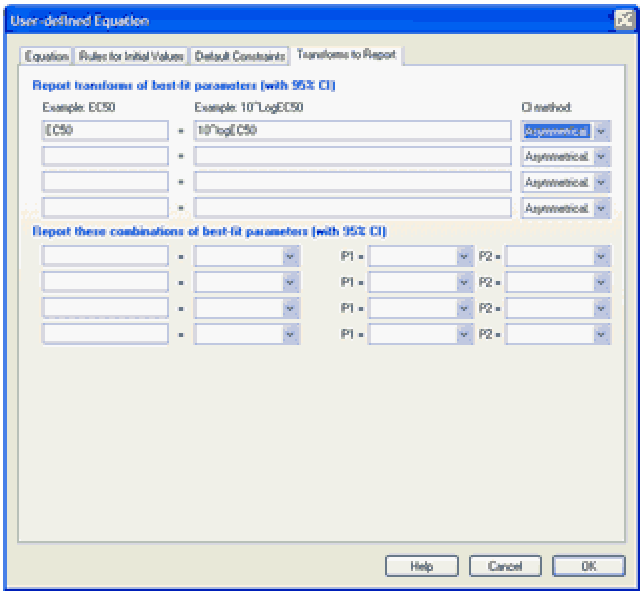
- When you write a model using several lines 732, Prism evaluates from the top down. In other words, you must first define intermediate variables and then use them lower down. The last line must define Y. This follows the conventions of all computer languages, but is opposite to how mathematical text books present complicated equations.

- After writing the equation, don't forget to define rules for initial values 721. If you don't do this, you'll need to enter initial estimated values for each parameter every time you fit the model to data.

- When you enter a new equation, Prism also lets you define default constraints 720 and transforms to report 715. These features are optional, so you can skip them when first learning about user-defined equations.

4.5.6.2 Choosing transforms of parameters to report

When defining an equation, you can also ask Prism to report transforms of the best-fit values on the "Transforms to report" tab.



Report transforms of best-fit parameters (with 95% CI)

Example: EC50 | Example: 10^LogEC50 | CI method

EC50 = 10^logEC50 Asymmetrical

= Asymmetrical

= Asymmetrical

= Asymmetrical

Report these combinations of best-fit parameters (with 95% CI)

= P1 = P2 =

= P1 = P2 =

= P1 = P2 =

= P1 = P2 =

Transforms of one parameter

Entering the transform

Enter simple transforms of one parameter. On the left side enter the name you want to give the results. On the right side enter the transform.

Example: You fit data to an equation that includes a parameter logEC50 that is the logarithm of an EC50, but you want to also report the EC50. Enter the label 'EC50' on the left and '10^logEC50' on the right (without the quotes).

Example: You fit data to an equation that reports a rate constant K, but also want to report the half-life. Enter the label 'HalfLife' on the left and 'ln(2)/K' on the right (without the quotes).

Confidence intervals - Symmetrical vs. asymmetrical

This choice only has an impact when you have chosen "symmetrical, asymptotic, approximate confidence intervals" in the Confidence tab.

- Symmetrical confidence interval. Prism also reports the SE of that parameter, and both SE and CI are exactly the same as they would have been if you had fit that transformed parameter directly. For example if you fit an exponential decay model to determine the rate constant, and transform that rate constant (take its reciprocal) to compute the time constant, the SE and CI of the time constant will be exactly the same as they would have been had you fit the time constant directly.

- Asymmetrical confidence interval. Prism transforms both confidence limits to create the confidence interval of the transformed parameter. In most cases, this transformed confidence interval will not be symmetrical.

Choosing between the two is often not straightforward. The symmetrical intervals give you the results you would have had if you had chosen to express the model differently. The asymmetrical results simply express the same confidence interval on a different scale. You want to compute the confidence intervals on the scale where the true uncertainty is most symmetrical. You can then transform those confidence limits to come up with asymmetrical confidence interval on a different scale.

If you really care to know the confidence intervals of the parameters and transforms of the parameters with precision, choose "asymmetrical, profile-likelihood" confidence intervals in the Confidence tab of nonlinear regression. When you make this choice, the choice about symmetrical or asymmetrical confidence interval is ignored, and you'll get the same results with either choice. You make the choice when defining the equation. At that point, Prism doesn't know if you are going to choose profile likelihood intervals or not, so the choice is always available. However the choice is ignored for fits where you choose profile likelihood intervals.

Notes:

- Prism 4 (and earlier) always reported the asymmetrical intervals for EC50 and half lives, but did not offer the choice of transforming parameters in user-defined models.

- If your transform simply changes units, then Prism still offers the two choices, but the choice doesn't matter as both end up being exactly the same. This is the case whenever the transform of parameter K is of the form a*K+b

- Don't mix up the two choices for reporting the CI of transformed parameters discussed here with the two methods for reporting the CI of the parameters themselves 343.

- Prism only reports results when the results of the transform is monotonic. Otherwise the results will be blank. This is true for both symmetric and asymmetric confidence intervals.

Interpolating transforms

How to interpolate points off the curve

You can also use these 'transforms' to report values from the curve. The interpolated value and its confidence interval will appear in the results, the same as other transformed parameters.

Use this syntax:

Y[value or parameter]	The Y value of the curve when X is the value you enter within the brackets. The Y value will be computed for any X, but confidence intervals will be calculated only when the X value is within the range of the X axis.
X[value or parameter]	The X value of the curve when Y is the value you enter within the brackets. Prism searches for the Y value you entered within the range the curve is plotted (Range tab) and extending in each direction a distance equal to half that range. It reports the smallest X value it finds within that range that corresponds to the Y value you entered, and doesn't alert you when the curve oscillates so there are several X values at a particular Y value. If both X and Y are within the axis range, a confidence interval is also calculated.

Within the brackets, enter one of the following:

- A number

- The name of a parameter

- The name of a transformed parameter (defined earlier on this page). This is new to Prism 8.

Example: You fit data to a log(dose) response curve and want to report the antilog of the X value (dose) when Y=50 (which is not always the same as the EC50). For the second example, you would enter "Dose at Y=50' on the left, and '10^X[50]' on the right.

Confidence intervals

The confidence interval for interpolating transforms is computed by interpolating the confidence bands of the regression curve. You don't have a choice of symmetrical vs. asymmetrical intervals, and your choice for symmetrical Wald or asymmetrical profile-likelihood confidence intervals 343 don't apply here (only to parameters and functions of parameters).

Don't mix up confidence intervals of transformed parameters with confidence interval of interpolated values

In spite of the fact that confidence intervals of transformed parameters and the confidence interval of interpolated values look the same in Prism's results tables, they are actually quite different. The transform of a parameter could have been a parameter had the equation been written differently. In contrast, an interpolating transform is the predicted value of the curve. The idea of a CI of interpolating transform is the same as the idea of the confidence band of the curve.

Note a potential point of confusion. The CI of interpolated values are shown in a results section with a title that depends on selected method: symmetrical Wald or asymmetrical profile likelihood. However the CI of interpolating transform are calculated using a method that has to do with confidence bands, and has nothing to do with the method used to compute the confidence of the parameters, even though it is in a section labeled that way.

Combining two parameters

Usefulness of combining parameters

The bottom half of the tab lets you define combinations of parameters to report. For example, you can report the sum of two parameters, or the ratio. Enter the name you want to give to the transform on the left, and then choose the calculation from the drop-down list (say "P1/P2" for a ratio). Then choose which parameter is P1 and which is P2 in the drop downs on the right.

Confidence intervals

When Prism combines two parameters to come up with a calculated value, it is smart about propagating the errors. The standard error and confidence interval it reports for the calculated variable are exactly the same as they would have been had you rearranged the equation to directly fit that calculated value.

When Prism reports the difference or ratio (etc) of two parameters, with standard error and confidence interval, the calculations account for the covariance of the two parameters. See this document.

4.5.6.3 Entering default constraints

What is a constraint?

Constraining parameters is often essential to getting useful results. Constraints can be used to fix parameters to constant 336 values, to share parameters among data sets (global 69 fitting), and to define one parameter to be a column constant 746 (whose value comes from the column titles in the data table).

Defining constraints when defining an equation

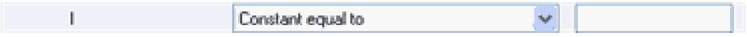
You can define default constraints as part of the equation definition. This is useful for constraints that will apply every time the equation is used.

Constraints can be for one parameter (Kfast must be greater than zero) or for the relationship between two parameters (Kfast must be greater than Kslow). But note that you cannot invoke a constraint between two parameters if both parameters also are themselves constrained. In a two phase exponential equation, you may want to constrain both parameters to be greater than zero, and also define one rate constant to be larger than the other (Kfast > Kslow). Prism won't let you do that. What you have to do is define one constraint that Kfast is greater than zero, and another that Kfast is greater than Kslow. But don't put in the constraint that Kslow is greater than zero. That is implied by being larger than Kfast.

Defining experimental constants

In some cases, one of the parameters in the equation is an experimental constant. It needs to be set to a constant value, but that value differs each time you run the experiment. Do that by going to the constraints tab of the nonlinear regression dialog every time that equation is used.

When you define the equation, you can add a constraint that reminds anyone using the equation to set that constant value. In the constraints tab when defining the equation, choose the constraint "Constant equal to" but leave the value (to the right) blank. With an equation defined this way, Prism won't fit data to that equation until the parameter is defined (or the constraint changed).



I Constant equal to

4.5.6.4 Entering rules for initial values

Why initial values?

Before it can perform nonlinear regression, Prism must have initial (estimated) values for each parameter in the equation. It then iteratively modifies these values to improve the fit of the model to the data. The initial values must be reasonable, so that the curve defined by those initial values has generally the shape of the data, and doesn't go awfully far from the data points. If the initial values are really bad, Prism may not be able to fit the model.

The need for initial values is not unique to Prism, but is intrinsic to nonlinear regression.

Why rules for initial values?

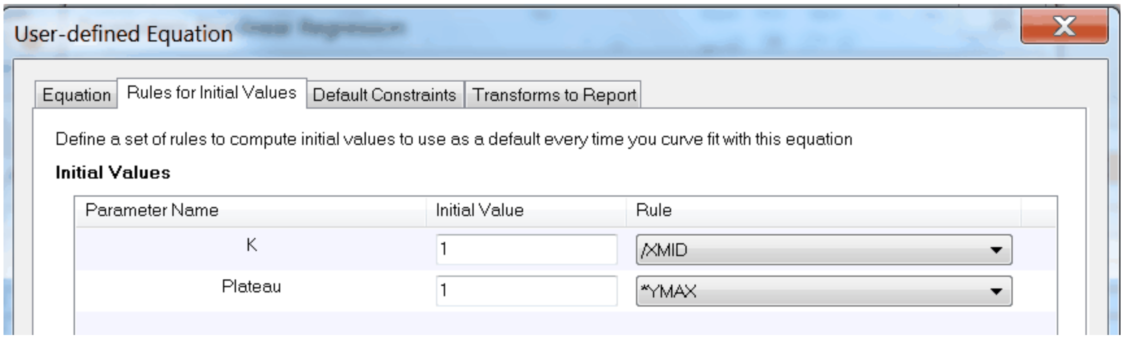
Prism can store rules (algorithms) to calculate initial values from the range of your data. If you choose a built-in equation, the rules are taken care of. If you enter your own (user-defined) equation, you can also enter rules to define the initial values of each parameter. If you enter a good set of rules, then your equation will work in the future with no need to think about initial values.

If you don't enter rules for initial values, you will need to enter the initial values for every variable, for every data set, every time you fit data.

If you enter a poor set of rules, then the initial values will not generate a curve that goes near the data points, and it is likely that Prism will not be able to fit the model to the data.

How to enter a rule

While entering or editing a user-defined equation, click on the tab labeled "Rules for initial values". For each parameter, enter a number in the first column and select a value to multiply or divide by.



Define a set of rules to compute initial values to use as a default every time you curve fit with this equation

Initial Values

Parameter Name	Initial Value	Rule
K	1	/XMID
Plateau	1	*YMAX

Most often, you'll use the choices based on YMIN, YMID or YMAX, the minimum, middle (mean of minimum and maximum), and maximum Y value in your data set, or the corresponding XMIN, XMID or XMAX.

You can also ask Prism to interpolate, for example to base the initial value on the interpolated value of X at the value of Y half way between the smallest and largest value (XatYmid) and related values.

Initial value, to be fit

The first choice on the drop-down list is "(Initial value, to be fit)". This means that the value you entered in the box to the left will be the initial value for all data sets. The initial value will not depend on the range of the data (it will not be calculated); it will be the actual value you entered.

Initial value based on the range of the data

Most of the rules are based on the range of the X and Y values.

The abbreviation YMIN is the minimum value of Y; YMAX is the maximum value, and YMID is the average of YMIN and YMAX. For example, if you enter "0.5" in the first column and select "YMAX" in the second column, Prism sets the initial value to half of YMAX (which differs for each data set).

The phrase "X at YMID" refers to the X value that corresponds to the mean of the highest and lowest Y values. Calculation details.

SIGN(YatXmax - YatXmin)

It equals +1 if the curve generally goes up as it goes from left to right: /

It equals -1 if the curve generally goes down as it goes from left to right: \

It is used as the initial value for the Hill Slope in dose response curves.

Here is an explanation of the math. The SIGN() function equals +1 when given a positive number, and -1 when given a negative number. YatXMax is the Y value at the largest X values. YatXmin is the Y value at the smallest X value. (YatXmax - YatXmin) will be positive when the curve goes up, and negative when the curve goes down.

Mean of column title values (or the log of that mean)

With some kinds of data, it makes sense to label the data sets numerically. Each column titles has a value.

With these kinds of data, you can access the column title values in two ways:

- Use the Constrain tab and choose "Data set constant (=column title)". That parameter in the model is now a constant, but the value of that constant will differ for each data set column, and gets it value from the column title.

- On the Rules for initial values tab in the dialog that defines an equation, set the initial value of a parameter to equal the mean of those column titles. Or to the logarithm of that mean.

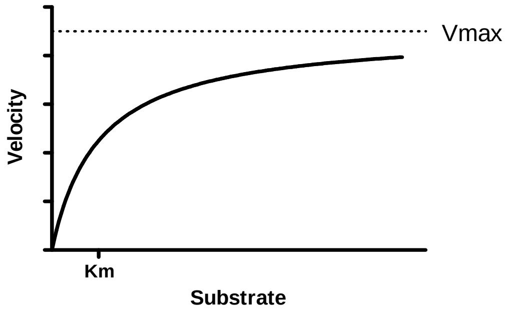
The best way to understand this is with an example. Bring up the sample XY data for competitive enzyme inhibition 622, and follow the instructions, and inspect the dialog settings.

Initial, middle and final slope

New to Prism 7: Base the initial value on the initial, middle, or final slope (dY/dX at the smallest, middle or largest value of X) or its reciprocal.

Example

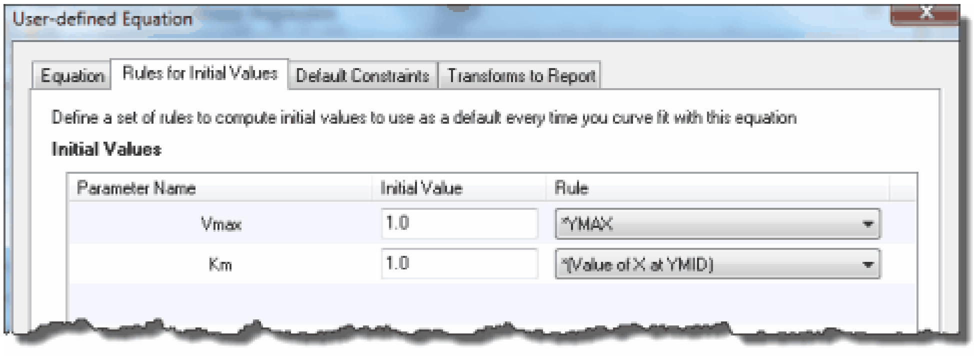
```
Y = Vmax*X/(Km + X)
```



The equation above is the Michaelis-Menten enzyme kinetics model built in to Prism. If you pick this equation as a built-in equation, the rules for initial values are provided automatically. But if you were entering this as a new user-defined equation, it would be easy to come up with rules.

Vmax is the Y value (enzyme velocity) at very high values of X (substrate). So a reasonable initial value would be to set Vmax equal to the largest Y value in the data set. To do this, enter the value 1.0 and choose the multiplier *Ymax. This means Prism will compute the initial value by multiplying the largest Y value by 1.0. You could argue that Vmax is usually higher than the largest Y value, so should compute the initial value by multiplying 1.25 times Ymax. There are often multiple rules one could justify. These rules only compute the initial value, so there is a lot of room for variation.

Km is the X value (substrate concentration) corresponding to a Y value (enzyme velocity) that is half Vmax. Compute its initial value by multiplying 1.0 times "Value of X at Ymid". Prism then averages the largest and smallest Y values to compute Ymid, and then interpolates from straight connecting lines to find the corresponding X value.



Define a set of rules to compute initial values to use as a default every time you curve fit with this equation

Initial Values

Parameter Name	Initial Value	Rule
Vmax	1.0	*YMAX
Km	1.0	*(Value of X at YMID)

Slope at Yleft, Ymid or Yright

These are the slopes (change in Y divided by change of X) at the left, middle or right of the graph. If you have 30 or more data points, Prism computes the slope from the first, middle or last 10% of the points. If you have fewer than 30 points, Prism computes the slope from the first, middle, or last two points.

If you have more than 30 points, the slope is determined by linear regression on the first, middle or last 10% of the points, rounded down. If there are fewer than 30 points, the slope is determined by the first or last two points. For slopeAtYmid with fewer than 30 points, Prism uses two middle points if there are an even number of points, and otherwise uses the three middle points.

What happens if you don't enter any rules?

If you don't enter rules for every parameter, Prism will insist that you enter initial values each time you fit data to the equation.

How to override the initial values defined by rule

The rules are stored with each equation, and the rules can be edited as part of the dialog that defines and edits equations.

The values defined by the rules are shown in the Initial Value tab of the nonlinear regression dialog. To override the initial values:

1. Select one or several data sets on top of the dialog. Or click "Select all" to select all data sets.

2. Uncheck the "Choose automatically" option.

3. Enter the initial value (as a value, a number, not a rule) .

To check how good the initial values are, go to the Diagnostics tab and check the option:

> (*)Don't fit the curve. Instead plot the curve defined by the initial values of the parameters.

If the resulting curve doesn't go near the data, revise the initial values (or the rules that generate those values) before asking Prism to fit the model to the data.

4.5.6.5 How to: View the equation used for a fit

When viewing a file with a nonlinear regression result, to view the details of that equation:

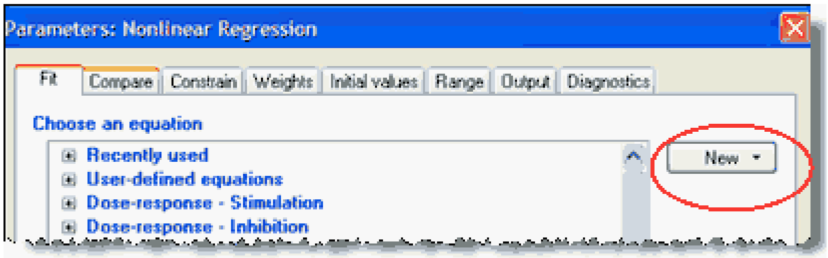
1. Go to the Model tab of nonlinear regression
2. If you use Windows, on the right side of the dialog, click the "Edit..." button.
   If you use Mac, on the right side of the dialog, click the button with the gear icon.
3. You'll see the Equation dialog that lets you see the equation, the rules for initial values, the default rules for constraints, and the list of transforms to report. If this is a user-defined equation, you can change anything you want. If it is a built-in equation, you can't change anything. If you want to change a built-in equation, you need to "clone 726" it. and then edit the copy.

4.5.6.6 How to: Clone an equation

If none of the built-in equations, suit your needs, you may not need to enter a new equation. Instead, consider cloning an existing equation. You may also clone an equation if your only goal is to rename some of the parameters.

To clone an equation:

1. Click the New button on the top of the Fit tab of the Nonlinear regression dialog.



2. A drop down menu lets you choose to enter a new equation, clone an existing one, or import an equation from a saved .PZF file. Choose to clone.

3. Edit the duplicated (cloned) copy. You can change the equation, the rules for initial values, constraints and transforms to report.

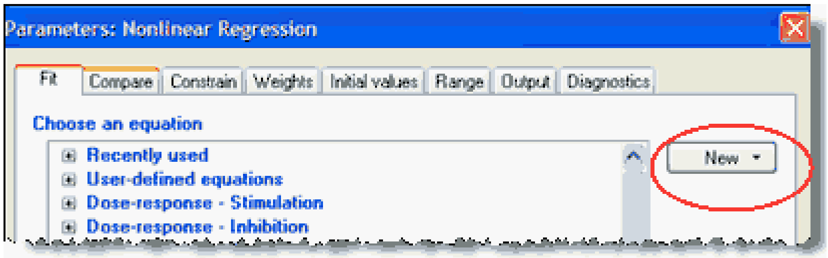
4. The default name of the cloned equation is the original name followed by a digit. Change to a more informative name.

The cloned equation is an exact duplicate of the equation you started with except for a digit appended to the equation name. Cloning does more than duplicate the math. It also includes the description, hint, rules for initial values, default constraints. and transforms to report.

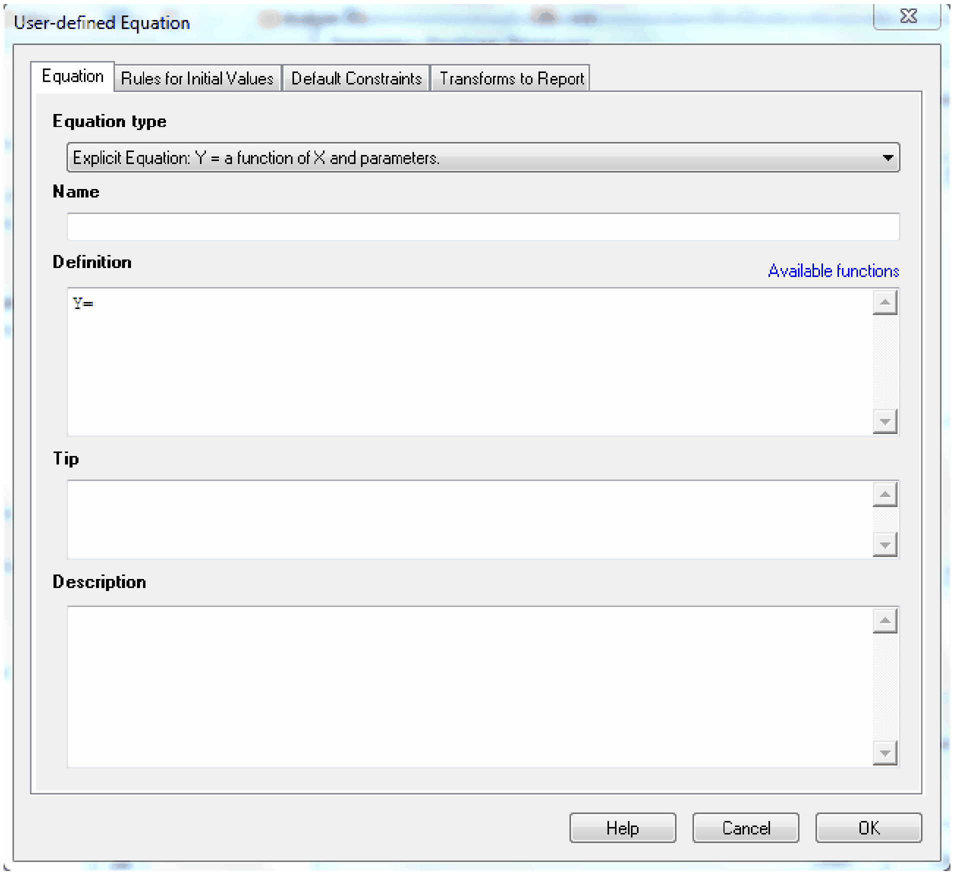
4.5.6.7 How to: Enter a new equation

Entering an equation

Prism comes with many built-in equations, but you will often want to enter a different equation. To do so, click the New button on the top of the Fit tab of the Nonlinear regression dialog.



A drop down menu lets you choose to enter a new equation, clone an existing one, or import an equation from a saved .PZF file.



Equation type

Explicit Equation: Y = a function of X and parameters.

Name

Definition

Available functions

Y=

Tip

Description

Name, definition, tip and description

Equation type

Choose whether you are entering an ordinary explicit equation (that defines Y as a function of X and parameters), an implicit equation (Y appears on both sides of the equals sign) or a differential equation (which defines the derivative of Y with respect to X).

Name

Enter a name you will recognize, so you can choose the equation again. The name also appears on the results table, so make it both descriptive and concise.

Definition

Write the math that defines Y as a function of X and one or more parameters. Read the details of equation syntax 730, a list of available math functions 738 you can use, and how to fit different models to different ranges of X values 743 and to different data sets 744.

Tip

The tip appears below the equation list in the Fit tab of the Nonlinear regression dialog. Use it to distinguish among similar equations. Tips are optional, and must be short (a few sentences).

Description

The description can be up to several paragraphs. Anyone choosing the equation can view the description (and the math) by clicking the Edit or View button. This is a good place to document the source of the equation, its assumptions, units of the parameters, etc.

Rules for initial values

Nonlinear regression must begin with initial estimated values for each parameter. You can enter values directly, or enter a rule whereby Prism can compute the initial value from the range of the X and Y values. Learn more 721.

The values or rules you enter will become the default set of initial values that will be used every time the equation is selected. But each time the equation is selected, you (or whoever is selecting the equation) can change the initial values 339 for that one fit.

Default constraints

Use the constraints tab to set default constraints. You can constrain a parameter to a constant value, constrain to a range of values, share among data sets (global fit), or define a parameter to be a column constant 746. These constraints will become the default every time the equation is selected. But each time the equation is selected, you (or whoever is selecting the equation) can change the constraints 336 for that one fit.

If a parameter has to be set constant, but the actual value is different for each experiment, set the constraint "Constant equal to" but leave the value blank. If someone chooses the equation but forgets to constrain that parameter to a constant value, Prism will prompt for one.

Transforms to report

Define transforms of the best-fit values on the Transforms to report 715 tab. Unlike initial values and constraints, you can not override these transforms each time you choose the equation. Transforms are defined with the equation definition and can not be tweaked each time the equation is selected.

4.5.6.8 Syntax used when enteirng a user-defined equation

Enter topic text here.

4.5.6.8.1 Overview of equation syntax

General syntax

- Variable and parameter names must not be longer than 13 characters.
- If you want to use two words to name a variable, separate with the underscore character, for example Half_Life. Don't use a space, hyphen or period.
- Prism does not distinguish between upper and lower case letters in variable, parameter or function names.
- Use an asterisk (*) to indicate multiplication. Prism does not always recognize implied multiplication. To multiply A times B, enter “A*B” and not “AB”.
- Use a caret (^) to indicate power. For example, "A^B" is A to the B power.
- Use parentheses as necessary to show the order of operations. To increase readability, substitute brackets [like this] or braces {like this}. Prism interprets parentheses, brackets, and braces identically. Don't make any assumptions about the order of precedence. Include

enough parentheses so there is no ambiguity about how the equation is evaluated.

- Use a single equals sign to assign a value to a variable.
- You don't need any special punctuation at the end of a statement.
- To enter a long line, type a backslash (\) at the end of the first line, then press Return and continue. Prism treats the two lines as one.
- To enter a comment, type a semicolon (;) and then the text. Comments can begin anywhere on a line.
- You can use many functions 738, most which are similar to those built-in to Excel.
- Be careful not to use the name of a built-in function as a parameter name. For example, since *beta* is the name of a function, you cannot name a parameter *beta*.

IF-THEN relationships

Prism allows you to introduce some branching logic through use of the IF function. The syntax is:

IF (conditional expression, value if true, value if false).

You can precede a conditional expression with NOT, and can connect two conditional expressions with AND or OR. Examples of conditional expressions:

```
Y>100
Ymax=Constraint
(A<B or A<C)
NOT(A<B AND A<C)
FRACTION<>1.0
X<=A and X>=B
```

Prism's syntax is that of most computer languages: "<>" means not equal to, "<=" means less than or equal to, and ">=" means greater than or equal to.

Here is an example.

```
  Y= If (X<X0, Plateau, Plateau*exp(-K*X))
```

In this example, if X is less than X0, then Y is set equal to Plateau. Otherwise Y is computed as Plateau*exp(-K*X). This approach is useful for segmental regression 744.

You may also insert a conditional expression anywhere in an equation, apart from an If function. A conditional expression evaluates as 1.0 if true and 0.0 if false. Example:

```
   Y=(X<4)*1 + (X>=4)*10
```

When X is less than 4, this evaluates to 1*1 + 0*10=1. When X is greater than 4, this evaluates to 0*1+1*10=10.

4.5.6.8.2 Multiline models

Equations can be written over several lines. Here is an example, the mixed model enzyme inhibition model built-in to Prism:

```
  VmaxApp=Vmax/(1+I/(Alpha*Ki))
  KmApp=Km*(1+I/Ki)/(1+I/(Alpha*Ki))
  Y=VmaxApp*X/(KmApp + X)
```

Prism follows the convention of all computer languages. It starts at the top and goes down.

First it computes the intermediate variable VmaxApp. It knows this is an intermediate variable, and not a parameter to fit, because it appears to the left of the equals sign.

Next Prism computes the value of KmApp.

Finally it uses those two values to compute Y.

Math textbooks tend to write equations in the opposite order. A math text might first define Y as a function of VmaxApp and KmApp, and then lower on the page define how to calculate VmaxApp and KmApp from

Vmax, Km, Alpha and I. Prism (like all computer languages) requires that you define an intermediate variable before you use it.

Here is a second example:

```
Specific=X*Bmax/(X+Kd)
Nonspecific=NS*X
<A>Y=Specific + Nonspecific
<B>Y=Nonspecific
```

The first line calculates the intermediate variable Specific. The second line defines the intermediate variable Nonspecific.

The third line is preceded by <A>. This means that this line only applies to data set A. The second line is preceded by <B> so only applies to data set B. This allows the model to fit a table of data where column A is the total binding and column B is the nonspecific binding. Read more about the syntax used to specify that a particular line only applies only to selected data sets 744.

You can define constants right in the multiline equation. This makes sense for defining true constants, whose value will never change. If the constant is something like a concentration that changes from experiment to experiment, it is better to not define it in the equation itself, but rather define it in the Constrain tab 336. If it first appears on the right side of the equation, Prism will treat it like a parameter. You can use the Constrain tab to fix that parameter to a constant value. If the variable name appears first to the left of the equals sign, it is used only within the equation and won't appear in the constraints tab. For example, this line defines Pi:

```
Pi=3.141529
```

4.5.6.8.3 Limitations when entering equations

Functions must be Y=f(X)

When you enter an equation into Prism, the independent variable must be 'X' and the dependent variable must be 'Y'. So if you measure a voltage as a function of time, you cannot enter an equation that defines **V** as a function of **t**. It must define **Y** as a function of **X**.

- Prism can fit an implicit equation 736 where Y is on both sides of the equation.

- Prism can fit a differential equation 735 that defines dY/dX.

- Prism cannot fit a model defined by a set of differential equations. For this reason, it cannot fit many compartmental models.

No models with more than one X variable

Prism does not calculate multiple nonlinear regression, so cannot fit models with two or more independent (X) variables. But note that you can define a parameter to be a column constant, in which case its value comes from the column titles. In some cases, you can think of these column constants as being a second independent variable.

Ways to evade this limitation.

Model complexity

Prism compiles your equation into an internal format it uses to calculate the math efficiently. If the compiled version of your equation won't fit in the space Prism sets aside for this purpose, it reports that the equation is "too complex" .

If you see this message, don't give up. You can usually rewrite an equation to make it less complex. Do this by defining an intermediate variable that defines combinations of variables. For example if your equation uses the term "K1+K2" four times, you reduce complexity (but keep exactly the same mathematical meaning) by defining an intermediate variable at the top of your equation (say, K1P2=K1+K2) and then using that intermediate later in the equation. That way Prism has fewer steps to store.

No loops or sums

Prism does not have any syntax to allow for summations or loops in user-defined equations.

If the loop or sum occurs a small and consistent number of times, a workaround would be to write the equation several times, once for n=1, then for n=2 ... and add up the results to compute the final value of Y.

An example. Exponential decay

A single phase exponential decay is defined by this equation.

Y=Y0*exp(-K*X)

Simple calculus shows that this has the property that the derivative of Y is proportional to the value of Y.

dY/dX = -K*Y

How to fit a differential equation with Prism.

Prism can fit a model defined by a differential equation.

Choose the differential equation type at the top of the Equation dialog, and define Y' (the derivative of Y with respect to X) as a function of X and parameters. For the example, enter:

Y' = -K*Y

That's it. Putting an apostrophe after the Y on the left side of the equation tells Prism that you are defining the derivative of Y with respect to X.

Notes

- Prism does not understand the other nomenclature for differential equations. Don't try to define an equation that starts with "dY/dX = ".
- Note that X doesn't actually appear in the equation. That's ok. It is there in spirit, since Y' defines the derivative of Y with respect to X.
- When you look at that equation, there appears to be only one parameter, K. In fact, the equation has two parameters. Prism generates a parameter Y[0], which is the value of Y at X=X0.
- When you go to add constraints and initial values, Y[0] appears just like the other parameter K.

- You can set X0 to any constant value you want, but it is usually set to 0.0. If you want to choose a different value, set it on the bottom of the initial values tab of the dialog that defines the equation. Note this is the dialog that defines the equation, not the dialog used for each fit.

- Fitting a differential equation requires more calculations, so it takes noticeably more time that fitting the usual kind of equation.

- It is only possible to define Y'. It is not possible to use differential equations to define intermediate variables. This would be useful for fitting compartmental models, but Prism cannot (yet) fit this kind of model.

- With models defined as differential equations, Prism 6 did not allow you do define different models for different data sets using the <A> ..<B>.. notation. Prism 7 does allow this. But note that in every case, you must define Y'. It is not possible to define Y' for some data sets and define Y for others.

- Prism 6 always fit the Y[0] value and shared its value for all data sets. Prism 7 lets you set it to a constant value, and lets you share it among data sets or not.

4.5.6.8.5 Entering an implicit equation

An implicit equation is one where Y appears on both sides of the equals sign. Prism will (try to) fit models defined by implicit equations to your data.

Example of an implicit equation

The model for total binding at equilibrium to a binding site that follows the law of mass action is

```
Y = NS*X + Bmax*X/(Kd + X)
```

That form of the model assumes that X is the free concentration of ligand. Since X is actually the concentration you entered, this model assumes that a tiny fraction of the ligand binds so even though X is concentration you applied it also is very close to the free concentration.

What if a substantial fraction of the ligand binds? A much more complicated model is built into Prism to handle this situation. But in other similar situations, the model may not have been derived and it may be very difficult or even impossible to derive one. Implicit equations to the rescue.

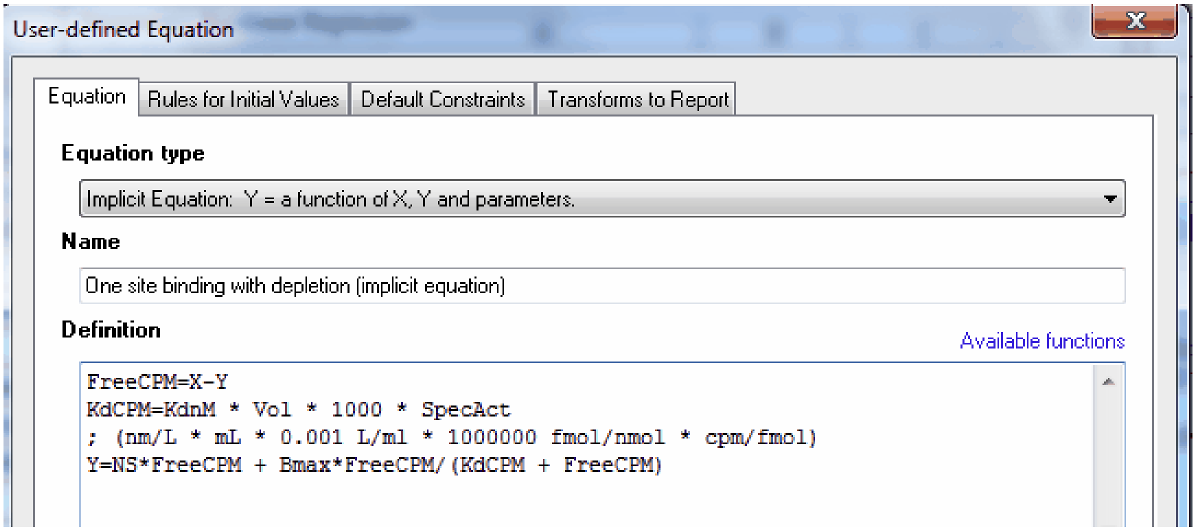
In the situation above, it is easy to write an implicit equation. There are three places in the model where X appears. The X values you enter are the total concentration of ligand. The value that enters into the model must be the free concentration, which is the total concentration minus the concentration that bound. Assuming X and Y are both in the same units, we simply replace X with X-Y in all three places:

```
Y = NS*(X-Y) + Bmax*(X-Y)/(Kd + (X-Y))
```

When you actually go to fit the data, you'll probably want to change the equation a bit (as shown below) to handle unit conversions so the Kd is reported in nM rather than cpm.

How to fit implicit equations with Prism

Choose the implicit equation type at the top of the Equation dialog, and define the equation with Y on both sides of the equals sign.



Equation type

Implicit Equation: Y = a function of X, Y and parameters.

Name

One site binding with depletion (implicit equation)

Definition

```
FreeCPM=X-Y
KdCPM=KdnM * Vol * 1000 * SpecAct
; (nm/L * mL * 0.001 L/ml * 1000000 fmol/nmol * cpm/fmol)
Y=NS*FreeCPM + Bmax*FreeCPM/(KdCPM + FreeCPM)
```

That's it. You'll need to define constraints and initial values as with any user-defined equation.

Notes

- In this example you don't actually see Y on both sides of the equals sign in the same equation line. But the first line puts Y on the right side of the equations sign, and the fourth line puts it on the right side. That makes the equation implicit.
- If you subtract Y from X, as in the example here, then X and Y must both be entered in the same units. Here both are entered as radioactivity counts per minute (cpm). Of course, it would make no sense to subtract Y from X if Y were in cpm and X were in nM.
- Prism finds it "harder" to fit implicit equations than ordinary ones. You may have to fuss with initial values and constraints to get it to work. The calculations take much longer, although this may not be noticeable with small data sets and fast computers.
- While Y appears on both sides of the equals sign in an explicit equation, X must appear only on the right side of the equals sign.
- In this particular example, the explicit equation has been derived and is even built into Prism. The file you can download (link below) fits the data both ways (explicit equation and implicit equation) and the results are identical. In other cases, it may be difficult or impossible to derive an explicit equation.

Example Prism file.

4.5.6.8.6 Available functions for user-defined equations

Allowed syntax 730

There are Excel functions with identical names (except Normal distribution functions, which are normsdist and normsinv) and behavior.

Function	Explanation	Excel equivalent
abs(k)	Absolute value.	abs(k)
arccos(k)	Arccosine. Result is in radians.	acos(k)
arccosh(k)	Hyperbolic arc cosine.	acosh(k)

Function	Explanation	Excel equivalent
arcsin(k)	Arcsine. Result is in radians.	asin(k)
arcsinh(k)	Hyperbolic arcsin. Result in radians.	asinh(k)
arctan(k)	Arctangent. Result is in radians.	atan(k)
arctanh(k)	Hyperbolic tangent. K is in radians.	atanh(k)
arctan2(x,y)	Arctangent of y/x. Result is in radians.	atan2(x,y)
besselj(n,x)	Integer Order J Bessel, N=0,±1, ±2…	besselj(x,n)
bessely(n,x)	Integer Order Y Bessel, N=0,±1, ±2…	bessely(x,n)
besseli(n,x)	Integer Order I Modified Bessel, N=0,±1, ±2…	besseli(x,n)
besselk(n,x)	Integer Order K Modified Bessel, N=0,±1, ±2…	besselk(x,n)
beta(j,k)	Beta function.	exp(gammaln(j) +gammaln(k) - gammaln(j+k))
binomial(k,n,p)	Binomial. Probability of k or more “successes” in n trials, when each trial has a probability p of “success”.	1 - binomdist(k,n,p,true) + binomdist(k,n,p,false)
chidist(x2,v)	P value for chi square equals x2 with v degrees of freedom.	chidist(x2,v)
chiinv(p,v)	Chi-square value for specified P value with v degrees of freedom.	chiinv(p,v)

Function	Explanation	Excel equivalent
ceil(k)	Nearest integer not smaller than k. Ceil (2.5)=3.0. Ceil(-2.5)=-2.0	(no equivalent)
cos(k)	Cosine. K is in radians.	cos(k)
cosh(k)	Hyperbolic cosine. K is in radians.	cosh(k)
deg(k)	Converts k radians to degrees.	degrees(k)
erf(k)	Error function.	2*normsdist(k*sqrt(2))-1
erfc(k)	Error function, complement.	2-2*normsdist(k*sqrt(2))
exp(k)	e to the kth power.	exp(k)
floor(k)	Next integer below k. Floor(2.5)=2.0. Floor(-2.5)=-3.0.	(no equivalent)
fdist(f,v1,v2)	P value for F distribution with v1 degrees of freedom in the numerator and v2 in the denominator.	fdist(f,v1,v2)
finv(p,v1,v2)	F ratio corresponding to P value p with v1 and v2 degrees of freedom.	finv(p,v1,v2)
gamma(k)	Gamma function.	exp(gammaln(k))
gammaln(k)	Natural log of gamma function.	gammaln(k)
hypgeometricm(a,b,x)	Hypergeometric M.	(no equivalent)
hypgeometricu(a,b,x)	Hypergeometric U.	(no equivalent)

Function	Explanation	Excel equivalent
hypgeometricf (a,b,c,x)	Hypergeometric F.	(no equivalent)
ibeta(j,k,m)	Incomplete beta.	(no equivalent)
if(condition, j, k)	If the condition is true, then the result is j. Otherwise the result is k. See details 730.	(similar in excel)
igamma(j,k)	Incomplete gamma.	gammadist(k, j, 1,TRUE)
igammac(j,k)	Incomplete gamma, complement.	1 - gammadist(k, j, 1,TRUE)
int(k)	Truncate fraction. INT(3.5)=3 INT(-2.3) = -2	trunc()
ln(k)	Natural logarithm.	ln(k)
log(k)	Log base 10.	log10(k)
max(j,k)	Maximum of two values.	max(j,k)
min(j,k)	Minimum of two values.	min(j,k)
j mod k	The remainder (modulus) after dividing j by k.	mod(j,k)
psi(k)	Psi (digamma) function. Derivative of the gamma function.	(no equivalent)
rad(k)	Converts k degrees to radians.	radians(k)
round(k,j)	Round the number k to show j digits after the decimal.	round(k,j)

Function	Explanation	Excel equivalent
sgn(k)	Sign of k. If k>0, sgn(k)=1. If k<0, sgn(k)= -1. If k=0, sgn(k)=0.	sign(k)
sin(k)	Sine. K is in radians.	sin(k)
sinh(k)	Hyperbolic sine. K is in radians.	sinh(k)
sqr(k)	Square.	k*k
sqrt(k)	Square root.	sqrt(k)
tan(k)	Tangent. K is in radians.	tan(k)
tanh(k)	Hyperbolic tangent. K is n radians.	tanh(k)

tdist(t,v)	P value (one-tailed) corresponding to specified value of t with v degrees of freedom. T distribution.	tdist(t,v,1)
		t.dist(t,v,true)

tinv(p,v)	t ratio corresponding to two-tail P value p with v degrees of freedom.	tinv(p,v)

zdist(z)	P value (one-tailed) corresponding to specified value of z. Gaussian distribution.	normsdist(z)
		norm.s.dist(z,true)

zinv(p)	Z ratio corresponding to one-tail P value	normsinv

4.5.6.8.7 Fitting different segments of the data to different models

What is segmental regression?

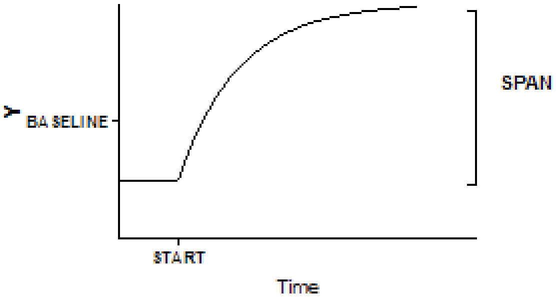
In some situations you may wish to fit different models to different portions of your data. This is called **segmental regression**. This often occurs in kinetic experiments where you add a drug or perform some sort of intervention while recording data. The values collected before the intervention follow a different model than those collected afterwards.

If you didn't perform an intervention at a particular time along the X axis, it is unlikely that this kind of segmental regression is the analysis of choice.

How to perform segmental regression with Prism

Although Prism has no built-in way to fit different equations to different portions of the data, you can achieve that effect using a user-defined equation containing the IF function.

In this example, you collected data that established a baseline early in the experiment, up to "Start". You then added a drug, and followed the outcome (Y) as it increased towards a plateau. Prior to the injection, the data followed a horizontal line; after the injection the data formed an exponential association curve.



```
Y1=BASELINE
Y2=BASELINE + SPAN*(1-exp(-K*(X-START)))
Y=IF[(X<START),Y1,Y2)]
```

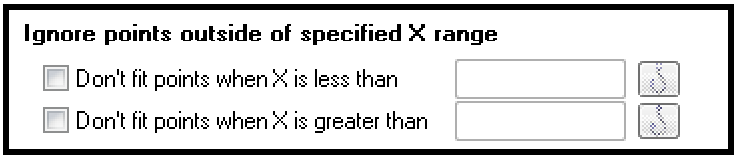
It is easiest to understand this equation by reading the bottom line first. For X values less than START, Y equals Y1, which is the baseline. Otherwise, Y equals Y2, which is defined by the exponential association equation.

This equation has two intermediate variables (Y1 and Y2). Prism can fit the four true variables: START, SPAN, K, and BASELINE.

In many cases, you will make START a constant equal to the time of the experimental intervention. If you want Prism to fit START, choose an initial value carefully.

This kind of model is most appropriate when X is time, and something happens at a particular time point to change the model. In the example above, a drug was injected at time=Start.

How to fit a model to only a portion of the data



Ignore points outside of specified X range

□ Don't fit points when X is less than

□ Don't fit points when X is greater than

The Range tab of the nonlinear regression dialog lets you define an X range that determines which points are fit and which are ignored.

4.5.6.8.8 Fitting different models to different data sets

Writing a model so some lines only apply to selected data sets

Prism can only fit data from one table at a time, and all the data sets must be fit by a single equation you select or create. But you can create this equation in such a way that different data sets are fit to mathematically distinct equations.

When entering a user defined equation, you use multiple lines to define intermediate variables. Each line can have a prefix that specifies which data sets that line will be used for:

Prefix	**This line applies to**
<C>	Data set C only
<~B>	All data sets, except B.
<A:D>	Data sets A-D.
<~A:D>	All data sets except A-D.
<A:J,3>	Data sets A,D,G, and J (every third data set between A and J.)
<~A:J,3>	All data sets except A,D,G, and J.

Here is an example. It fits column A to a model that defines total binding and column B to a model that defines nonspecific binding only. The first two lines of the equation are evaluated for all data sets, the third line is only evaluated for data set A, while the last line is only evaluated for data set B. To fit this model, you would want to set the constraint that the parameter NS is shared between data sets.

```
Specific=X*Bmax/(X+Kd)
Nonspecific=NS*X
<A>Y=Specific + Nonspecific
<B>Y=Nonspecific
```

Ambiguity when you select noncontinuous range of data sets to analyze

```
<A>Y=1/(1+Ka*X^h)
<C>Y=(Ka*X^h)/(1+Ka*X^h)
```

The equation above seems clear at first. The first line is for data set A, and the second line is one for data set C. But what if you asked Prism to only analyze data sets A and C, skipping B? Now it is a bit ambiguous. Does <C> mean data set C, or does it mean the third data set included in the analysis? Prism uses the second definition. So you need to enter the equation like this.

```
<A>Y=1/(1+Ka*X^h)
<B>Y=(Ka*X^h)/(1+Ka*X^h)
```

Now the second line is for the second data set, which in this example is data set C (since only A and C were selected in the Analyze dialog).

4.5.6.8.9 Different constants for different data sets

The Constrain tab lets you define a parameter to be a constant value, but one value for all data sets. What if you want a different constant value for each data set? Prism offers two ways to do this:

- Use the column title as the constant value 746.

- Use special notation within a user-defined equation to assign a different value for each data set. For example:

```
<A>Bottom=4.5
<B>Bottom=34.5
<C>Bottom=45.6
Y=Bottom + span*(1-exp(-1*K*X))
```

In this example, the parameter Bottom is set to 4.5 when fitting data set A, to 34.5 when fitting data set B, and to 45.6 when fitting data set C.

4.5.6.8.10 Column constants

What is a column constant?

When you fit a number of datasets at once, you can use the column title as a second independent variable. We call this constraining a parameter to be a column constant. This is best seen by example.

How to enter column constants

To see how column constants work, create a new XY table using the sample data file: Enzyme kinetics - Competitive inhibition.

The data table has one X column, and four Y columns, each representing a different concentration of inhibitor. The inhibitor concentrations are entered as column title.

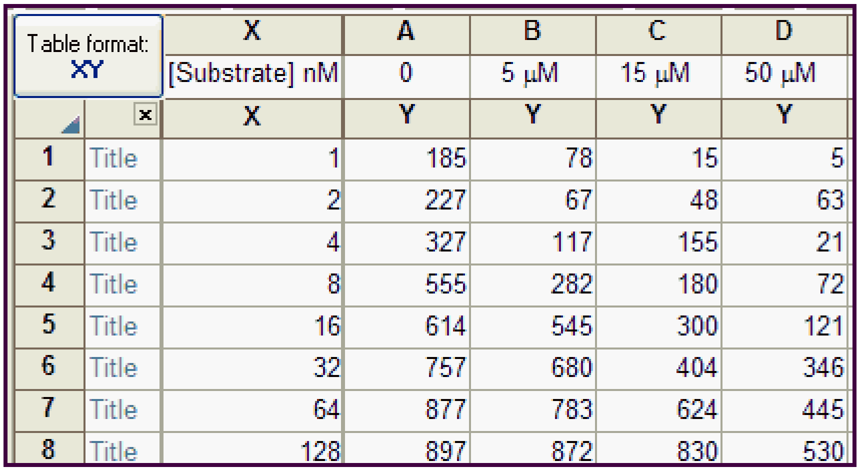
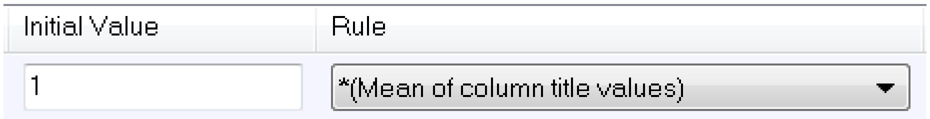


Table format: XY		X	A	B	C	D
		[Substrate] nM	0	5 µM	15 µM	50 µM
	☒	X	Y	Y	Y	Y
1	Title	1	185	78	15	5
2	Title	2	227	67	48	63
3	Title	4	327	117	155	21
4	Title	8	555	282	180	72
5	Title	16	614	545	300	121
6	Title	32	757	680	404	346
7	Title	64	877	783	624	445
8	Title	128	897	872	830	530

Note that Prism reads only the number in the column title. In this example, the units are specified as micromolar, but Prism ignores this and simply reads the numbers. It does not do any unit conversion.

Specifying a column constant when fitting data

When choosing an initial value, choose from the drop down list either "Mean of column title value" or "log(Mean of column title value".



Initial Value	Rule
1	*(Mean of column title values)

Example

To fit the sample data above, click Analyze, choose nonlinear regression, choose the Enzyme Kinetics panel of equations and choose Competitive enzyme kinetics. The equation is built in, but if you click the Details button you can see the math.

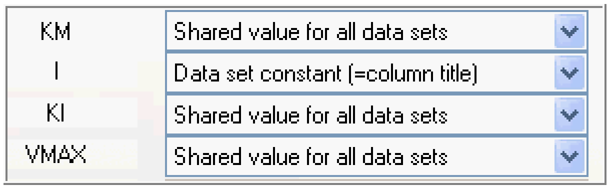
```
KmObs=Km(1+[I]/Ki)
Y=Vmax*X/(KmObs+X)
```

The first line defines an intermediate variable (KmObs, the observed Michaelis-Menten constant in the presence of a competitive inhibitor), which is a function of the Michaelis-Menten constant of the enzyme (Km), the concentration of inhibitor (I), and the competitive inhibition constant (Ki).

The second line computes enzyme velocity (Y) as a function of substrate concentration (X) and KMapp.

This model is defined with I constrained to being a data set constant, which means its value comes from the column titles. In this example, therefore, I=0 when fitting column A, I=5 when fitting column B, etc. The 'µM' in the title is ignored by Prism -- it doesn't do any unit conversions.

The other three parameters (Km, Ki and Vmax) are defined to be shared, so Prism fits one best-fit value that applies to the entire family of datasets.



KM	Shared value for all data sets
I	Data set constant (=column title)
KI	Shared value for all data sets
VMAX	Shared value for all data sets

Prism determined the maximum velocity of the enzyme with no inhibitor (Vmax in the same units as the Y values you entered), the Michaelis-Menten constant of the enzyme with no inhibitor (Km, in the units used for X values) and the competitive inhibition constant (Ki, in units used for the column constants). Note that I is not a parameter to be fit, but rather takes on constant values you entered into the column titles. KmObs is not a parameter to be fit, but is rather an intermediate variable used to define the model.

Learn more about competitive enzyme inhibition 622.

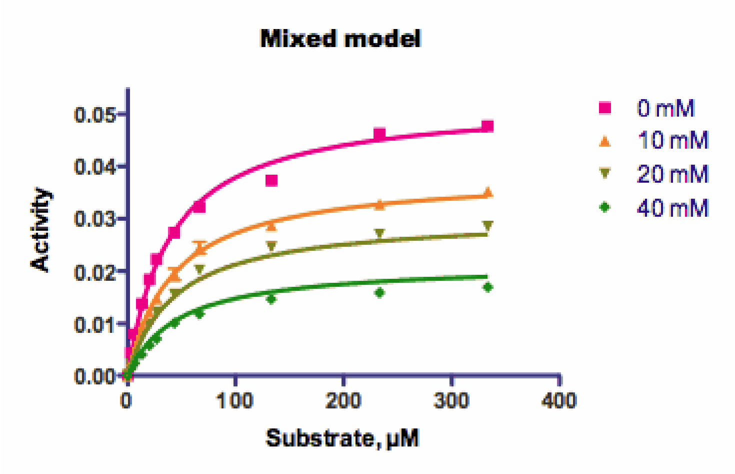
Summary. The advantage of column constants

By using column constants and global fitting (shared parameters), this example determined a parameter (Ki) whose value cannot be determined from any one dataset, but can only be determined by examining the relationships between datasets.

4.5.6.8.11 Defining equation with two (or more) independent variables

Prism is designed to perform nonlinear regression with one independent (X) variable. But, with a bit of cleverness, it is possible to also fit data with two independent variables. There are three separate cases:

Fitting a family of curves



The graph above shows the sample data for competitive enzyme kinetics, and shows how to fit a family of curves. Here, each curve shows enzyme activity as a function of substrate concentration. The curves differ by the presence of various amounts of an inhibitor. There are, essentially, two independent variables: substrate concentration, and inhibitor concentration.

The substrate concentration is entered into the X column and the inhibitor concentration into the column titles.

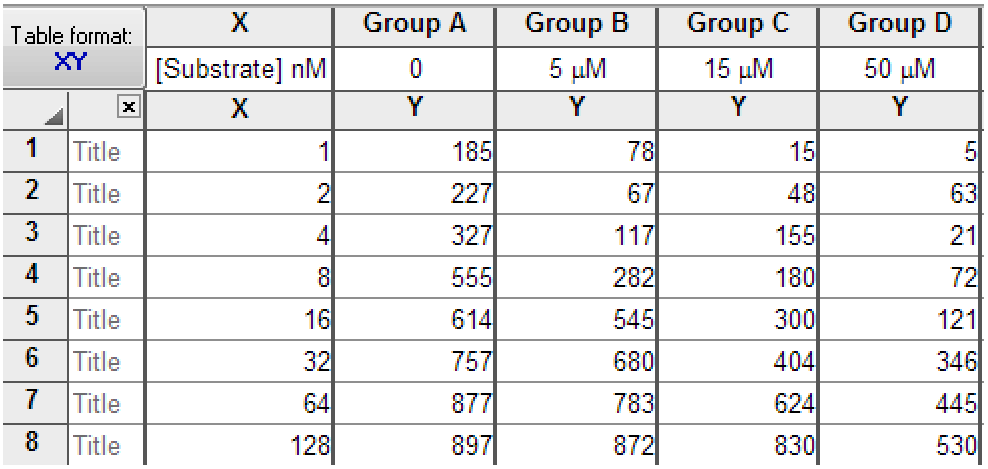
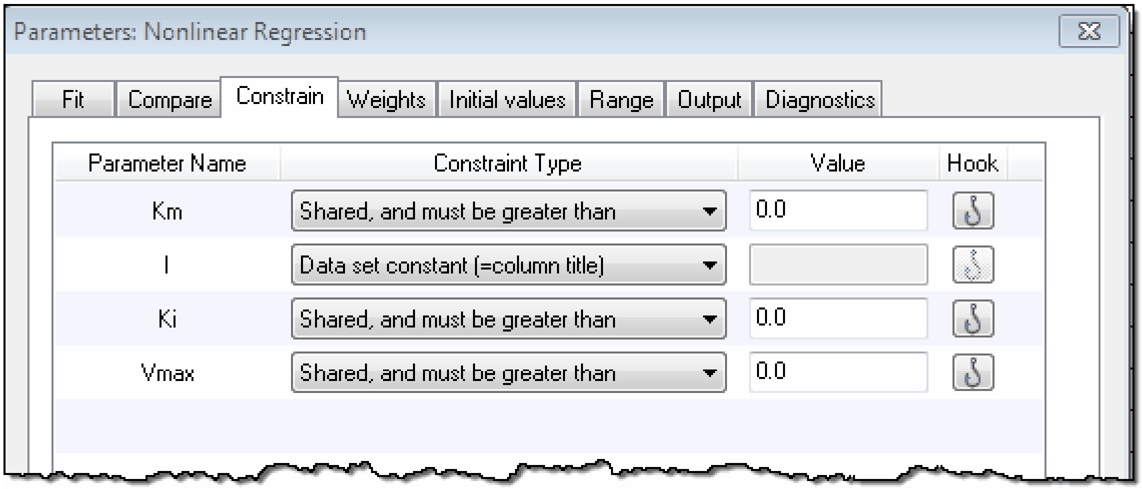


Table format: XY		X	Group A	Group B	Group C	Group D
		[Substrate] nM	0	5 µM	15 µM	50 µM
		X	Y	Y	Y	Y
1	Title	1	185	78	15	5
2	Title	2	227	67	48	63
3	Title	4	327	117	155	21
4	Title	8	555	282	180	72
5	Title	16	614	545	300	121
6	Title	32	757	680	404	346
7	Title	64	877	783	624	445
8	Title	128	897	872	830	530

When fitting the data, the Constrain tab of the nonlinear regression dialog is used to define the parameter I as dataset constant whose value comes from the column titles (so is not fit by regression), and to share the values of all the other parameters so there is one global fit of all the data.



Parameter Name	Constraint Type	Value	Hook
Km	Shared, and must be greater than	0.0	
I	Data set constant (=column title)		
Ki	Shared, and must be greater than	0.0	
Vmax	Shared, and must be greater than	0.0	

Fitting data where each value has two independent variables

But what if each value you collect has two independent variables? You don't have a family of curves. Rather you have one set of Y values, each associated with two X variables. Follow these steps to enter the data:

1. Create an XY data table, with whatever form of subcolumns fits your data.

2. Enter the first Y value in row 1 of column A.

3. Enter the first independent variable corresponding to that Y into first row of the X column.

4. Enter the other independent variable for that first Y value as the column title of column A.

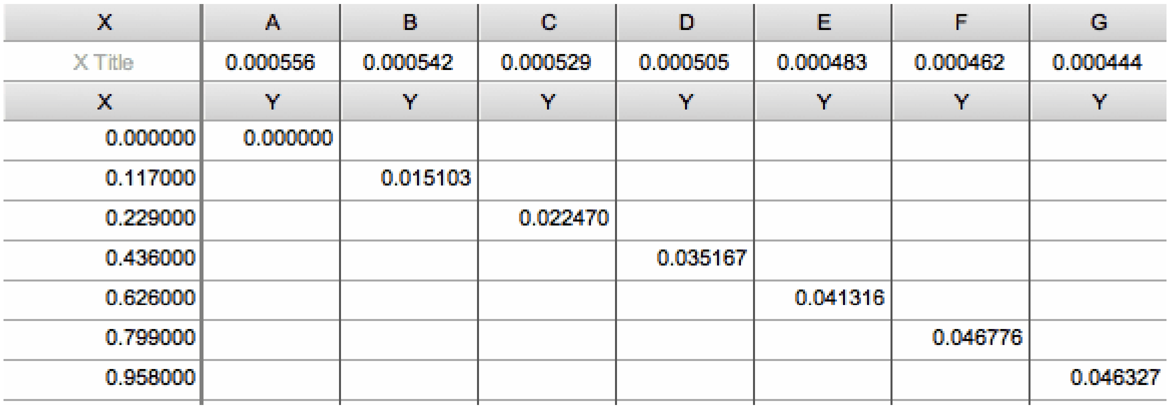
5. Enter the second Y value in row 2 of column B.

6. Enter the first independent variable for that Y into the second row of the X column.

7. Enter the other independent variable for that Y as the column title of column B.

8. Continue with the rest of the data. You'll be entering data diagonally down the table. Since Prism can only have 104 columns, you will be limited to 104 data points, each with a Y value and two X values.

9. When you are done, you'll have the same number of rows as Y columns.



X	A	B	C	D	E	F	G
X Title	0.000556	0.000542	0.000529	0.000505	0.000483	0.000462	0.000444
X	Y	Y	Y	Y	Y	Y	Y
0.000000	0.000000						
0.117000		0.015103					
0.229000			0.022470				
0.436000				0.035167			
0.626000					0.041316		
0.799000						0.046776	
0.958000							0.046327

When you fit the data, keep in mind the following:

- "X" refers to the values you enter into the X column. Use some other name ("X2" or whatever you want) for the other independent variable.

- In the constraints tab, define that parameter to be a data set constant equal to the column title. You don't want Prism to fit that parameter, but rather to get its value from the column titles.

- For all other parameters in the model, use the Constraint tab to share the value among data sets. Prism sees each column as a data set. In this data table, each data set has only one value so Prism can't possibly fit anything unless all the parameters are shared.

Prism will report the nonlinear regression results clearly. All the results you want will be in the final "global" column. Prism cannot graph the results since each data set has only one point, and Prism cannot plot 3D graphs.

Three independent variables

Prism is not really designed to fit models with three independent variables, but you can get it to do so in some circumstances.

Enter your data as above, with one independent variable as X and the second as column titles. Make sure each column of data has only one value for the third independent variable. Then define that third variable in a user-defined equation. Let's imagine that the third variable has the name Z. Add syntax like this to your equation:

```
<A> Z=2.34
<B> Z=45.34
<C>Z= -23.4
<D>Z=12.45
```

That tells Prism to assign Z one value for column A, a different value for column B, etc. This approach gives you three independent variables, with some constraints:

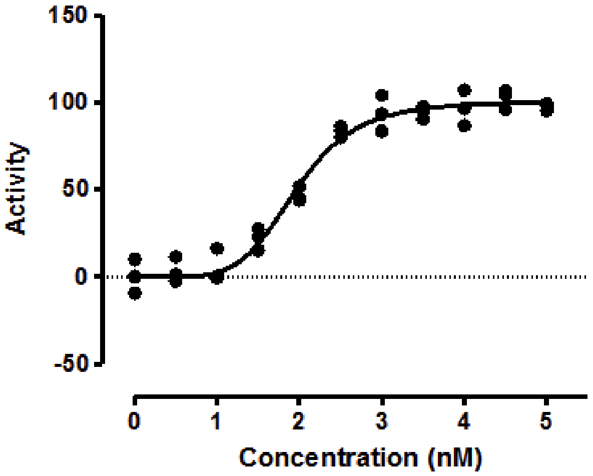
- The X column is one independent variable, with one value for each row.

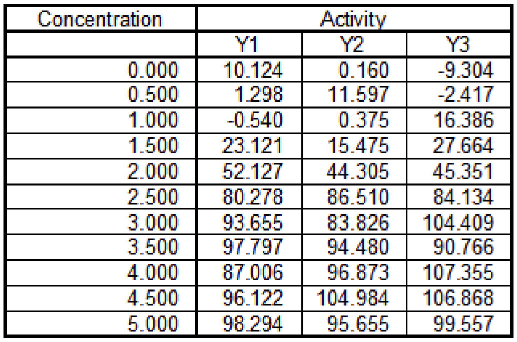
- The column titles form a second independent variable with one value per data set column.

- The Z values defined within the equation for a third independent variable, also with one value per data set.

4.5.6.8.12 Reparameterizing an equation

Reparameterizing an equation does not change the best-fit curve





Concentration	Activity		
	Y1	Y2	Y3
0.000	10.124	0.160	-9.304
0.500	1.298	11.597	-2.417
1.000	-0.540	0.375	16.386
1.500	23.121	15.475	27.664
2.000	52.127	44.305	45.351
2.500	80.278	86.510	84.134
3.000	93.655	83.826	104.409
3.500	97.797	94.480	90.766
4.000	87.006	96.873	107.355
4.500	96.122	104.984	106.868
5.000	98.294	95.655	99.557

You want to fit the sigmoidal enzyme kinetics data to a standard model. But there are two forms of that model that are commonly used:

```
Y=Vmax*X^h/(Khalf^h + X^h)
Y=Vmax*X^h/(Kprime + X^h)
```

The two are equivalent, with Kprime equal to Khalf^h , so the two fits will generate exactly the same curve, with the same sum-of-squares, the same R^2, and the same numbers of degrees of freedom. Even though the two equations express the same model, they are written differently. The fancy term is that they are *parameterized* differently.

They both fit Vmax (the maximum activity extrapolated to very high concentrations of substrate) and h (Hill slope, describing the steepness of the curve). But one model fits Khalf (the concentration needed to obtain a velocity half of maximal) and the other fits Kprime (a more abstract measure of substrate action).

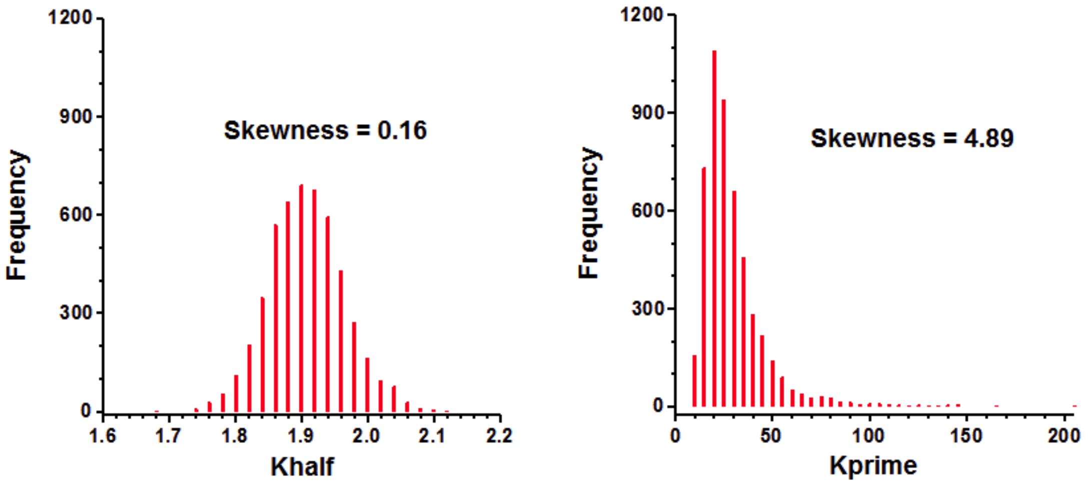
Which model is preferred?

One way to choose between the two models is to match other text books and papers, so your results can easily be compared to others. Another approach is to choose the form that fits the way you think. For this example, if you prefer to think graphically, choose the Khalf. If you think mechanistically, choose Kprime.

But the choice can be more than a matter of convenience and convention. The choice of model can determine the accuracy of the confidence intervals. Read on to see why.

Distribution of parameters are not always symmetrical

Simulations can determine parameter symmetry. I simulated sigmoidal enzyme kinetics using Vmax=100, h=5, Kprime=25, and Gaussian scatter with a SD equal to 7.5. The X values matched those in the figure above, with triplicate Y values at each X. Prism can do such simulations easily. Using Prism's Monte Carlo analysis, I repeated the simulations 5000 times, fit each curve to both forms of the models, and tabulated the best-fit values of Kprime and Khalf, and computed the skewness of each.



The distribution of Khalf is quite symmetrical and looks Gaussian. Accordingly, its skewness is close to zero. In contrast, the distribution of Kprime is quite skewed. Note that a few of the simulation data sets had best-fit values of Kprime greater than 100. The skewness value (4.89) confirms what is obvious by inspection -- the distribution is far from symmetrical.

Coverage of the symmetrical asymptotic confidence intervals

Prior versions of GraphPad Prism, like almost all nonlinear regression programs, computes the confidence interval of a best-fit parameters so the interval is symmetrical around the best fit values. If the uncertainty of a parameter is truly symmetrical, then these confidence intervals can be interpreted at face value. If the uncertainty is not symmetrical, then the confidence interval will not be accurate.

Ideally, a confidence interval is easy to interpret. A 95% CI has a 95% chance of including the true value of the parameter, and a 5% chance of missing it. When analyzing real data, we never know the value of the true parameter, so never know if the interval includes it or not. But when you simulate data, you know the true values of the parameters, so can quantify the coverage of the confidence intervals. I set up the same simulations mentioned above, fit each data set to both equations, and tabulated whether each confidence interval included the true parameter value or not. This table shows the fraction of 5,000 simulations where the confidence interval did not include the true parameter value (25 for Kprime, and 1.9037 for Khalf).

	"95% CI"	**"99% CI"**
Ideal	5.0%	1.0%
Kprime	8.8%	4.8%
Khalf	5.1%	1.0%

These results show that Khalf is well behaved, as expected given its symmetry (see above). The 95% confidence interval is expected to miss the true value in 5.0% of the simulations. In fact, it happened 5.1% of the time. Similarly, the 99% CI is expected to miss the true value in 1.0% of the simulations, which is exactly what happened. In contrast, Kprime is less well behaved. The intervals computed to be 95% confidence intervals were not wide enough so missed the true value in 8.8% of the simulations. The 99% intervals were similarly not wide enough so missed the true value in 4.8% of the simulations. Thus the confidence intervals computed to be 99% intervals, actually turned out to be 95% intervals.

These simulations show the advantage of choosing the equation that fits Khalf, rather than the one that fits Kprime. Khalf has a symmetrical distribution so the confidence intervals computed from these fits can be interpreted at face value. In contrast, Kprime has an asymmetrical distribution and its confidence intervals cannot be interpreted at face value.

Coverage of the asymmetrical likelihood confidence intervals

A new feature in Prism 7 is that it can compute asymmetrical likelihood confidence intervals 343. These intervals will have the same coverage no matter how the equation is parameterized. The 95% CI of Khalf failed to include the true value (25) in only 5.3% of 10,000 simulations. When the equation was parameterized to fit Kprime, the 95% CI failed to include the true value (1.9037) in 5.1% of 10,000 simulations.

This demonstrates the advantage of using asymmetrical confidence intervals. You'll get meaningful confidence intervals no matter how you parameterize the equation. You can choose the parameterization that fits parameters that match the way you think about the system or are easy to explain. You don't need to choose a parameterization that leads to a symmetrical distribution of parameters.

Hougaard's skewness

The results above were obtained by running numerous simulations. There is an easier way to figure out how symmetrical a parameter is. Hougaards skewness 399 quantifies the asymmetry of each parameter, computed from the equation, the number of data points, the spacing of the X values, and the values of the parameters.

For the simulated data set, Hougaard's skewness is 0.09 for Khalf and 1.83 for Kprime. A rule of thumb is to expect problems from asymmetry when the absolute value of the Hougaard's skewness is greater than 0.25, and big problems when the value is greater than 1.0. So Hougaard's skewness tells you that the confidence intervals will be accurate when you fit Khalf, but not be so accurate when you fit Kprime.

Note that Hougaard's skewness can be reported as part of the results of nonlinear regression (choose in the Diagnostics tab). No simulations are required.

Bottom line

Models can often be parameterized in multiple ways. You'll get the same curve either way, but choosing an optimum parameterization ensures that the confidence intervals for the parameters are believable. The best way to assess various parameterizations is to ask Prism to report the value of Hougaards 399 measure of skewness for every parameter. Simulations take a bit more work, but let you see how symmetrical a parameter is.

File used for this example Asymmetrical CI file.

4.5.6.9 Managing your list of user-defined equations

Enter topic text here.

4.5.6.9.1 How to: Manage your list of equations

Deleting user-defined equations

Every time you enter an equation, or edit an equation in a file you are working on, Prism adds that equation to your list of user-defined equations. Over time, this list can get too long to easily work with.

When working with the Fit tab of the nonlinear regression dialog, click on any user-defined equations, then click **Delete** or **Delete All.**

Don't hesitate to delete equations. It won't affect any files that use the equation you deleted. If you open one of these files, and change the parameters of the nonlinear regression, Prism will automatically add the equation back to your list. You can also import an equation from any Prism file onto your list of user-defined equations.

Changing the order of equations

Equations appear on the list in the order they were added. To change the order of equations in the list, select an equation and then click **Move up** or **Move down**. You can only change the order of user-defined equations, not the order of built-in equations.

Renaming an equation

The equation name helps you choose it in the future. It also appears on the analysis results. You are not stuck with the name you originally gave it. To **rename** an equation, select the equation, and then click Edit. Change the name and click OK.

Sending an equation to a colleague

The easiest way to send an equation to a colleague is to send a file that uses that equation. Ask your colleague to go to the results page, click the button in the upper left corner of the results table to bring up the Parameters: Nonlinear regression dialog. Then click OK. The equation will be appended to the list of user-defined equations.

4.5.6.9.2 How to add your user-defined equations to the list of built-in equations

Prism makes it easy to add your own equations, and these will appear in the "User defined equations" group of equations. But if you want to share a set of user-defined equations with others in your lab, you can create a new group of equations that will appear underneath the ones we provide.

How to create new Equation files:

1. Start a new Prism project. Go to preferences, file and printer tab, and make sure that you are saving in .pzf format and that the "Save compact" option is not checked.

2. For each equation, either fit a data set with nonlinear regression or simulate a theoretical curve (plot a function). Reorder the results pages to the order you want the equations displayed in the equation list.

3. If a user picks your equation and clicks 'Details' they will see a small preview of the graph that is linked to the equation. Make sure this graph is easy to comprehend at small sizes.

4. When entering each equation, also consider entering a description and tip. The tip appears on the Fit tab of nonlinear regression. The description appears on the Details page.

5. When saving your equation, use a name something like this: "[Smith Lab] Flow cytometry equations", so it is very clear to anyone using Prism that these equations did not come from GraphPad (we won't be able to answer questions about them).

6. Note that each file you create will become a "folder" of equations in Prism. You'll want each file, therefore, to include a set of related equations.

How to make those equations appear with the list of built-in equations:

- Windows: Move (or save) the files to the Equations folder within the Prism program folder.

- Mac: Find the Prism.app file. Right (or control) click and choose Show Package Contents. You'll see the "folders" within the package. Drill down to Contents: SharedSupport:Equations. Move your equation files there. Note that by changing the prism.app file (bundle) you have changed its digital signatures. So you'll need to configure Gatekeeper to let Prism run anyway.

- Both platforms: Please do not delete any equation files that came with Prism. They are "built in" so should always be available (to prevent confusion later).

If Prism is running, quit (exit) it, then restart it. You'll now see your list of equations in the Fit tab of nonlinear regression.

Editing those equations is a bit tricky. Quit Prism, and move the file out of the Equations folder. Now restart Prism, open the file and edit it. When done, save your changes and quit Prism. Finally, move the file back to the Equations folder, and restart Prism.

Don't worry about .pze files

For every "built-in" equation file, Prism creates a .pze file. These files contain, essentially, compiled equations and so let Prism load the equations more quickly. With Windows, you'll find these deep in the Application Data folder. With Mac, they are deep in the Application Support folder and also in the Cache folder in the Equations folder in the Prism bundle.

You don't need to manage these files, and shouldn't try to. Prism will automatically create and update the needed file when necessary.

4.6 Plotting a function

Graphing a family of theoretical curves, often called plotting a function, is useful in three contexts:

- To understand the properties of a function. After looking at the graph, go back and change the parameters to see how the graph changes. Or plot a family of curves, where one parameter varies from curve to curve.
- To create graphs for teaching theory.
- To understand what initial values would make sense when fitting a model to data with nonlinear regression.

4.6.1 How to: Plot a function

Prism offers a *Plot a function* analysis, but it doesn't in fact analyze any data. Rather it generates curves from an equation you choose and parameters you enter.

How to: Plot a function

1. Start from any data table or graph, click Analyze, open the **Generate Curve** folder, and then select **Plot a function**.
2. On the first tab (Function), choose the equation, the starting and ending values of X, and the number of curves you want to plot.
3. On the second tab (Options), choose whether you also want to plot the first derivative, second derivative or integral of the function. The "curve" is actually a set of X and Y coordinates that define a series of points that are connected to form the curve. You can choose the number of line segments that will define the curve. There is little

reason to change the default (150), unltess you want to plot only a portion of the curve on some graphs, in which case you should increase that value.

4. On the third tab (Parameter values), enter the parameter values (or click the fish hook icon to hook analysis or info constants).

Tips for plotting a function

Plotting a family of curves

If you choose to plot more than one curve (a choice on the first tab), the rest of the dialog works a bit differently.

You'll want one parameter in the equation to vary from curve to curve. Don't define this in the equation. Instead, define the values on the third tab.

You'll also want to label the columns of the computed table. There are two ways to do this. Specify on the bottom of the second (Options) tab whether you want to label each curve manually (enter the labels the top of the third tab) or you want each column labeled using the value of one of the parameters. The latter usually makes more sense and is easier.

If you need to do something more complicated, remember you can write the equation so some lines only apply to certain data sets 744. Put <A> in front of a line that only applies to column A, etc.

The top of the third tab lists all the curves you will generate. Select one or more of these curves (or click "select all") and then enter the parameter values below. Often you'll want to first click "select all" and enter most of the parameters. Then click on one curve at a time, and enter the value for the parameter that varies among curves.

Plotting each curve on a separate graph

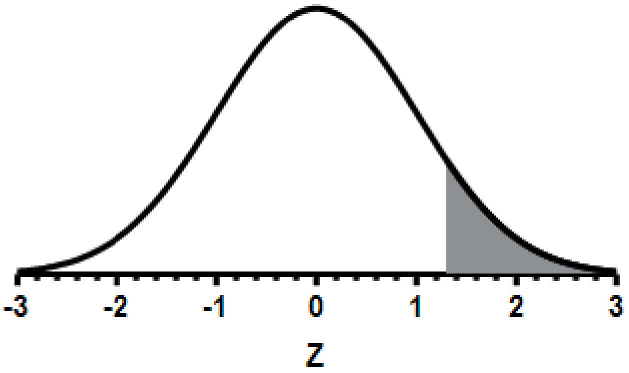
By default, Prism will make graph that contains all of the curves on one graph. If you want each curve to be on its on graph, go to the results table, click New and choose Graph of existing data. On the dialog that pops up, choose to make one graph for each data set (in this context, a data set is a curve).

If you plan to zoom in and plot only part of the curve

The curve is defined, by default, as 150 line segments. That creates a smooth looking curve. But if you then change the range of X values shown on the graph, only a fraction of those line segments will be visible, and the curve may seem coarse. To fix this problem, go back to the parameters dialog, to the Options tab, and increase the number of line segments to a much larger value.

Combining two curves on one graph

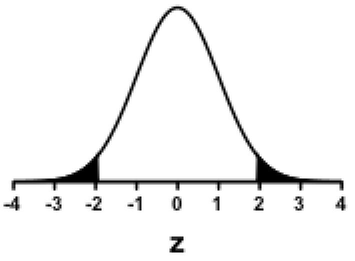
The graph below combines two plotted functions on one graph. The first time I plotted a function, I chose a Gaussian distribution, with the X ranging from -3 to 3. I set the mean to 0.0, the SD to 1.0, and the amplitude to 100.0 (arbitrary, since I hid that axis). I then repeated that analysis, but this time set the X range from 1.3 to 3.0. I put both curves on one graph (Change.. Add data sets -- remember that a curve generated by this analysis is a "data set" to Prism). For the shorter curve, I chose to create an area fill.

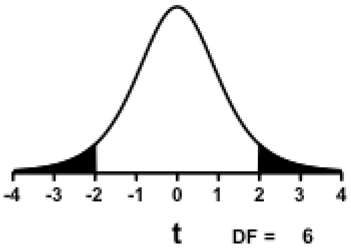


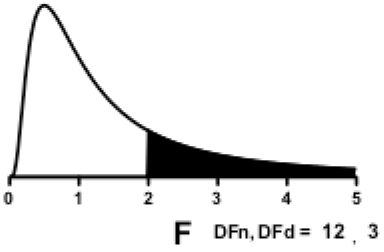
4.6.2 Plotting t, z, F or chi-square distributions

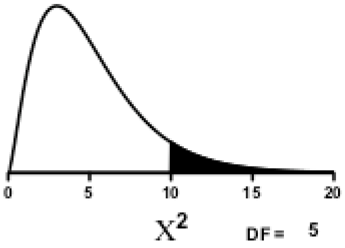
GraphPad Prism can generate probability distributions. This demonstrates Prism's ability to plot functions from user-defined functions, and also the use of hooking info constants to analyses.

Download this Prism file to generate and plot the graphs shown below.









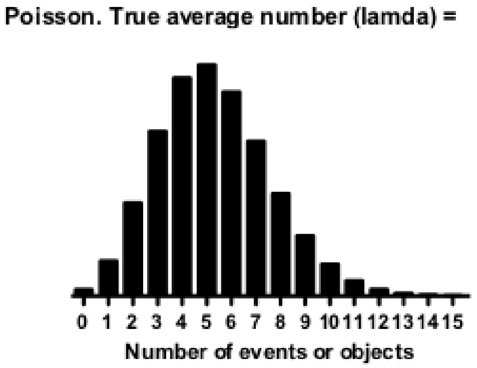
In each case, the simulation generates two (or three) data sets. The first (A) data set plots the entire curve. The equation is written so the second curve (data set B) only plots values where X is less than a specified cutoff value, and the third curve (data set B) only plots values when X is greater than the cutoff value. The second and third data set are plotted with area fill to shade the tails of the distributions. Remove data set B or C from the graph if you only want to shade one tail. For example, here is the equation used for the first graph (z distribution):

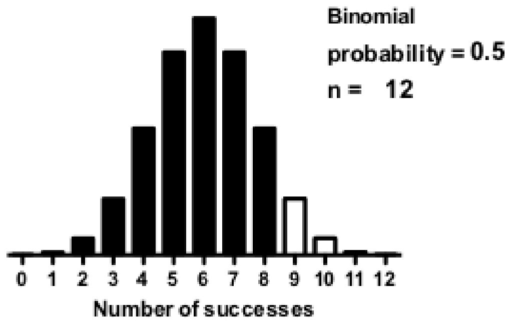
```
G=exp(-0.5*X^2)/sqrt(2*3.1415926)
<A>Y=G
<B>Y=IF(X<-z, G, 0)
<C>Y=IF(X>z, G, 0)
```

Change the numbers of degrees of freedom and the cutoff values (for shading) in the Info sheet. This demonstrates how values entered into an info sheet can be 'hooked' to constants used in analyses.

4.6.3 Plotting a binomial or Poisson distribution

How to plot a binomial or Poisson distribution





Download the Prism file.

To modify this file, change the value of lamda (for Poission) or the probability, n, and cutoff (Binomial) in the Info sheet. Enter new values there, and the graph updates. This is a good example of the usefulness of hooking an info constant to an analysis.

If you want to recreate graphs like these, keep in mind these points:

- As its name suggests, the analysis 'Create a Family of Theoretical Curves' is usually used to create curves, not bar graphs. When you choose the range of X values, specify the appropriate number of 'line segments' (points) so that the X interval equals 1.0. The binomial example on the left created 16 'line segments' starting at X=0 and ending at X=15. The Poisson on the right created 13 'segments' with X starting at 0 and ending at 13.

- The analysis will create a set of line segments (an attempt to create a curve). Click the change type of graph button, or drop the Change menu and choose Graph Type. Then choose the Grouped tab, and then choose interleaved bars.

- The binomial example on the right has two data sets. You don't want them plotted interleaved, as selected in the previous step. Double click to bring up Format Graph, then go to the middle tab, and choose to superimpose the second data set on the first (rather than interleave). And assign it a different color.

Mathematical details

Binomial distribution

The equation for the probability of exactly X successes in N trials, when each trial has probability P of success is:

```
R=INT(X+0.5)
ExactProb=(P^R)*(1-P)^(N-R)  ;exact probability of  successes in n trials
NRearrangments=exp(gammaln(N+1) - gammaln(R+1) - gammaln(N - R +1))
;gamma(J)=(J-1)factorial, or (J-1)! but factorial is not a function within Pri
;NRearrangments = N!/(R! (N-R)!)
H=ExactProb * NRearrangments
<A>Y=H
<B>Y=IF(X>cutoff, H, 0)
```

Poisson distribution

The equation for the Poisson distribution is:

```
Y=exp(-1*Lamda)*Lamda^X/gamma(X+1)
```

Note the definition of the gamma function:

```
gamma(i) = factorial(i-1)
gamma(x+1)= factorial(x) = X!
```

Index