GraphPad

PRISM®

Version 5.0

Regression Guide

Harvey Motulsky
President, GraphPad Software Inc.

This Regression Guide is a companion to GraphPad Prism 5. Available for both Mac and Windows, Prism makes it very easy to graph and analyze scientific data. Download a free demo from www.graphpad.com

The focus of this Guide is on helping you understand the big ideas behind regression so you can choose an analysis and interpret the results. Only about 10% of this Guide is specific to Prism, so you may find it very useful even if you use another regression program.

The companion Statistics Guide explains how to do statistical analyses with Prism,

Both of these Guides contain exactly the same information as the Help system that comes with Prism 5, including the free demo versrion. You may also view the Prism Help on the web at:

http://graphpad.com/help/prism5/prism5help.html

Citation: H.J. Motulsky, Prism 5 Statistics Guide, 2007, GraphPad Software Inc., San Diego CA, www.graphpad.com.

To contact GraphPad Software, email support@graphpad.com or sales@graphpad.com.

Contents

I. Correlation

II. Fitting a curve without a model

III. Generating curves and simulating data

IV. Linear regression

V. Nonlinear regression

Dose-response - Stimulation

Dose-response - Inhibition

Dose-response -- Special

Receptor binding - Key concepts

Receptor binding - Saturation binding

Receptor binding - Competitive binding

Receptor binding - Kinetics

Enzyme kinetics

Exponential

Lines

Polynomial

Gaussian

Sine waves

Classic equations from prior versions of Prism

I. Correlation

Correlation quantifies the association between two variables, without fitting a curve or line.

Key concepts: Correlation

What is correlation?

When two variables vary together, statisticians say that there is a lot of covariation or correlation. The correlation coefficient, r, quantifies the direction and magnitude of correlation.

Correlation is used when you measured both X and Y variables, and is not appropriate if X is a variable you manipulate.

The correlation analysis reports the value of the correlation coefficient. It does not create a regression line. If you want a best-fit line, choose linear regression. 32

Correlation vs. linear regression

Correlation and linear regression 32 are not the same. Consider these differences:

- Correlation quantifies the degree to which two variables are related. Correlation does not find a best-fit line. You simply are computing a correlation coefficient (r) that tells you how much one variable tends to change when the other one does.
- With correlation you don't have to think about cause and effect. You simply quantify how well two variables relate to each other. With regression, you do have to think about cause and effect as the regression line is determined as the best way to predict Y from X.
- With correlation, it doesn't matter which of the two variables you call "X" and which you call "Y". you will get the same correlation coefficient if you swap the two. With linear regression, the decision of which variable you call "X" and which you call "Y" matters a lot, as you will 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.
- Correlation is almost always used when you measure both variables. It rarely is appropriate when one variable is something you experimentally manipulate. With linear regression, the X variable is often something you experimentally manipulate (time, concentration...) and the Y variable is something you measure.

How to: Correlation

Prism can perform correlation analyses either from XY tables or Column tables. The analysis works a bit differently depending on which kind of table you analyze.

Correlation from XY tables

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: Correlation.

If you are entering your own data, choose to enter a single Y value for each point (no replicates, no error values).

2. Enter data

It matters which variable you place in the X column. If you enter Y values for several data sets (column A, B and C), Prism will report correlation results for X vs. YA, for X vs. YB, and for X vs. YC, but not YA vs. YB etc.

3. Analysis choices

Click Analyze and choose Correlation regression from the list of XY analyses.

As explained below, choose whether you want to use a nonparametric test and whether you want to switch to one-tail P values. If not sure, do not choose nonparametric and choose a two-tail P value.

Prism will compute correlation for X vs. YA, for X vs. YB, and for X vs. YC. However, Prism will not report the correlation of YA with YB, or YA with YC, etc.

If the data table contains subcolumns, Prism analyzes only the mean values.

Correlation from Column tables

1. Create a data table

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

If you are just getting started, choose the sample data: Correlation matrix

2. Enter data

It doesn't matter which variable goes in which column. Prism will compute the correlation of each variable (column) with every other variable.

3. Analysis choices

Click Analyze and choose Correlation regression from the list of Column analyses.

Prism gives you two choices:

- Choose one column to be "X" and another to be "Y". Prism then presents the full correlation results and creates a new XY graph of those two variables.

- Choose a correlation matrix. Prism computes the correlation coefficient for each pair of variables, and presents the results on two pages. One page presents the matrix of correlation coefficients and the other page presents the matrix of P values. Each P value is computed independently, with no correction for multiple comparisons. When you compute a correlation matrix, Prism does not create a graph.

Correlation choices

Nonparametric?

Prism offers two ways to compute correlation coefficients:

- **Pearson** correlation calculations are based on the assumption that both X and Y values are sampled from populations that follow a Gaussian distribution, at least approximately. With large samples, this assumption is not too important.

- **Spearman** correlation makes no assumption about the distribution of the values, as the calculations are based on ranks, not the actual values.

One- or two-tailed P values?

Prism can compute either a one-tailed or two-tailed P value. We suggest almost always choosing a two-tailed P value. You should only choose a one-tail P value when you have specified the anticipated sign of the correlation coefficient before collecting any data and are willing to attribute any correlation in the “wrong” direction to chance, no matter how striking that correlation is.

Interpreting results: Correlation

Correlation coefficient

The correlation coefficient, r, ranges from -1 to +1. The nonparametric Spearman correlation coefficient, abbreviated rs, has the same range.

Value of r (or rs)	**Interpretation**
1.0	Perfect correlation
0 to 1	The two variables tend to increase or decrease together.
0.0	The two variables do not vary together at all.
0 to -1	One variable increases as the other decreases.
-1.0	Perfect negative or inverse correlation.

If r or r_s is far from zero, there are four possible explanations:

- Changes in the X variable change the value of the Y variable.
- Changes in the Y variable change the value of the X variable.
- Changes in another variable influence both X and Y.
- X and Y don’t really correlate at all, and you just happened to observe such a strong correlation by chance. The P value quantifies the likelihood that this could occur.

If you choose Spearman nonparametric correlation, Prism computes the confidence interval of the Spearman correlation coefficient by an approximation. According to Zar (Biostatistical Analysis) this approximation should only be used when N>10. So with smaller N, Prism simple does not report the confidence interval of the Spearman correlation coefficient.

R squared

Perhaps the best way to interpret the value of r is to square it to calculate r^2. Statisticians call this quantity the coefficient of determination, but scientists call it "r squared". It is a value that ranges from zero to one, and is the fraction of the variance in the two variables that is “shared”. For example, if r^2=0.59, then 59% of the variance in X can be explained by variation in Y. Likewise, 59% of the variance in Y can be explained by variation in X. More simply, 59% of the variance is shared between X and Y.

Prism only calculates an r^2 value from the Pearson correlation coefficient. It is not appropriate to compute r^2 from the nonparametric Spearman correlation coefficient.

P value

The P value answers this question:

> If there really is no correlation between X and Y overall, what is the chance that random sampling would result in a correlation coefficient as far from zero (or further) as observed in this experiment?

If the P value is small, you can reject the idea that the correlation is due to random sampling.

If the P value is large, the data do not give you any reason to conclude that the correlation is real. This is not the same as saying that there is no correlation at all. You just have no compelling evidence that the correlation is real and not due to chance. Look at the confidence interval for r. It will extend from a negative correlation to a positive correlation. If the entire interval consists of values near zero that you would consider biologically trivial, then you have strong evidence that either there is no correlation in the population or that there is a weak (biologically trivial) association. On the other hand, if the confidence interval contains correlation coefficients that you would consider biologically important, then you couldn't make any strong conclusion from this experiment. To make a strong conclusion, you'll need data from a larger experiment.

If you entered data onto a one-grouping-variable table and requested a correlation matrix, Prism will report a P value for the correlation of each column with every other column. These P values do not include any correction for multiple comparisons.

Analysis checklist. Correlation

✓ Are the subjects independent?

Correlation assumes that any random factor affects only one subject, and not others. You would violate this assumption if you choose half the subjects from one group and half from another. A difference between groups would affect half the subjects and not the other half.

✓ Are X and Y measured independently?

The calculations are not valid if X and Y are intertwined. You'd violate this assumption if you correlate midterm exam scores with overall course score, as the midterm score is one of the components of the overall score.

✓ Were X values measured (not controlled)?

If you controlled X values (e.g., concentration, dose, or time) you should calculate linear regression rather than correlation.

✓ Is the covariation linear?

A correlation analysis would not be helpful if Y increases as X increases up to a point, and then Y decreases as X increases further. You might obtain a low value of r, even though the two variables are strongly related. The correlation coefficient quantifies linear covariation only.

✓ Are X and Y distributed according to Gaussian distributions?

To accept the P value from standard (Pearson) correlation, the X and Y values must each be sampled from populations that follow Gaussian distributions. Spearman nonparametric correlation does not make this assumption.

II. 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

Curve fitting without a model

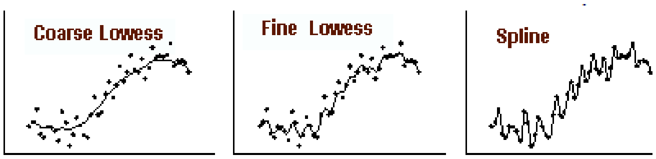
The term *curve fitting* is more general than *regression*. Your approach to curve fitting depends on your goal.

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.

Spline and 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 (middle panel below) reveals the fine structure of the data, but tends to wiggle a lot. A **cubic spline** curve (right panel below) goes through every data point, bending and twisting as needed.



Prism generates lowess curves using an algorithm adapted from reference 1. Don't select a lowess curve unless you have well over twenty data points. Prism generates the curve as a series of line segments. Enter the number of segments you want, and check the option box if you need to see the XY coordinates of each point, or if you want to use the resulting lowess, point-to-point, or spline curve as a standard curve.

These analyses generate curves, which are graphed and presented as a table of XY values. There are no tabular results to inspect.

Creating a point-to-point 'curve'

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 to calculate area under that curve

References

1. John Chambers et. al., *Graphical Methods for Data Analysis*, Wadsworth and Brooks, 1983.

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[17] go through every point, so may wiggle too much.

Lowess[17] 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[211], 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.

III. Generating curves and simulating data

Prism let's you plot a family of curves without fitting any data. Some programs call this "Plotting a function". It also lets you simulate data by adding random scatter to points on a simulated curve.

Plotting a function

This page explains how to plot a family of theoretical curves. Look elsewhere to simulate data 23 with random scatter, or to fit a curve to data 241

Key concepts: Plotting a function

Graph a family of theoretical curves to help learn the properties of the model. This is sometimes called *plotting a function*.

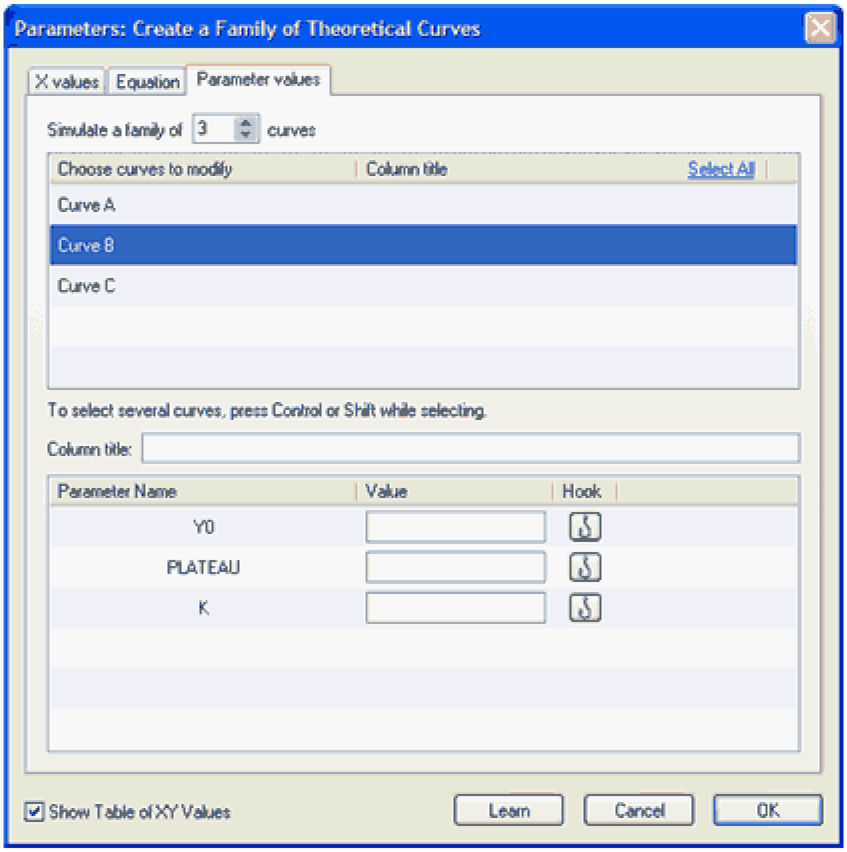
You choose the equation to plot, and the range of X values. Also choose the values of the parameters in the model.

After looking at the graph, go back and change the parameters to see how the graph changes.

Note that this analysis doesn't in fact analyze any data. Instead it generates curves from an equation.

How to: Graph a family of curves

1. Start from any data table or graph, click Analyze, open the **Simulate and generate** category, and then select **Create a family of theoretical curves**.
2. On the first tab, choose the number of line segments that will define the curve. For most purposes, the default value (150) will be fine. Also choose a starting and ending values of X.
3. On the second tab, select the equation (model) you want to plot.
4. On the third tab, choose how many curves will be in the family.
5. If you are generating only one curve, enter the parameters.
6. If you are simulating a family of curves, click 'Select all" in the dialog to enter the parameter values that apply for all curves. Then select the curves one at a time to enter the parameters that differ, as well as the titles you want to appear as column titles in the generated table.



Parameters: Create a Family of Theoretical Curves

Simulate a family of 3 curves

Choose curves to modify	Column title
Curve A	
Curve B	
Curve C	

To select several curves, press Control or Shift while selecting.

Column title:

Parameter Name	Value	Hook
Y0		
PLATEAU		
K		

☑ Show Table of XY Values

Simulating data with random error

This page explains how to simulate data sets, including random error. Look elsewhere to plot a family of theoretical curves 21.

Key concepts: Simulating data

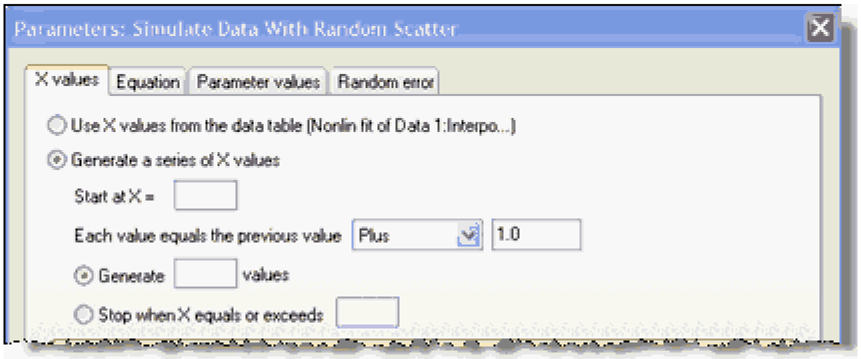
Simulation is an underused tool. It is a great way to understand models and plan experiments. Prism lets you combine an analysis that simulates data with a script 28 to do so many times as a way to perform Monte Carlo analyses.

Note that this analysis doesn't in fact analyze any data. Instead it generates curves from an equation.

How to: Simulate XY data

1. To simulate a family of data sets with random error, start from any data table or graph, click Analyze, open the **Simulate and generate** category, and then select **Simulate data with random scatter**.

2. **X values tab**. Generate a regular series (arithmetic or geometric) of X values or use the X values from the data table you are analyzing.



Parameters: Simulate Data With Random Scatter

○ Use X values from the data table (Nonlin fit of Data 1:Interpo...)

● Generate a series of X values

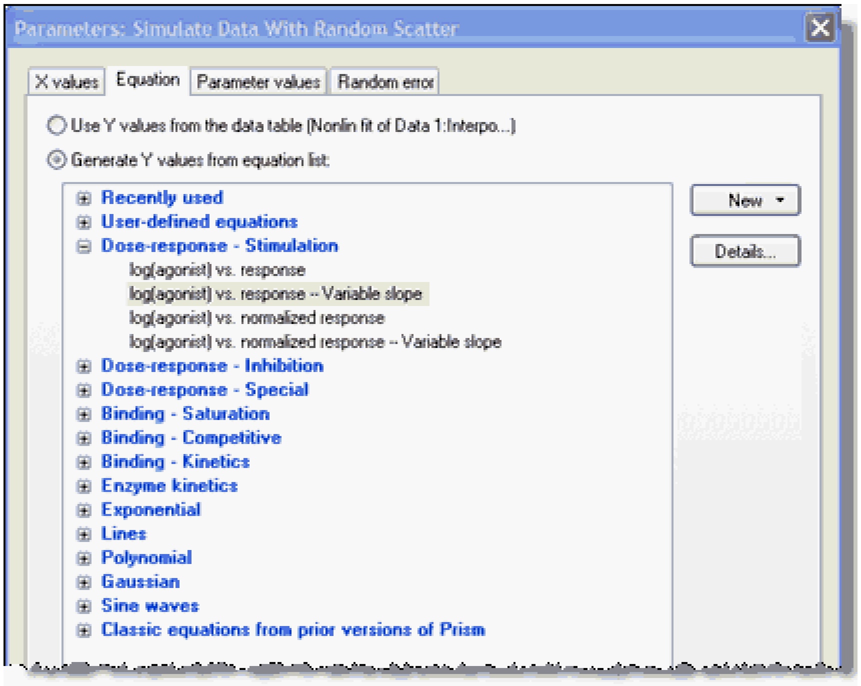
Start at X =

Each value equals the previous value Plus 1.0

● Generate values

○ Stop when X equals or exceeds

3. **Equation tab**. You can choose to use Y values from the data table you are analyzing, and then add random scatter. More often, you will choose an equation on this tab.



Parameters: Simulate Data With Random Scatter

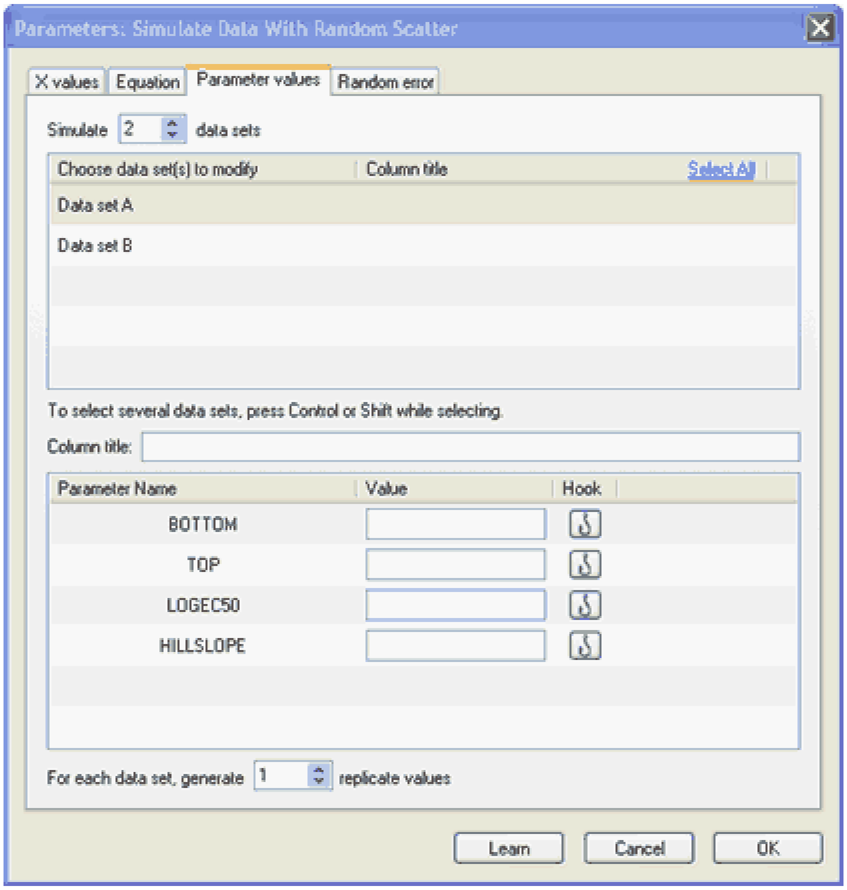
X values | Equation | Parameter values | Random error

○ Use Y values from the data table (Nonlin fit of Data 1:Interpo...)

● Generate Y values from equation list:

- ⊞ Recently used
- ⊞ User-defined equations
- ⊟ Dose-response - Stimulation
  - log(agonist) vs. response
  - log(agonist) vs. response -- Variable slope
  - log(agonist) vs. normalized response
  - log(agonist) vs. normalized response -- Variable slope
- ⊞ Dose-response - Inhibition
- ⊞ Dose-response - Special
- ⊞ Binding - Saturation
- ⊞ Binding - Competitive
- ⊞ Binding - Kinetics
- ⊞ Enzyme kinetics
- ⊞ Exponential
- ⊞ Lines
- ⊞ Polynomial
- ⊞ Gaussian
- ⊞ Sine waves
- ⊞ Classic equations from prior versions of Prism

4. **Parameter values tab**. On top of the tab, choose how many data sets you wish to simulate. At the bottom of the tab, choose how many replicates each data set will have. The main part of the tab is where you enter the values of each parameter. If you choose to simulate more than one data set, then you can choose to enter a parameter value just for one data set, or to enter a parameter that applies to several, or all, curves. Choose the data sets on the top part of the dialog, and enter the parameter values for that data set (or that group of data sets) below.



Simulate 2 data sets

Choose data set(s) to modify	Column title
Data set A	
Data set B	

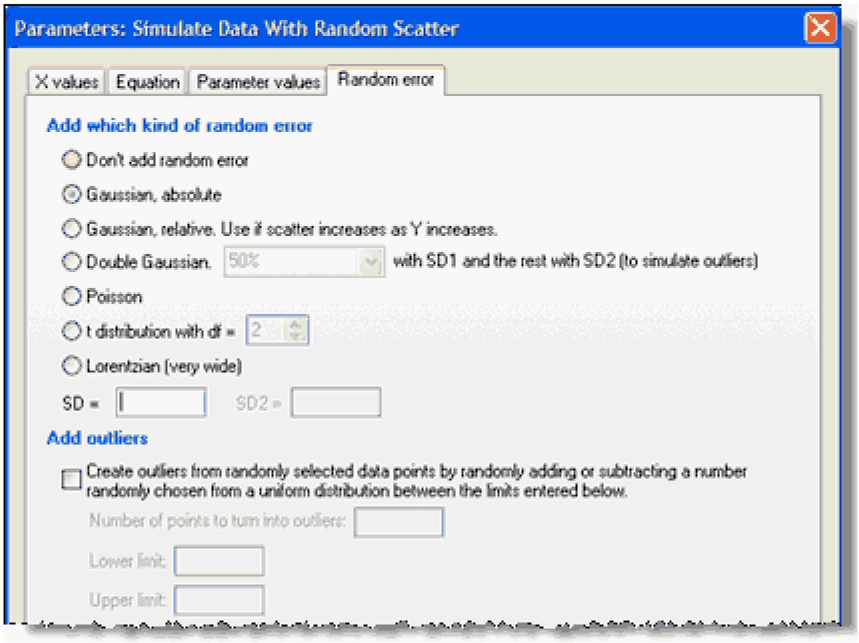
To select several data sets, press Control or Shift while selecting.

Column title:

Parameter Name	Value	Hook
BOTTOM		
TOP		
LOGEC50		
HILLSLOPE		

For each data set, generate 1 replicate values

5. **Random error tab**. Choose among several methods for generating random scatter and also adding outliers.



Add which kind of random error

○ Don't add random error

● Gaussian, absolute

○ Gaussian, relative. Use if scatter increases as Y increases.

○ Double Gaussian. 50% with SD1 and the rest with SD2 (to simulate outliers)

○ Poisson

○ t distribution with df = 2

○ Lorentzian (very wide)

SD = SD2 =

Add outliers

□ Create outliers from randomly selected data points by randomly adding or subtracting a number randomly chosen from a uniform distribution between the limits entered below.

Number of points to turn into outliers:

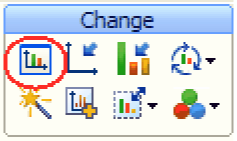
Lower limit:

Upper limit:

How to: Simulate column data

Prism can only simulate XY data. But you can simulate column data by following these steps.

1. In the first tab, choose a range of X values that generates the number of rows of data you want. The X values will be ignored, but you have to specify a range anyway.
2. In the second tab, choose the equation for a straight line from the lines section.
3. in the third tab, choose the number of data sets (columns) you want, and set the number of replicates (bottom of tab) to 1. Click "Select all" and set the slope equal to 0.0. Then set the intercept equal to the mean you want for each data set.
4. In the fourth tab, choose the random scatter which will be added to the mean values you entered (as 'intercept')
5. View the graph. It will be an XY graph, which is not useful. Click the Graph Type button and change to a column scatter graph.



How Prism generates random numbers

Prism can add random values to each of the calculated Y values to simulate experimental error.

The only way to generate truly random numbers is through a random physical process, such as tossing dice or measuring intervals between radioactive decays. Prism, like all computer programs, generates “random” numbers from defined calculations. Since the sequence of numbers is reproducible, mathematicians say that the numbers are “pseudo-random”. The difference between truly random and pseudo-random numbers rarely creates a problem. For most purposes, computer-generated random numbers are random enough to simulate data and test analytical methods.

Prism uses the time of day when calculating the first random number, so you will get a different series of random numbers every time you run the program.

Prism generates random values from a Gaussian distribution using routines adapted from ideas presented in Numerical Recipes in C, (W. H. Press et al, second edition, Cambridge Press, 1992). The function RAN3 (defined in Numerical Recipes) generates uniformly distributed random numbers and the function GASDEV transforms them to a Gaussian distribution with a mean of zero and a standard deviation you enter.

If you choose relative error, Prism first calculates a random number from a Gaussian distribution with a mean of zero and with a SD equal to the percent error you enter. It then multiplies that percentage times the ideal Y value to yield the actual random value that is added to the Y value.

When the Y values represent the number of objects you would observe in a certain space, or the number of events you would observe in a certain time interval, choose random numbers from a Poisson distribution. Again, our method is based on ideas from Numerical Recipes.

Prism also can generate random numbers from a t distribution with any number of degrees of freedom (df). This lets you simulate wider scatter than Gaussian. If df is low, this distribution is very wide. If df is high (more than 20 or so), it is almost indistinguishable from a Gaussian distribution. If df=1, the distribution is extremely wide (lots of outliers) and is identical to a Lorentzian distribution, also known as the Cauchy distribution. Prism uses this equation to generate random numbers from the t distribution with df degrees of freedom:

$$\frac{Rand}{\sqrt{\sum_{i=1}^{i=df} \frac{Rand^2}{df}}}$$

In this equation, Rand is a random number drawn from a Gaussian distribution with mean=0 and SD=1. To compute a random number from a t distribution with df degrees of freedom, Prism generates df+1 different random numbers drawn from a Gaussian distribution.

Using a script to simulate many data sets

Why simulate?

When testing analysis methods and experimental designs, it can be useful to simulate and analyze a large number of data sets. This can give you a sense of how precisely you can determine the parameter values, and whether the distribution of parameters is symmetrical.

Prism makes this easy.

How to: Running a Prism script

This example should help you get started.

1. Create a Prism project that simulates 23 a data set with random scatter (the first results sheet), and then fits 82 a curve through these data with nonlinear regression (second results sheet).
2. Make sure that the first data table is empty and formatted for column data (no X column; no Y subcolumns). The script will write the results into this table. You can click in the upper left corner of the data table to change its format, if needed.
3. Click the Prism button at the left of the toolbar, and choose Run Script. Then choose New Script, and start with a blank page.
4. Enter this script:

```
Table Prism 1 Clear
Foreach 100
 Goto R 1
 Regenerate
 Goto R 2
 WTable "logEC50",5,1
Next
```

5. Click the Run button on the Script dialog.
6. After the script is complete, go to the first data table and view the results.
7. Click analyze, and choose to do Column statistics (with normality test) or perhaps create a Frequency Distribution.

Understanding the script

The first line of the script specifies which data table will hold the results. Then the script loops 100 times. With each loop, it goes to the first results page (with the simulation) and regenerates with new random scatter. It then goes to the second results page (with curve fit results) and writes the value in the fifth row of the first column into the data table, and labels that column "logEC50". you will want to adjust the row number, and perhaps add additional lines to output additional results. You may also want to loop more than 100 times.

Simulations and script to assess confidence intervals

Meaning of 95% confidence

When you fit a curve with nonlinear regression, one of the most important set of results are the 95% confidence intervals of the parameters 253. These intervals are computed from the standard errors which are based on some mathematical simplifications. They are called "asymptotic" or "approximate" standard errors. 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.

How can you know whether the intervals really do have 95% confidence? There is no general way to answer this. But for any particular situation, you can get an answer using simulations.

Combining simulations and scripting: Monte Carlo analyses

1. Create a simulation analysis 23, to generate data similar to the data you plan to collect in your experiment, with reasonable choices for the range of X values, spacing of X values, number of replicates, and amount of scatter.

2. Fit the simulated data with nonlinear regression, choosing the appropriate model. In the Diagnostics tab, check the option to report the 95% confidence intervals of the parameters, and choose "separate lower and upper limits" (rather than a range).

3. Make sure the simulation is the first results sheet, and the nonlinear regression is the second. Create a column data table, and move it up to the top of the list of data tables (if there are others).

4. Run the script listed below. The first line sets things up so the Wtable commands write to the first data table in the project. The script loops 1000 times, each time regenerating the random scatter in the simulated data, recalculating the nonlinear regression results (by going to that results page), and writing out the two confidence limits (in the A column of rows 14 and 19) in the Wtable commands. You will have to change the row numbers to match the confidence limits you want to record, and change the titles ("lowerKM") accordingly.

   ```
   Table Prism 1 Clear
   ForEach 1000
    GoTo R 1
    Regenerate
    GoTo R 2
    Wtable "lowerKM", 14,1
    Wtable "uperKM", 19,1
   Next
   ```

5. Go to the first data table. Click Analyze, and choose Transform. Enter this user defined Y transform, and define the TrueValue in the dialog (the value used when simulating the data in step 1). The first line says that when transforming the Y values in column A, set the result equal to 0 if Y is less than TrueValue, and otherwise set Y equal to 1. The second line applies similar logic to column B.

   ```
   <A>Y=IF(Y<TrueValue, 0,1)
   <B>Y=IF(Y>TrueValue,0,1)
   ```

6. From the results table, click Analyze and choose Column Statistics. Accept the default choices. On the results page, note the bottom row, which is the sum of the columns.

The results table records whether the confidence intervals include the true value. Each row is from one simulated data set. Column A is 0 when that confidence interval started below the real value, and otherwise is 1. Column B is 0 when the confidence interval ended above the true value, and otherwise is 1. The number of 1's in both columns is the number of confidence intervals that do not include the true value. If the 95% confidence intervals are correct, and you ran 1000 simulations then you expect about 25 1 values in column A (the confidence interval started too high) and about 25 1 values in column B (the confidence interval ended too low) for a total of 50 intervals that did not include the true value (5%, leaving 95% that did include the true value). If the value is far from 5%, then you should distrust the confidence interval for that parameter.

Of course, you can obtain more precise answers by using more simulated data sets. Change the value in the second line of the script from 1000 to some larger value.

IV. Linear regression

Linear regression fits a straight line through your data to determine the slope and intercept

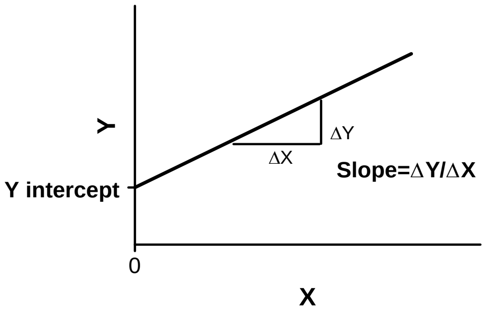
Key concepts: Linear regression

The goal of linear regression

What is linear regression?

Linear regression fits this model to your data:

$$Y=\text{intercept}+\text{slope}\times$$



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 vs. linear regression

Correlation 10 and linear regression are not the same. Consider these differences:

- Correlation quantifies the degree to which two variables are related. Correlation does not find a best-fit line (that is regression). You simply are computing a correlation coefficient (r) that tells you how much one variable tends to change when the other one does.
- With correlation you don't have to think about cause and effect. You simply quantify how well two variables relate to each other. With regression, you do have to think about cause and effect as the regression line is determined as the best way to predict Y from X.
- With correlation, it doesn't matter which of the two variables you call "X" and which you call "Y". you will get the same correlation coefficient if you swap the two. With linear regression, the decision of which variable you call "X" and which you call "Y" matters a lot, as you will 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.
- Correlation is almost always used when you measure both variables. It rarely is appropriate when one variable is something you experimentally manipulate. With linear regression, the X variable is often something you experimentally manipulate (time, concentration...) and the Y variable is something you measure.

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 = 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 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.

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.

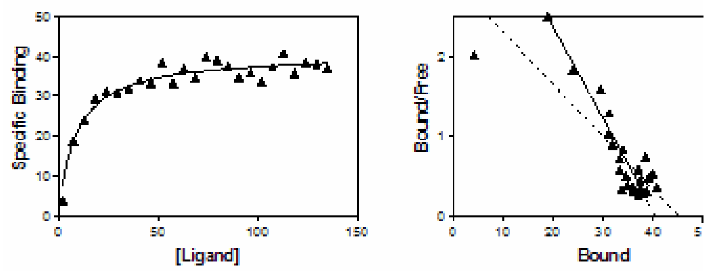
Before nonlinear regression was readily available, the best way to analyze nonlinear data was to transform the data to create a linear graph, and then analyze the transformed data with linear regression. Examples include Lineweaver-Burke plots of enzyme kinetic data, Scatchard plots of binding data,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

Advice: When to fit a line with nonlinear regression

Linear regression is a special case of nonlinear regression

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 linear regression.

Nonlinear regression offers more options

Using Prism's nonlinear regression analysis to fit a straight line makes sense when you want to:

- Fit to both a linear and nonlinear model, and compare the two models 243.
- Apply differential weighting 74.
- Automatically exclude outliers 70.
- Use a robust 70 fitting method.
- Perform a normality test 255 on the residuals.
- Inspect the correlation matrix or dependencies 261.
- Compare the scatter of points from the line with the scatter among replicates with a replicates test 259.

Nonlinear regression gives more choices if you enter averaged data

If you have replicates at each Y value, you can enter those directly into subcolumns. With both linear and nonlinear regression, Prism will fit the individual replicates unless you ask it to fit the means only.

If you manipulate your data in another program, you may enter your data as Mean, SD (or SEM) and N. In this case, Prism's linear regression analysis fits the means only, ignoring the scatter and sample size. In contrast, Prism's nonlinear regression gives you a choice (in the Weights tab 245) of fitting just the mean, or of accounting for scatter and sample size. With the latter choice, the results will be identical to what they would have been had you entered the raw data. If you want to account for the SD among replicates, use nonlinear regression.

Some fits that seem linear are really nonlinear

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' 204 with nonlinear regression.

If you want to fit two lines to different segments of the data, this cannot be done with Prism's linear regression analysis. However, Prism's nonlinear regression can fit segmental linear regression 206.

Using nonlinear regression is no harder than linear regression

Step-by-step instructions. 205

How to: Linear regression

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 35.

2. Enter data

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

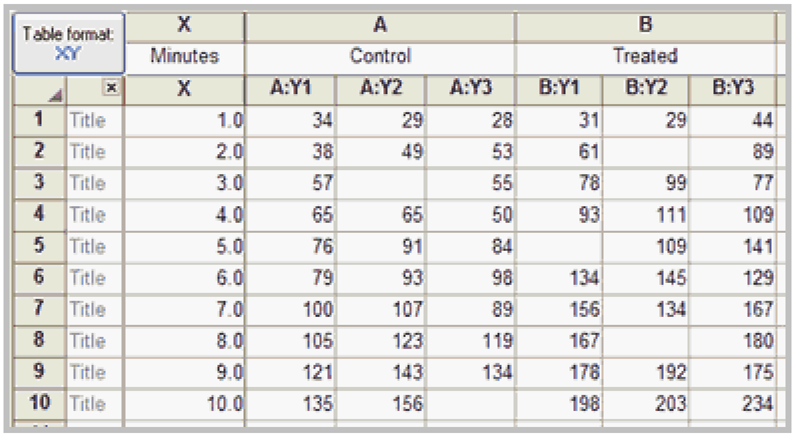
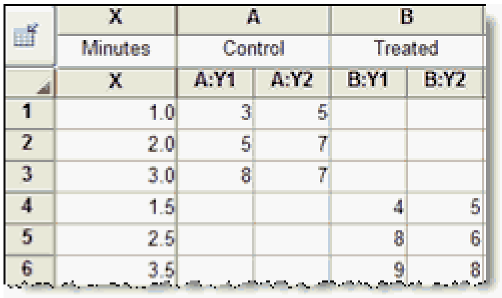


Table format: XY		X	A			B		
		Minutes	Control			Treated		
		X	A:Y1	A:Y2	A:Y3	B:Y1	B:Y2	B:Y3
1	Title	1.0	34	29	28	31	29	44
2	Title	2.0	38	49	53	61		89
3	Title	3.0	57		55	78	99	77
4	Title	4.0	65	65	50	93	111	109
5	Title	5.0	76	91	84		109	141
6	Title	6.0	79	93	98	134	145	129
7	Title	7.0	100	107	89	156	134	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 43

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 42.

Confidence and prediction bands

Learn about confidence and prediction bands 44.

Interpolating from a linear standard curve

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 -- 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 35.

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.

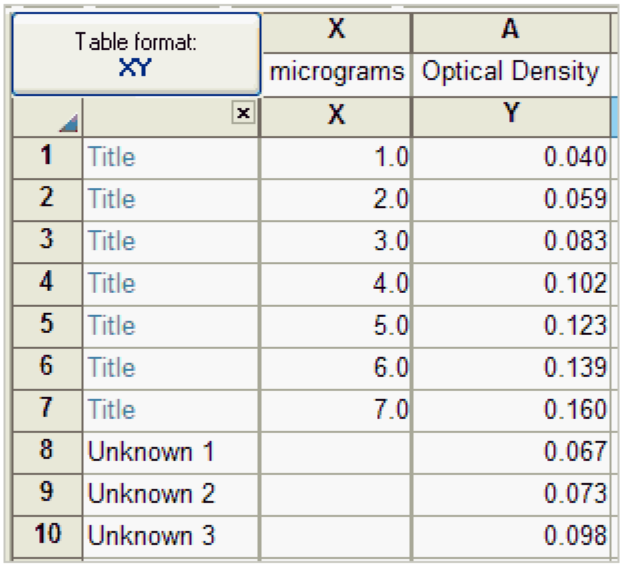


Table format: XY		X	A
		micrograms	Optical Density
		X	Y
1	Title	1.0	0.040
2	Title	2.0	0.059
3	Title	3.0	0.083
4	Title	4.0	0.102
5	Title	5.0	0.123
6	Title	6.0	0.139
7	Title	7.0	0.160
8	Unknown 1		0.067
9	Unknown 2		0.073
10	Unknown 3		0.098

3. Analysis choices

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

Choose Interpolate unknowns from standard curve.

Choose among the other analysis choices 36.

The results will appear in several pages, including one with the interpolated values, which will be in the same units as your original data. You can analyze that further using the Transform analysis to change units.

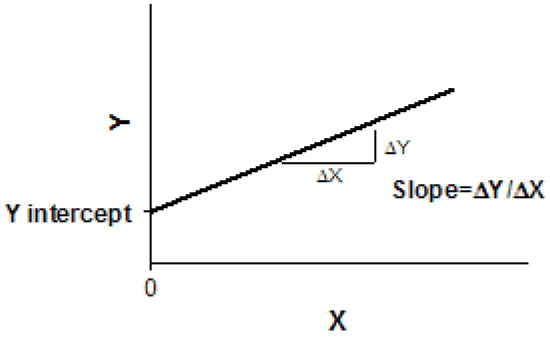
Results of linear regression

Slope and intercept

Prism reports the best-fit values of the slope and intercept, along with their standard errors and confidence intervals.

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.



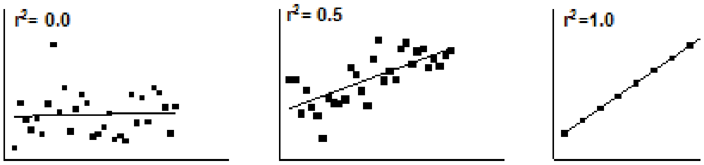
The standard error values of the slope and intercept can be hard to interpret, but their main purpose is to compute the 95% confidence intervals.

If you accept the assumptions of linear regression 43, 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.

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

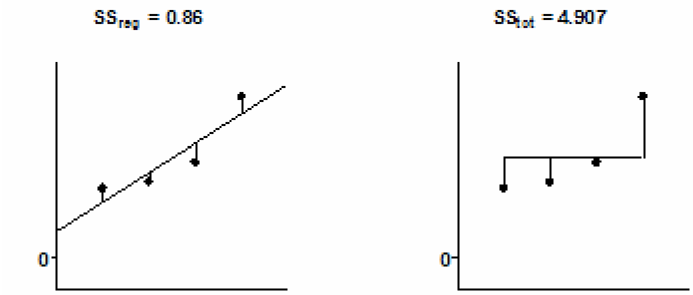
The meaning of r^2

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).

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.

Is the slope significantly different than zero?

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.

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 61 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.

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.

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.

.

Analysis checklist: 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.

✓ 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 ese data.

Graphing tips: Linear regression

Graphing the regression line

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

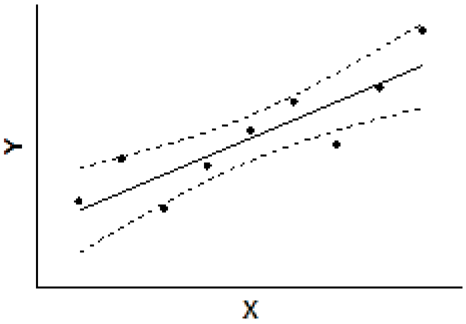
If you need to create additional graphs, or change which line is plotted on which graph, keep in mind that the line generated by linear regression is seen by Prism as a data set. You can add lines to a graph or remove lines from a graph on the 'Data sets on graph' tab of the Format Graph dialog.

Confidence and prediction bands

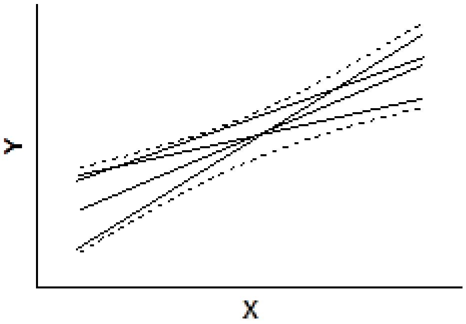
If you check the option box, Prism will calculate and graph either the 95% confidence band or 95% prediction band of the regression line.

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).

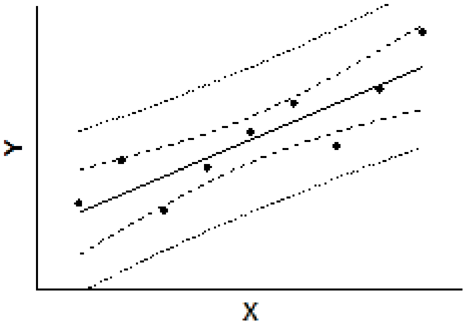


Given the assumptions of linear regression, you can be 95% confident that the two curved confidence bands enclose the true best-fit linear regression line, leaving a 5% chance that the true line is outside those boundaries.

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.

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

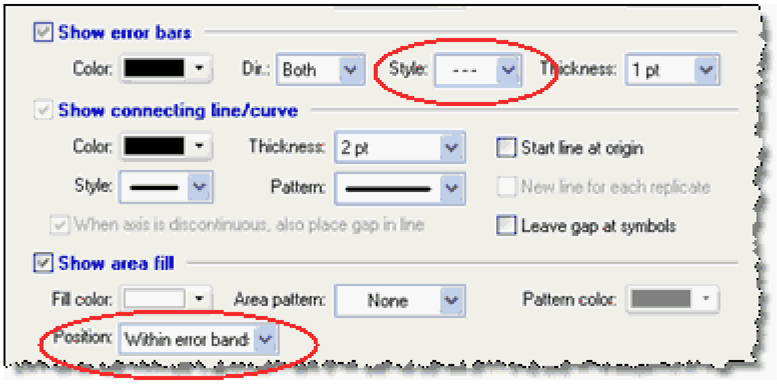
The confidence bands sort of combine the confidence intervals of the slope and intercept in a visual way. Use confidence bands to learn how precisely your data define the best-fit line.

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.

Fine-tuning the appearance of the confidence and prediction bands

If you check the option on the Linear regression, Prism will automatically superimpose the confidence or prediction band on the graph.

To adjust the appearance of the confidence or prediction bands, go to the Format Graph dialog, select the dataset that represents the best fit curve, 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

Residuals

If you check an option on the 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 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 283 of residuals from nonlinear regression.

Deming regression

Key concepts: Deming regression

Standard linear regression 32 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.

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).

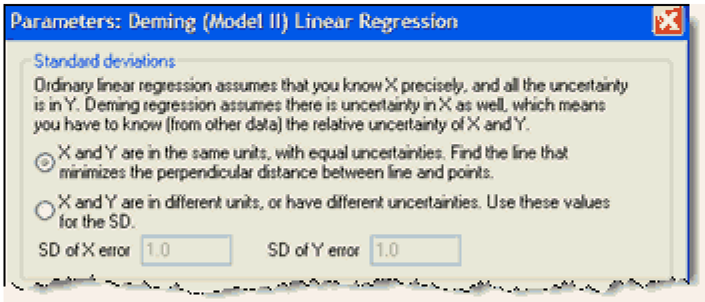
2. Enter data

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, 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).

$$SD_{error}\sqrt{\frac{\sum d_i^2}{N}}$$

Repeat this 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.

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.

V. Nonlinear regression

Nonlinear regression is one of the most powerful and useful features in Prism. Fit any model to your data to plot a curve and to determine best-fit values of the model' parameters.

Key concepts in nonlinear regression

Introducing nonlinear regression

The goal of nonlinear regression

The goal of nonlinear regression is to fit a model to XY data.

The model is expressed as an equation that defines Y as a function of X and one or more parameters.

Nonlinear regression finds the values of those parameters that generate the curve that comes closest to the data. Those best-fit values of the parameters are the best possible estimate of the values of those parameters.

To use nonlinear regression, therefore, you must choose a model 112 or enter one 228.

Nonlinear regression can also be used to compare two models 57 and to fit a family of cures at once (global fitting 61).

The differences between linear and 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 data to any equation that defines Y as a function of X and one or more parameters. 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 72 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 35 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 33 such as Scatchard or Lineweaver-Burke transforms whose only goal is to linearize your data.

Distinguishing nonlinear regression from other kinds of regression

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 51.

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.

GraphPad Prism 5 does not perform multiple regression. But by using column constants 64, you can effectively fit models with two independent variables in some circumstances.

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 5 does not perform logistic regression.

Proportional hazards regression

A proportional hazards regression 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 5 does not perform proportional hazards regression.

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 33 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).
- Don't smooth your data. 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.
- Avoid nonlinear Y transforms (reciprocals, logs) unless you have a very good reason.
- Enter raw replicates when possible, and not just mean and SD or SEM.
- If you have entered replicates, plot individual replicates rather than mean and error bar. You may want to later plot mean and error bar, but look at the plot of the raw data first.

Understanding models

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 and design new experiments.

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

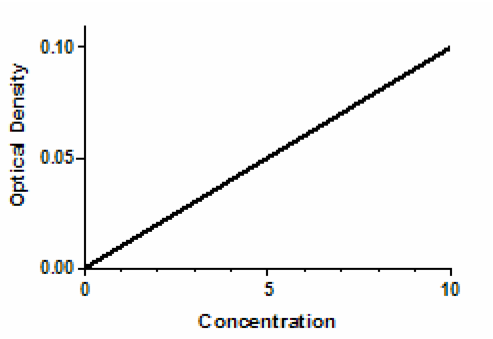
Background

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.

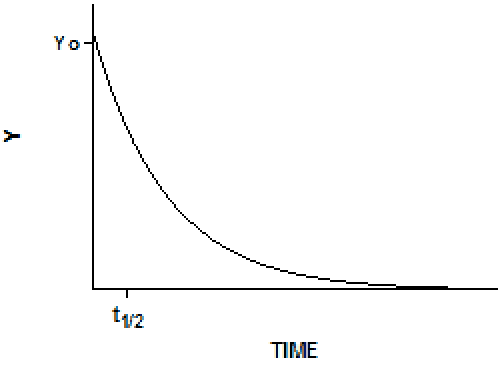
- Negative X values are meaningless, as concentrations cannot be negative.
- The model may fail at high concentrations of substance where the reaction is no longer limited by the concentration of substance.
- The model may also fail at high concentrations if the solution becomes so dark (the optical 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.

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

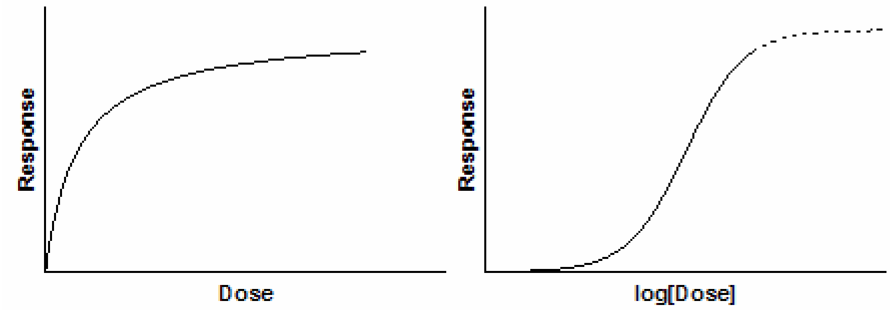
Equilibrium binding

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

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

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:

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



Why can't GraphPad Prism choose a model?

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.

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

Comparing models

Questions that can be answered by comparing models

Why compare?

When fitting biological data with regression, your main objective may be to *discriminate* between different models, or to ask if an experimental intervention changed a parameter.

Three kinds of comparisons are useful when analyzing data. Use the Compare tab[243] of the Nonlinear regression dialog to instruct Prism to perform any of these comparisons.

Three scenarios for comparing models

Prism can compare models to answer three distinct kinds of questions.

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 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).

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.

But that approach is too simple. A model with more parameters can have more inflection points, so of course comes closer to the points. A two-phase model almost always fits better than a one-phase model, and a three-phase 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.

Two statistical approaches to comparing models

Extra sum-of-squares F test

The Extra sum-of-squares F test 59 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 60.

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.

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.

Interpreting comparison of models 265

How the F test works to compare models

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.

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 significantly better than model 1. Otherwise, conclude that there is no compelling evidence supporting model 2, so accept the simpler

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.

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

The equation now makes intuitive sense. Like the F test, it balances the change in goodness-of-fit as assessed by sum-of-squares 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. If the difference in AIC is positive, then the change in sum-of-squares is not as large as expected from the change in number of parameters, so the simpler model is more likely to be correct.

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

Global nonlinear regression

What is global nonlinear regression?

The idea of global nonlinear regression

A global model defines a family of curves, rather than just a single curve and some parameters are shared between data sets. 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.

The uses of global nonlinear regression

Prism makes it easy to share a parameter across several data sets in order to enable global curve fitting. There are three uses for this.

- Test whether a parameter differs significantly between data sets. Prism tests this by comparing the goodness-of-fit when the parameter is shared, with the goodness-of-fit when the parameter is fit individually to each dataset. You set up this kind of comparison in the Compare tab 244, not the Constrain tab.
- Fit families of data where each dataset is incomplete, but the entire family of datasets defines the parameters. See an example 62.
- 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 63.

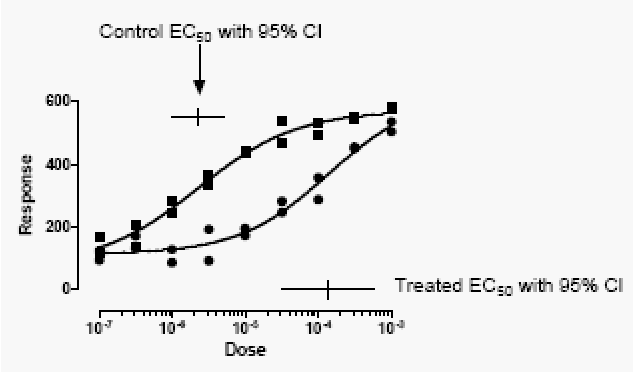
The first two uses of global fitting do not require writing special models. The third use requires that you write a model for this purpose.

Global nonlinear regression with Prism

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 244 of the Nonlinear regression dialog, choose which parameter(s) to share among data sets.

Using global regression to fit incomplete datasets

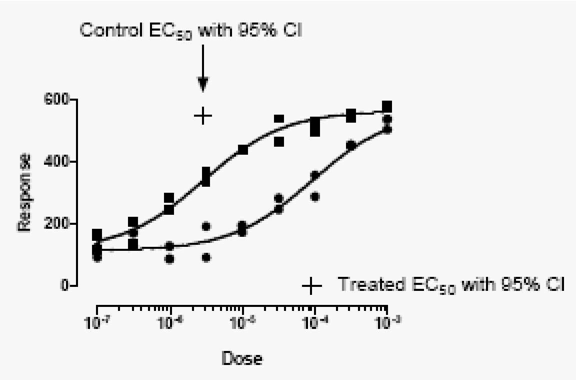
The graph below shows two dose-response curves. The goal of the experiment is to determine the two EC_{50} values. The EC_{50} is the concentration (dose) that gives a response half-way between the minimum and maximum responses. Each curve in the graph below was fit individually to one of the data sets.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

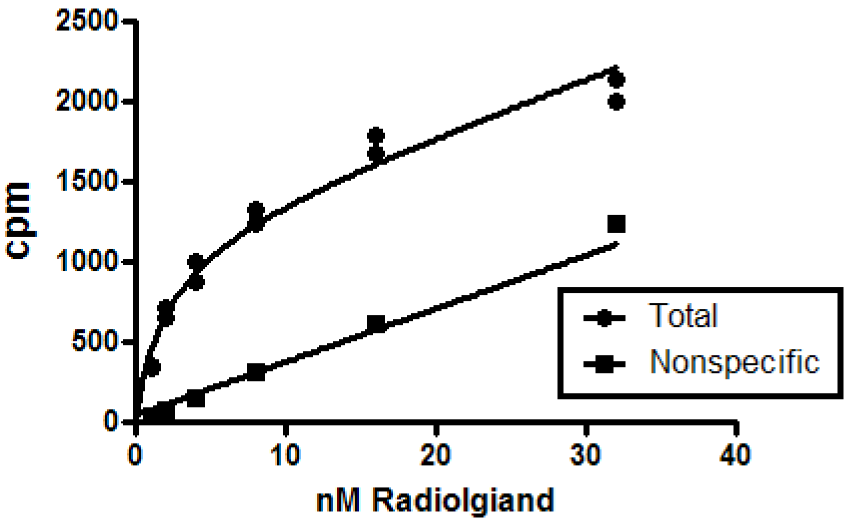
Fitting models where the parameters are defined by multiple data sets

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.

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.

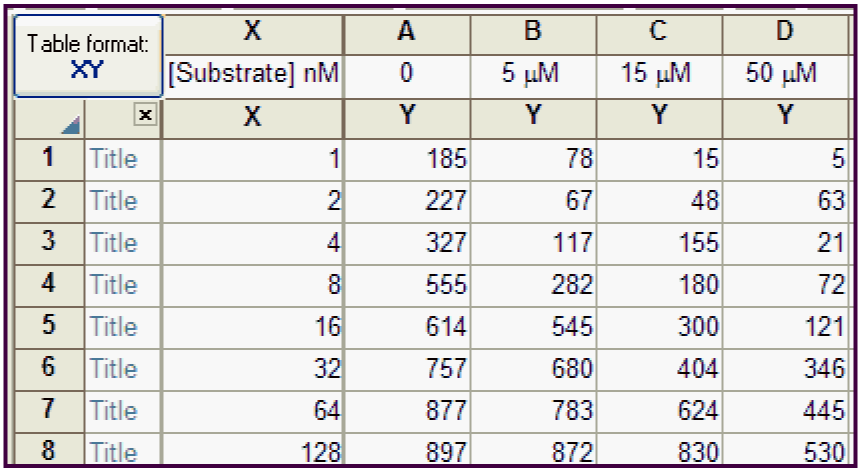


Table format: XY		X	A	B	C	D
		[Substrate] nM	0	5 µM	15 µM	50 µM
	☒	X	Y	Y	Y	Y
1	Title	1	185	78	15	5
2	Title	2	227	67	48	63
3	Title	4	327	117	155	21
4	Title	8	555	282	180	72
5	Title	16	614	545	300	121
6	Title	32	757	680	404	346
7	Title	64	877	783	624	445
8	Title	128	897	872	830	530

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.

Specifying a column constant when fitting data

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.

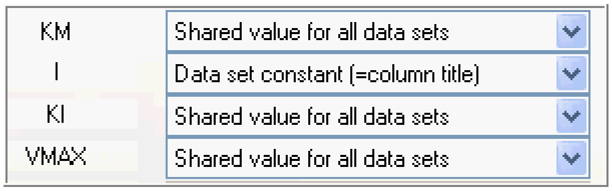
```
KmObs=Km(1+[I]/Ki)
Y=Vmax*X/(KmObs+X)
```

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 185.

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.

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.

Outlier elimination and robust nonlinear regression

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?

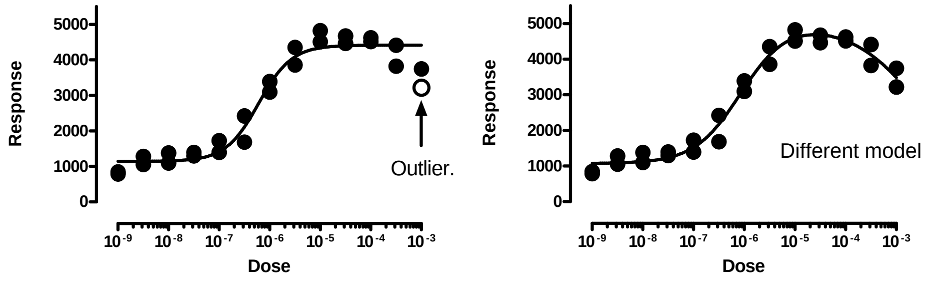
It depends on your experimental system.

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.

When to avoid automatic outlier removal

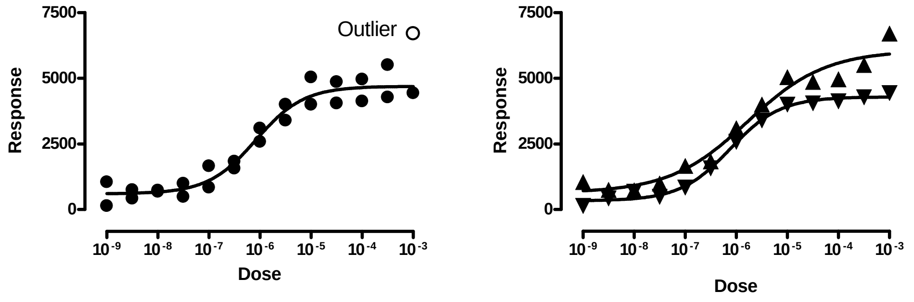
Outlier elimination is misleading when you are fitting the wrong model



The left panel above shows the data fit to a dose response curve 120. 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 131, 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).

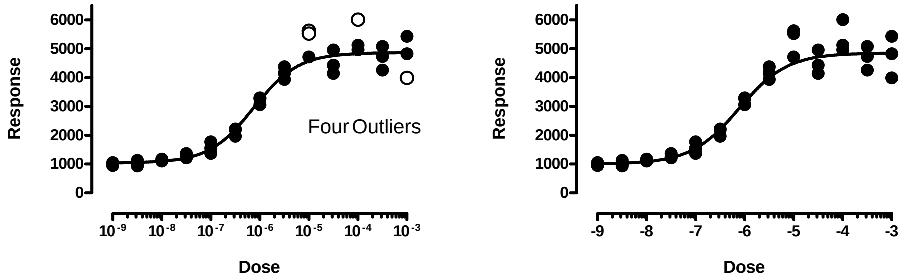
This example points out that outlier elimination is only appropriate when you are sure that you are fitting the correct model.

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 74. 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.

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 245 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.

The ROUT method of identifying outliers

How the ROUT method of removing outliers works

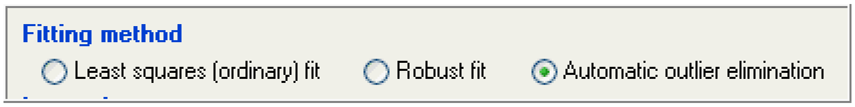
Prism offers a unique approach to identifying and removing outliers, detailed in reference 1. Because this method combines **Ro**bust regression and **Out**lier removal, we call it the **ROUT** method.

The ROUT method of regression follows these steps.

1. Our robust nonlinear regression 70 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 Weights tab 245 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.

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.

Robust nonlinear regression

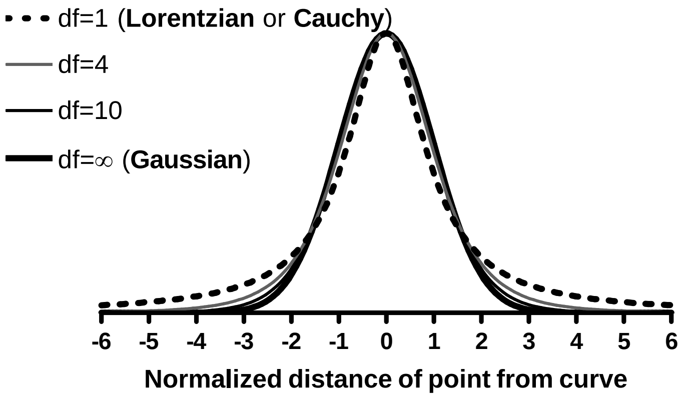
The need for robust regression

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 66 to identify and remove the outliers, and then run least-squares regression. Prism offers both choices.

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 72 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?

If your goal is just to obtain best-fit values of the parameters, robust regression works great. Outliers have little impact. Yet if all the data is Gaussian, robust regression and least-squares regression give almost identical results

Robust regression (as implemented by Prism) has three drawbacks:

- Robust regression cannot generate standard errors or confidence intervals for the parameters.
- Robust regression cannot generate confidence or prediction bands.
- Robust regression cannot compare the fits of two models or two datasets.

The main use of robust regression in Prism is as a 'baseline' from which to remove outliers 66. Its inability to compute standard errors or confidence intervals of the parameters greatly limits the usefulness of robust regression. We recommend it only to those who want to better understand the outlier-removal method (which begins with robust regression).

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.

How nonlinear regression works

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 = 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.

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.

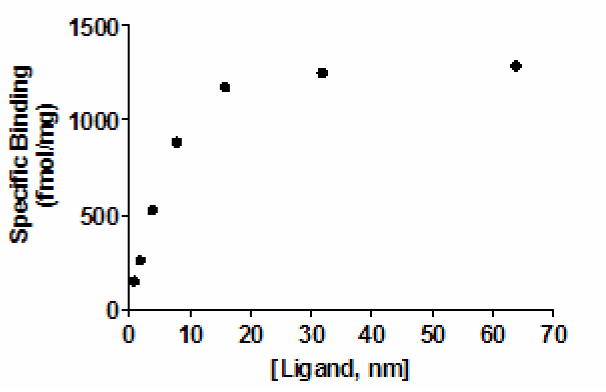
4. Adjust the parameters again so that the curve comes even closer to the points. Repeat.

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

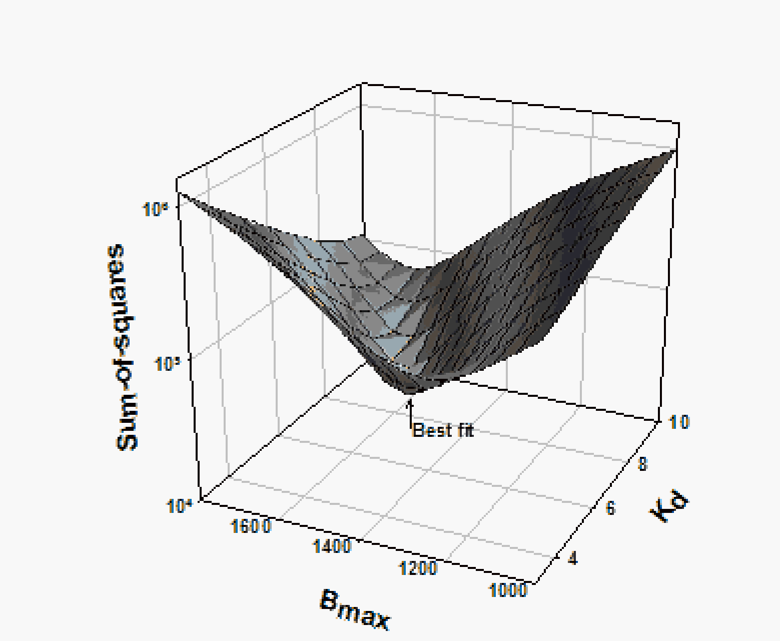
Step 3 is the only difficult one. Prism (and most other nonlinear regression programs) uses the method of Marquardt and Levenberg, which blends two other methods, the method of linear descent and the method of Gauss-Newton. The best way to understand these methods is to follow an example. Here are some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

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.

Unequal weighting in nonlinear regression

The idea of unequal 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.

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. To restore equal weighting to all the data points, you can choose a weighting method, as described below.

Prism offers six choices on the Weights tab 245 of nonlinear regression.

Relative weighting (weighting by 1/Y^2)

The weighting method used most often is called weighting by 1/Y^2. It is easier to think of this method as minimizing the sum-of-squares of the relative distances of the data from the curve. This method is appropriate when you expect the average distance of the points from the curve to be higher when Y is higher, but the relative distance (distance divided by Y) to be a constant. In this common situation, minimizing the sum-of-squares is inappropriate because points with high Y values will have a large influence on the sum-of-squares value while points with smaller Y values will have little influence. Minimizing the sum of the squares of the relative distances restores equal weighting to all points.

There are two ways to express the equation describing the quantity that nonlinear regression minimizes, shown below. The form on the left is easier to understand. You divide the distance of the data from the curve by the Y values of the data to obtain the relative distance, and then square that result. Most books on nonlinear regression use the equivalent form shown on the right – you first square the distance of the data from the curve, and then multiply that value times a weighting constant equal to 1/Y^2. That explains why relative weighting is often called weighting by 1/Y^2.

$$\sum\left(\frac{Y_{data}-Y_{curve}}{Y_{curve}}\right)^2=\sum\frac{1}{Y_{curve}^2}\left(Y_{data}-Y_{curve}\right)^2$$

Weighting by 1/Y

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.

$$\sum\left[\frac{Y_{data}-Y_{curve}}{\sqrt{Y_{curve}}}\right]^2=\sum\frac{1}{\left|Y_{curve}\right|}\left(Y_{data}-Y_{curve}\right)^2$$

Weighting by 1/X

The choices to weight by 1/X or 1/X^2 are rarely used. These choices are useful when you want to weight the points at the left part of the graph more than points to the right.

Weighting by observed standard deviation

Prism also offers the choice to weight by the reciprocal of the standard deviation squared. This means that data with little scatter (smaller standard deviation) get more weight that data with lots of scatter. This option will be useful if you understand how the scatter (or errors) arise in your experimental system, and can calculate appropriate weighting factors based on theory. Format a data table for entry of mean and SD, and enter (or paste) the weighting factors into the SD column. Don't use 1/SD^2 weighting if the SD values are computed from a few replicates. Random scatter can cause some SD values to be high and some low, and these differences may not reflect consistent differences in variability. You want to choose a weighting scheme to account for systematic differences in the predicted amount of variability if you were to repeat the experiment many times. You should not choose weighting based on variability you happened to observe in one small experiment. If you choose to weight by 1/SD ^{2}, Prism minimizes this quantity:

$$\sum\left(\frac{Y_{Data}-Y_{Curve}}{SD}\right)^2=\sum\frac{1}{SD^2}(Y_{Data}-Y_{Curve})^2$$

Caution: Beware of the option to weight by the observed standard deviation among replicates. That choice is much less useful than you might guess, especially when the SD is computed from only a few replicates.

How Prism 5 implements weighting

Prism 5 weights by the Y value of the curve. Previous versions 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. Compute Ycurve at all values of X based on the initial estimates of the parameter values.
2. Keeping those weights constant, use nonlinear regression to minimize the weighted sum-of-squares.
3. Recompute Ycurve based on the results of that curve fit.
4. Keeping those new weights constant, use nonlinear regression again to minimize the weighted sum-of-squares.
5. Repeat steps 3 and 4 until the weighted sum-of-squares no longer changes.

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

How standard errors and confidence intervals are computed

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 by multiplying $s_{y.x}$ by the parameter's diagonal element of the covariance matrix and taking the square root.

The confidence intervals are always centered on the best-fit value of the parameter, and extend above and below that value a distance equal to the parameter's standard error multiplied by a critical value from the t distribution. That value depends on how confident you want to be (95% is standard) and on the number of degrees of freedom, which equals the number of data points minus the number of parameters that are being fit. With 95% confidence and many degrees of freedom (more than a few dozen), this multiplier is very close to 1.96.

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. If you want to know the details, here they are:

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 261. 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.

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

Replicates

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.

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

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.

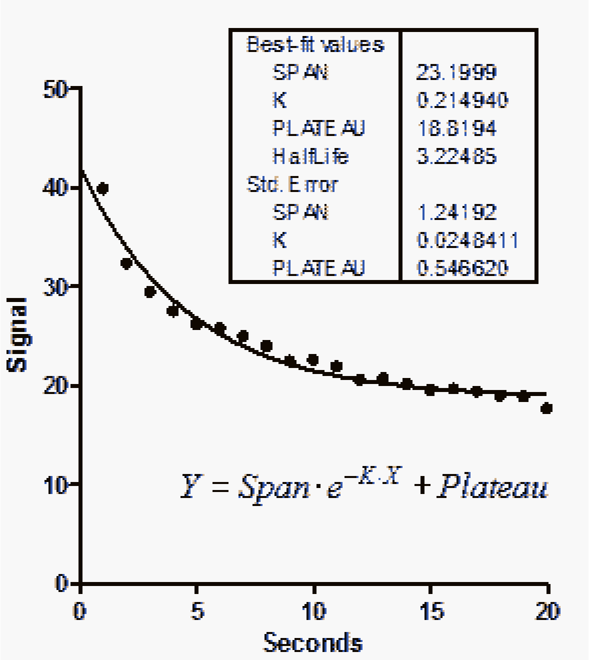
Interpreting dependency

You can interpret dependency 261 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

$$Y = Span \cdot e^{-K.X} + Plateau$$

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.

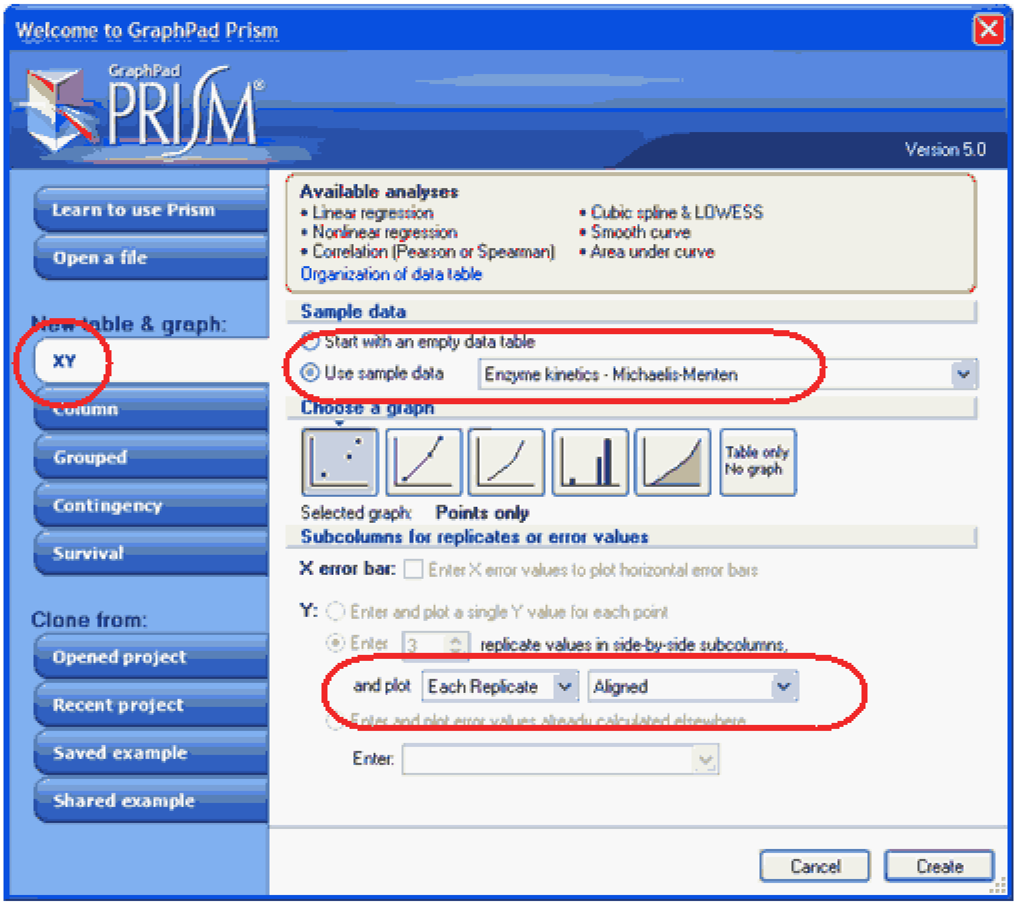
Each parameter has a distinct dependency (unless there are only two parameters). The dependency of Span is 0.613 and the dependency of Plateau is 0.813.

Nonlinear regression tutorials

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, select the sample data "Enzyme kinetics -- Michaelis-Menten", and choose to graph each replicate (rather than mean and error).



Welcome to GraphPad Prism

Available analyses
- Linear regression
- Nonlinear regression
- Correlation (Pearson or Spearman)
- Cubic spline & LOWESS
- Smooth curve
- Area under curve

Organization of data table

Sample data
○ Start with an empty data table
● Use sample data: Enzyme kinetics - Michaelis-Menten

Choose a graph
Selected graph: Points only

Subcolumns for replicates or error values
X error bar: □ Enter X error values to plot horizontal error bars
Y: ○ Enter and plot a single Y value for each point
● Enter 3 replicate values in side-by-side subcolumns,
and plot Each Replicate Aligned
○ Enter and plot error values already calculated elsewhere
Enter:

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.

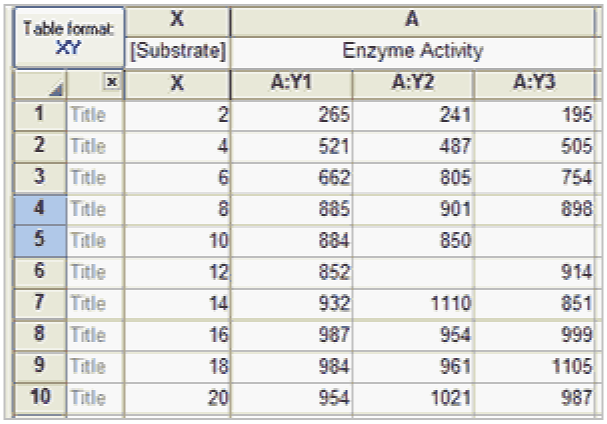
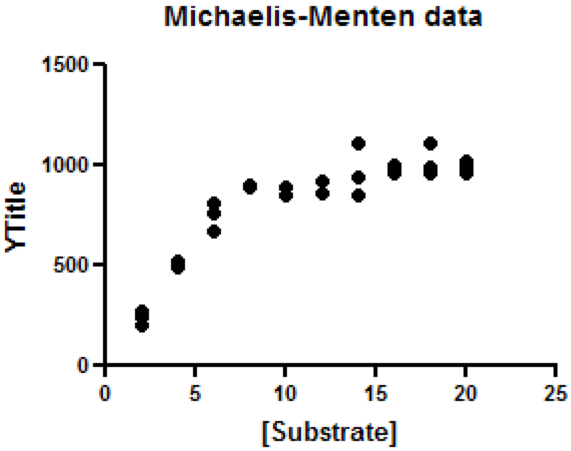


Table format: XY	X	A		
	[Substrate]	Enzyme Activity		
	X	A:Y1	A:Y2	A:Y3
1 Title	2	265	241	195
2 Title	4	521	487	505
3 Title	6	662	805	754
4 Title	8	885	901	898
5 Title	10	884	850	
6 Title	12	852		914
7 Title	14	932	1110	851
8 Title	16	987	954	999
9 Title	18	984	961	1105
10 Title	20	954	1021	987

The data are in triplicate. Some values are missing, and Prism always handles these just fine.

3. View the graph

Prism automatically assigns the graph the same name as the data table. Click on the Michaelis-Menten graph in the graphs section.

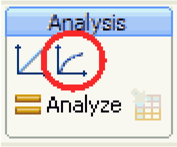


The graph Prism makes automatically is fairly complete. You can customize the symbols, colors, axis labels, position of legend, etc.

4. Choose nonlinear regression

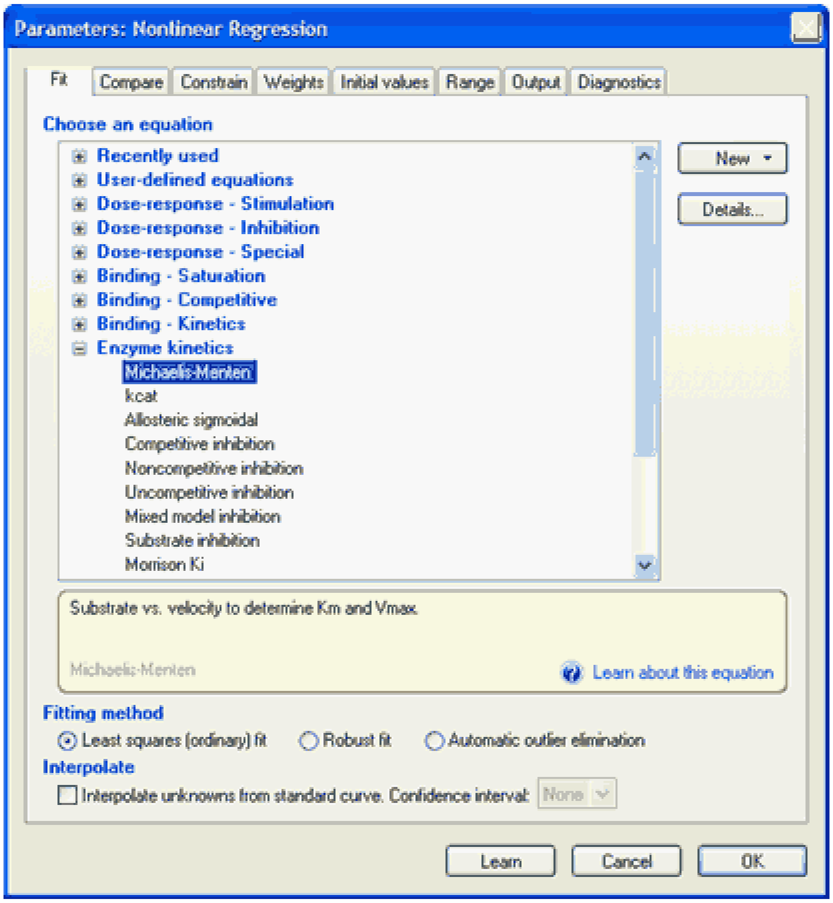
Click Analyze and choose Nonlinear regression from the list of XY analyses.

Even faster, click the shortcut button for nonlinear regression.



5. Choose a model

On the Fit tab of the nonlinear regression dialog, open the panel of Enzyme Kinetics equations and choose: Michaelis-Menten.



Choose an equation

- ⊞ Recently used
- ⊞ User-defined equations
- ⊞ Dose-response - Stimulation
- ⊞ Dose-response - Inhibition
- ⊞ Dose-response - Special
- ⊞ Binding - Saturation
- ⊞ Binding - Competitive
- ⊞ Binding - Kinetics
- ⊟ Enzyme kinetics
  - Michaelis-Menten
  - kcat
  - Allosteric sigmoidal
  - Competitive inhibition
  - Noncompetitive inhibition
  - Uncompetitive inhibition
  - Mixed model inhibition
  - Substrate inhibition
  - Morrison Ki

Substrate vs. velocity to determine Km and Vmax.

Michaelis-Menten

Learn about this equation

Fitting method

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

Interpolate

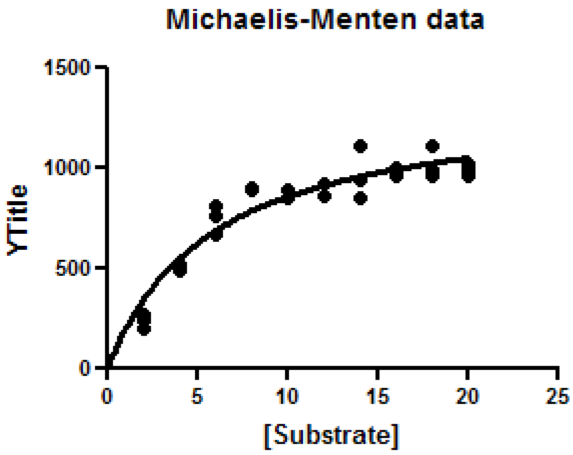
□ Interpolate unknowns from standard curve. Confidence interval: None

Learn more about the principles of enzyme kinetics 178 and about fitting Michaelis-Menten curves 180.

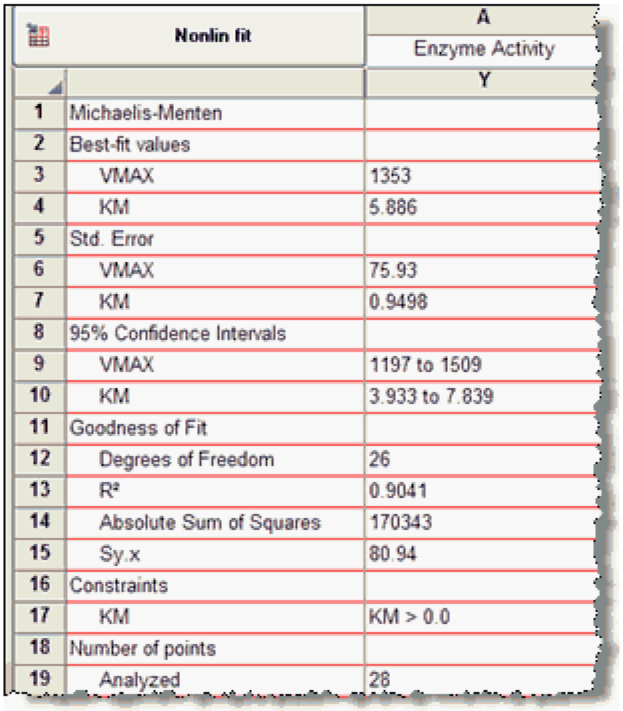
For this example, leave all the other settings to their default values.

Click OK to see the curves superimposed on the graph.

6. Inspect the graph



7. Inspect the results



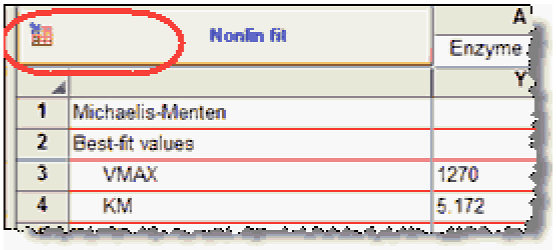
	Nonlin fit	A
		Enzyme Activity
		Y
1	Michaelis-Menten	
2	Best-fit values	
3	VMAX	1353
4	KM	5.886
5	Std. Error	
6	VMAX	75.93
7	KM	0.9498
8	95% Confidence Intervals	
9	VMAX	1197 to 1509
10	KM	3.933 to 7.839
11	Goodness of Fit	
12	Degrees of Freedom	26
13	R²	0.9041
14	Absolute Sum of Squares	170343
15	Sy.x	80.94
16	Constraints	
17	KM	KM > 0.0
18	Number of points	
19	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 253 .

8. Go back and perform the replicates test

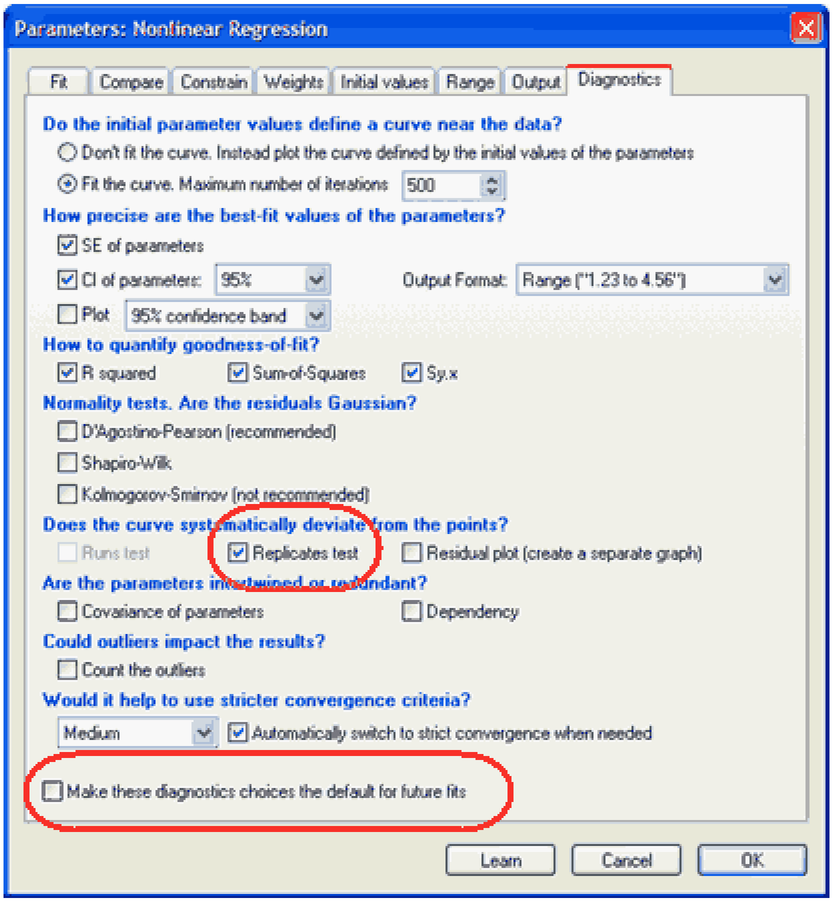
The replicates test assesses the adequacy of the fit by comparing the scatter among the triplicates with the scatter of points around the curve. It is not calculated by default, so the results do not appear in the results of step 7.

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.



	Nonlin fit	A Enzyme Y
1	Michaelis-Menten	
2	Best-fit values	
3	VMAX	1270
4	KM	5.172

Go to the Diagnostics tab 249, and check the option to perform the replicates test 259. Note that you can also check an option to make your settings here become the default for future fits.



Parameters: Nonlinear Regression

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 500

How precise are the best-fit values of the parameters?

☑ SE of parameters

☑ CI of parameters: 95% Output Format: Range ("1.23 to 4.56")

□ Plot 95% confidence band

How to quantify goodness-of-fit?

☑ R squared ☑ Sum-of-Squares ☑ Sy.x

Normality tests. Are the residuals Gaussian?

□ D'Agostino-Pearson (recommended)

□ Shapiro-Wilk

□ Kolmogorov-Smirnov (not recommended)

Does the curve systematically deviate from the points?

□ Runs test ☑ Replicates test □ Residual plot (create a separate graph)

Are the parameters intertwined or redundant?

□ Covariance of parameters □ Dependency

Could outliers impact the results?

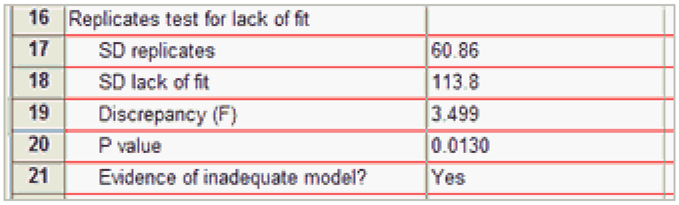
□ Count the outliers

Would it help to use stricter convergence criteria?

Medium ☑ Automatically switch to strict convergence when needed

□ 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 87.



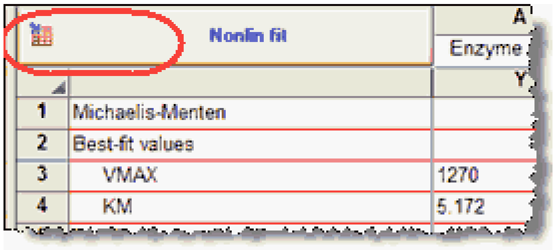
16	Replicates test for lack of fit	
17	SD replicates	60.86
18	SD lack of fit	113.8
19	Discrepancy (F)	3.499
20	P value	0.0130
21	Evidence of inadequate model?	Yes

Example: Comparing two enzyme kinetics models

1.Open analysis parameters dialog

If you are continuing from the prior example:

This example continues from the previous one 87. Click the analysis parameter button again.



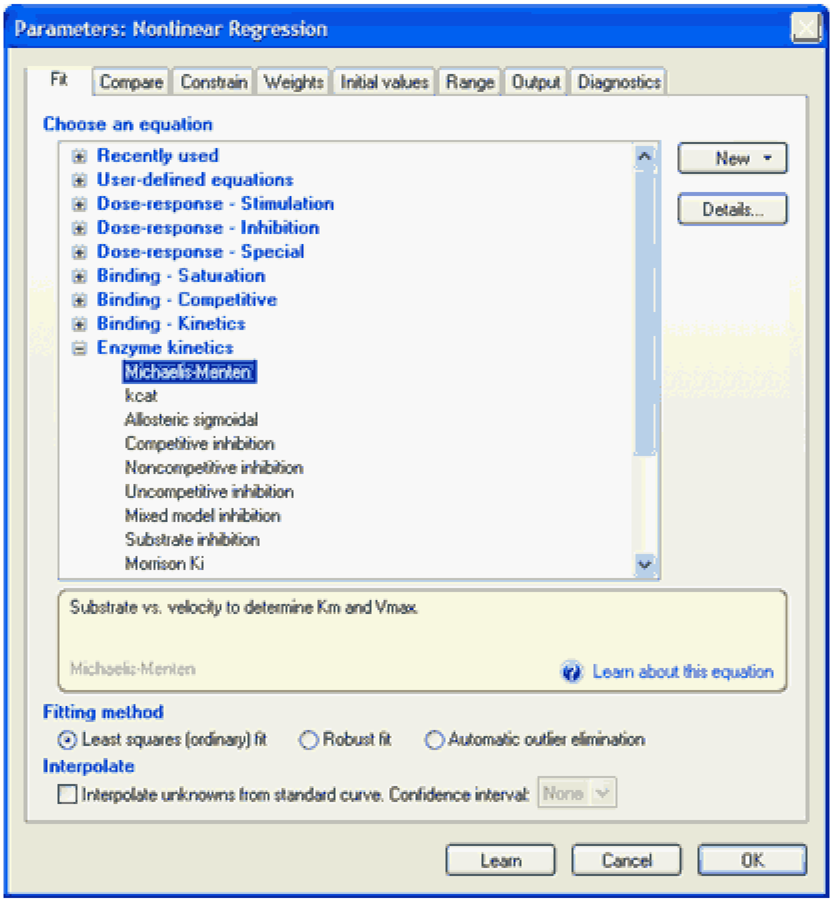
	Nonlin fit	A Enzyme Y
1	Michaelis-Menten	
2	Best-fit values	
3	VMAX	1270
4	KM	5.172

If you are starting with this example:

From the Welcome or New Table dialog, choose to create XY data table, select the sample data "Enzyme kinetics -- Michaelis-Menten", and choose to graph each replicate (rather than mean and error). Click Analyze and choose Nonlinear regression from the list of XY analyses.

2. Choose the first model

On the Fit tab, choose (or make sure you still have chosen) the Michaelis-Menten equation.



Substrate vs. velocity to determine Km and Vmax.

Michaelis-Menten

Fitting method

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

Interpolate

□ Interpolate unknowns from standard curve. Confidence interval: None

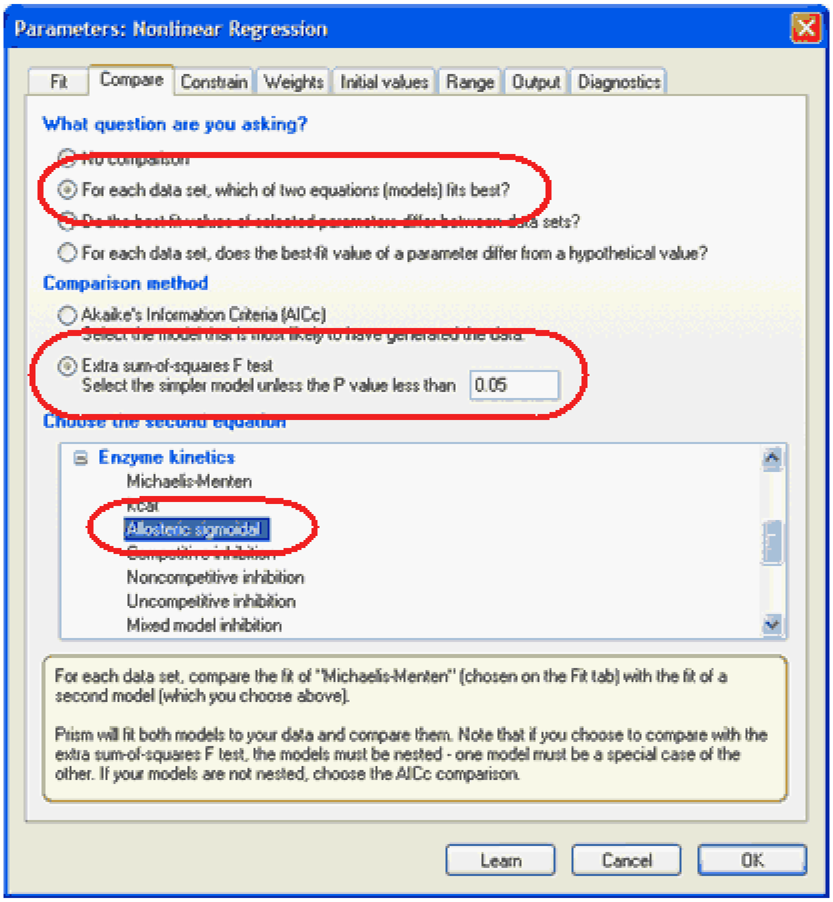
3. Choose the other model

Go to the Compare tab 243.

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 59.

For the second equation, choose "Allosteric sigmoidal 184" 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 parameters differ between data sets?

○ For each data set, does the best-fit value of a parameter differ from a hypothetical value?

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

Choose the second equation

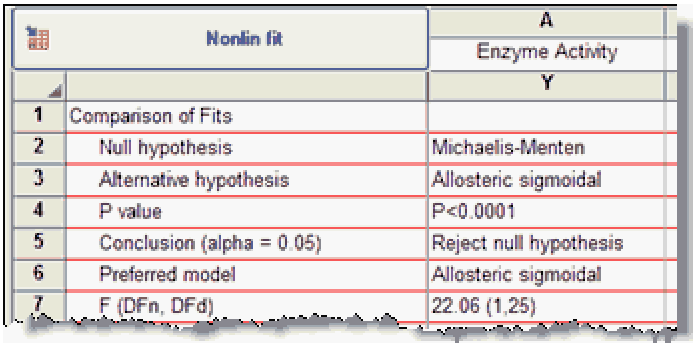
⊟ Enzyme kinetics
Michaelis-Menten
Kcat
Allosteric sigmoidal
Competitive inhibition
Noncompetitive inhibition
Uncompetitive inhibition
Mixed model inhibition

For each data set, compare the fit of "Michaelis-Menten" (chosen on the Fit 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

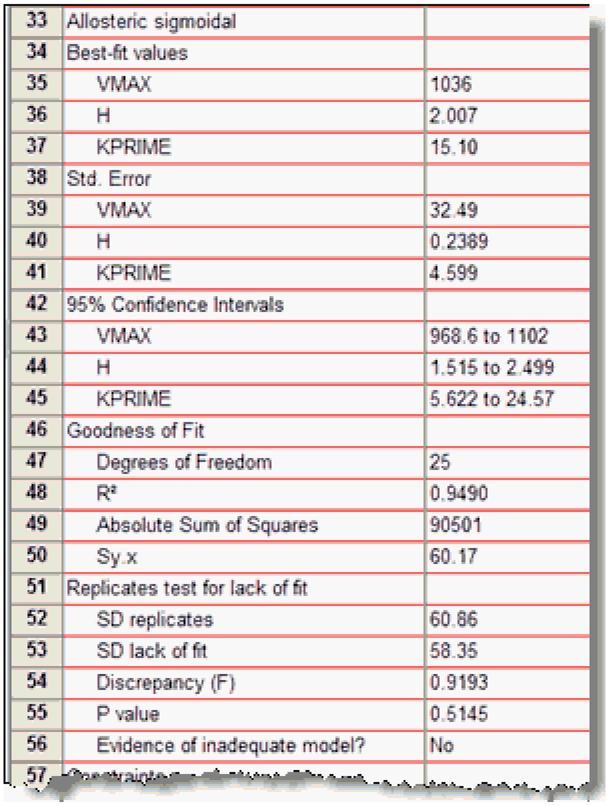
The top part of the results summarizes the comparison. The P value is low, suggesting that the simpler (Michaelis-Menten) model is too simple and should be rejected. The allosteric model fits significantly better.



	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.



33	Allosteric sigmoidal	
34	Best-fit values	
35	VMAX	1036
36	H	2.007
37	KPRIME	15.10
38	Std. Error	
39	VMAX	32.49
40	H	0.2389
41	KPRIME	4.599
42	95% Confidence Intervals	
43	VMAX	968.6 to 1102
44	H	1.515 to 2.499
45	KPRIME	5.622 to 24.57
46	Goodness of Fit	
47	Degrees of Freedom	25
48	R^2	0.9490
49	Absolute Sum of Squares	90501
50	Sy.x	60.17
51	Replicates test for lack of fit	
52	SD replicates	60.86
53	SD lack of fit	58.35
54	Discrepancy (F)	0.9193
55	P value	0.5145
56	Evidence of inadequate model?	No
57	Constraints	

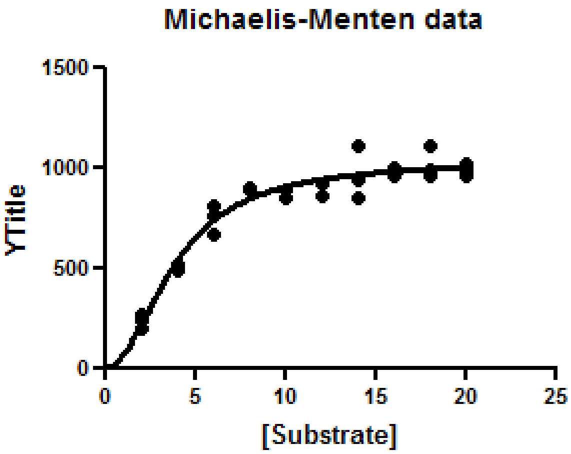
The P value of the replicates test is high, which means the scatter of points around the curve is consistent with variability of replicates from each other.

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

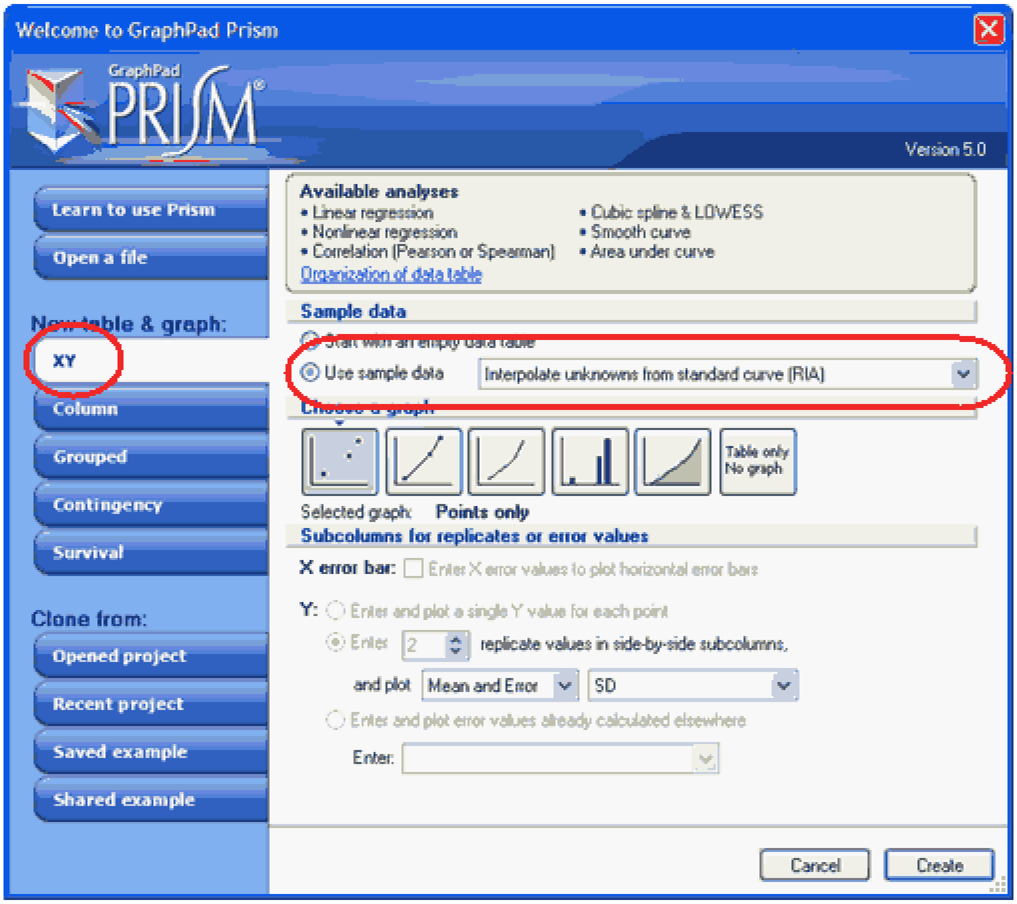
Since the allosteric model fit significantly better, that is the model that Prism uses when it plots a curve on the graph. You can barely see that it has a sigmoidal shape. You can also see that you should collect more data with substrate concentrations between 0 and 5 to fully define this curve.


Michaelis-Menten data
YTitle
1500
1000
500
0
0
5
10
15
20
25
[Substrate]

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 "Interpolate unknowns from standard curve (RIA)".



Available analyses
- Linear regression
- Nonlinear regression
- Correlation (Pearson or Spearman)
- Cubic spline & LOWESS
- Smooth curve
- Area under curve

Organization of data table

Sample data

○ Start with an empty data table

● Use sample data: Interpolate unknowns from standard curve (RIA)

Choose a graph

Table only No graph

Selected graph: Points only

Subcolumns for replicates or error values

X error bar: □ Enter X error values to plot horizontal error bars

Y: ○ Enter and plot a single Y value for each point

● Enter 2 replicate values in side-by-side subcolumns,

and plot Mean and Error SD

○ Enter and plot error values already calculated elsewhere

Enter:

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.

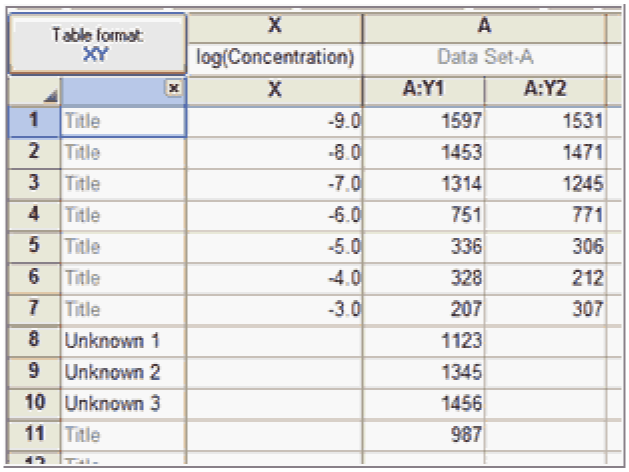
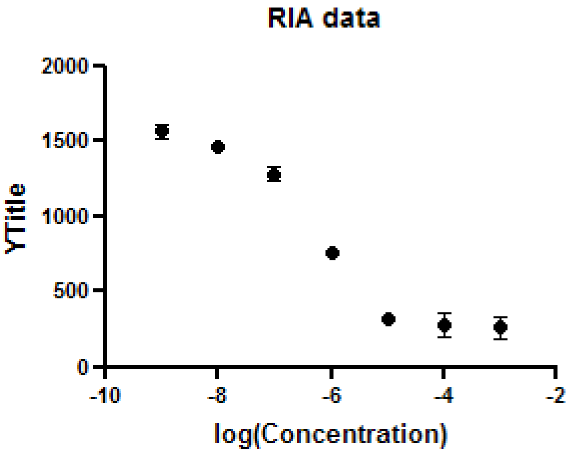


	Table format: XY	X	A	
		log(Concentration)	Data Set-A	
		X	A:Y1	A:Y2
1	Title	-9.0	1597	1531
2	Title	-8.0	1453	1471
3	Title	-7.0	1314	1245
4	Title	-6.0	751	771
5	Title	-5.0	336	306
6	Title	-4.0	328	212
7	Title	-3.0	207	307
8	Unknown 1		1123	
9	Unknown 2		1345	
10	Unknown 3		1456	
11	Title		987	
12	Title			

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.

3. View the graph

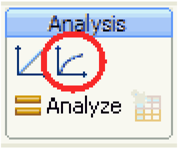


The graph Prism makes automatically is fairly complete. You can customize the symbols, colors, axis labels, etc. Since the unknowns have no X value, they are not included on the graph.

4. Choose nonlinear regression

Click Analyze and choose Nonlinear regression from the list of XY analyses.

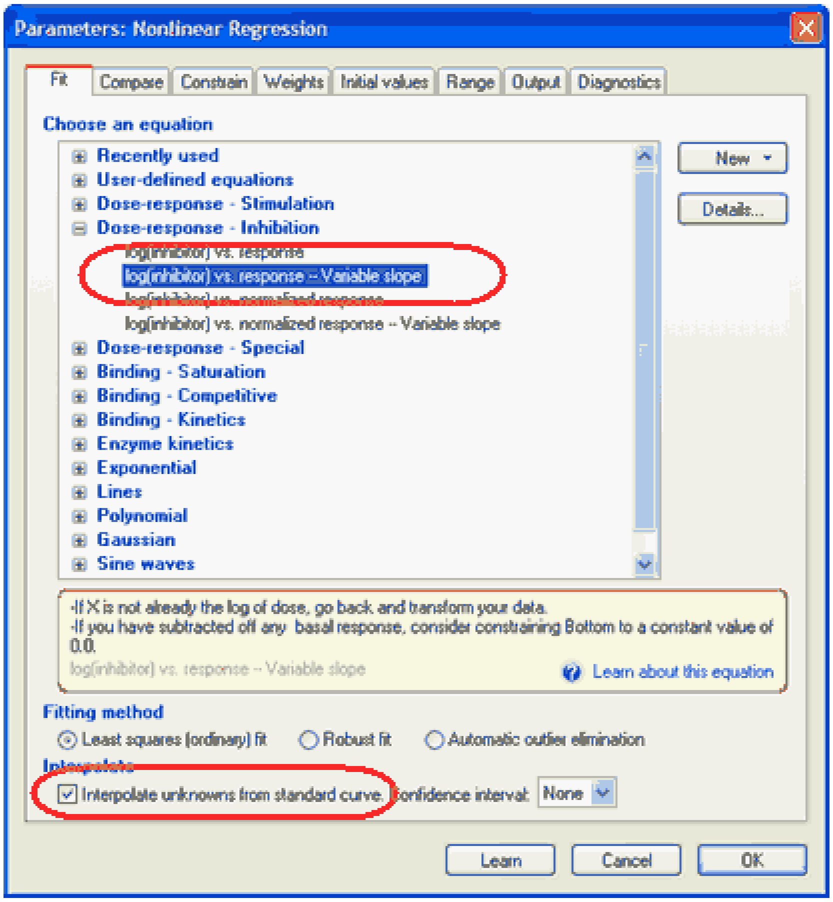
Alternatively, click the shortcut button for nonlinear regression.



5. Choose a model

On the Fit tab of the nonlinear regression dialog, open the panel of inhibitory dose-response models and choose: log(inhibitor) vs. response -- variable slope.

At the bottom of the dialog, check the option: Interpolate unknowns from standard curve.


Parameters: Nonlinear Regression
Fit
Compare
Constrain
Weights
Initial values
Range
Output
Diagnostics
Choose an equation
⊞ Recently used
⊞ User-defined equations
⊞ Dose-response - Stimulation
⊟ Dose-response - Inhibition
log(inhibitor) vs. response
log(inhibitor) vs. response -- Variable slope
log(inhibitor) vs. normalized response
log(inhibitor) vs. normalized response -- Variable slope
⊞ Dose-response - Special
⊞ Binding - Saturation
⊞ Binding - Competitive
⊞ Binding - Kinetics
⊞ Enzyme kinetics
⊞ Exponential
⊞ Lines
⊞ Polynomial
⊞ Gaussian
⊞ Sine waves
New
Details...
Learn about this equation
Learn
Cancel
OK

-If X is not already the log of dose, go back and transform your data.
-If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.0.
log(inhibitor) vs. response -- Variable slope

Fitting method

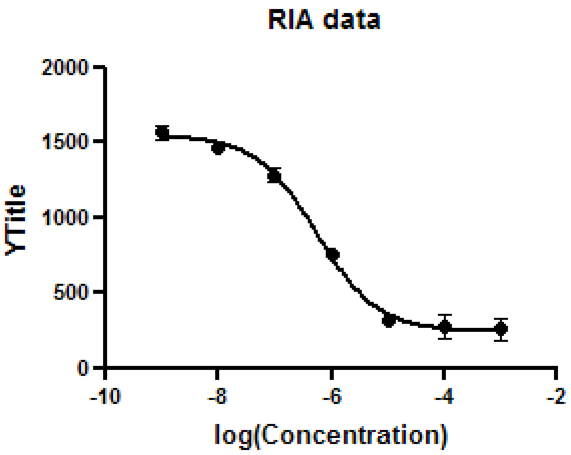
● Least squares (ordinary) fit ○ Robust fit ○ Automatic outlier elimination

Interpolate

☑ Interpolate unknowns from standard curve. Confidence interval: None

For this example, leave all the other settings to their default values.

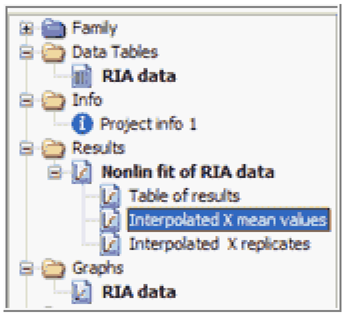
Click OK to see the curves superimposed on the graph.

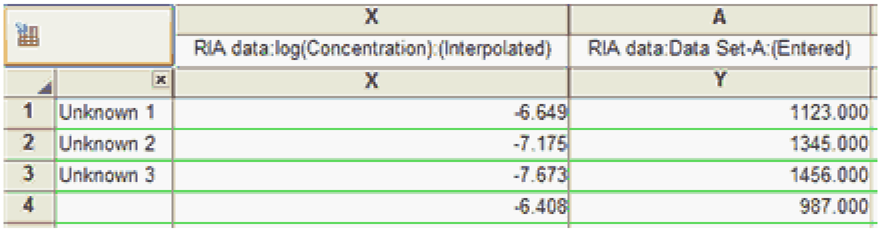


6. Inspect the results

The results appear on several pages. The first page is the table of results. It tabulates the best-fit values of the parameters and much more. For this example, we aren't too interested in these results. The next two pages show the results we care about here -- the interpolated X values

Since the data table was formatted for entry of duplicate Y values, Prism presents the interpolated values on two pages. One page averages the duplicates and interpolates the average. The other page reports the interpolation of each entered Y value. For this example, we only entered unknown values into one subcolumn so the two interpolated results pages are identical.





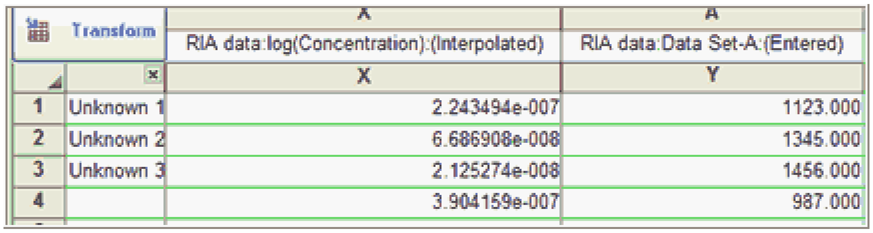
		X	A
		RIA data:log(Concentration):(Interpolated)	RIA data:Data Set-A:(Entered)
		X	Y
1	Unknown 1	-6.649	1123.000
2	Unknown 2	-7.175	1345.000
3	Unknown 3	-7.673	1456.000
4		-6.408	987.000

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 Analyze 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.

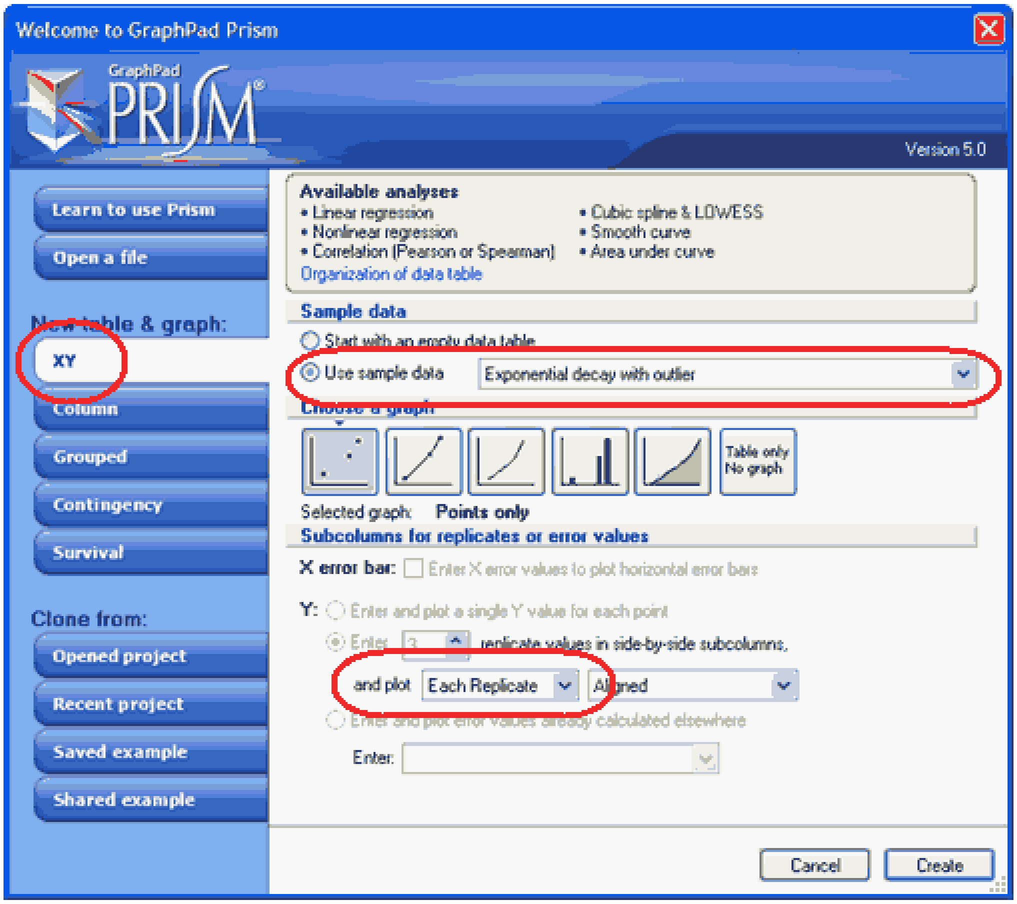
Click and edit the column title to "Concentration (M)".

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 69, when it is useful 66, and when it should be avoided 67.

1. Create the data table

From the Welcome or New Table dialog, choose to create XY data table, select the sample data "Exponential decay with outlier", and choose to graph each replicate (rather than mean and error).



Welcome to GraphPad Prism

Available analyses
- Linear regression
- Nonlinear regression
- Correlation (Pearson or Spearman)
- Cubic spline & LOWESS
- Smooth curve
- Area under curve

Organization of data table

Sample data

○ Start with an empty data table

● Use sample data: Exponential decay with outlier

Choose a graph

Selected graph: Points only

Subcolumns for replicates or error values

X error bar: □ Enter X error values to plot horizontal error bars

Y: ○ Enter and plot a single Y value for each point

● Enter 3 replicate values in side-by-side subcolumns, and plot Each Replicate, Aligned

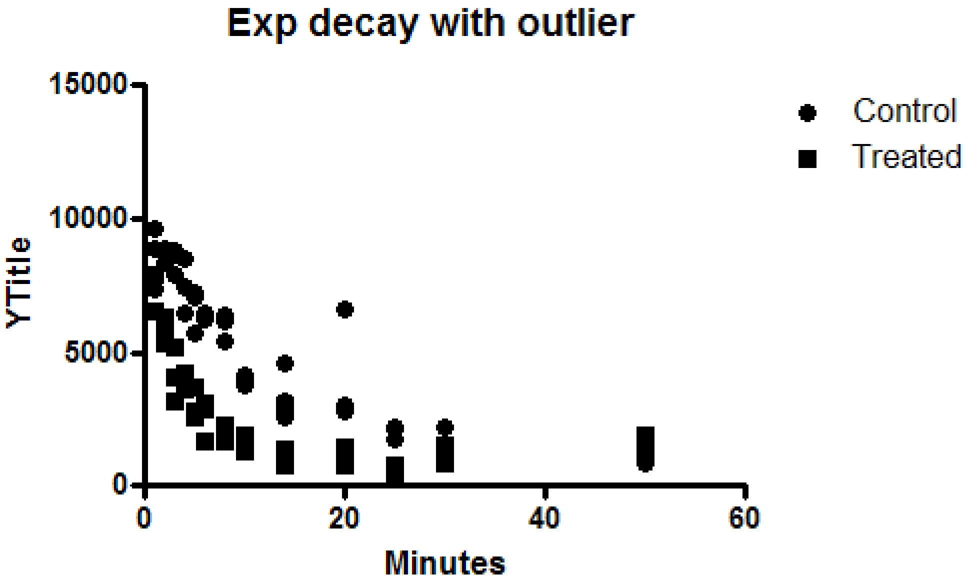
○ Enter and plot error values already calculated elsewhere

Enter:

2. Inspect the data table and graph

The data table has values for two data sets (control and treated) with triplicate subcolumns.

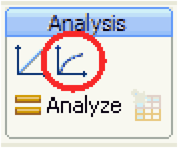
Here is the graph that Prism makes automatically:



3. Choose nonlinear regression

Click Analyze and choose Nonlinear regression from the list of XY analyses.

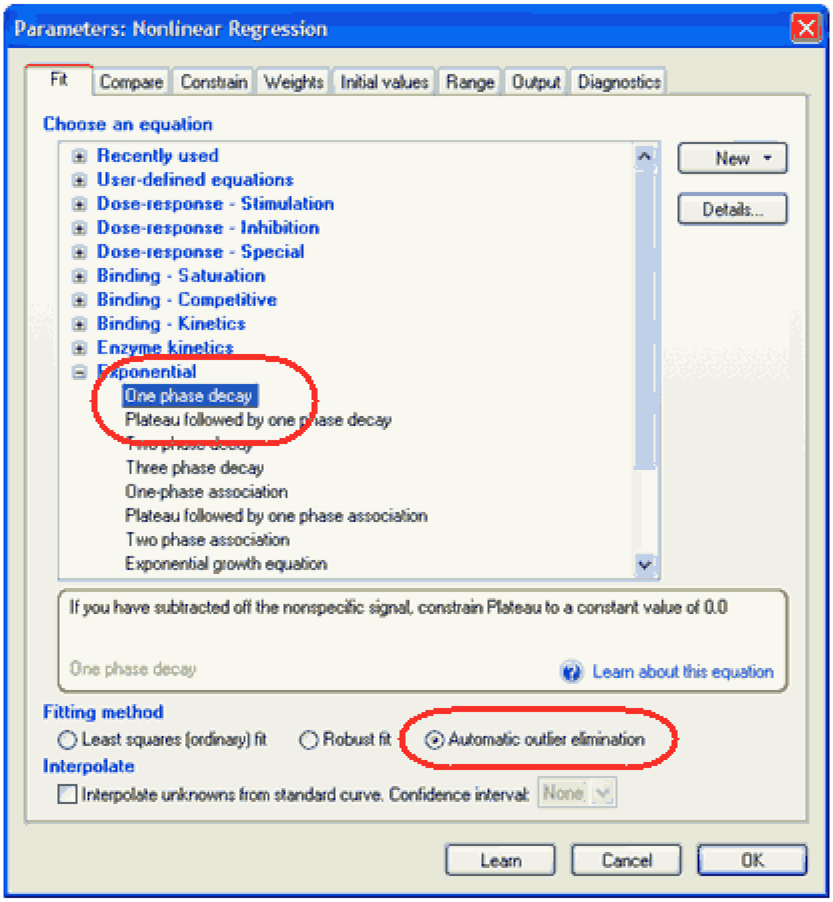
Alternatively, click the shortcut button for nonlinear regression.



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.

Also choose automatic outlier elimination. Learn about when outlier elimination is helpful 66, when it should be avoided 67, and how it works 69.



Choose an equation

- ⊞ Recently used
- ⊞ User-defined equations
- ⊞ Dose-response - Stimulation
- ⊞ Dose-response - Inhibition
- ⊞ Dose-response - Special
- ⊞ Binding - Saturation
- ⊞ Binding - Competitive
- ⊞ Binding - Kinetics
- ⊞ Enzyme kinetics
- ⊟ Exponential
  - One phase decay
  - Plateau followed by one phase decay
  - Two phase decay
  - Three phase decay
  - One-phase association
  - Plateau followed by one phase association
  - Two phase association
  - Exponential growth equation

If you have subtracted off the nonspecific signal, constrain Plateau to a constant value of 0.0

One phase decay

Learn about this equation

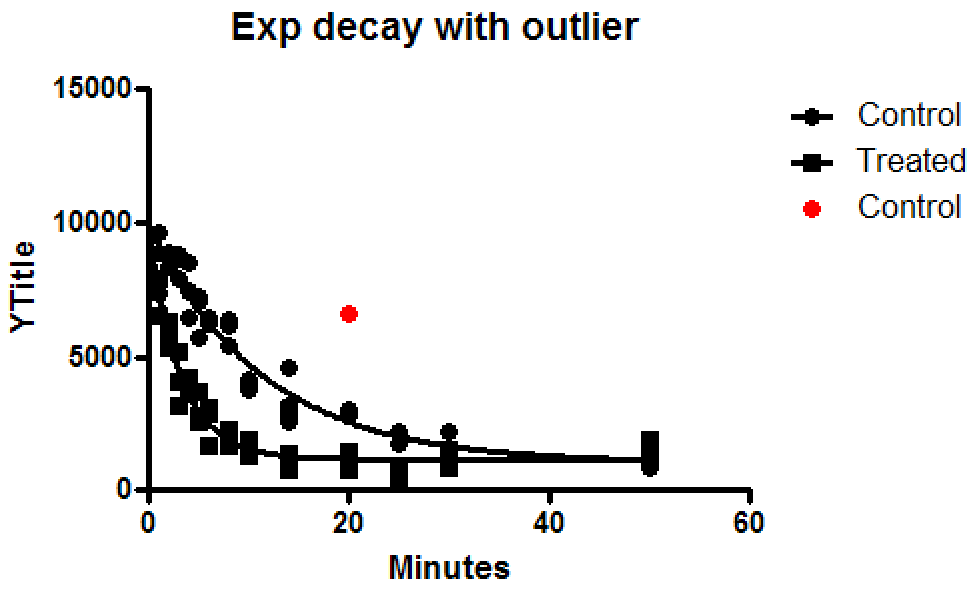
Fitting method

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

Interpolate

□ Interpolate unknowns from standard curve. Confidence interval: None

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.

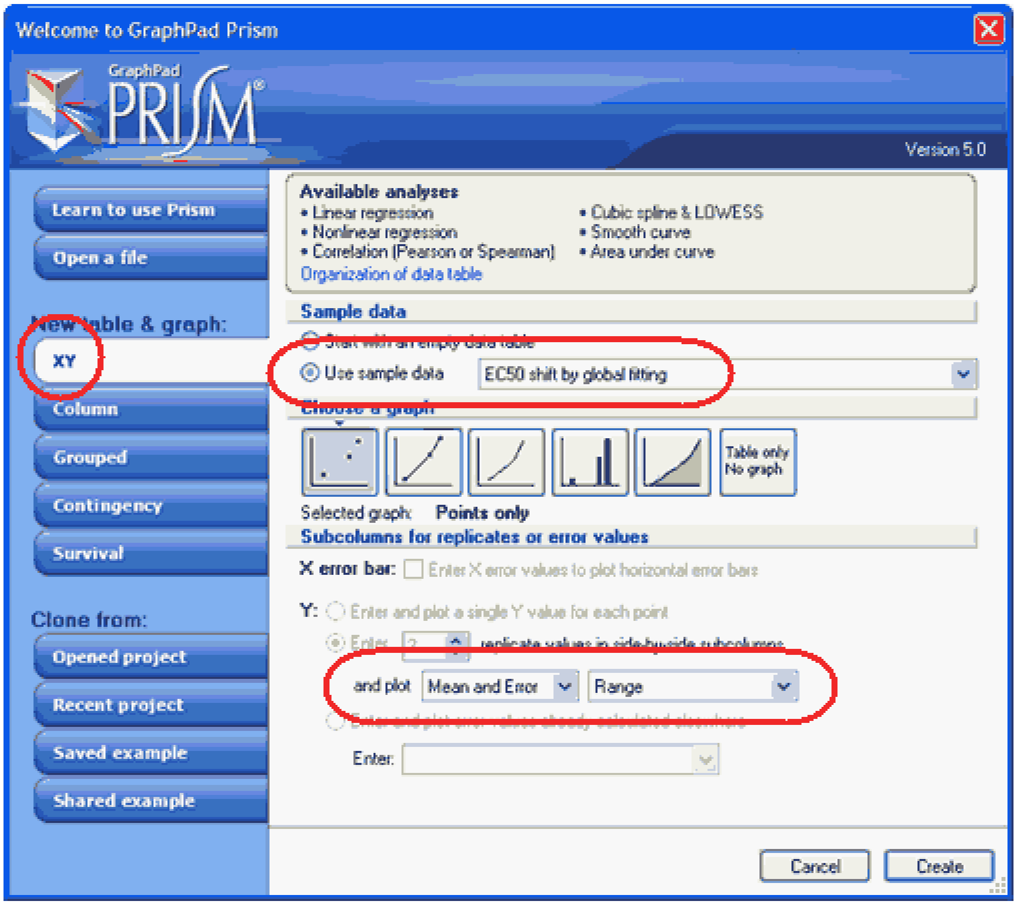
Before accepting the results, think about why the point was an outlier. Remember, not all outliers are "bad" points 68.

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 284.

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, and select the sample data "EC50 shift by global fitting". Choose to plot the mean with error bars defined by the range.



Welcome to GraphPad Prism

Version 5.0

Available analyses
- Linear regression
- Nonlinear regression
- Correlation (Pearson or Spearman)
- Cubic spline & LOWESS
- Smooth curve
- Area under curve

Organization of data table

New table & graph: XY

Sample data
○ Start with an empty data table
● Use sample data: EC50 shift by global fitting

Choose a graph

Selected graph: Points only

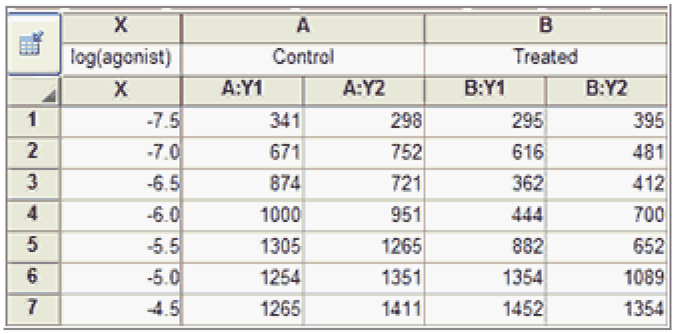
Subcolumns for replicates or error values

X error bar: □ Enter X error values to plot horizontal error bars

Y: ○ Enter and plot a single Y value for each point
● Enter 2 replicate values in side-by-side subcolumns
and plot Mean and Error, Range
○ Enter and plot error values already calculated elsewhere
Enter:

2. Inspect the data

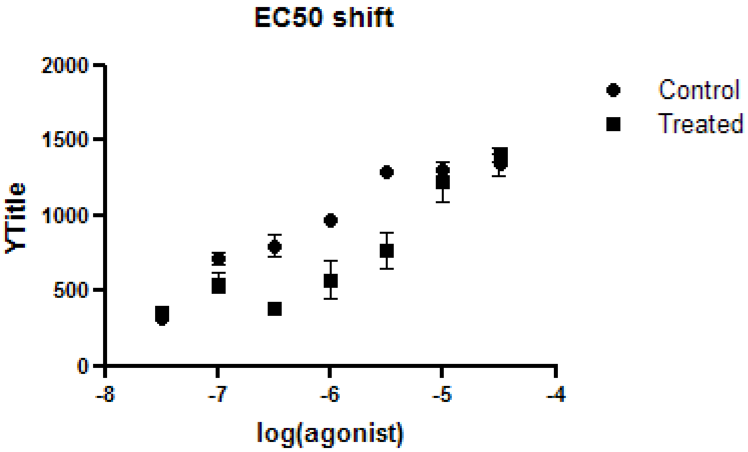
The sample data may be partly covered by a floating note explaining how to fit the data (for people who are not reading this help page). You can move the floating note out of the way, or minimize it.



	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. The Y values are responses, in duplicate, in two conditions.

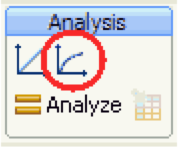
3. View the graph



4. Choose nonlinear regression

Click Analyze and choose Nonlinear regression from the list of XY analyses.

Alternatively, click the shortcut button for nonlinear regression.

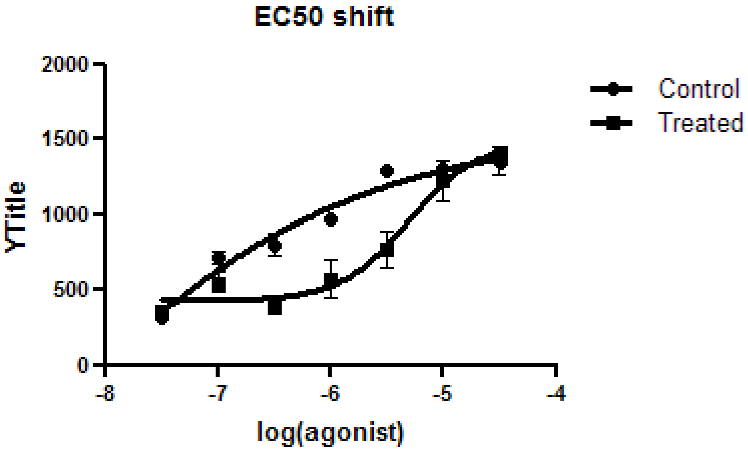


5. Choose a model

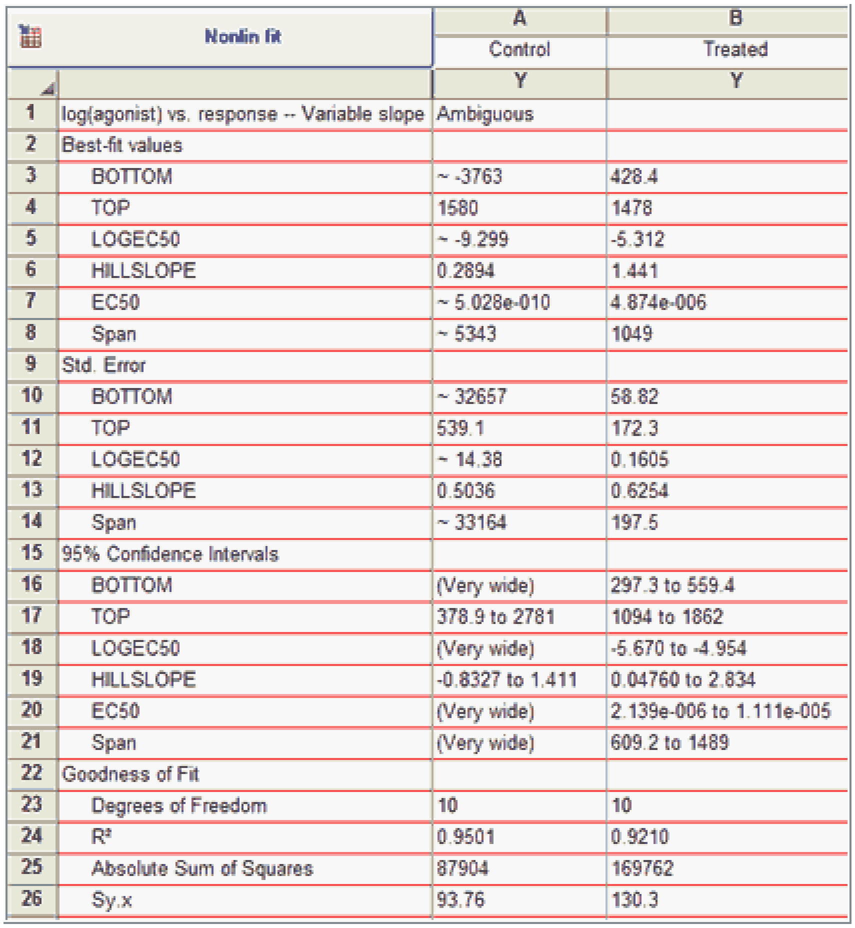
On the Fit tab of the nonlinear regression dialog, open the panel of inhibitory dose-response models and choose: log(inhibitor) vs. response -- variable slope.

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	A	B
		Control	Treated
		Y	Y
1	log(agonist) vs. response -- Variable slope	Ambiguous	
2	Best-fit values		
3	BOTTOM	~ -3763	428.4
4	TOP	1580	1478
5	LOGEC50	~ -9.299	-5.312
6	HILLSLOPE	0.2894	1.441
7	EC50	~ 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	LOGEC50	~ 14.38	0.1605
13	HILLSLOPE	0.5036	0.6254
14	Span	~ 33164	197.5
15	95% Confidence Intervals		
16	BOTTOM	(Very wide)	297.3 to 559.4
17	TOP	378.9 to 2781	1094 to 1862
18	LOGEC50	(Very wide)	-5.670 to -4.954
19	HILLSLOPE	-0.8327 to 1.411	0.04760 to 2.834
20	EC50	(Very wide)	2.139e-006 to 1.111e-005
21	Span	(Very wide)	609.2 to 1489
22	Goodness of Fit		
23	Degrees of Freedom	10	10
24	R²	0.9501	0.9210
25	Absolute Sum of Squares	87904	169762
26	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.

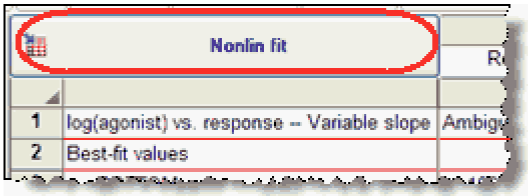
The data do not define a bottom plateau for the control (circles) data set, so its best-fit value is ambiguous. The EC50 is the concentration that gives a response half way between the bottom and top plateaus of the curve. If the bottom is ambiguous, so is the EC50.

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

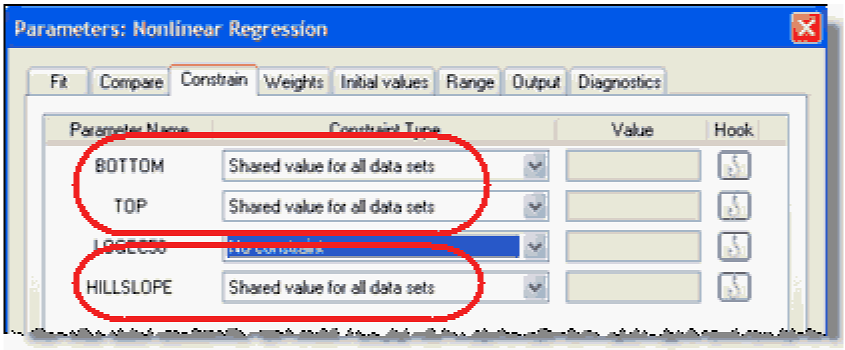
You can get much better results from this data set if you are willing to assume that that the top and bottom plateaus, and the slope, are the same under control and treated conditions. In other words, you assume that the treatment shifts the EC50 but doesn't change the basal response, the maximum response, or the Hill slope.

Return to the nonlinear regression dialog by clicking the button in the upper left of the results table.



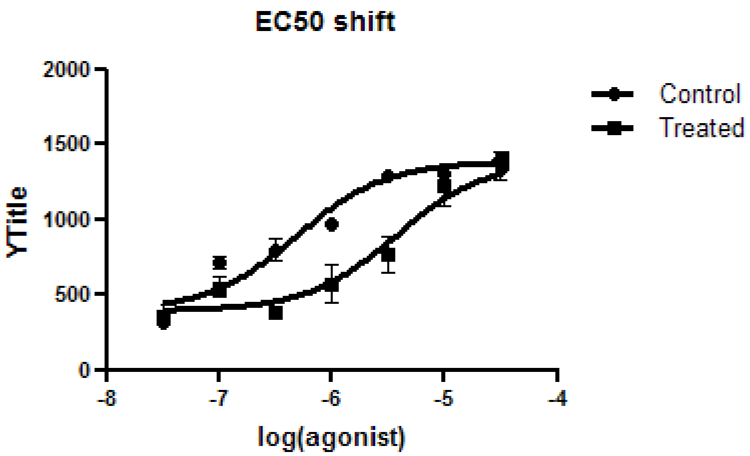
	Nonlin fit	R
1	log(agonist) vs. response -- Variable slope	Ambig
2	Best-fit values	

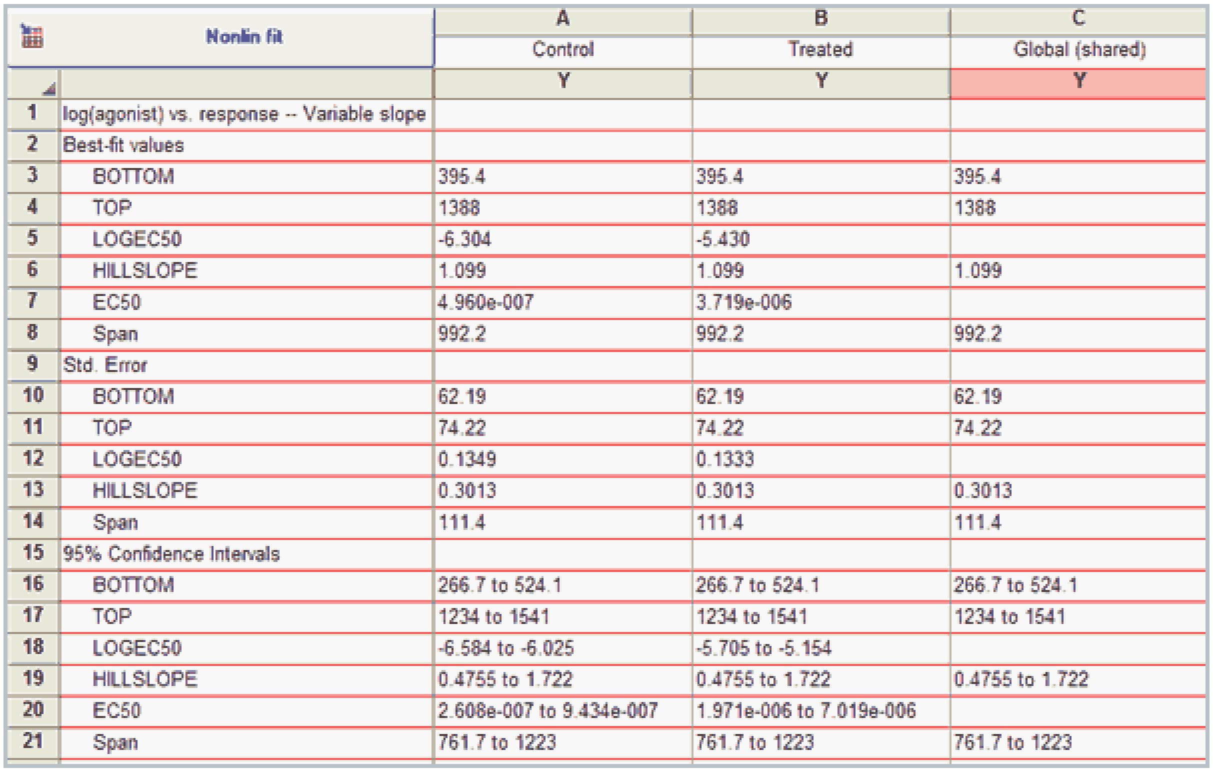
Go to the constraints tab and choose to share the value of Bottom, Top, and HillSlope. When you share these parameters, Prism fits the data sets globally to find one best-fit value for Bottom, Top and HillSlope (for both data sets) and separate best-fit values for the logEC50.



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	A	B	C
		Control	Treated	Global (shared)
		Y	Y	Y
1	log(agonist) vs. response -- Variable slope			
2	Best-fit values			
3	BOTTOM	395.4	395.4	395.4
4	TOP	1388	1388	1388
5	LOGEC50	-6.304	-5.430	
6	HILLSLOPE	1.099	1.099	1.099
7	EC50	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	LOGEC50	0.1349	0.1333	
13	HILLSLOPE	0.3013	0.3013	0.3013
14	Span	111.4	111.4	111.4
15	95% Confidence Intervals			
16	BOTTOM	266.7 to 524.1	266.7 to 524.1	266.7 to 524.1
17	TOP	1234 to 1541	1234 to 1541	1234 to 1541
18	LOGEC50	-6.584 to -6.025	-5.705 to -5.154	
19	HILLSLOPE	0.4755 to 1.722	0.4755 to 1.722	0.4755 to 1.722
20	EC50	2.608e-007 to 9.434e-007	1.971e-006 to 7.019e-006	
21	Span	761.7 to 1223	761.7 to 1223	761.7 to 1223

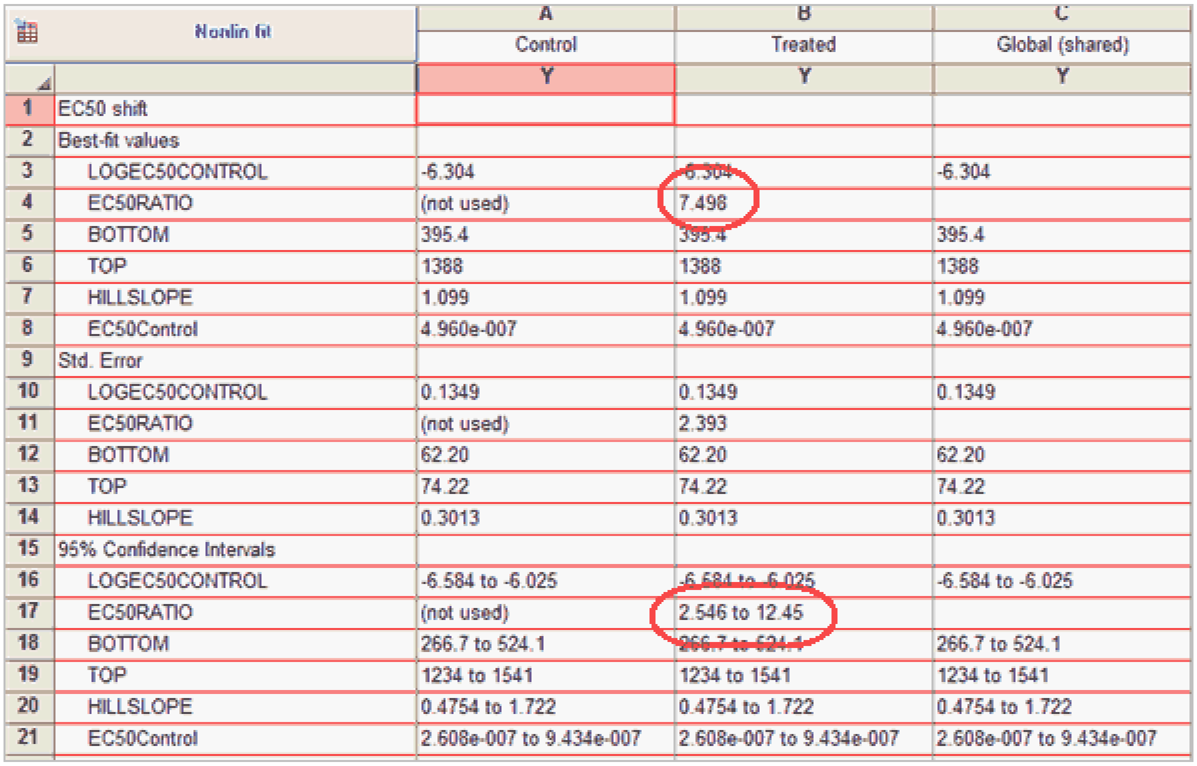
The fit is no longer labeled 'ambiguous' and the confidence intervals are much tighter.

9. View the revised graph and results

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" (also in the specialized dose response section). 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.



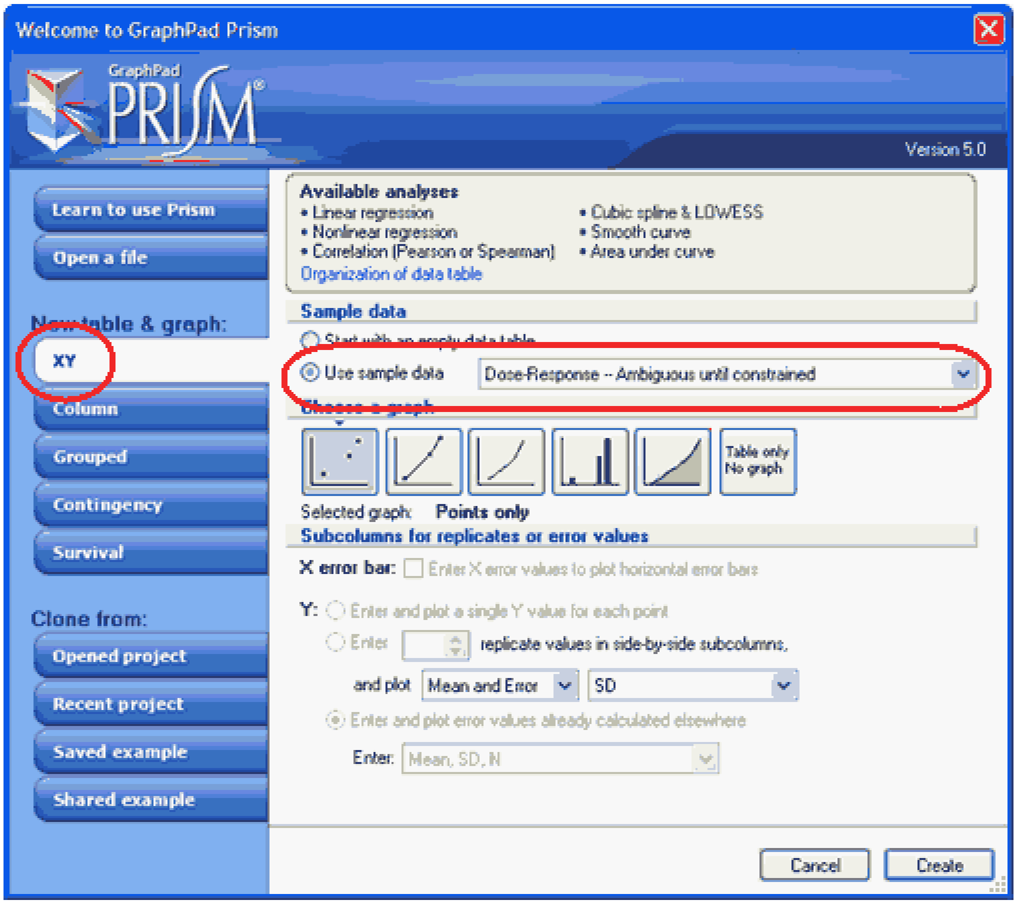
	Nonlin fit	A	B	C
		Control	Treated	Global (shared)
		Y	Y	Y
1	EC50 shift			
2	Best-fit values			
3	LOGEC50CONTROL	-6.304	-6.304	-6.304
4	EC50RATIO	(not used)	7.498	
5	BOTTOM	395.4	395.4	395.4
6	TOP	1388	1388	1388
7	HILLSLOPE	1.099	1.099	1.099
8	EC50Control	4.960e-007	4.960e-007	4.960e-007
9	Std. Error			
10	LOGEC50CONTROL	0.1349	0.1349	0.1349
11	EC50RATIO	(not used)	2.393	
12	BOTTOM	62.20	62.20	62.20
13	TOP	74.22	74.22	74.22
14	HILLSLOPE	0.3013	0.3013	0.3013
15	95% Confidence Intervals			
16	LOGEC50CONTROL	-6.584 to -6.025	-6.584 to -6.025	-6.584 to -6.025
17	EC50RATIO	(not used)	2.546 to 12.45	
18	BOTTOM	266.7 to 524.1	266.7 to 524.1	266.7 to 524.1
19	TOP	1234 to 1541	1234 to 1541	1234 to 1541
20	HILLSLOPE	0.4754 to 1.722	0.4754 to 1.722	0.4754 to 1.722
21	EC50Control	2.608e-007 to 9.434e-007	2.608e-007 to 9.434e-007	2.608e-007 to 9.434e-007

This equation was designed to do exactly what is needed for this example. Read about how this equation was set up 137, so you can construct your own equations when necessary.

Example: Ambiguous fit (dose-response)

1. Create the data table

From the Welcome or New Table dialog, choose to create XY data table, select the sample data "Dose-Response - Ambiguous until constrained".



Welcome to GraphPad Prism

GraphPad PRISM®

Version 5.0

Available analyses

- Linear regression
- Nonlinear regression
- Correlation (Pearson or Spearman)
- Cubic spline & LOWESS
- Smooth curve
- Area under curve

Organization of data table

Sample data

○ Start with an empty data table

● Use sample data: Dose-Response -- Ambiguous until constrained

Choose a graph

Selected graph: **Points only**

Subcolumns for replicates or error values

X error bar: □ Enter X error values to plot horizontal error bars

Y: ○ Enter and plot a single Y value for each point

○ Enter [ ] replicate values in side-by-side subcolumns,

and plot Mean and Error, SD

● Enter and plot error values already calculated elsewhere

Enter: Mean, SD, N

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.

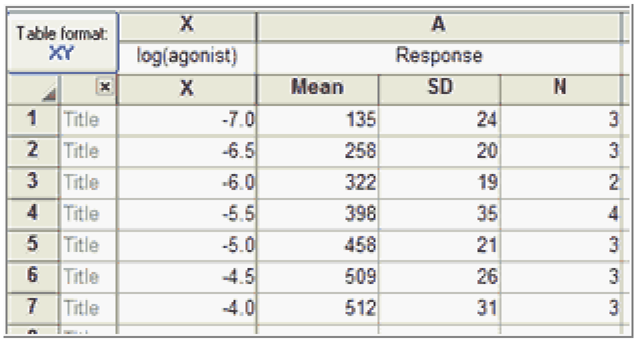
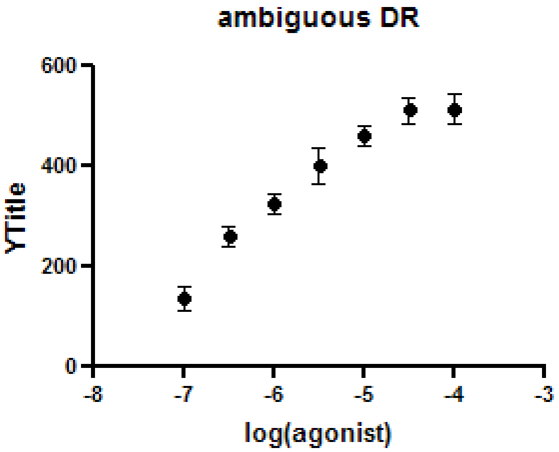


Table format: XY		X	A		
		log(agonist)	Response		
		X	Mean	SD	N
1	Title	-7.0	135	24	3
2	Title	-6.5	258	20	3
3	Title	-6.0	322	19	2
4	Title	-5.5	398	35	4
5	Title	-5.0	458	21	3
6	Title	-4.5	509	26	3
7	Title	-4.0	512	31	3

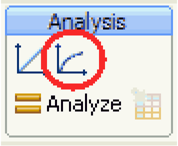
3. View the graph



4. Choose nonlinear regression

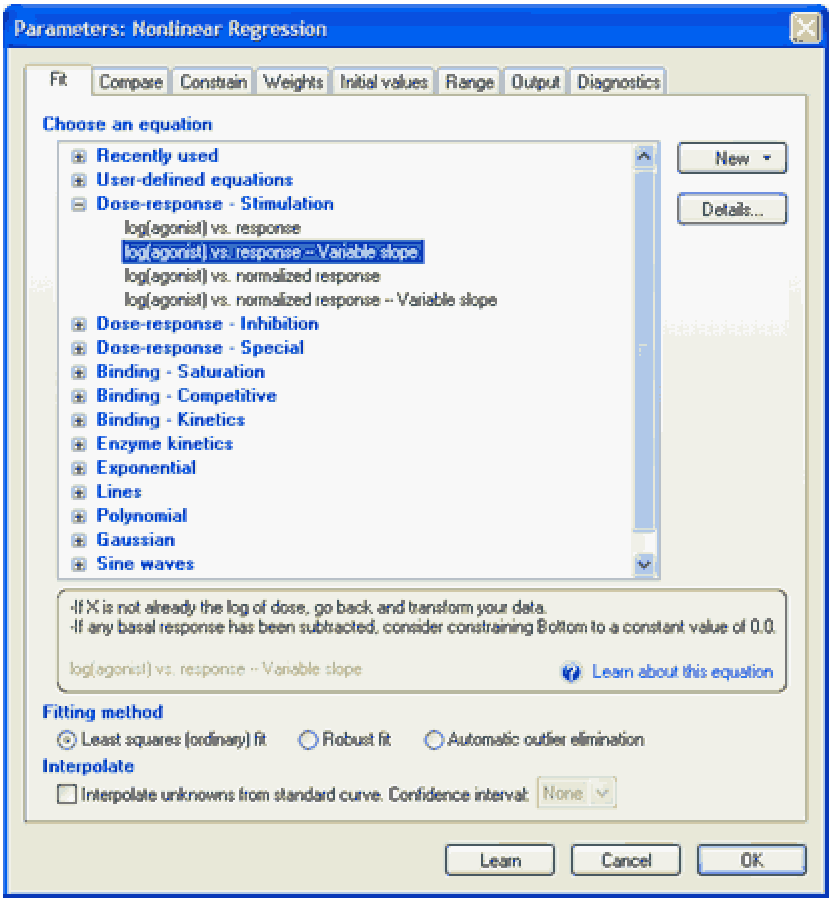
Click Analyze and choose from the list of XY analyses.

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.



-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

Fitting method

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

Interpolate

□ Interpolate unknowns from standard curve. Confidence interval: None

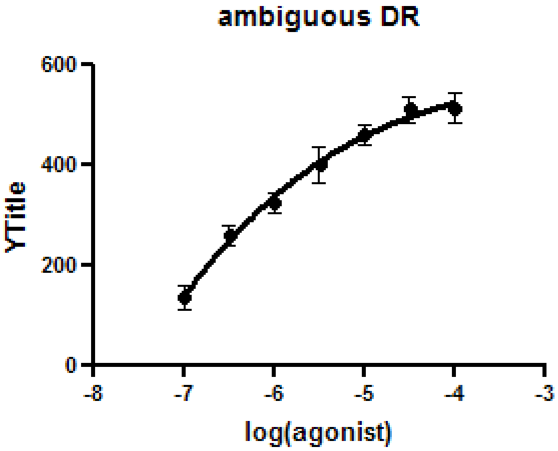
Learn more about fitting dose-response curves 120.

For this example, leave all the other settings to their default values.

Click OK to see the curves superimposed on the graph.

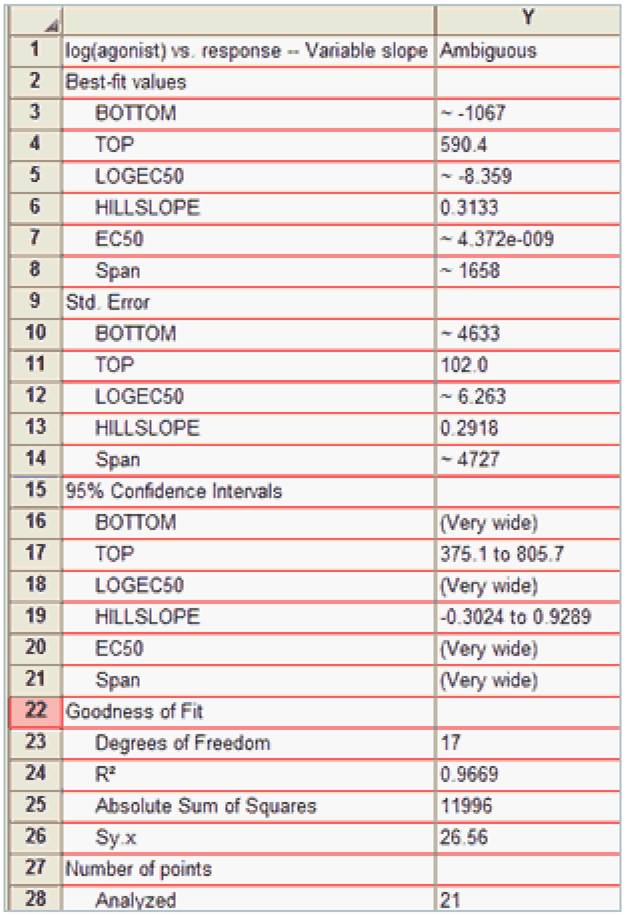
6. Inspect the graph

The curve goes through the point nicely, and looks fine.



7. Inspect the results

Go to the results page, and view the results.



		Y
1	log(agonist) vs. response -- Variable slope	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% Confidence Intervals	
16	BOTTOM	(Very wide)
17	TOP	375.1 to 805.7
18	LOGEC50	(Very wide)
19	HILLSLOPE	-0.3024 to 0.9289
20	EC50	(Very wide)
21	Span	(Very wide)
22	Goodness of Fit	
23	Degrees of Freedom	17
24	R^2	0.9669
25	Absolute Sum of Squares	11996
26	Sy.x	26.56
27	Number of points	
28	Analyzed	21

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 276.

Note that 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.

Also note that 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.

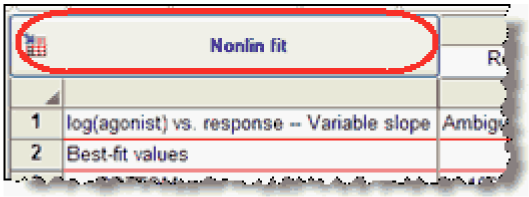
Even though the curve is close to the points, the best-fit parameter values are useless.

8. 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.

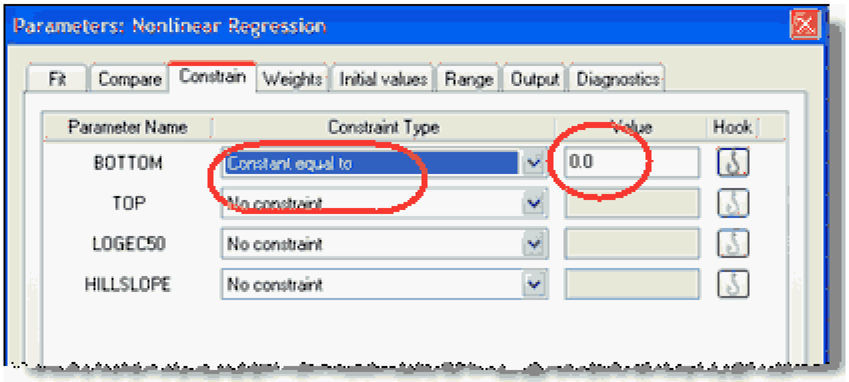
These data were calculated so the basal nonspecific response was subtracted away. This means that you know that the response (Y) at very low concentrations of agonist (very low values of X) has to be zero. Prism needs to know this to fit the data sensibly.

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.



1	log(agonist) vs. response -- Variable slope	Ambigu
2	Best-fit values	

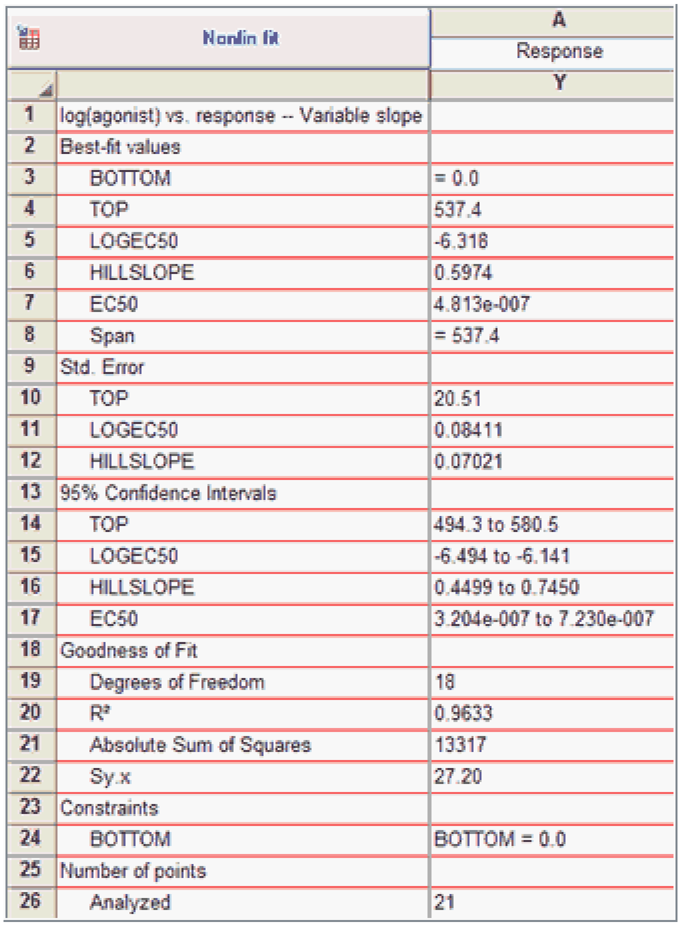
Go to the Constrain tab, and check constrain the parameter Bottom to have a constant value which you set to 0.0.



Parameter Name	Constraint Type	Value	Hook
BOTTOM	Constant equal to	0.0	
TOP	No constraint		
LOGEC50	No constraint		
HILLSLOPE	No constraint		

9. 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	A
		Response
		Y
1	log(agonist) vs. response -- Variable slope	
2	Best-fit values	
3	BOTTOM	= 0.0
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% Confidence Intervals	
14	TOP	494.3 to 580.5
15	LOGEC50	-6.494 to -6.141
16	HILLSLOPE	0.4499 to 0.7450
17	EC50	3.204e-007 to 7.230e-007
18	Goodness of Fit	
19	Degrees of Freedom	18
20	R²	0.9633
21	Absolute Sum of Squares	13317
22	Sy.x	27.20
23	Constraints	
24	BOTTOM	BOTTOM = 0.0
25	Number of points	
26	Analyzed	21

Nonlinear regression with Prism

Choosing a built-in model

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 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.

If you entered concentrations, instead of the logarithm of concentrations, you can perform the transformation with Prism 118.

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.

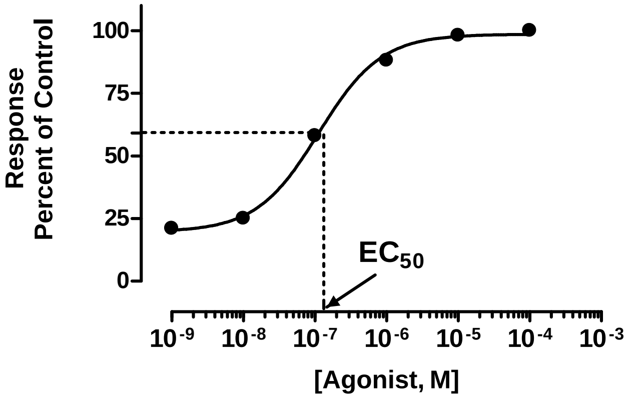
You can transform the Y values to new units by multiplying or dividing by a constant. Use Prism's Transform analysis for this. Transforming to new units will not fundamentally change the results of a curve fit.

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.

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

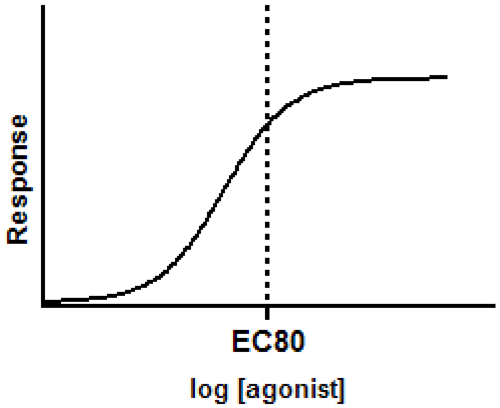
The pEC50 is defined as the negative logarithm of the EC50. If the EC50 equals 1 micromolar (10-6 molar), the log EC50 is –6 and the pEC50 is 6. There is no particular advantage to expressing potency this way, but it is customary in some fields.

The IC50

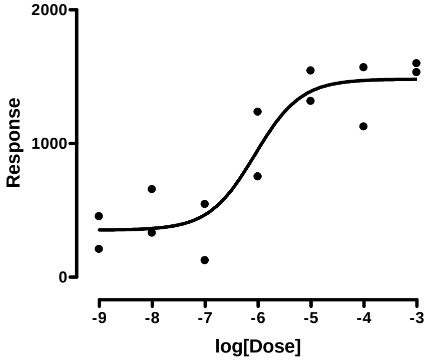
In many experiments, you vary the concentration of an inhibitor. With more inhibitor, the response decreases, so the dose-response curve goes downhill. With such experiments, the midpoint is often called the IC50 ("I" for inhibition) rather than the EC50 ("E" for effective). This is purely a difference in which abbreviation is used, with no fundamental difference.

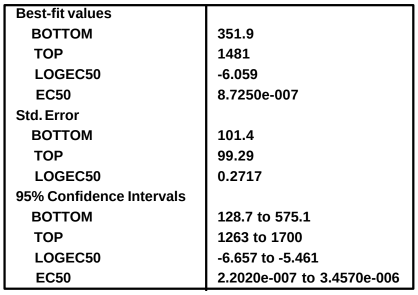
ECanything

A simple rearrangement of the equation lets you fit EC80 (or EC90 or ECanything) instead of the EC50. Prism includes an equation 140 that fits any EC value directly.



Two ways to compute the 95% confidence interval of the EC50





Best-fit values	
BOTTOM	351.9
TOP	1481
LOGEC50	-6.059
EC50	8.7250e-007
Std. Error	
BOTTOM	101.4
TOP	99.29
LOGEC50	0.2717
95% Confidence Intervals	
BOTTOM	128.7 to 575.1
TOP	1263 to 1700
LOGEC50	-6.657 to -5.461
EC50	2.2020e-007 to 3.4570e-006

The sample data above were fit to a dose-response curve with a Hill slope of 1. The best-fit value for logEC50 is -6.059. Converting to the EC50 is no problem – simply take the antilog, which is 0.87 mM.

The standard error of the logEC50 is 0.2717. It is used to calculate a 95% confidence interval, which ranges from -6.657 to -5.461. Take the antilog of both of those limits to express that confidence interval on a concentration scale -- from 0.22 to 3.46 mM. This is the interval that Prism presents. Note that it is **not** centered on the best-fit value (0.87 mM). Switching from linear to log scale converted the symmetrical confidence interval into an asymmetrical interval.

If you fit the same data to an equation describing a dose-response curve in terms of the EC50 rather than the logEC50, the EC50 remains 0.87 mM. But now Prism computes the SE of the EC50 (0.5459 mM), and uses this to compute the 95% confidence interval of the EC50, which ranges from -0.3290 to +2.074 mM. Note that the lower limit of the confidence interval is negative! Since the EC50 is a concentration, negative values are nonsense. Even setting aside the negative portion of the confidence interval, it includes all values from zero on up, which isn't terribly useful.

The problem is that the uncertainty of the EC50 really isn't symmetrical, especially when you space your doses equally on a log scale. It only makes sense to compute the 95% CI of the logEC50, and then transform both confidence limits to a concentration scale, knowing that the confidence interval will not be symmetrical on the concentration scale.

Do not transform the standard error of the logEC50

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.

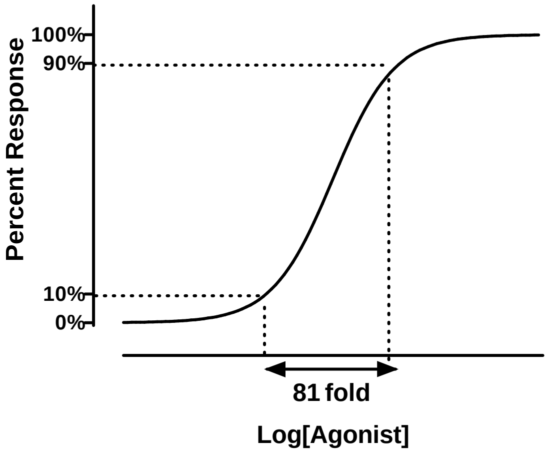
In the example, the SE of the logEC50 is 0.2717. The antilog of 0.2717 equals 1.869. What does this mean? It certainly is not the SE of the EC50. The SE does not represent a point on the axis; rather it represents a distance along the axis. A distance along a log axis does not represent a consistent distance along a linear (standard) axis. For example, increasing the logEC50 1 unit from -9 to -8 increases the EC50 9nM, while increasing the logEC50 1 unit from -3 to -2 increases the EC50 by 9 mM, which equals 9,000,000 nM.

Averaging the EC50 from several experiments

The uncertainty is symmetrical when you express the midpoint of a dose-response curve as a logEC50, but is far from symmetrical (often) 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 logEC50 values.

The usual steepness of a dose-response curve

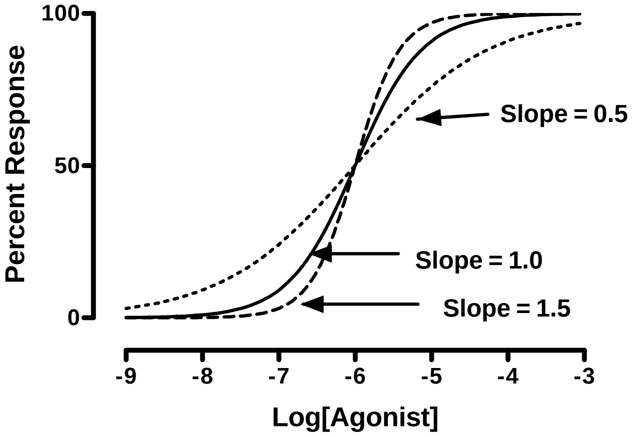
Many dose-response curves follow the shape of a receptor binding curve. As shown below, 81 times more agonist is needed to achieve 90% response than a 10% response.



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.

To choose a dose-response model in Prism, you need to answer three questions:

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?

A huge variety of dose-response curves follow the same steepness as receptor binding, so have a Hill slope 115 of 1.00 (for stimulation) or -1.00 (for inhibition). 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.

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.

Normalized or not?

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.

One way to solve the problem is to constrain the Top or Bottom, or both, to control values.

An alternative is to normalize your data so responses run from 0 to 100, and then choose a "normalized response" model. These models don't fit the bottom and top plateaus, but rather assume that the bottom plateau is 0 and the top plateau is 100. Only choose a 'normalized response' equation when you have determined the values that define 0 and 100 very precisely.

Just because you have normalized your data to run from 0 to 100, you don't have to pick a 'normalized response' model.

Special dose-response models

Prism has a set of special models used for special dose-response situations:

Asymmetrical (five parameter) 128

Biphasic dose-response 130

Bell-shaped dose-response 131

Operational model - Depletion 132

Operational model - Partial agonist 134

Gaddum/Schild EC50 shift 136

Allosteric EC50 shift 138

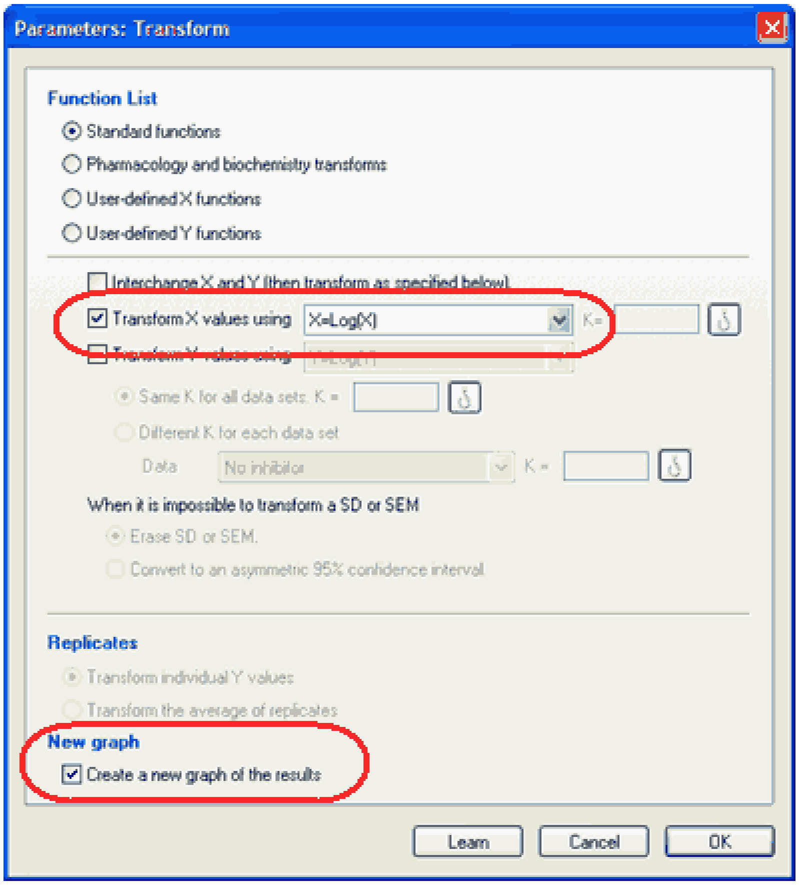
ECanything 140

Transforming X values with Prism

All the dose-response equations built-in to Prism expect the X values to be the logarithm of dose or concentration.

If you entered actual doses or concentrations, instead of their logarithms, use Prism to transform the X values.

1. From the data table, click Analyze and then choose Transform, which is the very first analysis listed on the Analyze dialog.
2. On the Transform dialog, check the option to transform X, and choose X=log(X) .At the bottom of the dialog, check the option: Create a new graph of the results.



Parameters: Transform

Function List

● Standard functions
○ Pharmacology and biochemistry transforms
○ User-defined X functions
○ User-defined Y functions

□ Interchange X and Y (then transform as specified below)
☑ Transform X values using X=Log(X) K=
□ Transform Y values using Y=Log(Y)

● Same K for all data sets. K =
○ Different K for each data set
Data No inhibitor K =

When it is impossible to transform a SD or SEM
● Erase SD or SEM.
○ Convert to an asymmetric 95% confidence interval

Replicates
● Transform individual Y values
○ Transform the average of replicates

New graph
☑ Create a new graph of the results

3. Go to the newly created graph.
4. Click Analyze, choose nonlinear regression, and then choose the dose-response curve.

Dose-response - Stimulation

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.

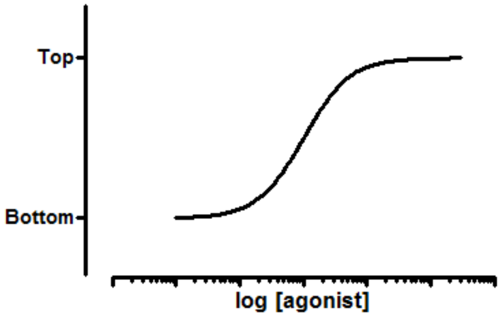
If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 118.

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.

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.

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 118.

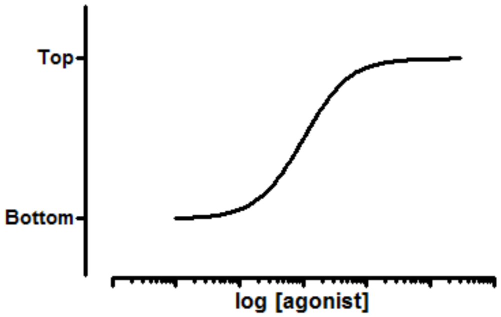
From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Stimulation" and then choose the equation "*log(Agonist) vs. response -- Variable slope*".

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.

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 120. 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

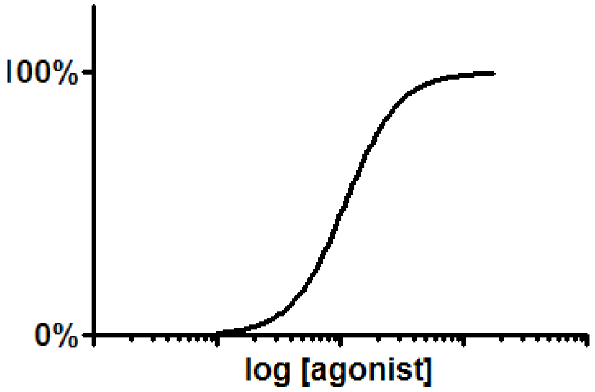
Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 118.

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.

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 120. 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

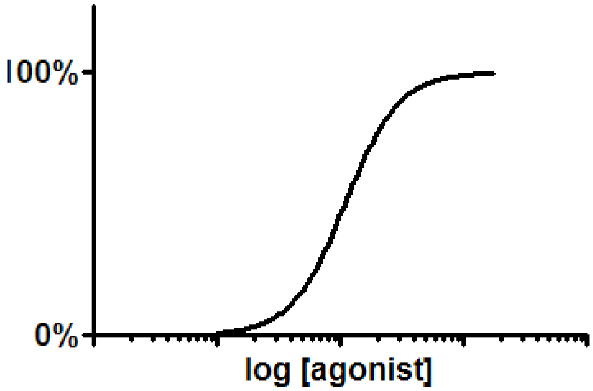
Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 118.

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.

Dose-response - Inhibition

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the EC50 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.

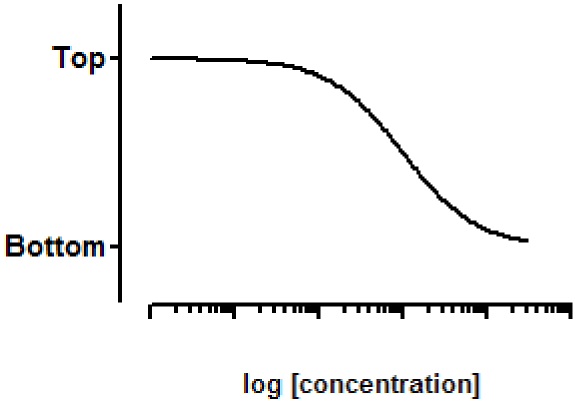
If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 118.

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

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.

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the EC50 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.

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.

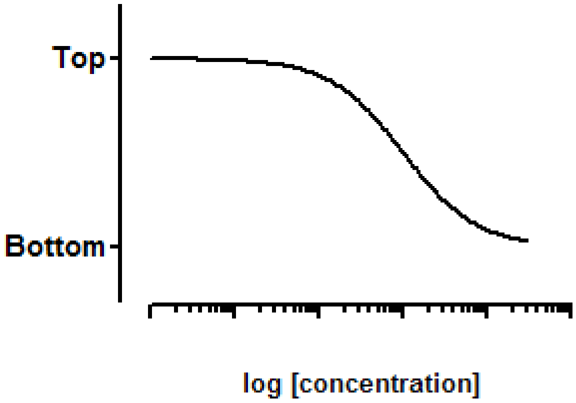
If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 118.

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

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 more negative than -1 (say -2) is steeper.

Top and **Bottom** are plateaus in the units of the Y axis.

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 EC50 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 123. 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

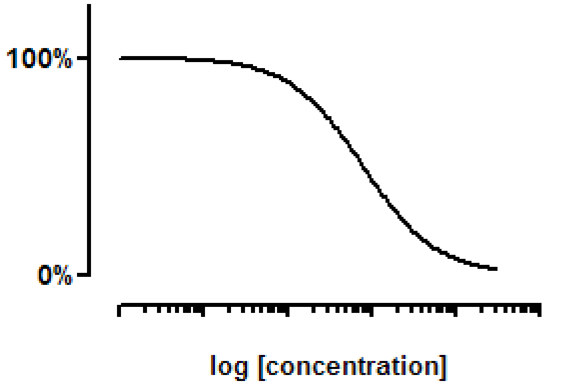
Create an XY data table. Enter the logarithm of the concentration of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 118.

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

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.

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 EC50 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 123. 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

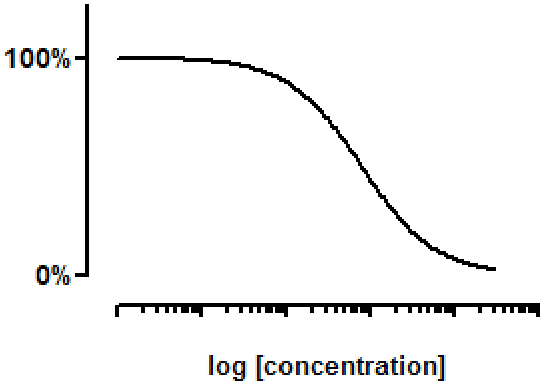
Create an XY data table. Enter the logarithm of the concentration of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to transform the X values to logs 118.

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

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 more negative than -1 (say -2) is steeper.

Dose-response -- Special

.

Introduction

The standard dose-response curve is sometimes called the **four-parameter logistic equation**. It fits the bottom and top plateaus of the curve, the EC50, 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**.

Step by step

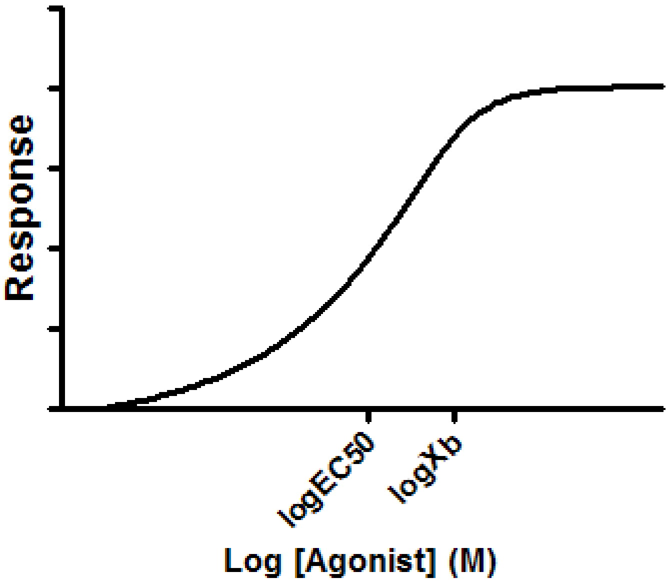
Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special. Then choose *Asymmetrical (five parameter).*

Consider constraining the Hill Slope to a constant value of 1.0 (stimulation) or -1 (inhibition).

Model

```
LogXb = LogEC50 + (1/HillSlope)*Log((2^(1/S))-1)
Numerator = Top - Bottom
Denominator = (1+10^((LogXb-X)*HillSlope))^S
Y = Bottom + (Numerator/Denominator)
```



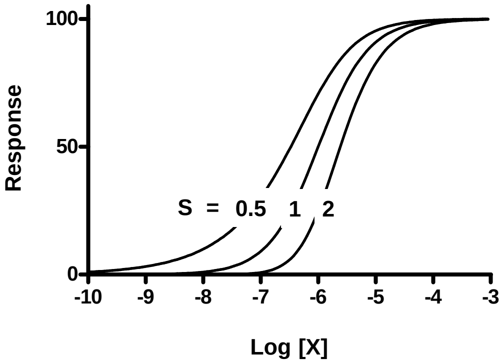
Interpret the parameters

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 the inflection point Xb (see below).

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

The inflection point is called LogXb. It is not the same as the logEC50. Prism does not fit logXb, but you can do so using this equation:

$$\mathrm{LogX_b} = \mathrm{LogEC_{50}} + \left(\frac{1}{\mathrm{HillSlope}}\right)\cdot \mathrm{Log}\left(2^{1/S}-1\right)$$

Reference

Girlado et al. (Pharmacol. Ther., 95, 21-45, 2002)

Introduction

A common deviation from the standard monotonic sigmoid shape is the biphasic sigmoid shape.

Step by step

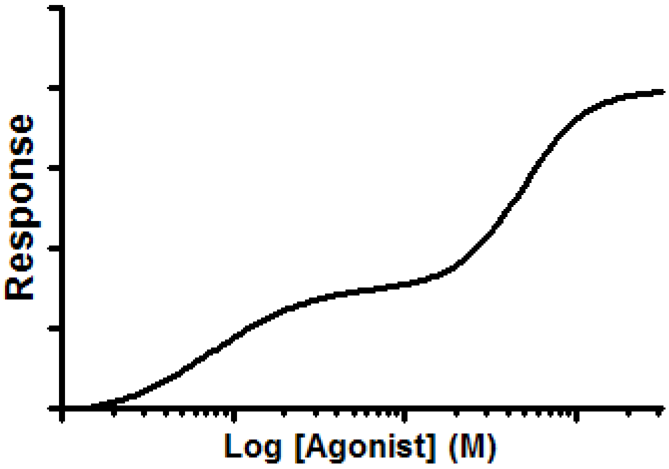
Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special. Then choose *Biphasic dose-response.*

Consider constraining nH1 and nH2 to constant values of 1.0 (stimulation) or -1 (inhibition).

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.

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

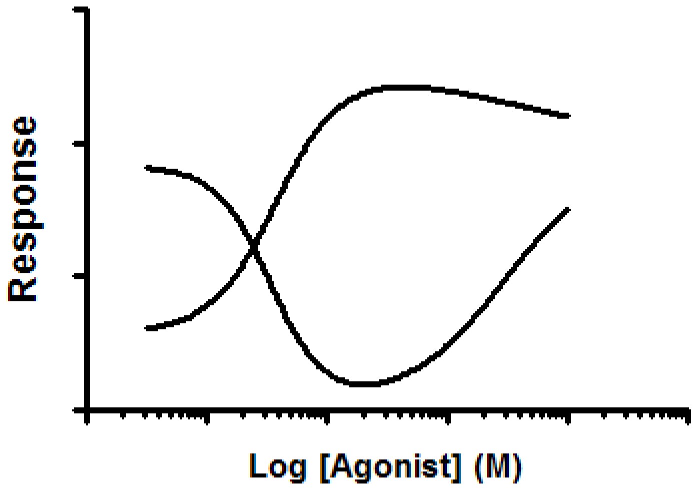
Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special. Then choose *Bell-shaped dose-response.*

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

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).

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).

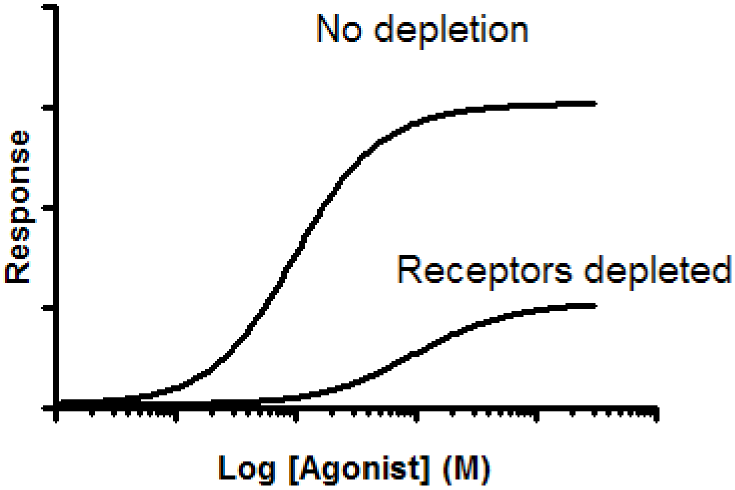
From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special. Then choose *Operational Model - Depletion.*

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 t 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

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. Then choose *Operational Model - Partial agonists.*

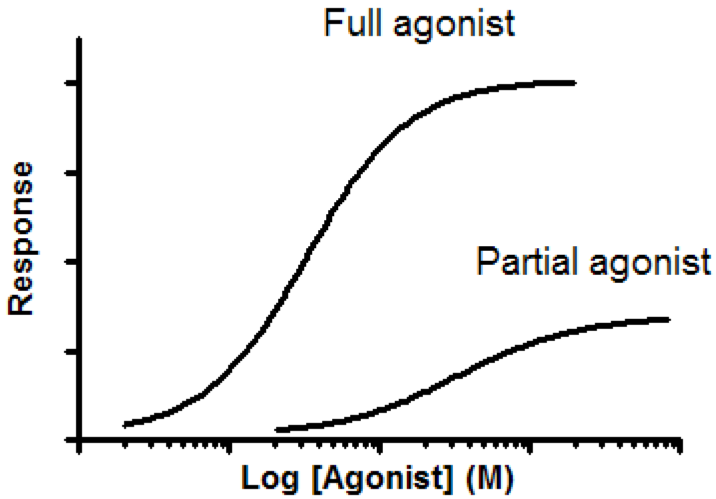
If you have subtracted off any basal response, consider constraining the parameter Basal to a constant value of zero.

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.

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.

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.

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.

Reference

Black and Leff (Proc. R. Soc. Lond. B, 220: 141-162, 1983

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

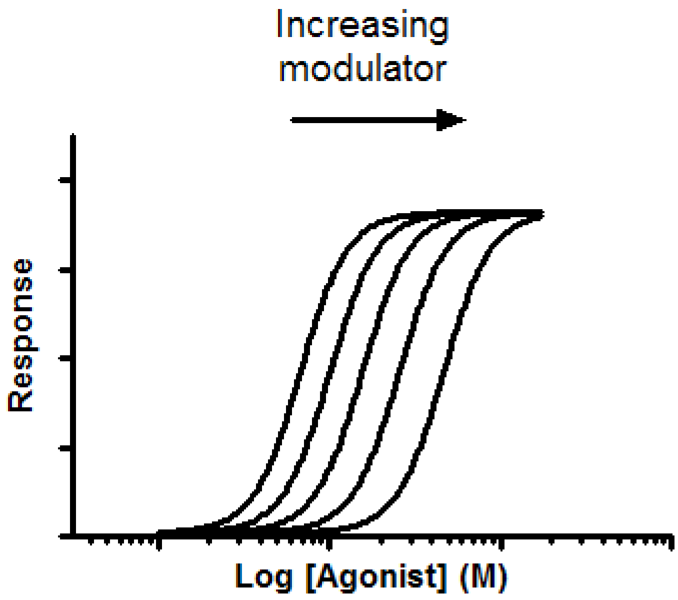
Create an XY data table. Enter the logarithm of the concentration of the agonist ligand into X. Enter response into Y in any convenient units. Enter data with no inhibitor into column A. Enter data collected with a constant concentration of inhibitor into column B. Repeat, if you have data, for column C, D, E, ..., each with a different concentration of inhibitor. Enter the inhibitor 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.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special. Then choose *Gaddum/Schild EC50 shift.*

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.

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.

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.

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.

Introduction

An 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 the two dose response curves and determines the fold shift.

Step by step

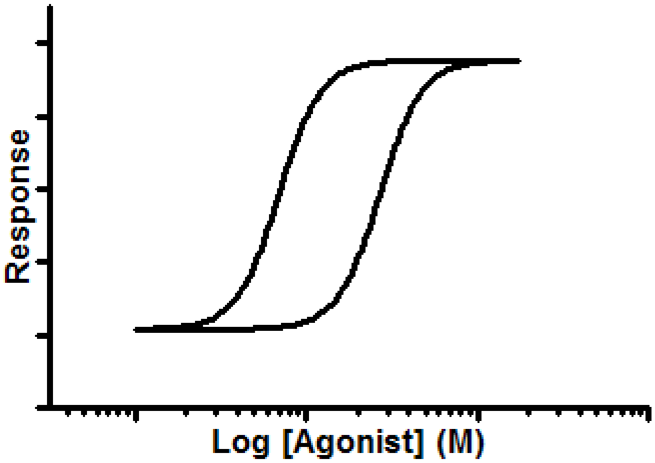
Create an XY data table. Enter the logarithm of the concentration of the agonist ligand into X. Enter response into Y in any convenient units. Enter data with no inhibitor into column A. Enter data collected with a constant concentration of inhibitor into column B.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special. Then choose *Dose shift.*

If you have subtracted off any basal 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 136 and determine the pA2.

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

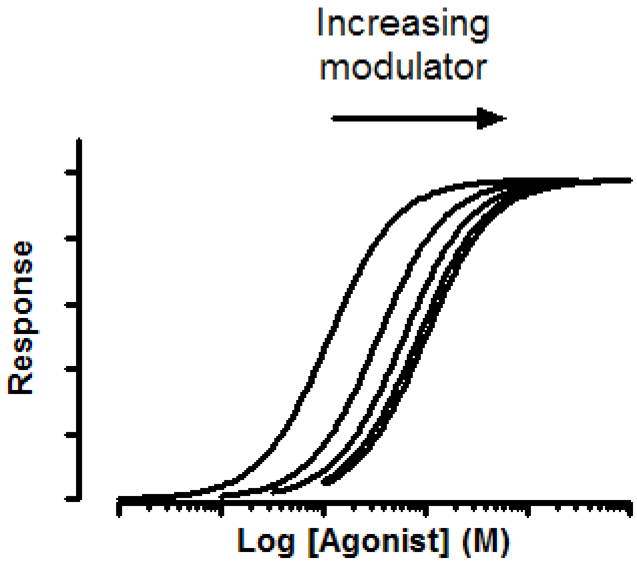
Create an XY data table. Enter the logarithm of the concentration of the agonist ligand into X. Enter response into Y in any convenient units. Enter data with no modulator into column A. Enter data collected with a constant concentration of modulator into column B. Repeat, if you have data, for column C, D, E, ..., each with a different concentration of modulator. Enter the modulator concentration (in 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.

From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special. Then choose *Allosteric EC50 shift.*

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))
```



EC50 is the concentration of agonist 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 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.

Reference

A. Christopoulos and T. Kenakin, Pharmacol Rev, 54: 323-374, 2002

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. Then choose "*log(Agonist) vs. response -- Find ECanything*"*.*

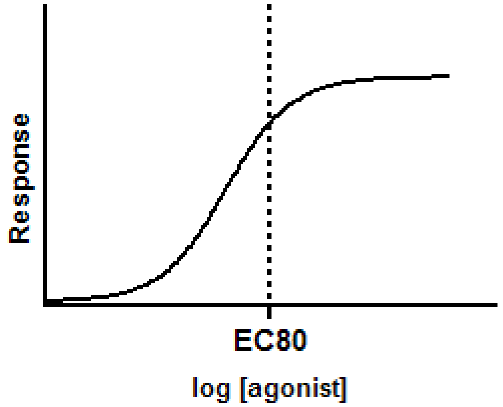
You must constrain the parameter F to have a constant value between 0 and 100. Set F to 80 if you want to fit the EC80. If you constrain F to equal 50, then this equation is the same as a variable slope dose-response curve.

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.

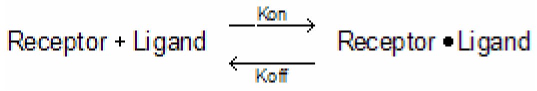
Notes

This equation can also fit inhibitory data where the curve goes downhill rather than uphill. The this case.

Receptor binding - Key concepts

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:

$$[\text{Ligand}]\cdot[\text{Receptor}]\cdot k_{on} = [\text{Ligand}\cdot\text{Receptor}]\cdot k_{off}$$

Meaning of Kd

Rearrange that equation to define the equilibrium dissociation constant Kd.

$$\frac{[\text{Ligand}]\cdot[\text{Receptor}]}{[\text{Ligand}\cdot\text{Receptor}]} = \frac{k_{off}}{k_{on}} = K_d$$

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.

$$\text{Fractional occupancy} = \frac{[\text{Ligand}\cdot\text{Receptor}]}{[\text{Total Receptor}]} = \frac{[\text{Ligand}\cdot\text{Receptor}]}{[\text{Receptor}] + [\text{Ligand}\cdot\text{Receptor}]}$$

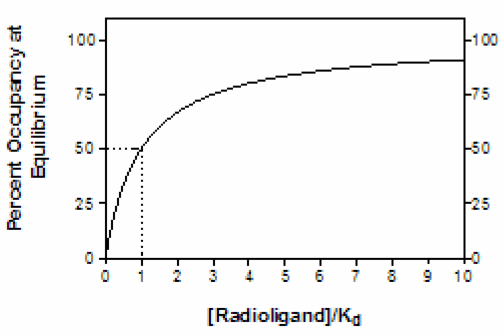
This equation is not useful, because you don't know the concentration of unoccupied receptor, [Receptor]. A bit of algebra creates a useful equation.

$$\text{Fractional occupancy} = \frac{[\text{Ligand}]}{[\text{Ligand}] + K_d}$$

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.
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.
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 149 and competition 165 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.

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 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 counted. The result is the concentration of radioligand in nM.

Receptor binding - 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 151.
- Analyze the total binding only 147, inferring the amount of nonspecific binding from the shape of the total binding curve.
- Globally fit total and nonspecific binding together 148.

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 149 (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.

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.

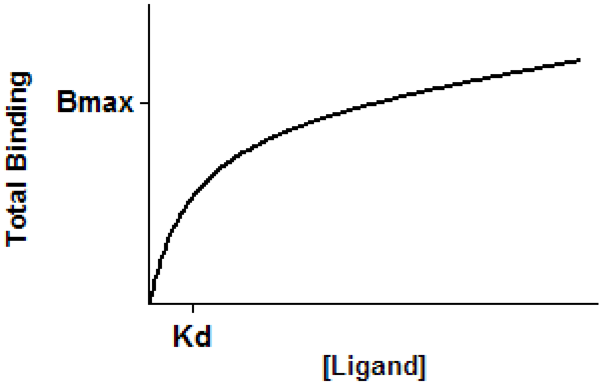
Use any convenient units for X and Y. The Kd will be reported in the same units as X, and the Bmax will be reported in the same units as Y.

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 binding 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 149.

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 151.
- Analyze the total binding only, inferring the amount of nonspecific binding from the shape of the total binding curve. Learn more 147.
- 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.*

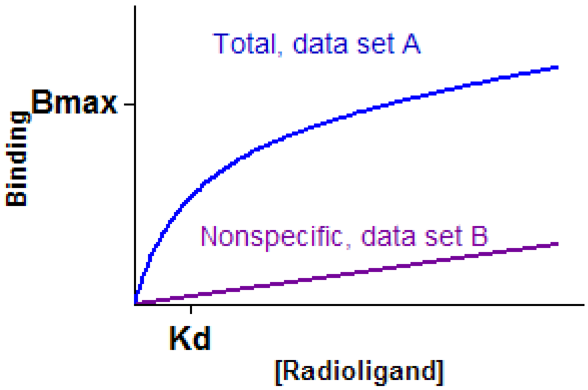
Consider constraining the parameter Background to a constant value of zero. This is the measured 'binding' when there is no radioligand binding added, so represents the counter background, if there is any.

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 61 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 binding 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.

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 147.

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.

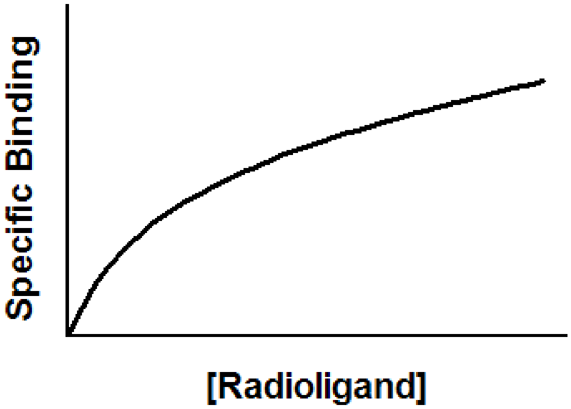
From the table of total binding, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *One site -- Total, accounting for ligand depletion.*

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 binding constant in nM. It is the radioligand concentration needed to achieve a half-maximum specific binding at equilibrium.

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 147.

Reference

This equation came from S. Swillens (Molecular Pharmacology, 47: 1197-1203, 1995)

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 147.
- Globally analyze the total and nonspecific binding at one time. Learn more. 148

Step by step

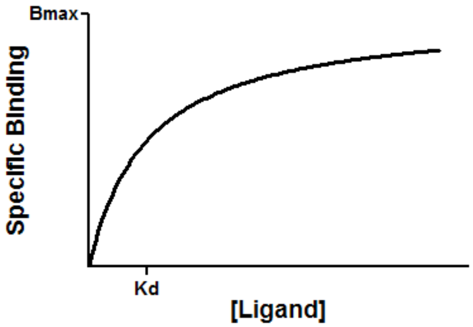
Create an XY data table. Enter radioligand concentration into X, and specific binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

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)
```



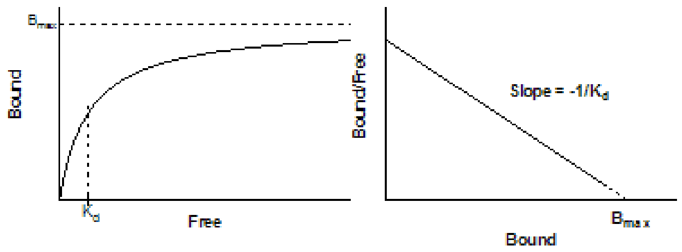
Interpret the parameters

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 binding 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 148, 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).

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 151.

Step by step

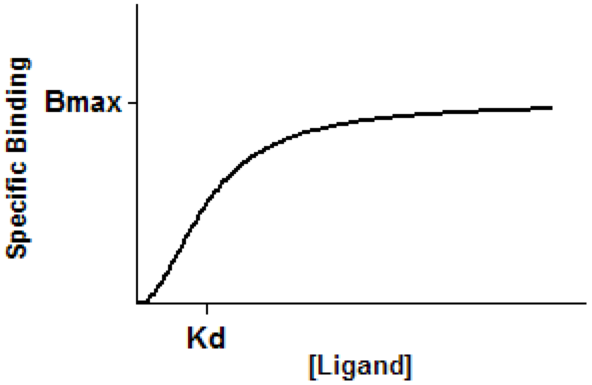
Create an XY data table. Enter radioligand concentration into X, and specific binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

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 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 binding 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.

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. 156

Step by step

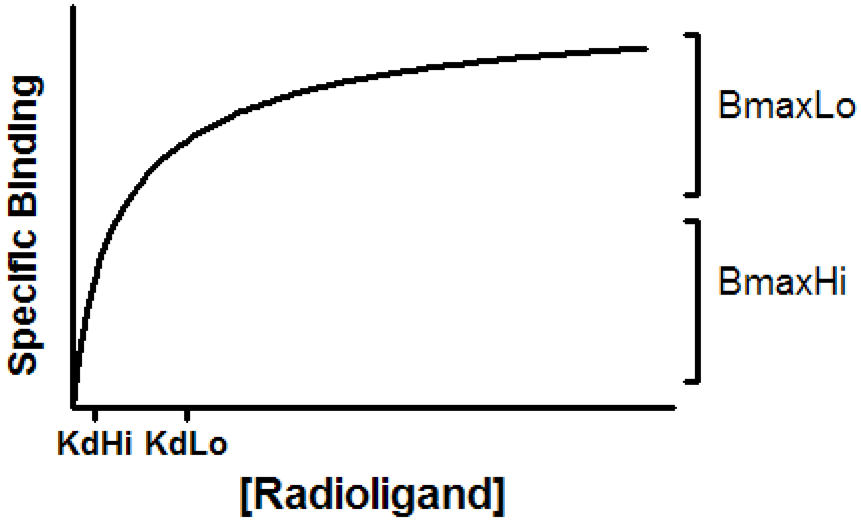
Create an XY data table. Enter radioligand concentration into X, and specific binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

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

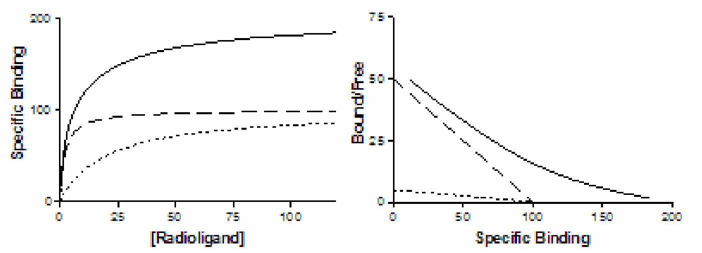
BmaxHi and **BmaxLo** are the maximum specific bindings to the two sites in the same units as Y.

KdHi and **KdLo** are the equilibrium binding 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.

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 154.
- 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.*

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

```
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 binding 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.

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

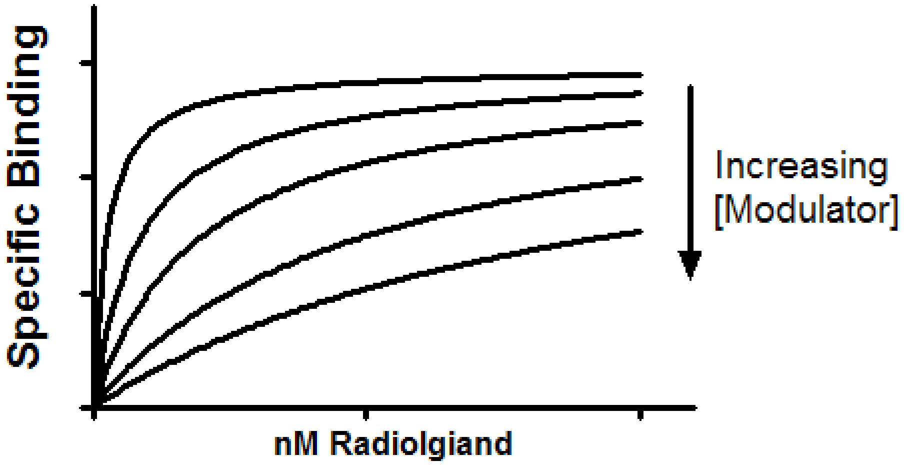
Create an XY data table. Enter the concentration of the labeled ligand into X, using any convenient units (maybe nM). Enter specific binding into Y in any convenient units. Enter data with no modulator into column A. Enter data collected with a constant concentration of modulator into column B. Repeat, if you have data, for column C, D, E, ..., each with a different concentration of modulator. Enter the modulator concentration (in 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.

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

Receptor binding - 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 receptor depletion 165.

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 166.

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 168 (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.

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.

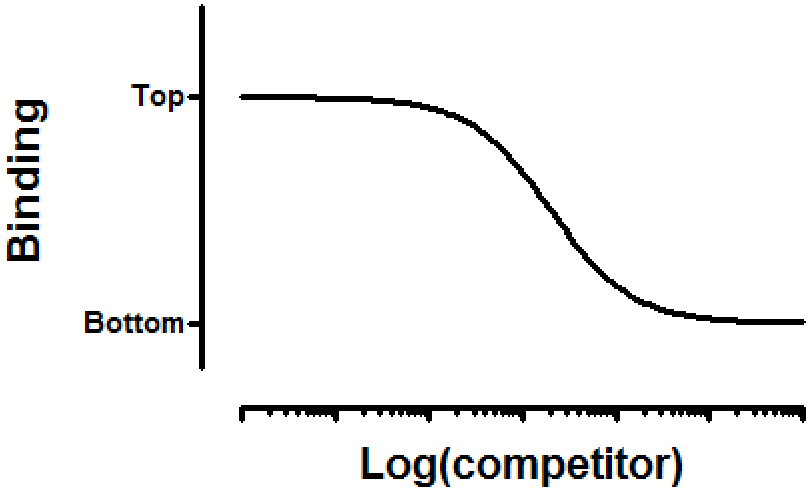
From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *One site - Fit Ki.*

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 Prussoff correction. 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 161.

The analysis assumes that you have one site, and that the binding is reversible and at equilibrium.

Introduction

You can determine the equilibrium dissociation constant of an unlabelled ligand by measuring its competition for radioligand binding.

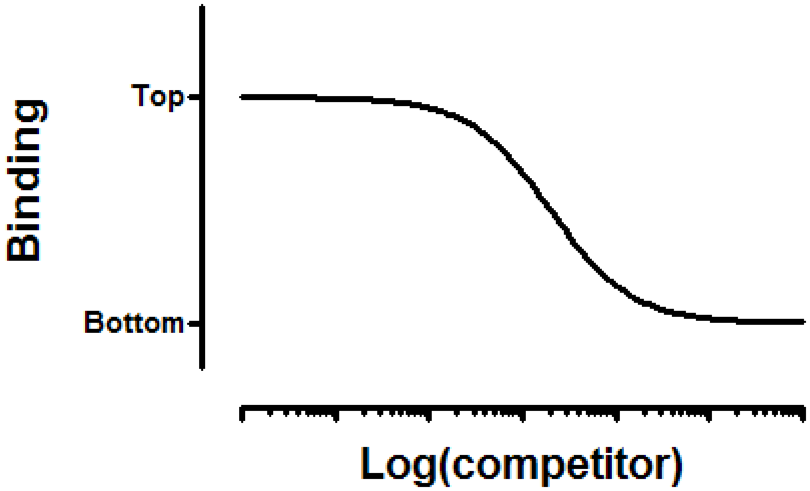
Step by step

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 160.

The analysis assumes that you have one site, and that the binding is reversible and at equilibrium.

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 identical affinity for the radioligand, but different affinities for the competitor.

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.

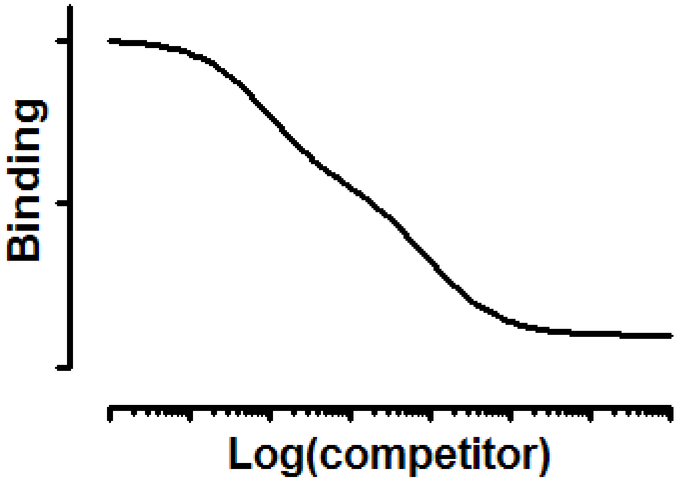
From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *Two sites - Fit Ki.*

You must constrain three 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.
- **KdHi** is the equilibrium dissociation constant of the labeled ligand for the high-affinity site in nM.
- **KdLo** is the equilibrium dissociation constant of the labeled ligand for the low-affinity site in nM.

Model

```
ColdnM=10^(X+9)
KIHinM = 10^(LogKI_Hi+9)
KILonM = 10^(LogKI_Lo+9)
SITE1= HotnM*(Top-Bottom)/(HotnM + KDHotNM_Ki*(1+coldnM/KiHinM))
SITE2= HotnM*(Top-Bottom)/(HotnM + KDHotNM_Lo*(1+coldnM/KiLonM))
Y = SITE1*FractionHi + SITE2*(1-FractionHi) + Bottom
```



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 Prussoff correction. 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 164.

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.

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 identical affinity for the radioligand, but different affinities for the competitor.

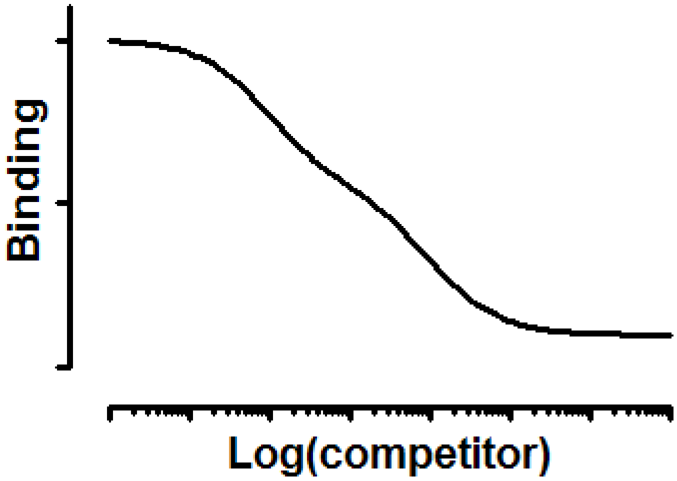
Step by step

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 *Two sites - Fit logIC50.*

Model

```
Span=Top-Bottom
Section1=Span*FractionHi/(1+10^((X-LogIC50HI)))
Section2=Span* (1-FractionHi)/(1+10^((X-LogIC50Lo)))
Y=Bottom + Section1 +Section2
```



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 162.

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.

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.

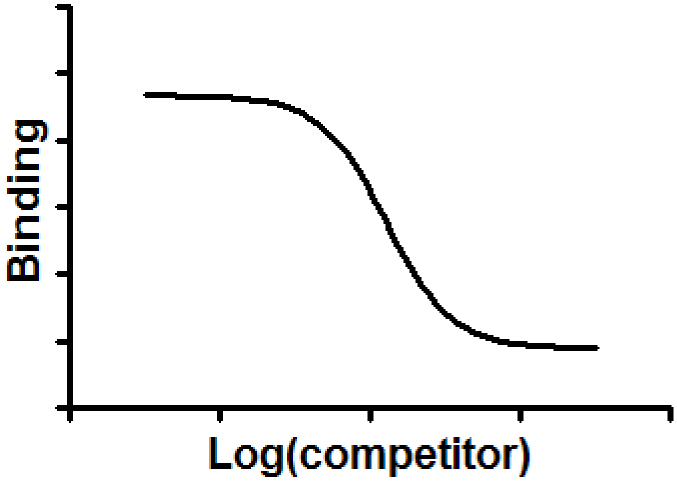
From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *One site -- Heterologous with depletion.*

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 160.

Reference

This equation came from S. Swillens (Molecular Pharmacology, 47: 1197-1203, 1995)

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

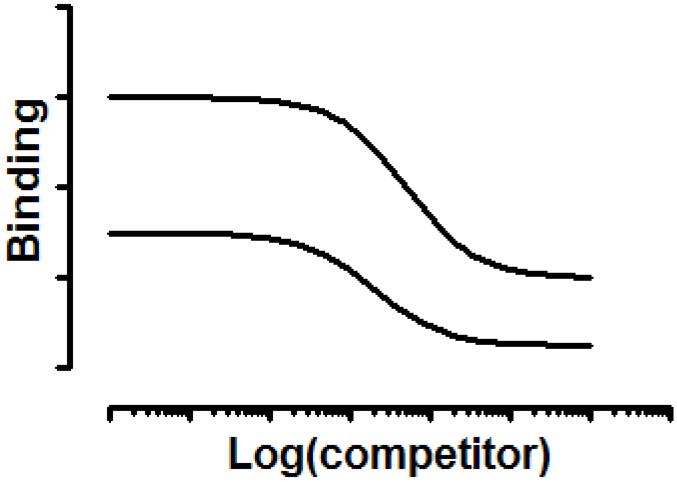
Create an XY data table. Enter the logarithm of the concentration of the unlabeled compound into X, and binding into Y. Enter the concentration of labeled ligand (in nM) as the column title. You will get better results if you use two different concentrations of labeled ligand.

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 binding 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 a 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.

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 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.*

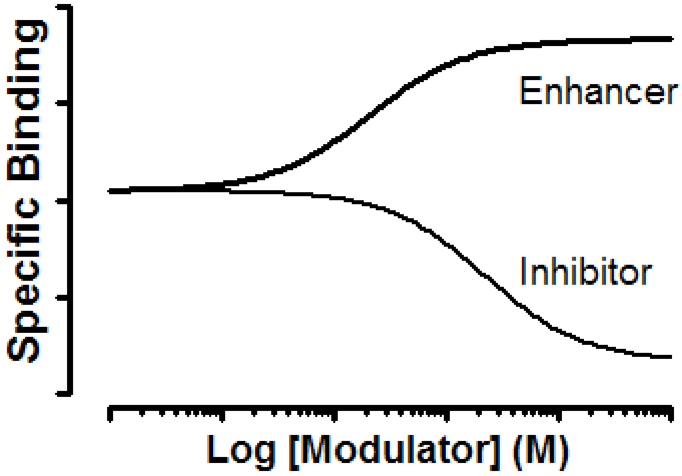
You must constrain two parameters to constant values based on your experimental design:

- 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

Receptor binding - Kinetics

acol Rev, 54: 323-374, 2002

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:

$$\text{Receptor} + \text{Ligand} \underset{\text{Koff}}{\overset{\text{Kon}}{\rightleftarrows}} \text{Receptor}\cdot\text{Ligand}$$

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 172, measure association kinetics with two or more concentrations of radioligand 173, or determine the rate of association and dissociation in one experiment 174.

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.

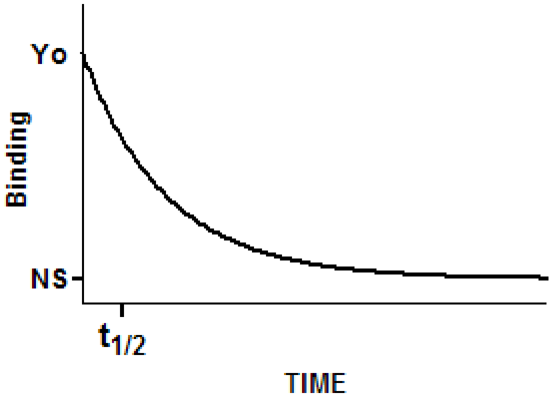
Step by step

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

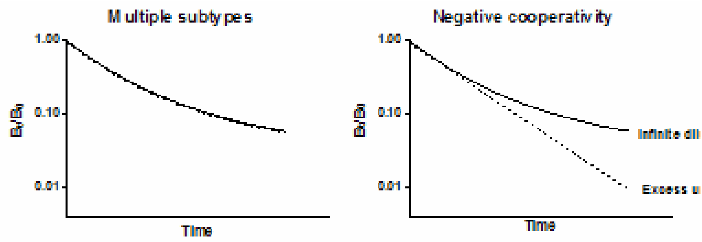
Y0 is the binding at time zero, in the units of the Y axis.

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.

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? 170) 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 173, or to analyze an experiment that measuresboth association and dissociation 174 rate sequentially.

Step by step

Create an XY data table. Enter time into X, and specific binding into Y.

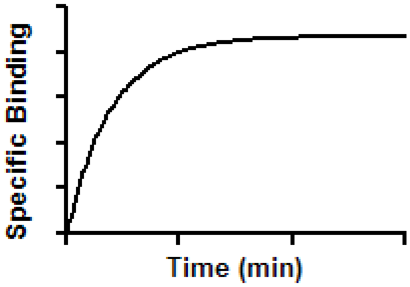
From the table of specific binding, click Analyze, choose nonlinear regression, choose the panel of Kinetics Binding equations, and choose *Association kinetics - One conc. of hot.*

You must constrain Hotnm ([radioligand] in nM) and Koff (dissociation rate constant) 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 inverse molar time inverse time.

Kd is the equilibrium binding constant, in Molar, computed as Koff/Kon

Bmax is the maximal binding at equilibrium, at maximal radioligand concentration.

Notes

Introduction

You cannot determine k_{on} from an association binding measured at a single concentration of radioligand. 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

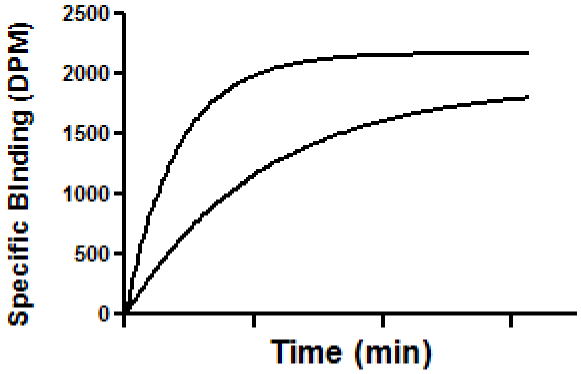
Create an XY data table. Enter time into X, and total binding into Y. Enter binding at one concentration of radioligand into the column A, binding at another concentration into column B, etc. Enter the concentrations, in nM, into the column titles.

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 inverse minutes.

Kon is the association rate constant, in units of inverse molar time inverse time.

Kd is the equilibrium binding constant, in Molar, computed as Koff/Kon

Bmax is the maximal binding at equilibrium, at maximal radioligand concentration.

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.

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 172. 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

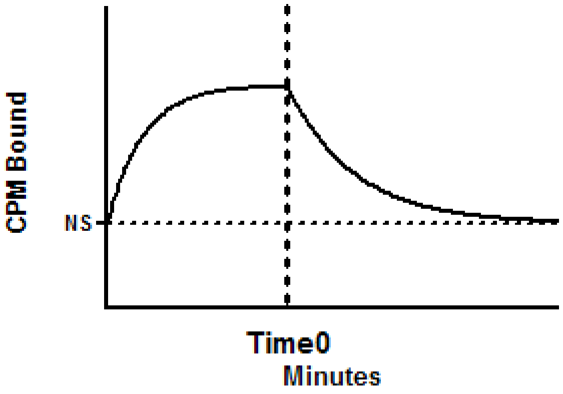
Create an XY data table. Enter time into X, and total binding into Y.

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 inverse time units.

Kon is the association constant in inverse time 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

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. 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.*

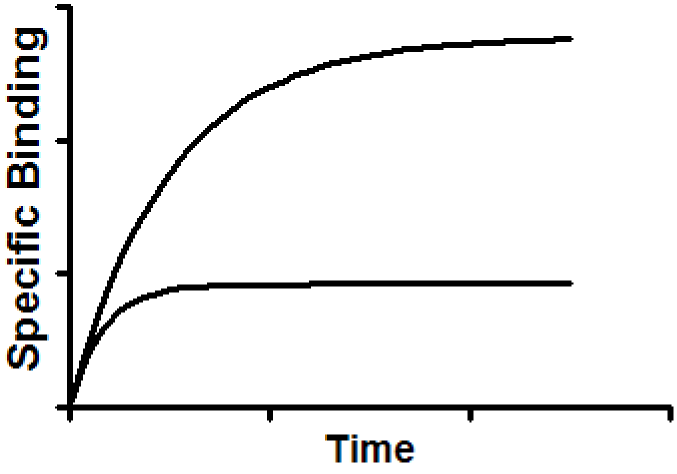
Constrain k1 and k2 to constant values determined from kinetic binding experiments. k1 is the association rate constant of the hot ligand in M^{-1} $min^{-1}$and k2 is its dissociation rate constant in units of min^{-1}.

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.

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.

Reference

This method was described by Motulsky and Mahan in Molecular Pharmacology 25:1-9, 1984.

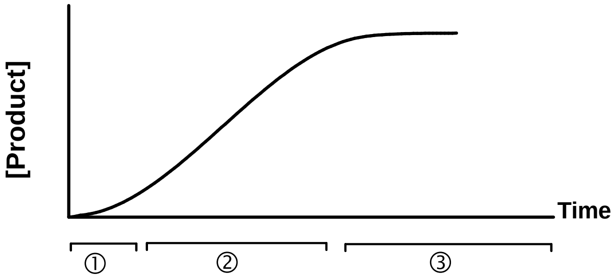
Enzyme kinetics

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.

- 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.

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.

Fitting a substrate-velocity curve

The simplest experiment in enzyme kinetics is to vary the substrate concentration and measure enzyme velocity.

The standard way to fit these data is to fit the Michaelis-Menten model 180 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 182. The curve will be identical to the Michaelis-Menten fit.

Fitting results from 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 three models of enzyme inhibition, plus a more general model which includes the first three as special cases:

- A competitive 185 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 186 inhibitor reversibly binds to the enzyme-substrate complex, but not to the enzyme itself. This means that the inhibition is not surmountable by increasing substrate concentration, but there is no change in the concentration of substrate needed to get half maximal activity. The substrate-velocity curve is shifted down but neither to the right or left.
- An uncompetitive 187 inhibitor binds with equal affinity to the enzyme, and the enzyme-substrate complex. The inhibition is not surmountable by increasing substrate concentration. Because the enzyme-substrate complex is stabilized, it takes less substrate to get to half-maximal activity. The substrate-velocity curve is shifted down and to the left.
- The mixed 189 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 this parameter tells you about the mechanism of inhibition.

Substrate inhibition

In some cases, the substrate of an enzyme also inhibits the enzyme by binding to a second site on the enzyme. Prism offers a model to fit substrate-velocity curves when the substrate also inhibits the enzyme 190.

Reference

RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005.

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 182 of the equation.

Step by step

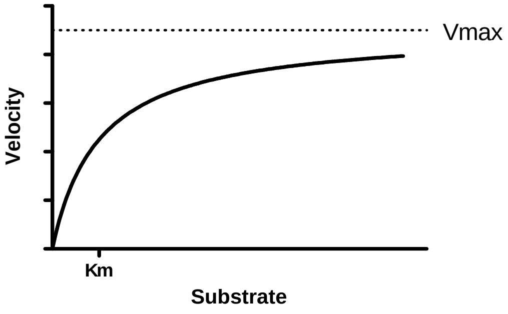
Create an XY data table. Enter substrate concentration into X, and enzyme velocity into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

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)
```



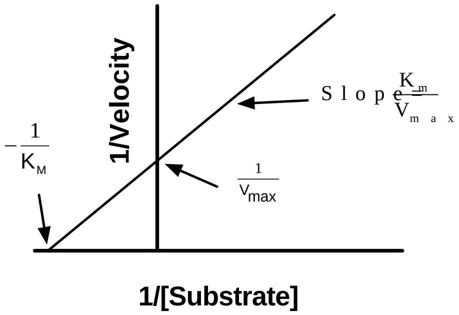
Interpret the parameters

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=0, Y=-1/KM (previously determined by nonlinear regression).
3. Into row 2 enter X=1/Smax (Smax is the largest value of [substrate] you want to include

on the graph) and Y=(1/Vmax)(1.0 + KM/Smax).

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 178 of all analyses of enzyme kinetics.
- This equation fits exactly the same curve as the equation that fits the turnover number Kcat 182 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 184. 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.

Introduction

The Vmax of an enzyme is the maximum enzyme velocity extrapolated to maximum substrate concentrations. The Vmax is determined by how many enzyme sites are present and the rate at which the enzyme can convert substrate to product. If you know the concentration of enzyme sites, you can fit kcat, the rate at each enzyme site can convert substrate to product.

Entering 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.

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 *Kcat*.

Constrain Et to a constant value

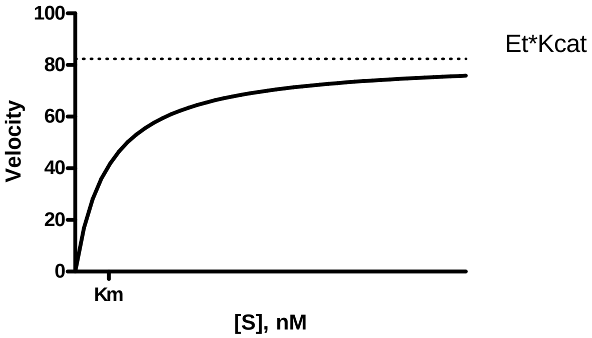
You must constrain **Et** to a constant value, based on other experiments. Et is the concentration of enzyme catalytic sites. If the enzyme has multiple subunits, note that Et is the concentration of catalytic sites, which can be larger than the concentration of enzyme molecules. The Y values are entered in units of concentration per time, and Et must be entered in those same concentration units.

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 180. 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.

The model

```
Y = Et*kcat*X/(Km + X)
```



Interpret the parameters

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 Et 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.

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 is almost always higher than any velocity measured in your experiment. It is computed by multiplying Et times kcat.

Notes

- See the list of assumptions 178 of all analyses of enzyme kinetics.
- This equation fits exactly the same curve as the equation that fits Vmax 180, 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.

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.

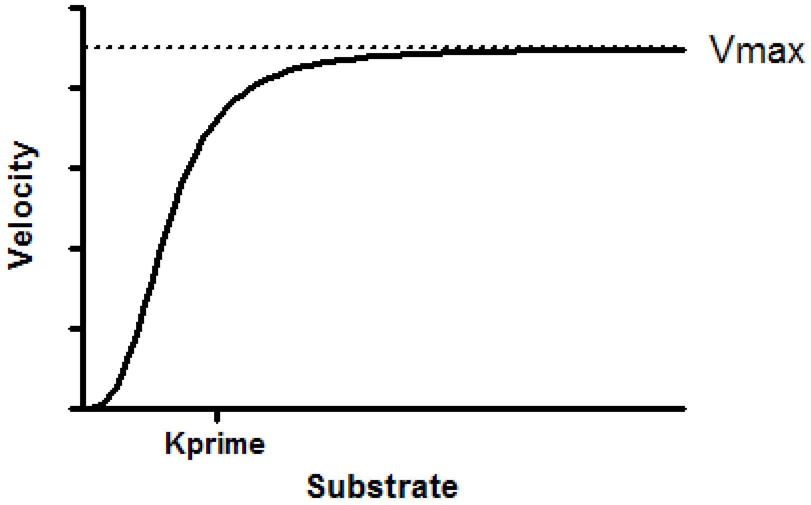
How to enter data

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/(Kprime + 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.

Kprime is related to the Km, but is not equal the substrate concentration needed to achieve a half-maximum enzyme velocity (unless h=1). It is expressed in the same units as X.

h is the Hill slope. When n=1, this equation is identical to the standard Michaelis-Menten equation 180. When it is greater than 1.0, the curve is sigmoidal due to positive cooperativity. The variable n does not always equal the number of interacting binding sites, but its value can not exceed the number of interacting sites. Think of n as an empirical measure of the steepness of the curve and the presence of cooperativity.

Reference

Equation 5.47, in RA Copeland, *Enzymes,* 2nd edition, Wiley, 2000.

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.

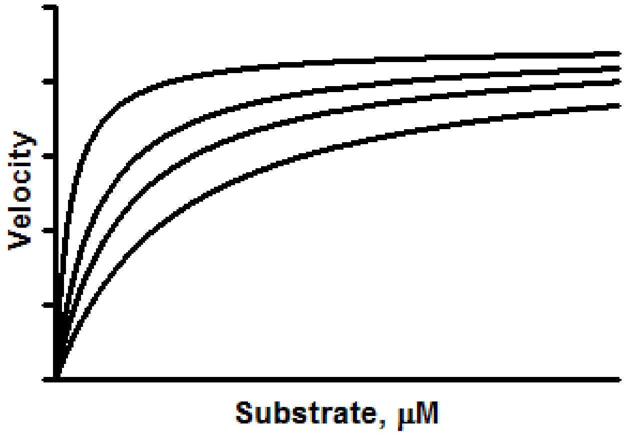
After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Competitive enzyme inhibition*.

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.

If the data don't fit the model well, consider instead fitting to a noncompetitive 186 or uncompetitive 187 model. Or fit to the more general equation for mixed-model inhibition 189.

Reference

Equation 8.11, in RA Copeland, *Enzymes,* 2nd edition, Wiley, 2000.

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 189.

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.

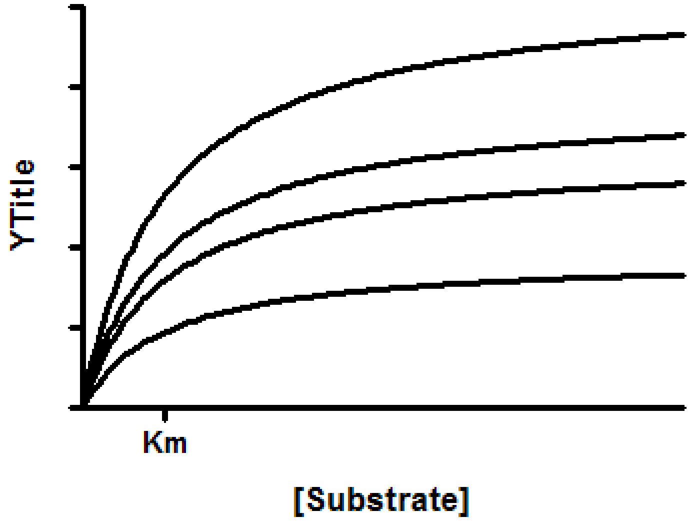
After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Noncompetitive enzyme inhibition*.

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 without inhibitor, expressed in the same units as Y.

Km is the Michaelis-Menten constant (without inhibitor), expressed in the same units as X.

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 185 or uncompetitive 187 model. Or fit to the more general equation for mixed-model inhibition 189.

Reference

Equation 8.15, in RA Copeland, *Enzymes,* 2nd edition, Wiley, 2000.

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.

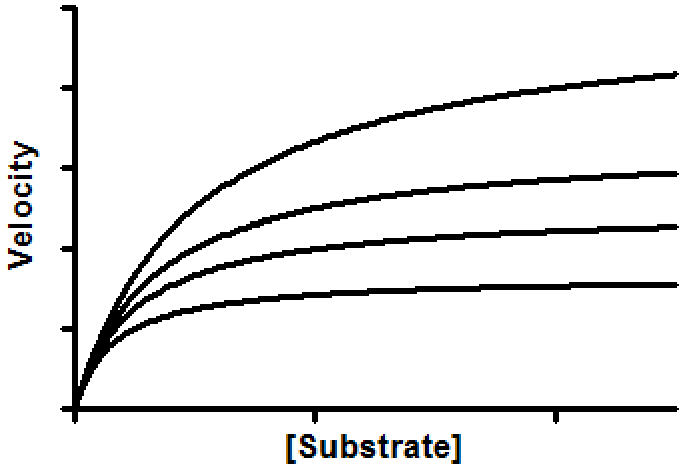
After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Uncompetitive enzyme inhibition*.

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 without inhibitor, expressed in the same units as Y.

Km is the Michaelis-Menten constant (without inhibitor), expressed in the same units as X.

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 185 or noncompetitive 186 model. Or fit to the more general equation for mixed-model inhibition 189.

Reference

Equation 8.17, in RA Copeland, *Enzymes,* 2nd edition, Wiley, 2000.

Introduction

The mixed model is a general equation that includes competitive 185, uncompetitive 187 and noncompetitive 186 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 without inhibitor, expressed in the same units as Y.

Km is the Michaelis-Menten constant, expressed in the same units as X.

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 does not alter binding of substrate to the enzyme, and the mixed-model is identical to noncompetitive 186 inhibition. When alpha is very large, binding of inhibitor prevents binding of the substrate and the mixed-model becomes identical to competitive 185 inhibition. When Alpha is very small (but greater than zero), binding of the inhibitor enhances substrate binding to the enzyme, and the mixed model becomes nearly identical to an uncompetitive 187 model.

Reference

Equation 8.17, in RA Copeland, *Enzymes,* 2nd edition, Wiley, 2000.

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.

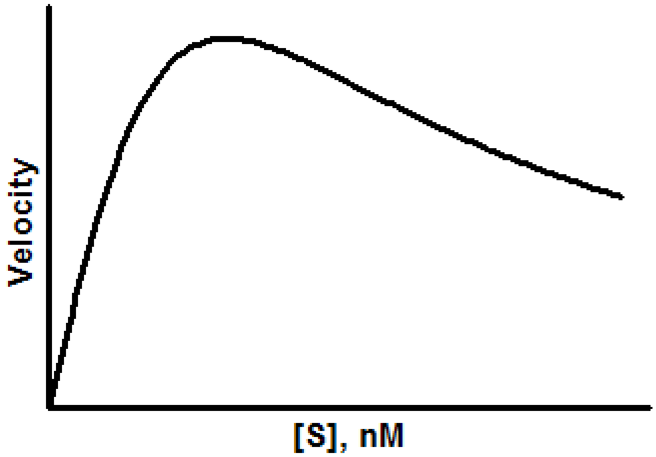
Step by step

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.

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.

Reference

Equation 5.44, in RA Copeland, *Enzymes,* 2nd edition, Wiley, 2000.

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 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 *Morrison Ki*.

Constrain Et, S and Km to constant values

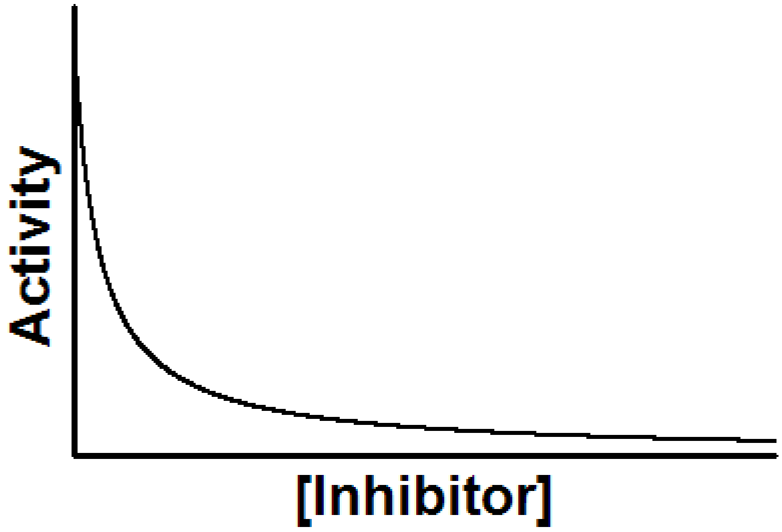
You must constrain three parameters to constant values. To constrain the values, go to the Constrain tab of the nonlinear regression dialog, make sure that the drop down next to Et, S and Km is set to "Constant equal to" and enter the values.

- **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 180 without competitor.

Prism cannot fit 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 with no 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.

Reference

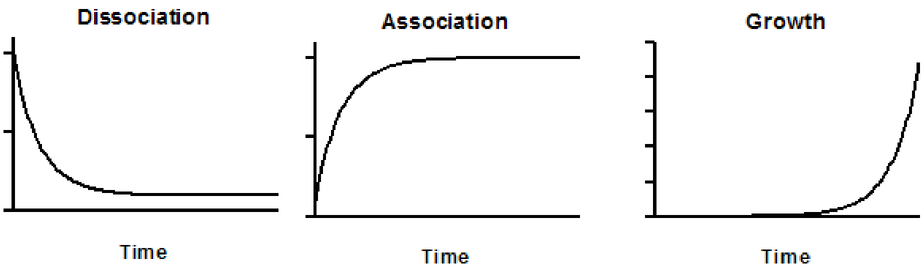
Equation 9.6, in RA Copeland, *Enzymes,* 2nd edition, Wiley, 2000.

Exponential

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.

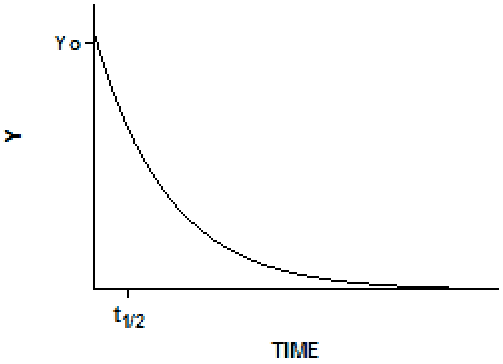
Define X to be time, and Y to be the outcome you are measuring. Three examples:

- 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.

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.

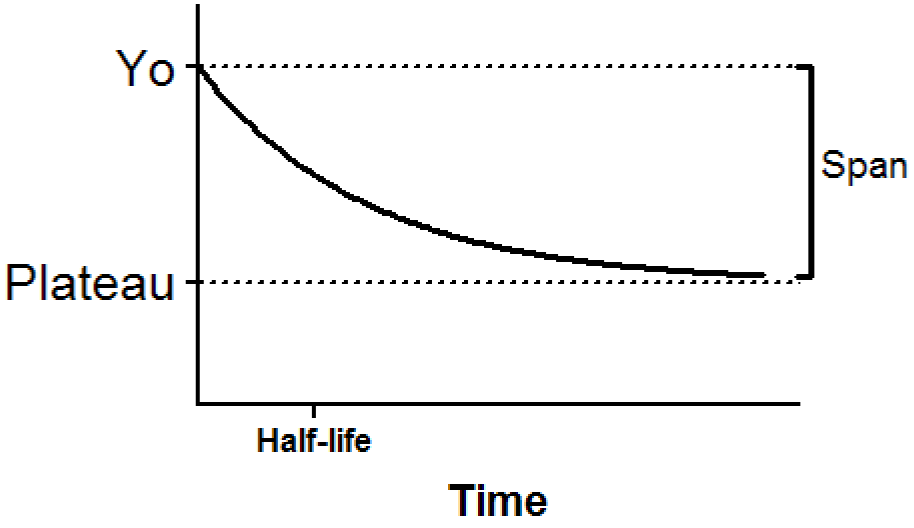
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *One phase decay*.

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.

Introduction

In the standard one-phase decay 194 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

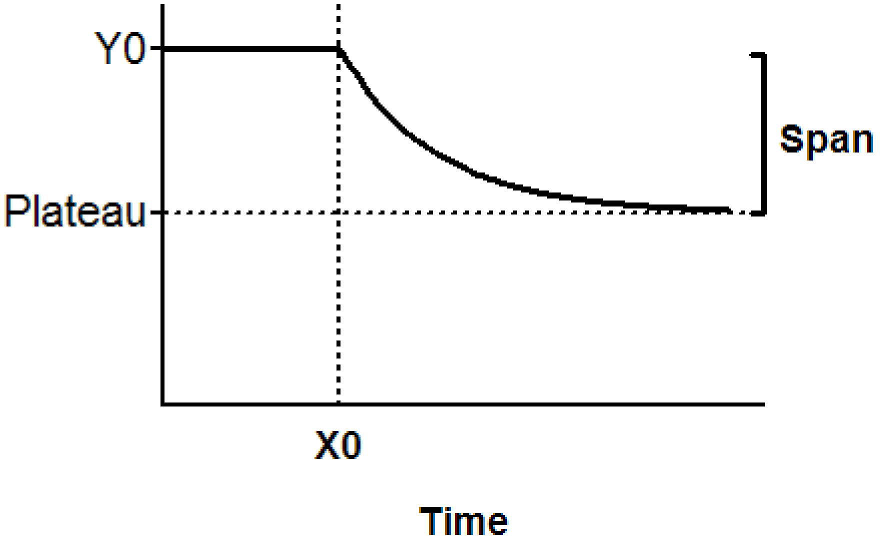
If you know the time at which you initiated the decay, you should constrain X0 to that value.

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.

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.

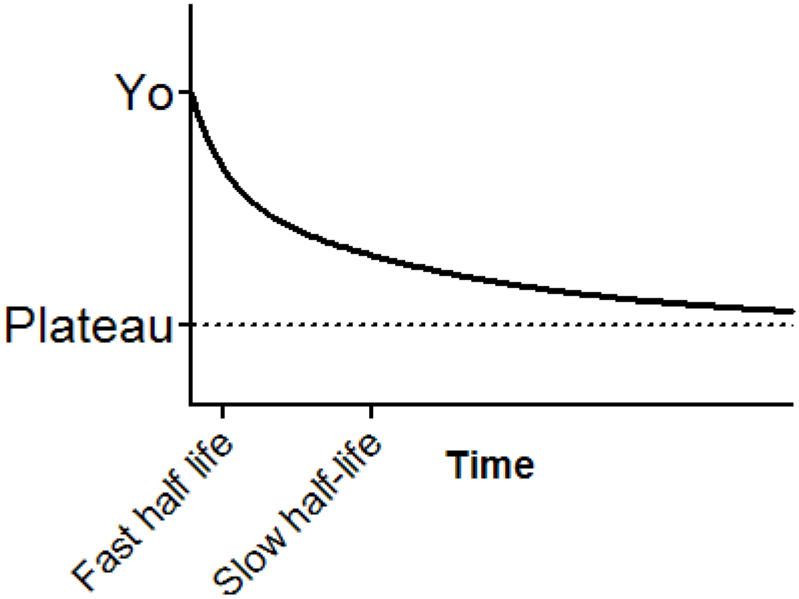
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Two phase decay*.

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

```
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.

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.

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.

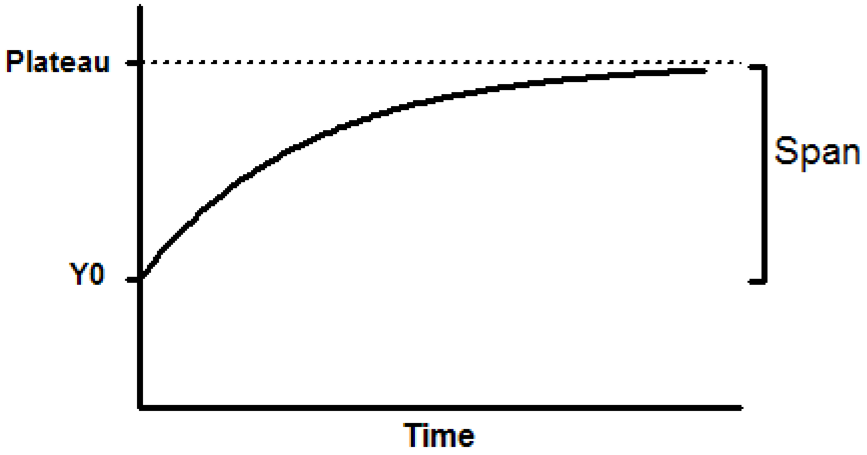
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *One phase association*.

Consider constraining 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-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.

Introduction

In the standard one-phase association 199 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.

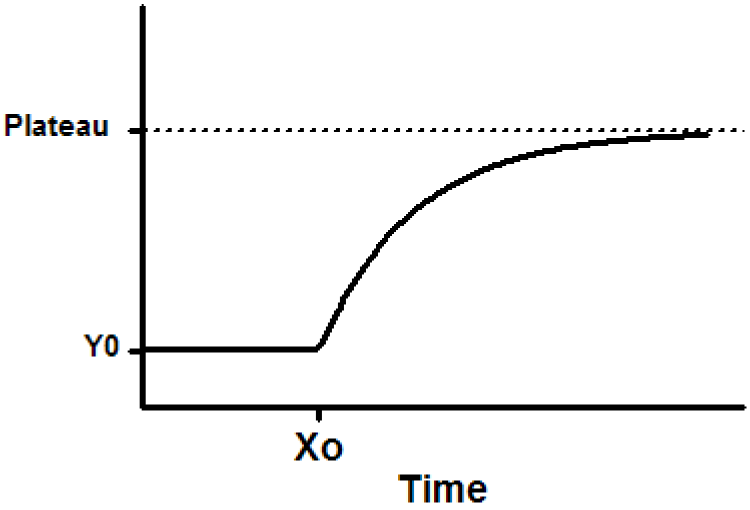
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Plateau followed by one phase association.*

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.

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.

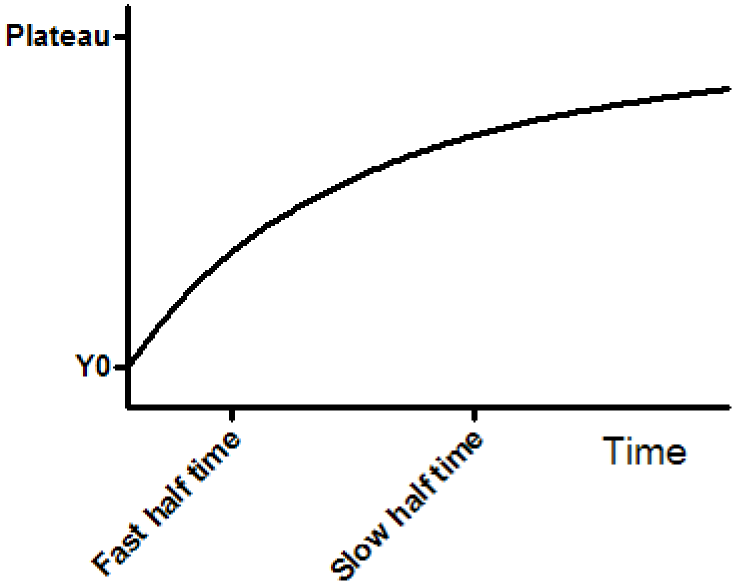
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Two phase association*.

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.

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.

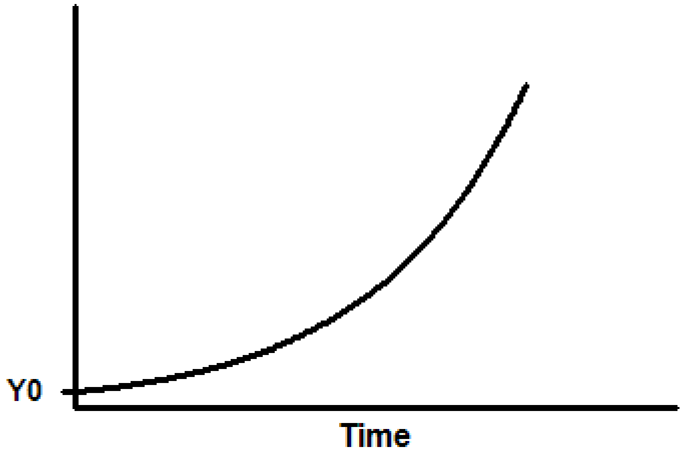
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Exponential growth*.

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.

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 243, apply weighting 244, or automatically exclude outliers 70.

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 206 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.

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 32, such as the ability to compare two models 243, apply weighting 244, automatically exclude outliers 70 and perform normality tests 255 on the residuals.

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" model built in:

```
Y=Slope*X
```

The only parameter is the slope.

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, 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.

Don't use segmental linear regression to analyze a biphasic Scatchard 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

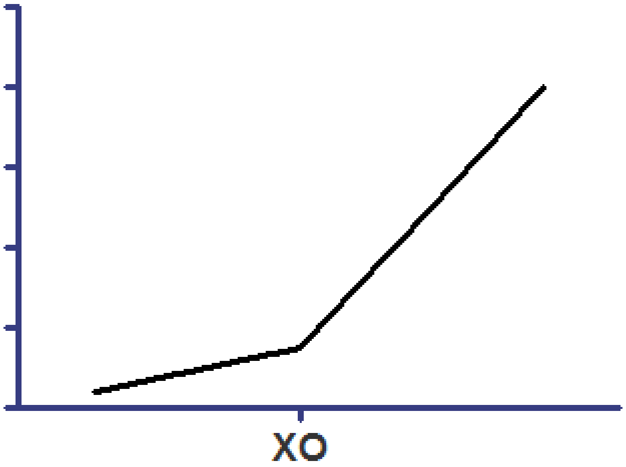
```
Y1 = intercept1 + slope1*X
YatX0 = slope1*X0 + intercept1
Y2 = YatX0 + slope2*(X – X0)
Y = IF(X<X0, Y1, Y2)
```

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.

Intercept2 is the Y value where the second (rightmost) line segment intersects the Y axis.

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))
```

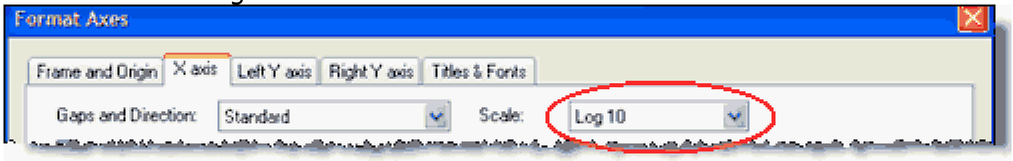
.

Fitting straight lines on graphs with nonlinear axes

The nonlinear regression analysis fits the data, not the graph. Since Prism lets you choose logarithmic, 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.

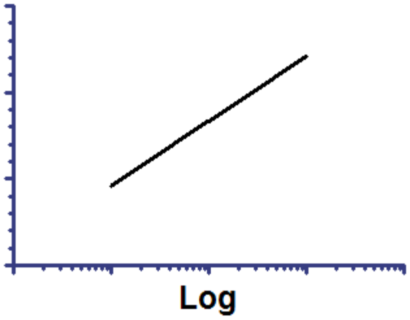
Equations

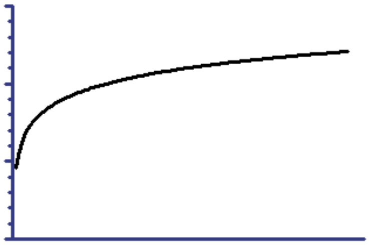
Semilog line -- X axis is logarithmic, Y axis is linear

Y=Yintercept + Slope*log(X)

On semilog axis

On linear axes



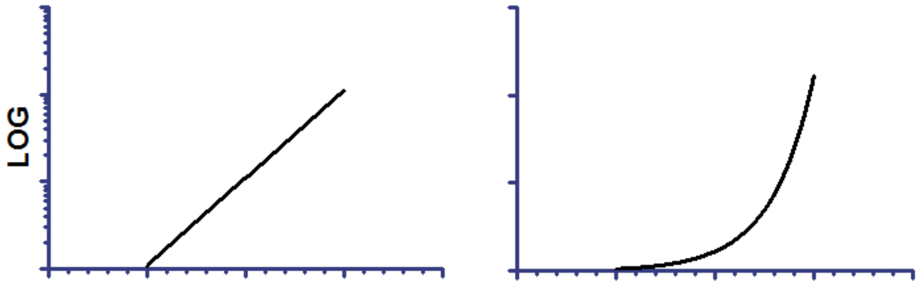


Semilog line -- X axis is linear, Y axis is logarithmic

Y=10^(Slope*X + Yintercept)

On semilog axis

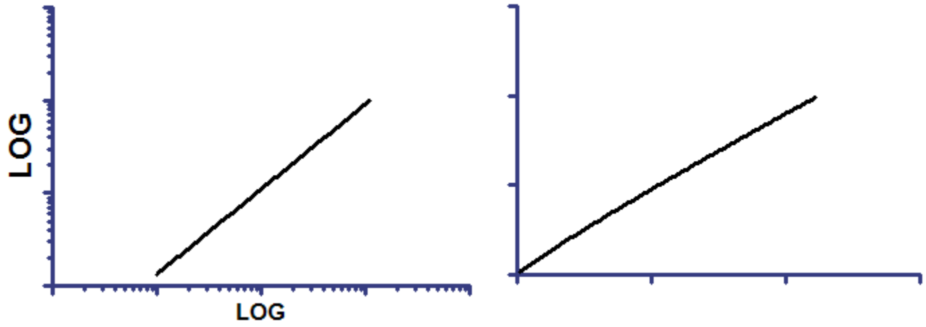
On linear axes



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.

Parameters

In all three equations, Y intercept is in units of the Y values, and Slope is in units of the Y values divided by units of the X values.

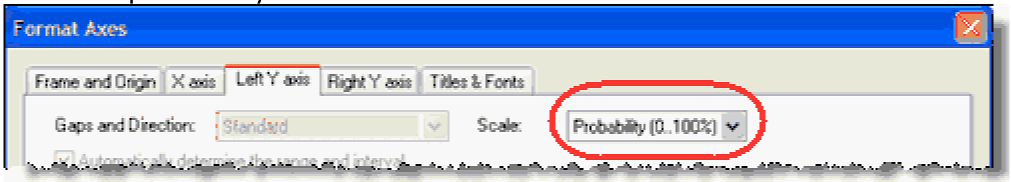
Fitting straight lines on graphs with nonlinear axes

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.



Format Axes

Gaps and Direction: Standard

Scale: Probability (0..100%)

☒ Automatically determine the range and interval

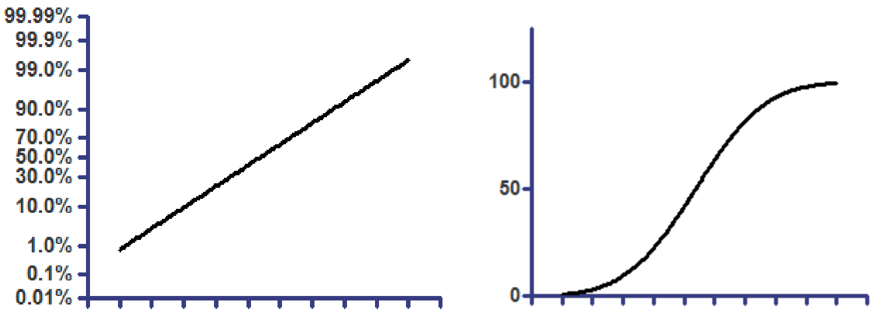
3. Click Analyze, choose Nonlinear regression (not Linear regression) and then choose one of the Cumulative Gaussian distribution equations from the "Lines" section of equations.

Equations

Cumulative Gaussian - Y values are percentages

```
Top=100
z=(X-Mean)/SD
Y=Top * zdist(z)
```

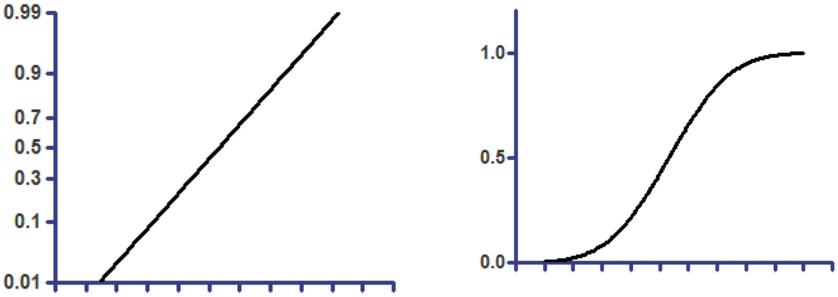
On probability axis **On linear axis**



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.

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 263.

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.

The model

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.

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.

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.

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

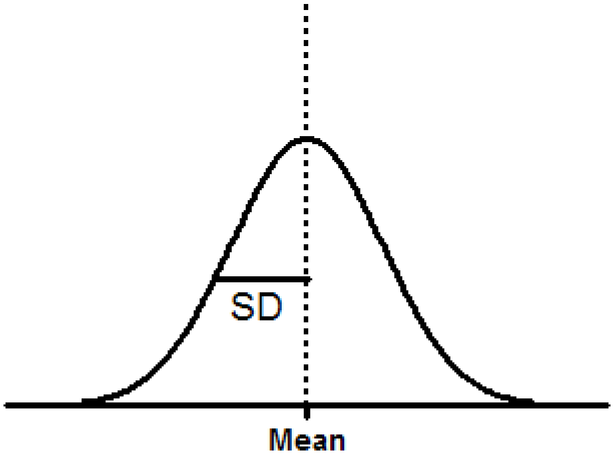
The data you fit must be in the form of a frequency distribution on an XY table. The X values are the bin center and the Y values are the number of observations.

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.

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 log Gaussian 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

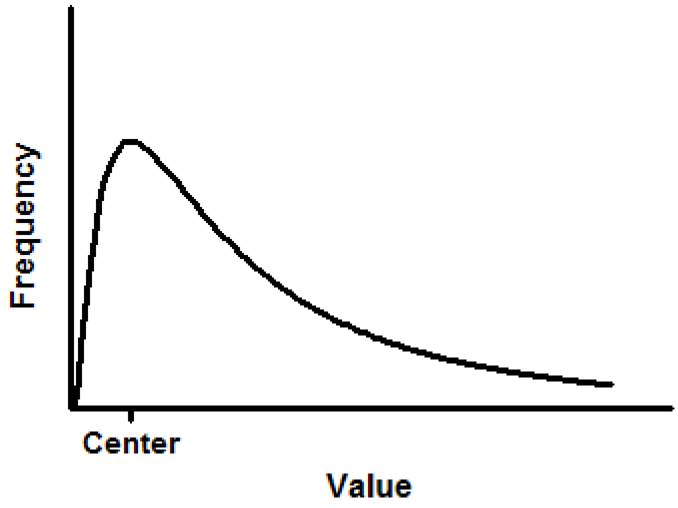
The data must be in the form of a frequency distribution on an XY table. The X values are the bin center and the Y values are the number of observations.

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 "logGaussian" equation from the "Gaussian" family of equations.

Model

```
Y=Amplitude*exp(-0.5*(ln(X/Center)/Width)^2)
```



Amplitude is the height of the center of the distribution in Y units.

Center is the X value at the peak of the distribution.

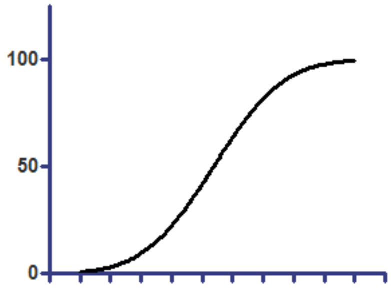
Width is a measure of the width of the distribution, in the same units as X.

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.

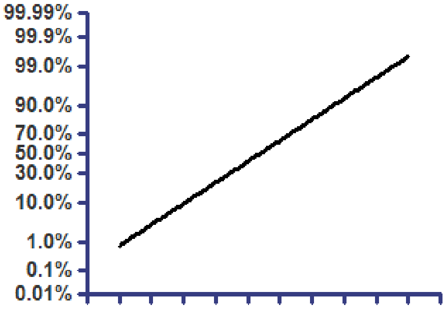
SD is the standard deviation of the original distribution.

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.



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 del to fit, try fussing with the initial parameter values.

Sine waves

Introduction

Sine waves describe many oscillating phenomena.

Step by step

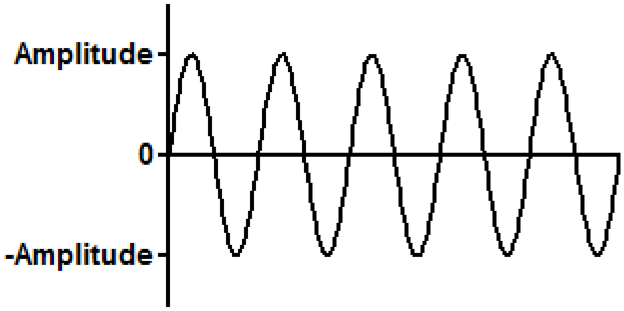
Create an XY data table. There is one X column, and many Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel of equations for sine waves, and choose *Standard sine wave*.

If you know the Y value is zero at time zero, then constrain PhaseShift to a constant value of zero.

Model

```
Y= Amplitude*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 is the earliest time when Y=0, in time units of X axis.

Introduction

Sine waves describe many oscillating phenomena. Often the peak of each wave decreases or dampens as time goes on.

Step by step

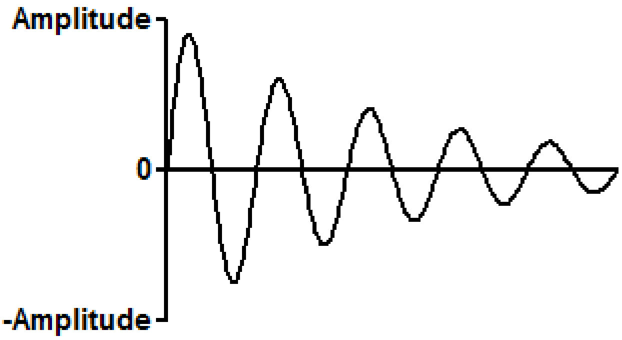
Create an XY data table. There is one X column, and many Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

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.

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 is the earliest time when Y=0, in time units of X axis.

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.

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.

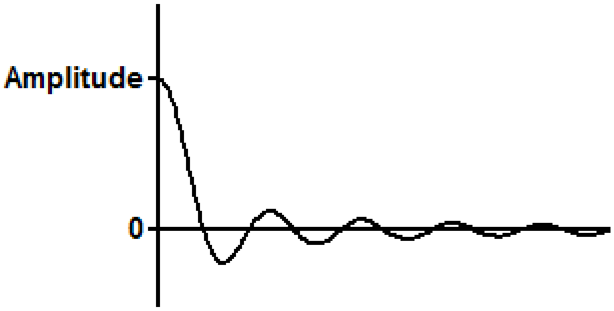
Step by step

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.

Model

```
Y=IF(X=0,Amplitude,Amplitude*sin(2*pi*X/Wavelength)/(2*pi*X/Wavelength) )
```



Interpret the parameters

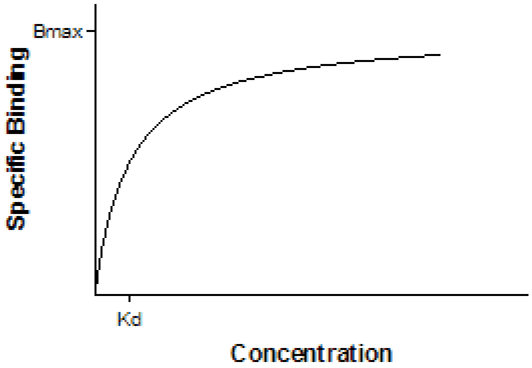
Amplitude is the height of top of the waves, in Y units.

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.

Classic equations from prior versions of Prism

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



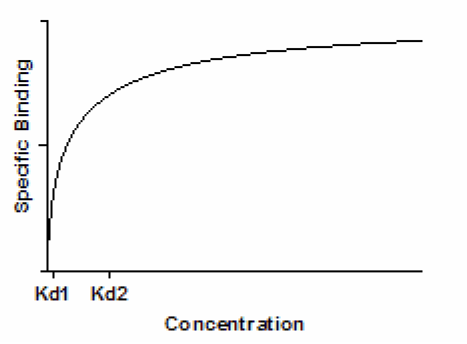
This curve is known as a *rectangular hyperbola*, *binding isotherm*, or *saturation binding curve*. Y is zero initially, and increases to a maximum plateau value B_{max}.

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.

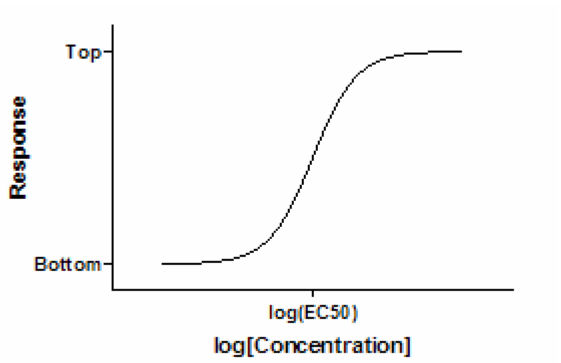
This equation also describes the activity of an enzyme as a function of substrate concentration. In this case, the variable labeled Bmax is really Vmax, the maximum enzyme activity, and the variable labeled Kd is really Km, the Michaelis-Menten constant.

$$Y = \frac{B_{max1} \cdot X}{K_{d1} + X} + \frac{B_{max2} \cdot X}{K_{d2} + X}$$



This equation is an extension of the one site binding curve 221. 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.

$$Y = Bottom + \frac{(Top - Bottom)}{1 + 10^{LogEC50-X}}$$

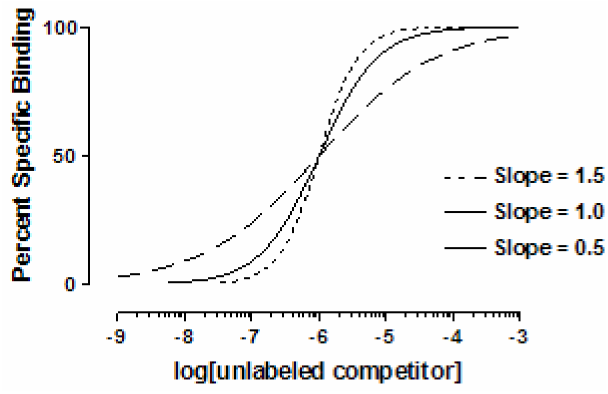


This is a general equation for a dose-response curve. It shows response as a function of the logarithm of concentration. X is the logarithm of agonist concentration and Y is the response. This equation is also called a three-parameter logistic equation.

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 223 allows for a 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.

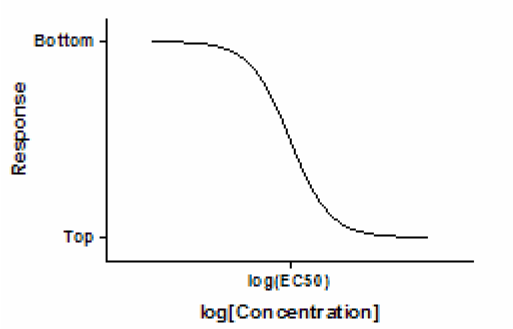
- Bottom is the Y value at the bottom plateau.
- Top is the Y value at the top plateau.
- LogEC50 is the X value when the response is halfway between Bottom and Top. With different kinds of variables, this variable is sometimes called ED50 (effective dose, 50%), or IC50 (inhibitory concentration, 50%, 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.

$$Y = \text{Bottom} + \frac{(\text{Top} - \text{Bottom})}{1 + 10^{X - \text{LogEC50}}}$$

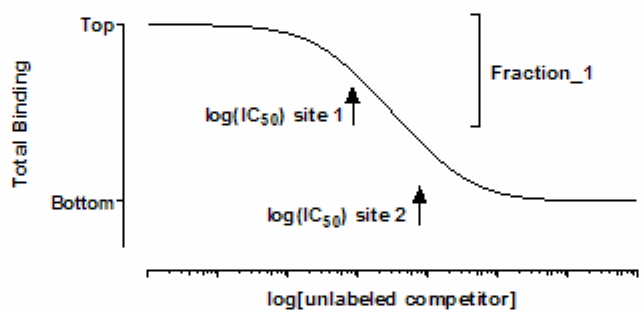
This equation describes the competition of a ligand for receptor binding. It is identical to the sigmoid dose-response curve with HILLSLOPE = -1.0.

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.



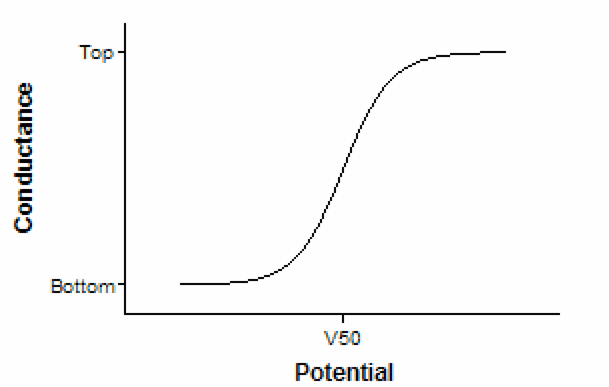
$$Y = \text{Bottom} + (\text{Top} - \text{Bottom})\left[\frac{\text{Fraction\_1}}{1 + 10^{X - \text{LogEC50\_1}}} + \frac{1 - \text{Fraction\_1}}{1 + 10^{X - \text{LogEC50\_2}}}\right]$$



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.

$$Y = \text{Bottom} + \frac{(\text{Top} - \text{Bottom})}{1 + \exp\left(\frac{\text{V50} - X}{\text{Slope}}\right)}$$



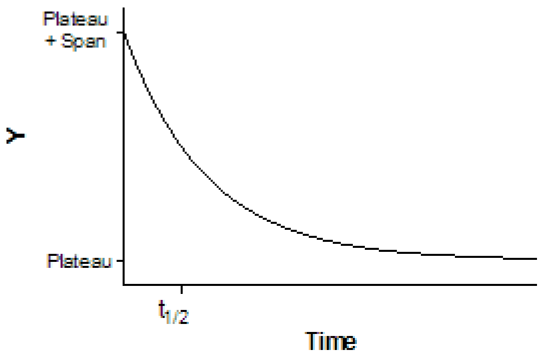
This equation describes voltage dependent activation of ion channels. It describes conductance

(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.

$$Y = \mathrm{Span} \cdot e^{-K \cdot X} + \mathrm{Plateau}$$



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.

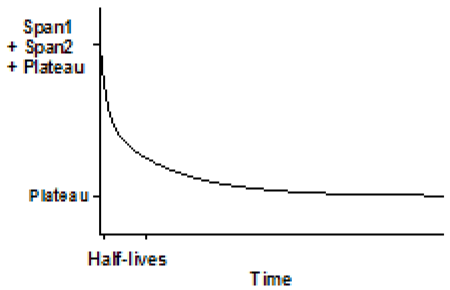
X is time.

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.

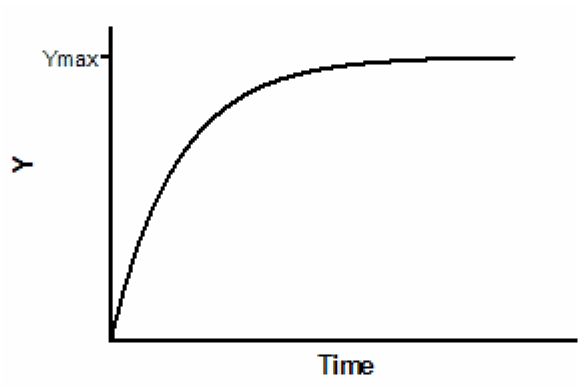
SPAN and PLATEAU are expressed in the same units as the Y axis. K is expressed in the inverse of the units used by the X axis. In many circumstances, the plateau equals zero. When fitting data to this equation, consider fixing the plateau to a constant value of zero.

$$Y = \mathrm{Span1} \cdot e^{-K_1 \cdot X} + \mathrm{Span2} \cdot e^{-K_2 \cdot X} + \mathrm{Plateau}$$



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.

$$Y = Ymax\cdot\left(1-e^{-K\cdot X}\right)$$



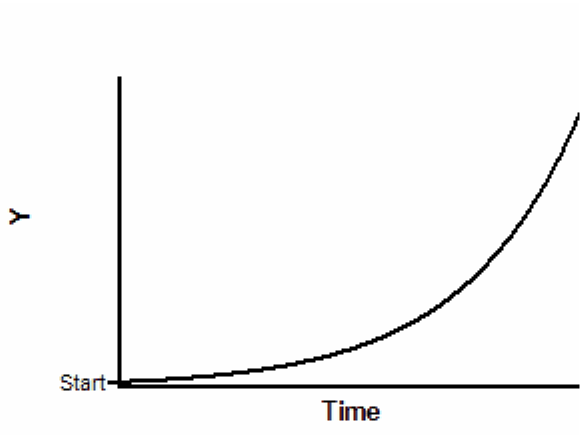
This equation describes the pseudo-first order association kinetics of the interaction between a ligand and its receptor, or a substrate and an enzyme. Y is either binding or enzyme activity. X is time.

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.

$$Y = Ymax_1\cdot\left(1-e^{-K_1\cdot X}\right)+Ymax_2\cdot\left(1-e^{-K_2\cdot X}\right)$$

This is an extension of the exponential association to two phases, corresponding to a radioligand binding to two independent sites.

$$Y = Start\cdot e^{K\cdot X}$$



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.

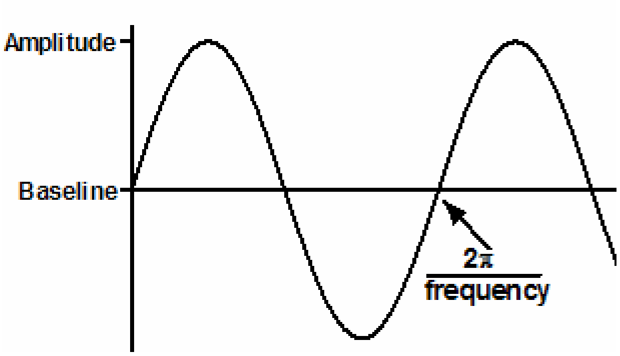
This versatile equation has many uses.

$$Y = A \cdot X^{B} + C \cdot X^{D}$$

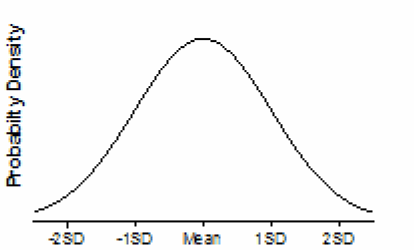
Fitting data to a power series model can be difficult. The initial values generated automatically by Prism are not very helpful (all four parameters are set to 1.0). you will probably need to enter better initial values in order to fit this equation to data. The initial values of B and D are important, 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.

$$Y = \text{Baseline} + \text{Amplitude} \cdot \sin(\text{Frequency} \cdot X + \text{Offset})$$



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.



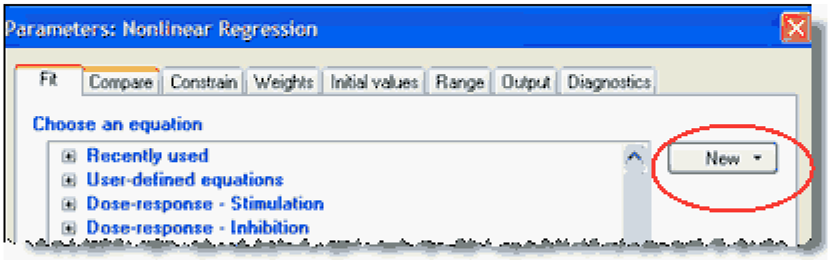
Cumulative probability distribution of a Gaussian bell-shaped distribution with specified mean and SD. The area under the entire curve is AREA. A standard probability distribution is scaled so that AREA equals 1.0. The units of the Y-axis are arbitrary, determined by your choice of AREA.

Writing your own model (equation)

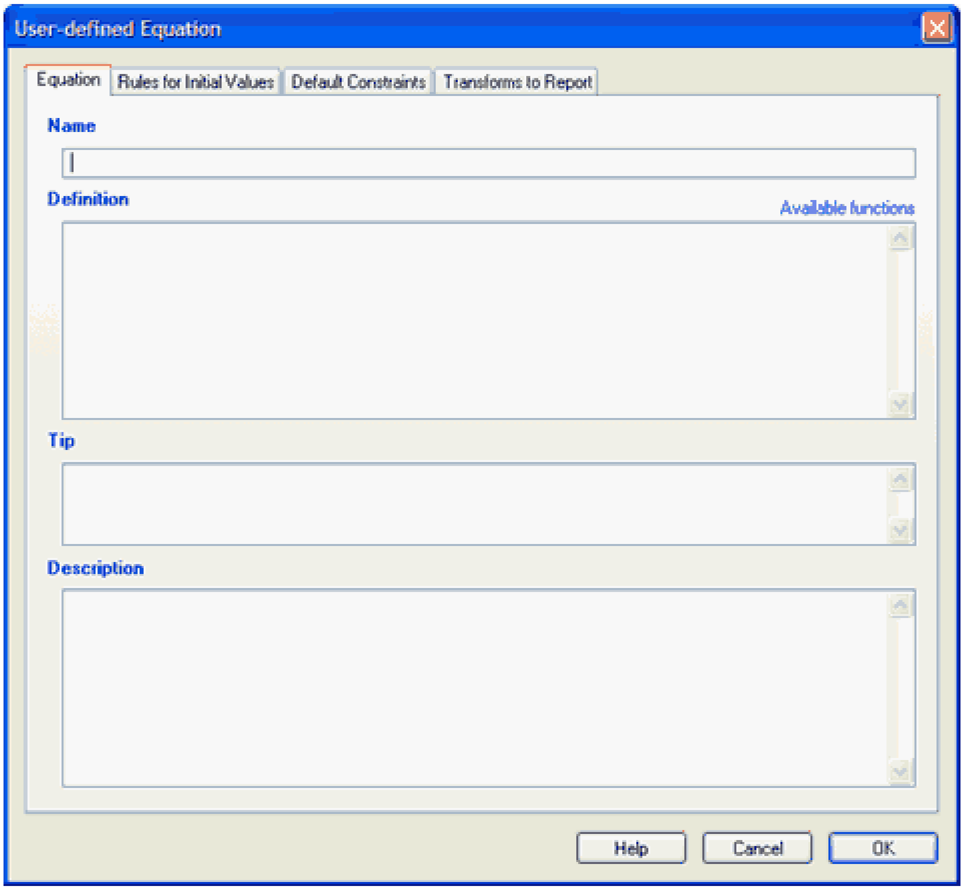
How to: Enter a new equation

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.



Name, definition, tip and description

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 232, a list of available math functions 233 you can use, and how to fit different models to different ranges of X values 235 and to different data sets 236.

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 237.

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 247 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 64. 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 244 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

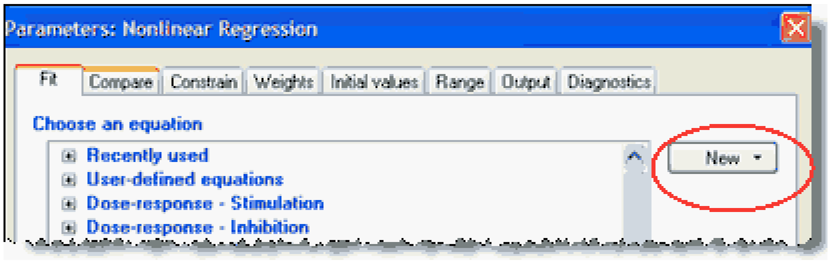
Define transforms of the best-fit values on the Transforms to report 239 tab. Unlike initial values and constraints, you can not override these transforms each time you choose the equation. Transforms are defined with the equation definition and can not be tweaked each time the equation is selected.

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.

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.

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 equations.

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.**

Models Prism cannot fit

Prism gives you a lot of flexibility when entering equations, but you must beware of these limitations:

- **No implicit equations**. Y must be defined as a function of X and one or more parameters. The variable Y can only appear once, on the left side of the last line of the equation. If Y also appears on the right side of the equation, you have an implicit equation, which Prism cannot handle. In many cases, you will be able to algebraically rearrange the equation.
- **No differential equations.**You must define Y as a function of X and one or more variables. It is not sufficient to define the derivatives.
- **No models with more than one X variable**. Prism does not calculate multiple 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.

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.

Syntax of user-defined equations

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.
- 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 233, most which are similar to those built-in to Excel.

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:

```
MAX>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 236.

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.

Available functions for user-defined equations

Allowed syntax 232

Function	Explanation	Excel equivalent
abs(k)	Absolute value. If k is negative, multiply by 1.	abs(k)
arccos(k)	Arccosine. Result is in radians.	acos(k)
arccosh(k)	Hyperbolic arc cosine.	acosh(k)
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)
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)

Function	Explanation	Excel equivalent
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)
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)
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 below.	(similar in excel)
igamma(j,k)	Incomplete gamma.	(no equivalent)
igammac(j,k)	Incomplete gamma, complement.	(no equivalent)
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)

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)
zdist(z)	P value (one-tailed) corresponding to specified value of z. Gaussian distribution.	normsdist(z)

Fitting different segments of the data to different models

What is segmental regression?

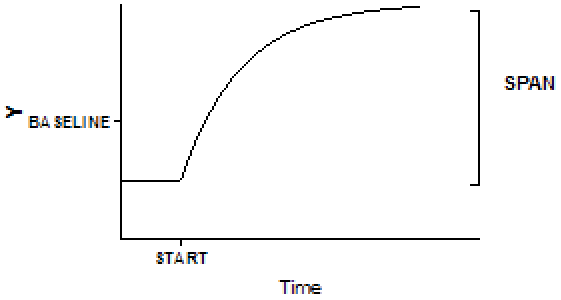
In some situations you may wish to fit different models to different portions of your data. This is called **segmental regression**. This often occurs in kinetic experiments where you add a drug or perform some sort of intervention while recording data. The values collected before the intervention follow 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 do 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.

Fitting different data sets to different equations

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 can specify that certain lines in the equation pertain to only certain data sets. For example:

- Precede a line in the equation with <C> if you want it to only apply to data set C.
- Precede with <~A> so it applies to all data sets except data set A.

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
```

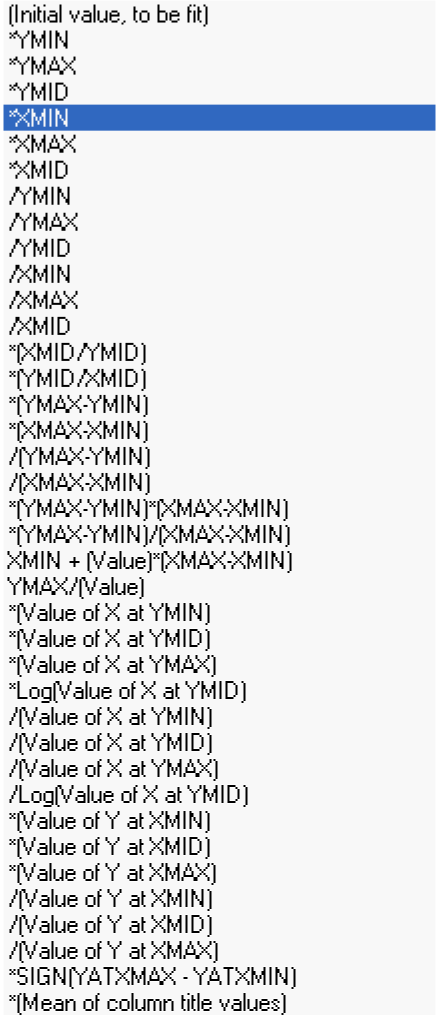
Rules for initial values

Why rules?

Before it can perform nonlinear regression, Prism must have initial values for each parameter in the equation. You can enter initial values at the time you fit curves, but it is helpful to define rules for generating the initial values at the time you enter a new equation. Then Prism will calculate the initial values automatically. 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.

Rule syntax

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 multiplier from the drop-down list in the second column.



```
(Initial value, to be fit)
*YMIN
*YMAX
*YMID
*XMIN
*XMAX
*XMID
/YMIN
/YMAX
/YMID
/XMIN
/XMAX
/XMID
*(XMID/YMID)
*(YMID/XMID)
*(YMAX-YMIN)
*(XMAX-XMIN)
/(YMAX-YMIN)
/(XMAX-XMIN)
*(YMAX-YMIN)*(XMAX-XMIN)
*(YMAX-YMIN)/(XMAX-XMIN)
XMIN + (Value)*(XMAX-XMIN)
YMAX/(Value)
*(Value of X at YMIN)
*(Value of X at YMID)
*(Value of X at YMAX)
*Log(Value of X at YMID)
/(Value of X at YMIN)
/(Value of X at YMID)
/(Value of X at YMAX)
/Log(Value of X at YMID)
*(Value of Y at XMIN)
*(Value of Y at XMID)
*(Value of Y at XMAX)
/(Value of Y at XMIN)
/(Value of Y at XMID)
/(Value of Y at XMAX)
*SIGN(YATXMAX - YATXMIN)
*(Mean of column title values)
```

The first choice on the drop-down list is "(Initial value, to be fit)". This means that the value you entered will be the initial value for all data sets. The initial value will not depend on the range of the data.

The rest of the choices on the drop-down list are used to calculate the initial value from your data. 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).

No 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.

Default constraints

What is a constraint?

Constraining parameters is often essential to getting useful results. Constraints can be used to fix parameters to constant 244 values, to share parameters among data sets (global 61 fitting), and to define one parameter to be a column constant 64 (whose value comes from the column titles in the data table).

Defining constraints when defining an equation

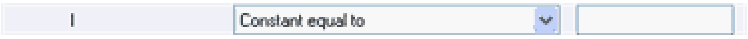
You can define default constraints as part of the equation definition. This is useful for constraints that will apply every time the equation is used.

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.

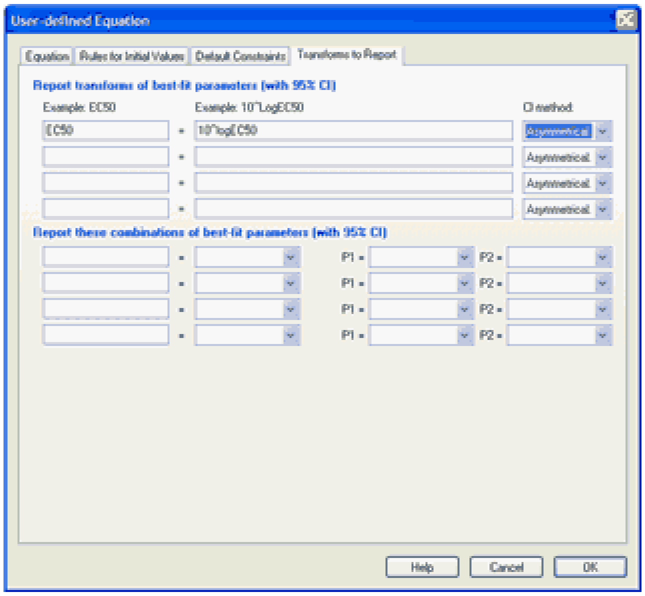
When you define the equation, you can add a constraint that reminds anyone using the equation to set that constant value. In the constraints tab when defining the equation, choose the constraint "Constant equal to" but leave the value (to the right) blank. With an equation defined this way, Prism won't fit data to that equation until the parameter is defined (or the constraint changed).



I Constant equal to

Transforms 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)

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

When transforming a single parameter, choose between symmetrical and asymmetrical confidence intervals via a drop down to the right of the transform.

If you choose to report a 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.

If you choose to report an 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.

Prism 4 (and earlier) always reported the asymmetrical intervals for EC50 and half lives, which were automatically calculated for built-in equations, and 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

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]	The Y value of the curve when X is the value you enter within the brackets. You must enter a number within the brackets, not a mathematical expression. 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]	The X value of the curve when Y is the value you enter within the brackets. You must enter a number within the brackets, not a mathematical expression. 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.

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 interpolation off the confidence bands of the regression curve. You don't have a choice of symmetrical vs. asymmetrical intervals.

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.

Nonlinear regression choices

Which choices are essential?

So many choices!

Prism's nonlinear regression dialog has seven 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...

... 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.

If your goal is to fit a model to determine best-fit parameters....

... 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..

Fit tab

> If you are new to nonlinear regression, focus on picking a model in the Fit tab and making appropriate choices on the Constrain tab. When getting started, it is ok to skip over choices on the Compare, Weights, Initial values, Range and Output tab.

Choose an equation

Selecting an equation is the most important step in fitting a curve. The choice cannot be automated 56.

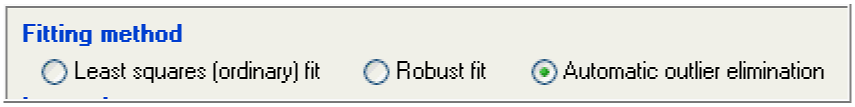
If your goal is to fit a model in order to understand your data and make comparisons, then choosing a model is a scientific decision that you must make with care. If your goal is to interpolate unknowns from a standard curve, then it matters less which equation you pick, so long as it ends up creating a smooth curve through your data.

Some tips:

- Prism provides a long list of equations that you can choose. But if these don't fit your needs, don't be afraid to create or clone an equation 230 to fit your needs.
- Part of choosing a model is choosing constraints 244. 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 244), Prism finds one best-fit value for the parameter that applies to all the data sets. Read more 61 about shared parameters (global fitting).

Fitting method

Prism offers three methods to fit curves.



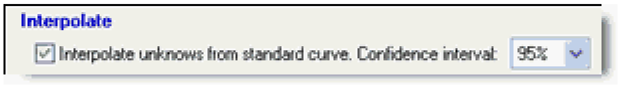
Fitting method

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

If you aren't sure which method to choose, pick least-squares regression, as it is standard. Robust regression 70 is less affected by outliers, but it cannot generate confidence intervals for the parameters, so has limited usefulness. Automatic outlier removal is extremely useful, but can lead to invalid (and misleading) results in some situations 66, so should be used with caution.

Interpolate

Check this option in order to interpolate the concentration of unknown samples from the best-fit curve. Learn more. 92



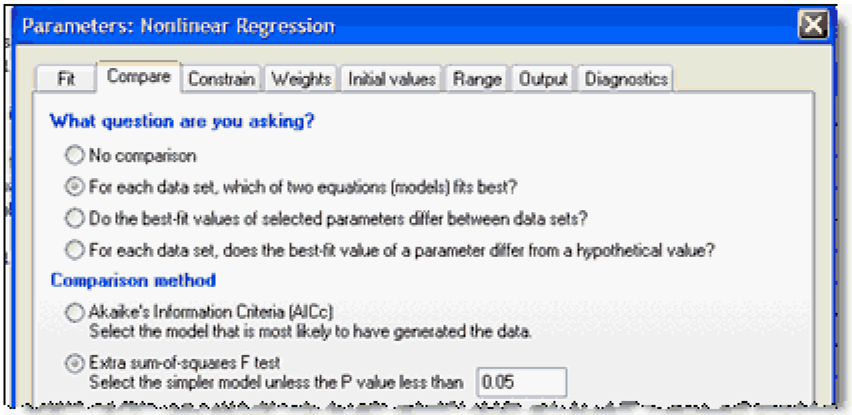
Interpolate

☑ Interpolate unknowns 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 is).

Compare tab

When you are first learning nonlinear regression, you can skip the choices on this Compare tab. But don't forget to come back and learn how Prism can help you compare models and datasets, as your scientific goals will often include comparing models.



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 parameters differ between data sets?

○ For each data set, does the best-fit value of a parameter differ from a hypothetical value?

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

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 ask if the best-fit value of a parameter differs significantly from a theoretical value. Learn more about these three kinds of comparisons 57. Your choice, of course, has to be based on your experimental goals.

Prism can perform the comparison using two alternative methods 58: the extra sum-of-squares F test 59, and using Akaike's information criteria 60. 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 AIC_c 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 AIC_c.

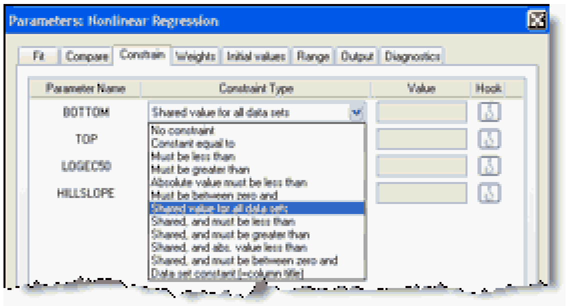
- If the models are not nested, then the F test is not valid so you should choose AIC_c. Note that Prism does not enforce this. It will calculate the F test even if the models are not nested, but the results won't be useful.

Constrain tab

Setting constraints is often essential to getting useful results.

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.

Remember that Prism has no "common sense", does not know how you did the experiment, and can't read your mind. Setting constraints is your job.

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 three effects:

- If nonlinear regression finds the best-fit values without ever running into the constraint, then setting the constraint had no effect. you would get exactly the same results without the constraint.
- If nonlinear regression would get "confused" and set parameters to values that make no sense, then setting a constraint can be very helpful. During the iterations, the curve fitting process may hit a constraint, but then it 'bounces back' and the best-fit value ends up in the allowed zone. Setting the constraint prevented nonlinear regression iterations from going astray, and you can interpret all the results normally. The only way to know this happened is to compare results with and without the constraint, but there is no point in

doing this as the results are interpreted the same either way.

- In some cases, the curve fitting process hits a constraint but isn't able to 'bounce back'. Prism reports that the best-fit value is right at the constraint border, and reports (at the top of the results) that the fit "Hit constraint 278". This usually means that you set the constraint incorrectly.

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. 63
- Each dataset is incomplete, but the entire family of datasets defines the parameters. See example. 62

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. Its value then 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 64.

Weights tab

ROUT coefficient

If you ask Prism to automatically exclude outliers (a choice on the Fit tab) or to count (but not remove from the analysis) outliers (a choice on the Diagnostics tab), set the ROUT coefficient to determine how aggressively Prism defines outliers.

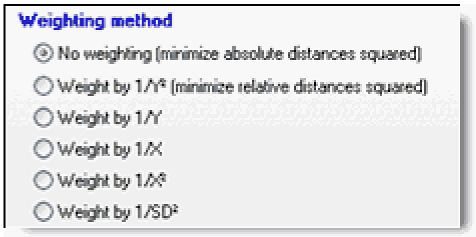
We recommend using a value of 1%. Our simulations have shown that if all the scatter is Gaussian, Prism will falsely find one or more outliers in about 2-3% of experiments. If there really are outliers present in the data, Prism will detect them with a False Discovery Rate less than 1%. See reference 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.

Unequal weighting

Prism offers six choices on the Weight tab of nonlinear regression:



Weighting method

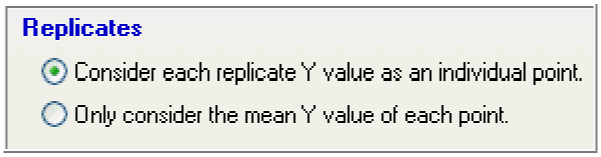
- ● No weighting (minimize absolute distances squared)
- ○ Weight by $1/Y^2$ (minimize relative distances squared)
- ○ Weight by 1/Y
- ○ Weight by 1/X
- ○ Weight by $1/X^2$
- ○ Weight by $1/SD^2$

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.

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.

Learn more about weighting 74. If you are confused by weighting, just stick to the default choice (no weighting; minimize sum-of-squares). Changing to a different weighting method rarely has a huge impact on the results.

Replicates



Replicates

- ● Consider each replicate Y value as an individual point.
- ○ Only consider the mean Y value of each point.

In most experiments, it is fair to consider each replicate to be an independent data point, and you should choose that first choice unless you have a strong reason not to.

Here is an example where 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. If one animal happens to respond more than the others, that will affect all the replicates. Since the replicates are not independent, you should fit the means.

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..

Initial values tab

> The choices on this tab rarely make a huge impact on the results. If you are new to nonlinear regression, you can leave this tab set to default values. If you have trouble getting fitting to work, then you might want to come back here and fuss with initial values.

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 237 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.

Range tab

> The choices on this tab are rarely needed. If you are new to nonlinear regression, you can leave this tab set to default values.

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.

Table of XY coordinates

Check this option if you want to see a table of XY coordinates of line segments defining the curve.

In prior versions of Prism, this table was critical to plotting the curve and interpolating unknowns. But Prism 5 can graph the curve and interpolate unknowns from the curve, without creating this table. Check the option to create this table only if you want to copy (or export) the table into another program. In most cases, you will have no need to view this table, and so should leave the option unchecked.

Output tab

> The choices on this tab rarely impact the results. If you are new to nonlinear regression, you can leave this tab set to default values.

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.

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

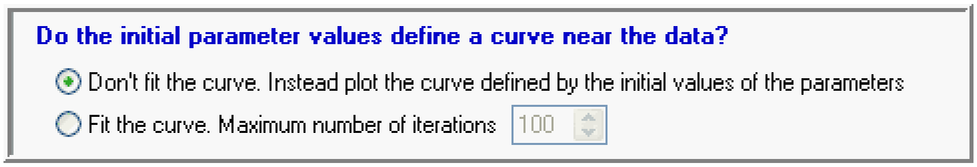
Prism offers two options here (dose ratios, and Ki from IC50). These are available only to be compatible with earlier versions of Prism, and are only useful if you select equations from the classic equations list.

With Prism 5, you can calculate a Gaddum/Schild EC50 shift 136 directly, without need to separately compute dose ratios. Similarly, you can fit competitive binding curves directly to determine the Ki for one 160 or two 164 sites, without a separate calculation of Ki from IC50.

Diagnostics tab

The choices on this tab usually won't impact the results, but will help you understand the results. If you are new to nonlinear regression, you can leave this tab set to default values. But as you gain more experience with nonlinear regression, you will want to get more complete results by checking options on this dialog.

Do the initial parameter values define a curve near the data?



Do the initial parameter values define a curve near the data?

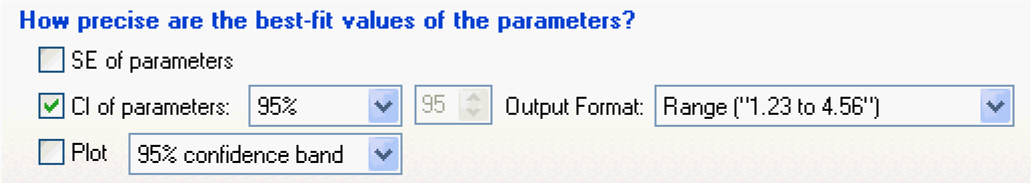
● Don't fit the curve. Instead plot the curve defined by the initial values of the parameters

○ Fit the curve. Maximum number of iterations 100

Nonlinear regression works iteratively, and begins with initial values 247 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 Diagnostics tab and check "Fit the curve".

While fitting a curve, Prism will stop after the maximum number of iterations set here. If you are running a script to automatically analyze many data tables, you might want to lower the maximum number of iterations so Prism won't waste time trying to fit impossible data.

How precise are the best-fit values of the parameters?



How precise are the best-fit values of the parameters?

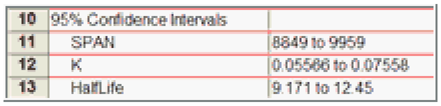
□ SE of parameters

☑ CI of parameters: 95% | 95 | Output Format: Range ("1.23 to 4.56")

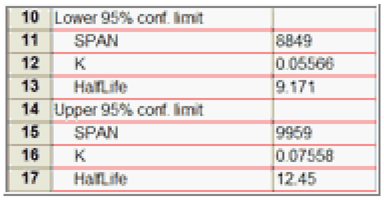
□ Plot | 95% confidence band

If your goal is to find the best-fit value of the parameters, you will also want to know how precise those estimates are. We suggest that you report **confidence intervals**, as inspecting the confidence intervals of best-fit parameters is an essential part of evaluating any nonlinear fit. **Standard errors** are intermediate values used to compute the confidence intervals, but are not very useful by themselves. Include standard errors in the output to compare Prism's results to those of another program that doesn't report confidence intervals.

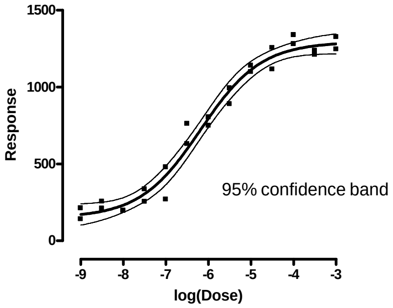
Choose to report confidence intervals as a range or separate blocks of lower and upper confidence limits (useful if you want to paste the results into another program).

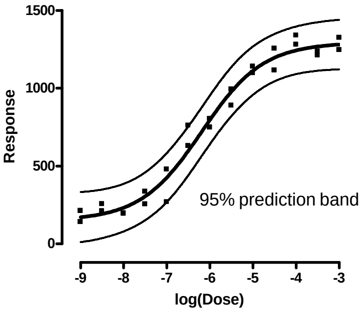


10	95% Confidence Intervals	
11	SPAN	8849 to 9959
12	K	0.05566 to 0.07558
13	HalfLife	9.171 to 12.45



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

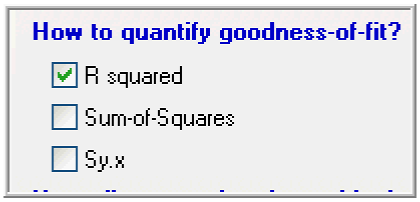




The 95% **confidence bands** enclose the area that you can be 95% sure contains the true curve. It gives you a visual sense of how well your data define the best-fit curve.

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.

How to quantify goodness-of-fit



How to quantify goodness-of-fit?

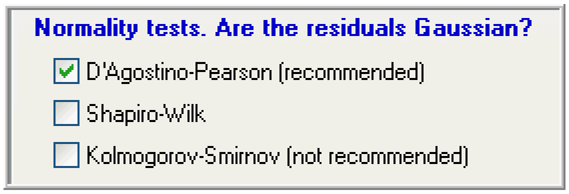
☑ R squared

□ Sum-of-Squares

□ Sy.x

you will probably want to ask Prism to report R2 256, simply because it is standard to do so, even though knowing R^2 doesn't really help you interpret the results. Reporting the sum-of-squares and $s_{y.x}$ will only be useful if you want to compare Prism's results to those of another program, or you want to do additional calculations by hand.

Normality tests



Normality tests. Are the residuals Gaussian?

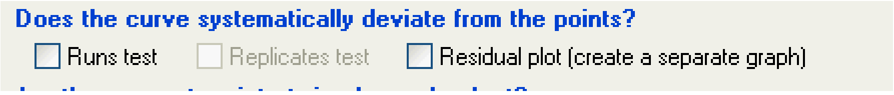
☑ D'Agostino-Pearson (recommended)

□ Shapiro-Wilk

□ Kolmogorov-Smirnov (not recommended)

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 255 on the residuals. Prism offers three normality tests. We recommend the D'Agostino-Pearson test.

Does the curve systematically deviate from the points?



Does the curve systematically deviate from the points?

□ Runs test □ Replicates test □ Residual plot (create a separate graph)

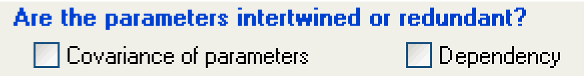
Does the curve follow the trend of the data? Or does the curve systematically deviate from the trend of the data? Prism offers two tests that answer these questions.

If you have entered replicate Y values, choose the replicates test 259 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 258 is available if you entered single Y values (no replicates) or chose to fit only the means rather than individual replicates (weighting 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 choose a residual plot 283, Prism creates a new graph. The X axis is the same as the graph of the data, while the Y axis plots the distance of each point from the curve (the residuals). Points with positive residuals are above the curve; points with negative residuals are below the curve. 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?



Are the parameters intertwined or redundant?

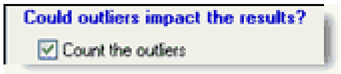
□ Covariance of parameters □ Dependency

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' 276.

We suggest that you report the dependency 261, and not bother with the covariance matrix 261.

When you are getting started with curve fitting, it is OK to leave both options unchecked.

Could outliers impact the results?



Could outliers impact the results?

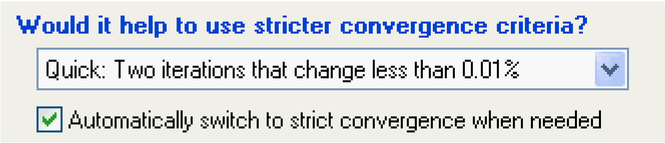
☑ Count the outliers

Nonlinear regression is based on the assumption that the scatter of data around the ideal curve follows a Gaussian distribution. 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.

Check this option to count the outliers, but leave them in the calculations. Choose how aggressively to define outliers by adjusting the ROUT coefficient 245.

If you chose the option in the Fit tab 242 to exclude outliers from the calculations, then this option to simply count outliers (in the Diagnostics tab) is not available.

Would it help to use stricter convergence criteria?



Would it help to use stricter convergence criteria?

Quick: Two iterations that change less than 0.01%

☑ Automatically switch to strict convergence when needed

Nonlinear regression is an iterative process. It starts with initial values 247 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.

Prism lets you define the convergence criteria in three ways. The medium choice is default, and will work fine in most cases. With this choice, nonlinear regression ends when five iterations in a row change the sum-of-squares by less than 0.0001%. If you are having trouble getting a reasonable fit, you might want to try the stricter definition of convergence: five iterations in a row change the sum-of-squares by less than 0.00000001%. It won't help very often, but is 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.

If you are fitting huge data sets, you can speed up the fit by using the 'quick' definition of convergence: Two iterations in a row change by less than 0.01%.

Nonlinear regression results

Interpreting results: Nonlinear regression

Standard errors and confidence intervals of parameters

Standard errors of best-fit 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 249). The calculation of the standard errors 77 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.

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 77 from the standard errors of the parameters.

How accurate are the standard errors and confidence intervals?

The standard errors reported by Prism (and virtually all other nonlinear regression programs) are based on some mathematical simplifications. They are called "asymptotic" or "approximate" standard errors. 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. You can test the accuracy of a confidence interval using simulations 29.

Sometimes Prism reports "very wide" instead of reporting the confidence interval

If you see the phrase 'very wide' instead of a confidence interval, you will also see the phrase 'ambiguous' 276 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.

Confidence intervals of transformed parameters

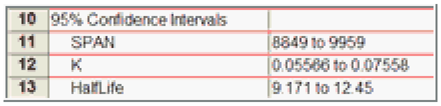
In addition to reporting the confidence intervals of each parameter in the model, Prism can also report confidence intervals for transforms 239 of those parameters. For example, when you fit an exponential model to determine the rate constant, Prism also fits the time constant tau, which is the reciprocal of the rate constant.

When you write your own equation, or clone an existing one, choose 239 between two ways to compute the confidence interval of each transformed parameter. If you pick a built-in equation, Prism always reportd asymmetrical confidence intervals of transformed parameters.

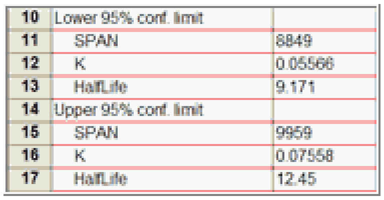
Do not mix up confidence intervals and confidence bands

It is easy to mix up confidence intervals and confidence bands. Choose both on the Diagnostics tab 249.

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:

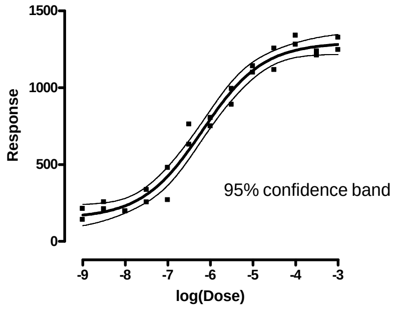


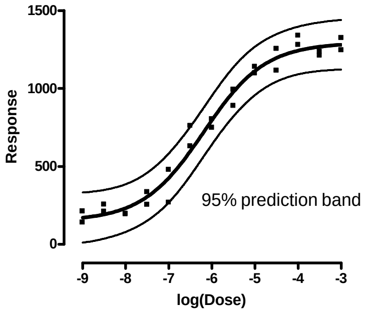
10	95% Confidence Intervals	
11	SPAN	8849 to 9959
12	K	0.05586 to 0.07558
13	HalfLife	9.171 to 12.45



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.





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?

A small P value is evidence that your data don't follow one of the assumptions of the regression. Things to consider:

- Fit a different model
- Weight 245 the data differently.
- Exclude outliers 70.

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.

Reference

[1] RB D'Agostino, "Tests for Normal Distribution" in *Goodness-Of-Fit Techniques* edited by RB

Goodness of fit of nonlinear regression

R²

Meaning of R²

The value R^2 quantifies goodness of fit. It is a fraction between 0.0 and 1.0, and has no units. Higher values indicate that the model fits the data better. 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 R^2=1.0, all points lie exactly on the curve with no scatter. If you know X you can calculate Y exactly. 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. How low is too low? There is really no general answer to that question. If you repeat an experiment many times, you will know what values of R^2 to expect, and can investigate further when R^2 is much lower than the expected value.

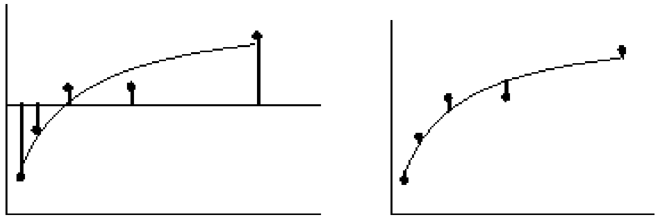
By tradition, statisticians use uppercase (R^2) for the results of nonlinear and multiple regression and lowercase (r^2) for the results of linear regression, but this is a distinction without a difference.

Tip: Don't make the mistake of using R^2 as your main criteria for whether a fit is reasonable. 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.

How R² is calculated

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, SSreg 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 (SSreg) equals 4165.



R^2 is calculated using this equation.

$$R^2 = 1.0 - \frac{SS_{reg}}{SS_{tot}} = 1.0 - \frac{4165}{62735} = 1.0 - 0.0664 = 0.9336$$

Even if you choose weighted nonlinear regression, Prism still computes R^2 using the formula above. In other words, it computes R^2 from the unweighted sum-of-squares

R^2 can be negative!

Note that R^2 is not really the square of anything. If SSreg is larger than SStot, R^2 will be negative. While it is surprising to see something called "squared" have a negative value, it is not impossible (since R^2 is not actually the square of R). R^2 will be negative when the best-fit curve fits the data even worse than does a horizontal line.

If R^2 is negative, check that you picked an appropriate model, and set any constraints correctly. 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.

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.

Standard deviation of residuals

The value sy.x is the standard deviation of the residuals, expressed in the same units as Y.

Prism calculates sy.x from the sum-of-squares (SS) and degrees of freedom (df, equal to number of data points minus the number of parameters fit) as:

$$s_{yx} = \sqrt{\frac{SS}{df}}$$

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/DF, where N is the number of points fit and DF is the number of degrees of freedom (equal to N -K, where K is the number of parameters fit).

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 259 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.

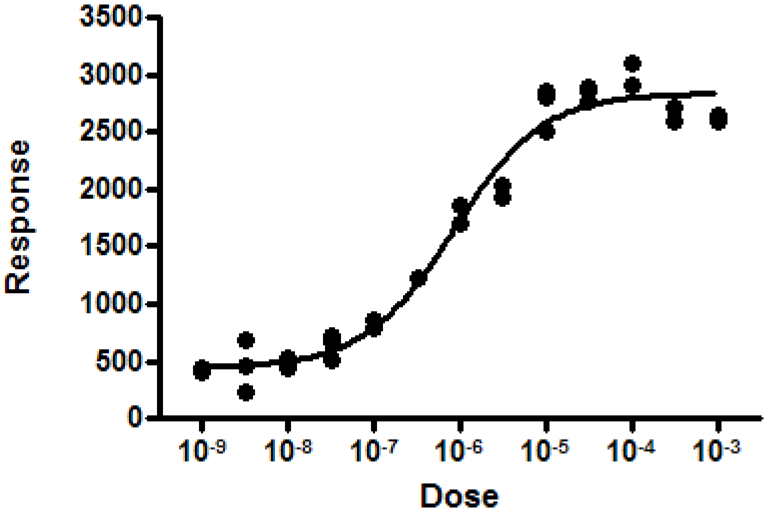
Replicates test

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

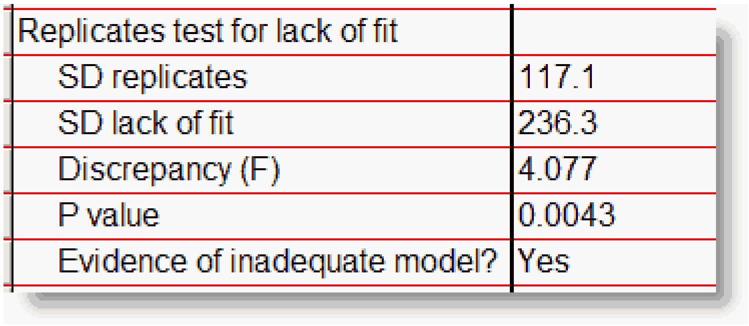


The response at the last two doses dips down a bit. Is this coincidence? Or evidence of a biphasic response?

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*, GAF Seber and CJ Wild, Wiley Interscience, 2003, pages 30-32.

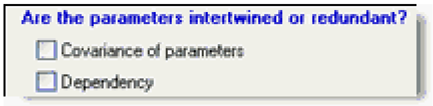
Dependency and covariance matrix

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. 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 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?

Dependency is reported for each parameter, and quantifies the degree to which that parameter is intertwined with others. Its value ranges from 0.0 to 1.0. A high value means that the value of a parameter is intertwined with the values of other parameters. Changing the value of the parameter will make the curve fit the data much worse. But changing other parameters can bring the curve back.

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 calls the fit 'ambiguous 276'.

With experimental data, of course, the value lies between 0.0 and 1.0. 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 79 will help you understand how Prism computes dependency.

Covariance matrix

If you want to see the covariance matrix, check an option on the Diagnostics tab. Prism will then report the normalized covariance between every pair of parameters. The value 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.

Each value in the covariance matrix tell you how much two parameters are intertwined. In contrast, each dependency value tells you how much that parameter is intertwined with all other parameters.

Tip. Rather than interpreting the covariance matrix, we suggest you focus on the dependency instead.

For the example, Prism reports these values for the covariance matrix

SPAN & K	0.5390

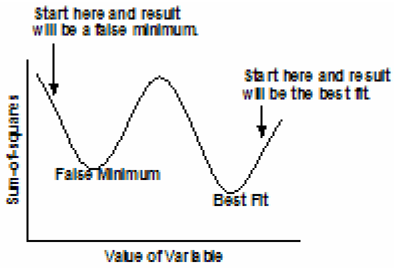
SPAN & PLATEAU	0.1048
K & PLATEAU	0.8109

With only three parameters, there are three values for dependency (one per parameter) and three values in the covariance matrix (one for each pair of parameters). With more parameters, the covariance matrix grows faster than the list of dependencies, which is why we

Could the fit be a local minimum?ıl.

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.



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:

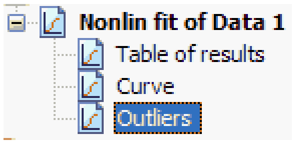
1. Note the values of the variables and the sum-of-squares from the first fit.
2. Make a large change to the initial values of one or more parameters and run the fit again.
3. Repeat step 2 several times.

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.

Outliers

The outliers table

If you choose to exclude or count outliers, one of the results tables will list all outliers.



Nonlin fit of Data 1
Table of results
Curve
Outliers

Keep in mind an important distinction:

- If you chose to count outliers (on the Diagnostics tab 249), then the outliers are still included in the fit.
- If you chose to exclude outliers (on the Fit tab 242), then the outliers are ignored by the fit, but are still included on the graph 284.

What does it mean to be an outlier

Prism uses the ROUT algorithm 66, using the value of Q that you can adjust in the Weights tab 245, 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 67.
- 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

Why results in Prism 5 can differ from Prism 4

Prism 5 does not use precisely the same algorithm as did Prism 4, so curve fitting results can be different in rare cases:

- If your fit is labeled "Ambiguous" by Prism 5, you know that you have a problem. See these suggestions 275 for dealing with ambiguous fits. 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 245, 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 has 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 handles weighting differently than did Prism 4. Prism 5 weights by the Y value of the *curve*, while Prism 4 (and earlier releases) weighted by the Y value of the *data.* The method used by Prism 5 is better, so the results of Prism 5 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 5 does an extra step. If one of the models is ambiguous 276, then Prism chooses the other model, without doing the F test or AIC comparison.
- Prism 5 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.

Interpreting results: Comparing models

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 276, 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 266 or using information theory and computation of AIC 267.

Interpreting the extra sum-of-squares F test

The extra sum-of-squares F test 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 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.

The F test is only valid if the two models being compared are nested. If you are comparing whether parameters are different between treatments or testing whether a parameter value is different than a hypothetical value, then the models are nested. if you are comparing the fits of two equations you chose, the models may or may not be nested. "Nested" means that one model is a simple case of the other. If the models are not nested, the F test is not valid. Prism does not test whether the models are nested or not.

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.

Prism names the null and alternative hypotheses, and reports the likelihood that each is correct It also reports the difference between the AICc values, 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.

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 Fit tab 242), 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.

Analysis checklists: Nonlinear regression

Analysis checklist: Fitting a model

This checklist is for when the nonlinear regression finished normally and your goal was to interpret the best-fit parameter values. Look elsewhere if your goal was to interpolate from a standard curve 272 or compare models 270.

View other checklists if your curve fit ended with a completion code: Bad initial values 274, Interrupted 274, Not converged 275, Ambiguous 276 Hit Constraint 278, Don't fit 279, Too few points 280, Perfect Fit 280 or Impossible weights 281.

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 272 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 258 and replicates 259 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 283 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 R^2, and inspecting the results can only be useful when you have done similar experiments in the past so know what values to expect.

Analysis checklist: Comparing nonlinear fits

This checklist is for when the nonlinear regression finished normally and your goal was to compare models. Look elsewhere if your goal was to interpret parameters 268 or interpolate from a standard curve 272.

View other checklists if your curve fit ended with a completion code: Bad initial values 274, Interrupted 274, Not converged 275, Ambiguous 276 Hit Constraint 278, Don't fit 279, Too few points 280, Perfect Fit 280 or Impossible weights 281.

Approach to comparing models

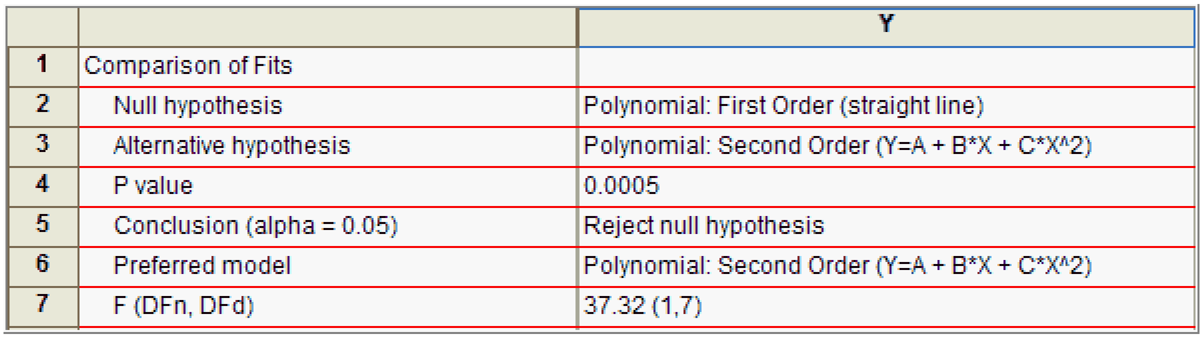
Which model is 'best'? At first, the answer seems simple. The goal of nonlinear regression is to minimize the sum-of-squares, so the model with the smaller sum-of-squares clearly fits best. But that is too simple.

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 58 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 276 or perfect 280, 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 268 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?**

Analysis checklist: Interpolating from a standard curve

This checklist is for when the nonlinear regression finished normally and your goal was to interpolate from a standard curve. Look elsewhere if your goal was to interpret parameters 268 or compare models 270.

View other checklists if your curve fit ended with a completion code: Bad initial values 274, Interrupted 274, Not converged 275, Ambiguous 276 Hit Constraint 278, Don't fit 279, Too few points 280, Perfect Fit 280 or Impossible weights 281.

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 268.

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 prediction bands too wide?

The prediction bands 282 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.

"Bad initial values"

This checklist is for when the nonlinear regression was not able to finish, but rather reported "Bad initial values" at the top of the results. View other checklists if your curve fit ended with a different completion code: Completed, 268 Interrupted 274, Not converged 275, ambiguous 276, Hit Constraint 278, Don't fit 279, Too few points 280, Perfect Fit 280 or Impossible weights 281.

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?

"Interrupted"

This checklist is for when the nonlinear regression was not able to finish, but rather reported "Interrupted" at the top of the results. View other checklists if your curve fit ended with a different completion code: Completed, 268 Bad initial values 274, Not converged 275, Ambiguous 276 Hit Constraint 278, Don't fit 279, Too few points 280, Perfect Fit 280 or Impossible weights 281.

What does 'interrupted' mean?

Prism reports 'interrupted' in two situations:

- Nonlinear regression takes more iterations than the maximum entered on the diagnostics tab (the default is 500).
- 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 the curve defined by your initial values come near your data? Check the option on the diagnostics tab 249 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?

arameter over all the data sets (global fitting)?

"Not converged"

This checklist is for when the nonlinear regression was not able to finish, but rather reported "Not coverged" at the top of the results. View other checklists if your curve fit ended with a different completion code: Completed, 268 Bad initial values 274, Interrupted 274, Ambiguous 276 Hit Constraint 278, Don't fit 279, Too few points 280, Perfect Fit 280 or Impossible weights 281.

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 249 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?

"Ambiguous"

e a parameter over all the data sets (global fitting)?

This checklist is for when the nonlinear regression finished, but reported that the fit was "ambiguous" at the top of the results. View other checklists if your curve fit ended with a different completion code: Completed, 268 Bad initial values 274, Interrupted 274, Not converged 275, Hit Constraint 278, Don't fit 279, Too few points 280, Perfect Fit 280 or Impossible weights 281.

Ambiguous fits

"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.

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.

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 the fit is 'ambiguous' you really can't interpret the best-fit values of some parameters.

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.

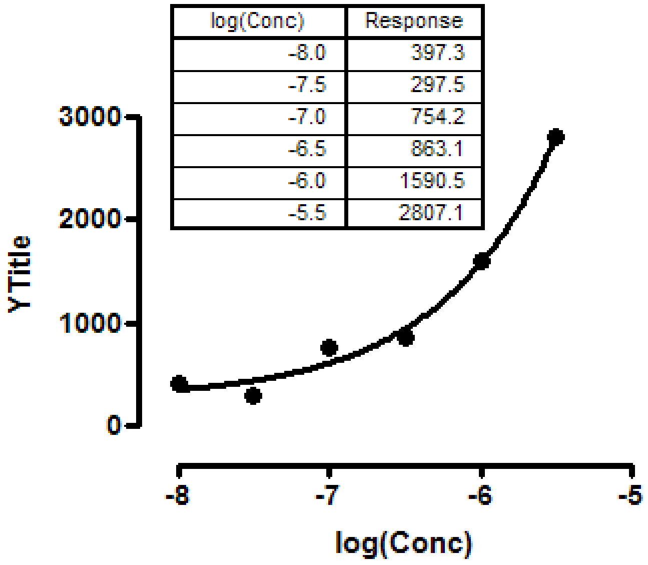
If your goal is to learn about your data by inspecting the values of the parameters, then you've got a real problem. At least one of the parameters has a best-fit value that you should not rely upon.

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

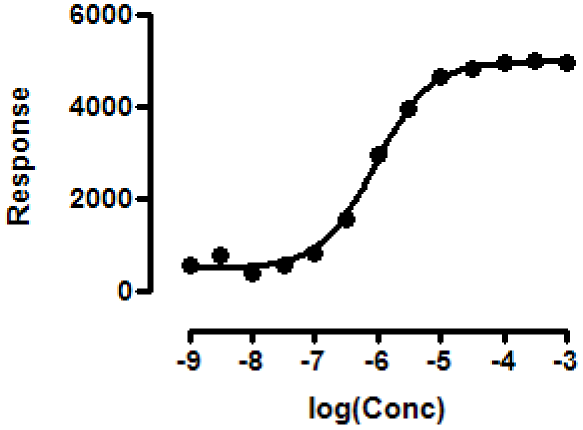
Data not collected over a wide enough range of X values



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

"Hit constraint"

This checklist is for when the nonlinear regression was not able to finish, but rather reported "Hit constraint" at the top of the results. View other checklists if your curve fit ended with a different completion code: Completed, 268 Bad initial values 274, Interrupted 274, Not converged 275, Ambiguous 276, Don't fit 279, Too few points 280, Perfect Fit 280 or Impossible weights 281.

Constraints

Prism reports "Hit constraint" when the best-fit value of one or more parameters is right at the limit of a constraint.

The Constrain tab lets you tell Prism that a parameter can have values only within a certain range of values. Constraints are most helpful when they guide the iterations (prevent impossible values), but the best-fit value ends up within the allowed zone.

In some cases, the curve fitting process hits a constraint but isn't able to 'bounce back'. In that case, the best-fit value is right at the constraint border. 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.

Note that hitting a constraint is **not** the same as constraining a parameter to a constant value. The best-fit values of all the parameters will be the same, but the standard errors, confidence intervals, covariance matrix, and dependencies will be different. Prism still treats the parameter that hit the constraint to be a parameter when it calculates these values (but doesn't when a parameter is constrained to a constant value).

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, and not right at one of the limits. This usually happens when you set the constraint incorrectly.

Checklist

✓ Did you enter the constraint correctly? Did you mix up "<" and ">".

✓ Did you enter the model correctly?

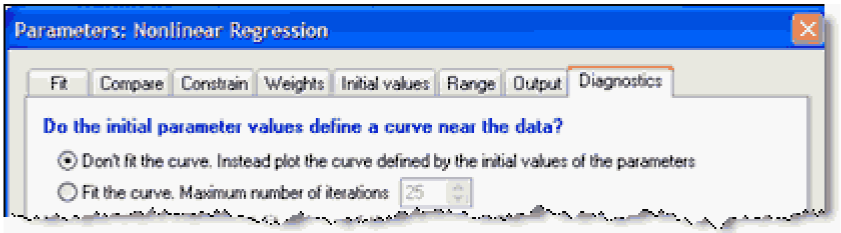
✓ Are you asking Prism to do the impossible -- too many parameters for your data?

✓ If your model has two phases or components, consider switching to a one phase or odel.

"Don't fit"

This checklist is for when the nonlinear regression process never started, and instead reports "Don't fit" at the top of the results. View other checklists if your curve fit ended with a different completion code: Completed, 268 Bad initial values 274, Interrupted 274, Not converged 275, ambiguous 276, Hit constraint 278, Too few points 280, Perfect Fit 280 or Impossible weights 281.

The top of the Diagnostics tab offers the choice to plot the curve defined by the initial values of the parameters.



Parameters: Nonlinear Regression

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.").

"Too few points"

This checklist is for when the nonlinear regression process never started, and instead reports "Too few points" at the top of the results. View other checklists if your curve fit ended with a different completion code: Completed, 268 Bad initial values 274, Interrupted 274, Not converged 275, ambiguous 276, Hit constraint 278, Don't fit 279, Perfect Fit 280 or Impossible weights 281.

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 280, 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 276 or not converged 275 fit.

"Perfect fit"

This checklist is for when nonlinear regression reports "Perfect fit" at the top of the results. View other checklists if your curve fit ended with a different completion code: Completed, 268 Bad initial values 274, Interrupted 274, Not converged 275, ambiguous 276, Hit constraint 278, Don't fit 279 , Too few points 280 or Impossible weights 281.

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.

"Impossible weights"

This checklist is for when nonlinear regression reports "Impossible weights" at the top of the results. View other checklists if your curve fit ended with a different completion code: Completed, 268 Bad initial values 274, Interrupted 274, Not converged 275, ambiguous 276, Hit constraint 278, Don't fit 279, Too few points 280, or Perfect fit 280.

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 weights tab and choose a different weighting method 245.

Graphing nonlinear regression

How to: Graphing nonlinear regression 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.

Prism 5 plots curves differently than did early versions

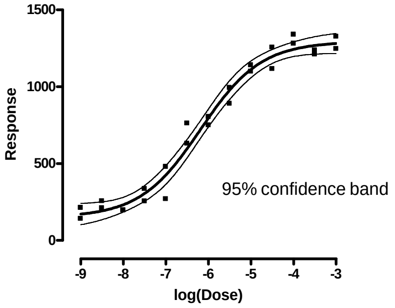
If you have used earlier versions of Prism, beware of these two differences:

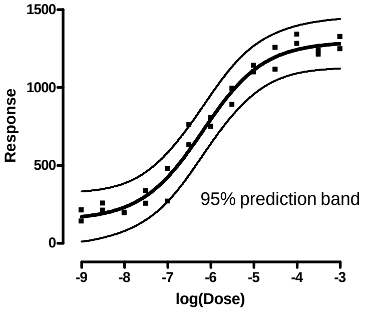
- With Prism 5, it is Ok to stretch an axis to a log scale, a choice on top of the Format Axis dialog. With prior versions, this didn't work well. The curve was defined as a series of line segments, and these would be far from equally spaced on a log axis. Prism 5 makes it work, so the curve will look smooth on either a linear or logarithmic axis.
- Prism plots the curve by drawing a set of small line segments. In earlier versions of Prism, you chose how many segments Prism should create. Prism 5 decides this automatically. If you want to create a table of XY coordinates 248 for exporting, you decide how many rows this table should have, but that choice only matters for creating the table and has no impact on how the curve is plotted.

Confidence and prediction bands

Distinguishing confidence from prediction bands

Prism can superimpose on your graph either confidence or prediction bands for the regression curve. Choose in the Diagnostics tab 249 of the Nonlinear regression parameters dialog. Learn how they are calculated 77.





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.

With each nonlinear fit, Prism can plot either the confidence bands or the prediction bands, but not both. To plot both on one graph, fit the curve twice and superimpose both fits on the same graph.

When to plot confidence and prediction bands

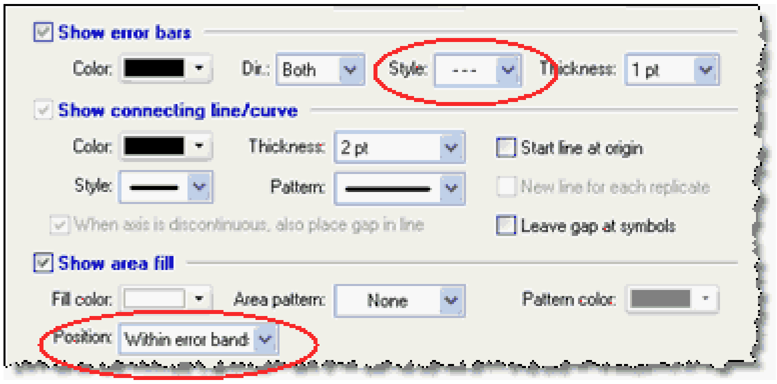
Confidence bands let you see how well your data define the curve. The confidence intervals for the parameters define the precision of the fit. The confidence bands sort of combine these in a visual way. Use confidence bands to learn how well your data define the best-fit curve.

Prediction bands are wider, to also include the scatter of the data. Use prediction bands when your main goal is interpolation from a standard curve.

Fine-tuning the appearance of the confidence and prediction bands

If you check the option on the Fit tab, Prism will automatically superimpose the confidence or prediction band on the graph.

To adjust the appearance of the confidence or prediction bands, go to the Format Graph dialog, select the dataset that represents the best fit curve, 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

How to see the confidence or prediction bands numerically

If you want to see the confidence or prediction bands as numbers, first go to the Range tab and check the option to create a table of XY values. Then view the "Curve" results page. At each X value, it shows you the Y value of the curve, as well as plus and minus error values ediction or confidence band.

Residual plot

When to plot residuals

A residual is the distance of a point from the curve. A residual is positive when the point is above the curve, and is negative when the point is below the curve. The residual table has the same X values as the original data, but the Y values are the vertical distances of the point from the curve.

Create a residual plot if you aren't sure that your data really follow the model you selected. Mild deviations of data from a model are often easier to spot on a residual plot.

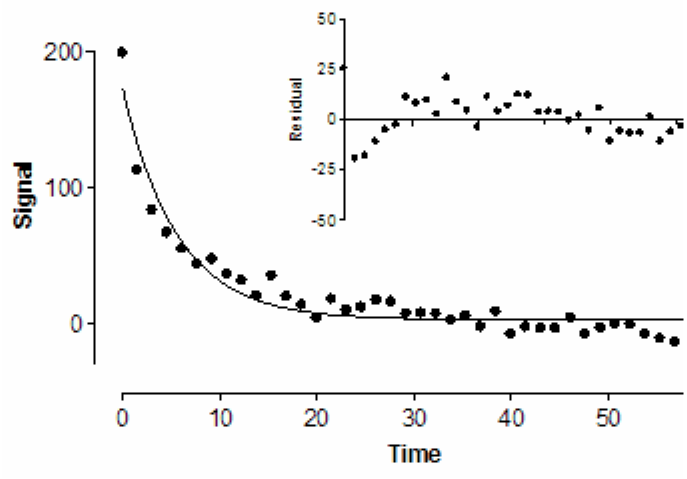
How to graph residuals

Choose to create a residual plot by checking an option on the Diagnostics tab of the nonlinear regression dialog. Prism will automatically make a new graph. The residuals are tabulated in a separate page of results, and you can use these like any other table (make additional graphs, plot on other graphs, transform...).

Interpreting a residual plot

The X axis of the residual plot is the same as the graph of the data, while the Y axis is the distance of each point from the curve. Points with positive residuals are above the curve; points with negative residuals are below the curve.

An example is shown below, with a graph of the data and curve combined with a residual plot in a layout. If you look carefully at the curve on the left, you will see that the data points are not randomly distributed above and below the curve. There are clusters of points at early and late times that are below the curve, and a cluster of points at middle time points that are above the curve. This is much easier to see on the graph of the residuals in the inset. The data are not randomly scattered above and below the X-axis.



Graphing outliers

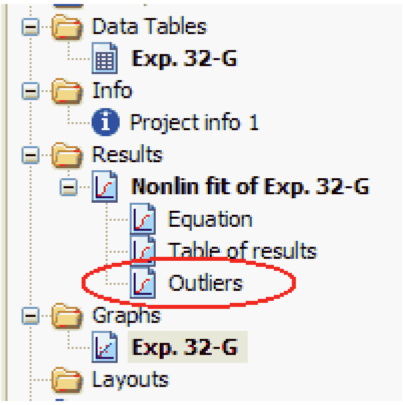
How to identify or eliminate outliers

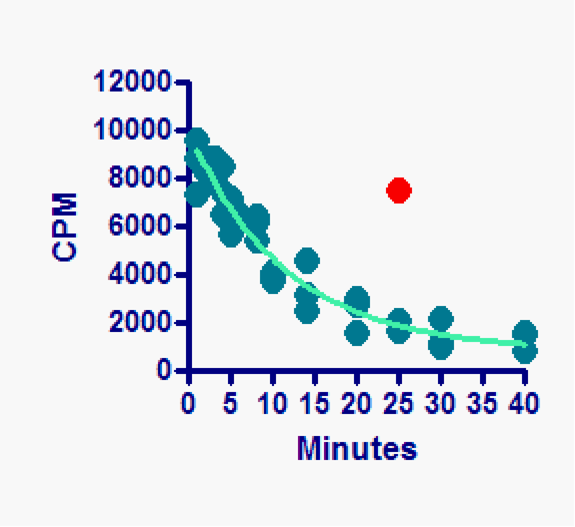
Prism can identify outliers from the curve fit, and you can choose whether these should simply be plotted or also ignored by the curve fitting process. Choose to count the outliers on the Diagnostics tab 249, and to eliminate outliers on the Fit tab 70 of the nonlinear regression dialog.

You can adjust the value of Q 245 to redefine how aggressively Prism defines outliers.

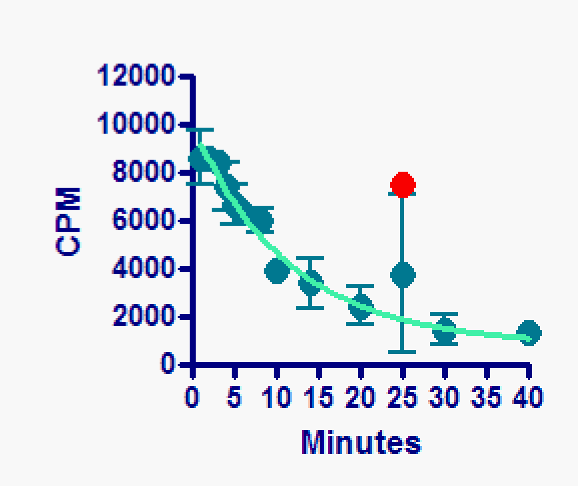
How Prism plots outliers

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).



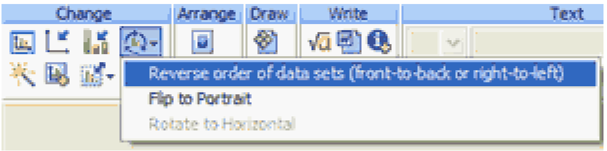


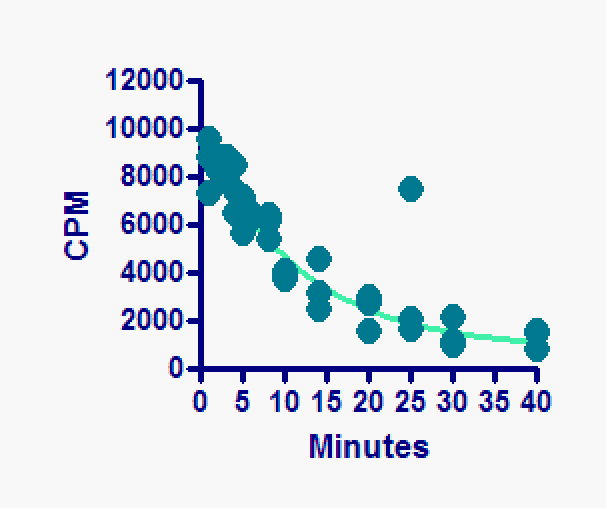
The example above shows a graph where each replicate is plotted individually. The example below shows a graph which plots mean and SD error bars. The error bars are computed from **all the data**, including any outliers. The outliers are then superimposed on the graph as well.



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).

Index