

GraphPad Curve Fitting Guide

**GraphPad Software Inc.
www.graphpad.com**

© 1995-2014 GraphPad Software, Inc.

This is one of three companion guides to GraphPad Prism 6.
All are available as web pages on graphpad.com.

Table of Contents

Foreword	0
Part I PRINCIPLES OF CURVE FITTING	8
1 Understanding mathematical models.....	8
What is a model?	9
Three example models	9
The problem with choosing models automatically	12
Advice: How to understand a model	12
2 Principles of linear regression.....	13
The goal of linear regression	14
How linear regression works	15
Comparing linear regression to correlation	16
Comparing linear regression to nonlinear regression	17
Advice: Look at the graph	18
Advice: Avoid Scatchard, Lineweaver-Burke and similar transforms	19
3 Getting started with nonlinear regression.....	21
Distinguishing nonlinear regression from other kinds of regression	21
The goal of nonlinear regression	22
The six steps of nonlinear regression	24
Preparing data for nonlinear regression	25
Don't fit a model to smoothed data	26
Reparameterizing an equation can help	27
4 Weighted nonlinear regression.....	30
The need for unequal weighting in nonlinear regression	30
Math theory of weighting	31
Don't use weighted regression with normalized data	35
What are the consequences of choosing the wrong weighting method	37
5 The many uses of global nonlinear regression.....	40
What is global nonlinear regression?	41
The uses of global nonlinear regression	41
Using global regression to fit incomplete datasets	42
Fitting models where the parameters are defined by multiple data sets	43
Advice: Don't use global regression if datasets use different units	45
6 Comparing fits of nonlinear models.....	45
Questions that can be answered by comparing models	46
Approaches to comparing models	48
How the F test works to compare models	50
How the AICc computations work	51
7 Outlier elimination and robust nonlinear regression.....	53
When to use automatic outlier removal	53
When to avoid automatic outlier removal	54
Outliers aren't always 'bad' points	56
The ROUT method of identifying outliers	57
Robust nonlinear regression	59
8 How nonlinear regression works.....	60
Why minimize the sum-of-squares?	61
How nonlinear regression works	62

Nonlinear regression with unequal weights	65
How standard errors and confidence intervals are computed	66
How confidence and prediction bands are computed	67
Replicates	68
How dependency is calculated	70
How confidence and prediction bands are computed	72
Who developed nonlinear regression?	74
Part II CURVE FITTING WITH PRISM 6	74
1 What's new in Prism 6 (regression)?.....	74
2 Linear regression with Prism.....	75
How to: Linear regression	76
Finding the best-fit slope and intercept.....	76
Interpolating from a linear standard curve.....	79
Advice: When to fit a line with nonlinear regression.....	82
Results of linear regression	84
Slope and intercept.....	84
r ² , a measure of goodness-of-fit of linear regression.....	86
Is the slope significantly different than zero?.....	88
Comparing slopes and intercepts.....	89
Runs test following linear regression.....	90
Analysis checklist: Linear regression.....	91
Graphing tips: Linear regression.....	92
Questions and answers.....	95
Deming regression	100
Key concepts: Deming regression.....	100
How to: Deming regression.....	100
Q&A: Deming Regression.....	102
Analysis checklist: Deming regression.....	103
3 Interpolating from a standard curve.....	104
Key concept: Interpolating	104
How to interpolate	105
Example: Interpolating from a sigmoidal standard curve	107
Equations used for interpolating	111
The results of interpolation	113
Interpolating with replicates in side-by-side subcolumns	114
Interpolating several data sets at once	116
When X values are logarithms	117
Analysis checklist: Interpolating	118
Reasons for blank (missing) results	120
Q&A: Interpolating	121
How Prism interpolates	122
Standard Addition Method	123
4 Nonlinear regression tutorials.....	125
Example: Fitting an enzyme kinetics curve	126
Example: Comparing two enzyme kinetics models	131
Example: Automatic outlier elimination (exponential decay)	135
Example: Global nonlinear regression (dose-response curves)	138
Example: Ambiguous fit (dose-response)	145
5 Nonlinear regression with Prism.....	151
How to fit a model with Prism	152
Which choices are essential?	154

Nonlinear regression choices	154
Fit tab	155
Compare tab.....	157
Constrain tab.....	158
Weights tab.....	160
Initial values tab.....	162
Range tab.....	164
Output tab.....	164
Diagnostics tab.....	165
Graphing tips: Nonlinear regression	170
Graphing best-fit curves.....	171
Graphing confidence and prediction bands.....	171
Adding the equation to the graph.....	175
Graphing outliers.....	176
Residual plot.....	179
6 Interpreting nonlinear regression results.....	181
Interpreting results: Nonlinear regression	181
Standard errors and confidence intervals of parameters.....	183
Normality tests of residuals.....	185
R squared.....	187
Sum-of-squares.....	191
Why Prism doesn't report the chi-square of the fit.....	191
Runs test.....	193
Replicates test.....	194
Dependency of each parameter.....	197
Covariance matrix.....	199
Confidence and prediction bands.....	200
Hougaard's measure of skew ness.....	204
Could the fit be a local minimum?.....	206
Outliers	207
Troubleshooting nonlinear regression.....	209
Why results in Prism 5 and 6 can differ from Prism 4.....	211
Interpreting results: Comparing models	212
Interpreting comparison of models.....	212
Interpreting the extra sum-of-squares F test.....	213
Interpreting AIC model comparison.....	215
How Prism compares models when outliers are eliminated.....	215
Interpreting the adjusted R2.....	216
Analysis checklists: Nonlinear regression	217
Analysis checklist: Fitting a model.....	218
Analysis checklist: Comparing nonlinear fits.....	221
Analysis checklist: Interpolating from a standard curve.....	223
Error messages from nonlinear regression	225
"Bad initial values".....	226
"Interrupted".....	226
"Not converged".....	227
"Ambiguous".....	228
"Hit constraint".....	231
"Don't fit".....	232
"Too few points".....	232
"Perfect fit".....	233
"Impossible weights".....	233
"Equation not defined".....	233
"Can't calculate".....	234

7 Models (equations) built-in to Prism.....	234
Dose-response - Key concepts	235
What are dose-response curves?.....	235
The EC50.....	236
Confidence intervals of the EC50.....	238
Hill slope	240
Choosing a dose-response equation.....	241
Pros and cons of normalizing the data.....	243
Converting concentration to log(concentration).....	244
The term "logistic".....	245
50% of what? Relative vs absolute IC50.....	248
Fitting the absolute IC50.....	251
Incomplete dose-response curves.....	253
Troubleshooting fits of dose-response curves.....	254
Dose-response - Stimulation	255
Equation: log(agonist) vs. response.....	255
Equation: log(agonist) vs. response -- Variable slope.....	256
Equation: log(agonist) vs. normalized response.....	258
Equation: log(agonist) vs. normalized response -- Variable slope.....	259
Dose-response - Inhibition	261
Equation: log(inhibitor) vs. response.....	261
Equation: log(inhibitor) vs. response -- Variable slope.....	262
Equation: log(inhibitor) vs. normalized response.....	264
Equation: log(inhibitor) vs. normalized response -- Variable slope.....	265
Dose-response -- Special	267
Asymmetrical (five parameter).....	267
Equation: Biphasic dose-response.....	269
Equation: Bell-shaped dose-response.....	271
Equation: Operational model - Depletion.....	272
Equation: Operational model - Partial agonist.....	275
Equation: Gaddum/Schild EC50 shift.....	277
Equation: EC50 shift.....	279
Equation: Allosteric EC50 shift.....	281
Equation: ECanything.....	283
Receptor binding - Key concepts	285
Law of mass action.....	285
Nonspecific binding.....	288
Ligand depletion.....	289
The radioactivity web calculator.....	290
Receptor binding - Saturation binding	291
Key concepts: Saturation binding.....	291
Equation: One site -- Total binding.....	292
Equation: One site -- Fit total and nonspecific binding.....	293
Equation: One site -- Total, accounting for ligand depletion.....	295
Equation: One site -- Specific binding.....	297
Equation: One site -- Specific binding with Hill slope.....	300
Binding potential.....	302
Equation: Two sites -- Specific binding only.....	304
Equation: Two sites -- Fit total and nonspecific binding.....	307
Equation: One site with allosteric modulator.....	309
Receptor binding - Competitive binding	311
Key concepts: Competitive binding.....	311
Equation: One site - Fit Ki.....	312
Equation: One site - Fit logIC50.....	314

Equation: Two sites - Fit K_i	315
Equation: Two sites - Fit $\log IC_{50}$	317
Equation: One site - Ligand depletion.....	318
Equation: One site - Homologous.....	321
Equation: Allosteric modulator.....	322
Receptor binding - Kinetics	324
Key concepts: Kinetics of binding.....	325
Equation: Dissociation kinetics.....	326
Equation: Association kinetics (one ligand concentration).....	328
Equation: Association kinetics (two ligand concentrations).....	329
Equation: Association then dissociation.....	331
Equation: Kinetics of competitive binding.....	333
Enzyme kinetics -- Key concepts	335
Key concepts: Terminology.....	335
Key concepts: Assumptions.....	336
Enzyme kinetics - Substrate vs. velocity	337
Key concepts: Substrate vs. velocity.....	337
Equation: Michaelis-Menten model.....	338
Equation: Determine k_{cat}	340
Equation: Allosteric sigmoidal.....	342
Enzyme kinetics -- Inhibition	344
Key concepts: Enzyme inhibition.....	344
Equation: Competitive inhibition.....	345
Equation: Noncompetitive inhibition.....	347
Equation: Uncompetitive inhibition.....	349
Equation: Mixed-model inhibition.....	350
Equation: Substrate inhibition.....	352
Equation: Tight inhibition (Morrison equation).....	353
Exponential	355
Key concepts: Exponential equations.....	355
Key concepts: Derivation of exponential decay.....	356
Equation: One phase decay.....	357
Equation: Plateau followed by one phase decay.....	359
Equation: Two phase decay.....	360
Equation: Three phase decay.....	362
Equation: One phase association.....	364
Equation: Plateau followed by one phase association.....	365
Equation: Two phase association.....	367
Equation: Exponential growth.....	368
Lines	370
Key concepts: Fitting lines.....	370
Equation: Fitting a straight line with nonlinear regression.....	371
Equation: Line through origin.....	372
Equation: Segmental linear regression.....	374
Equation: Fitting a straight line on a semi-log or log-log graph.....	376
Equation: Fitting a straight line on a graph with a probability axis.....	379
Polynomial	381
Key concepts: Polynomial.....	381
Centered polynomial equations.....	382
Equations: Polynomial models.....	384
Gaussian	385
Key concepts: Gaussian.....	385
Equation: Gaussian distribution.....	386
Equation: Log Gaussian distribution.....	388

Equation: Cumulative Gaussian distribution.....	389
Equation: Lorentzian.....	391
Sine waves	393
Standard sine wave.....	393
Damped sine wave.....	394
Sinc wave.....	395
Classic equations from prior versions of Prism	396
Equation: One site binding (hyperbola).....	396
Equation: Two site binding.....	397
Equation: Sigmoidal dose-response.....	397
Equation: Sigmoidal dose-response (variable slope).....	398
Equation: One site competition.....	399
Equation: Two site competition.....	400
Equation: Boltzmann sigmoid.....	400
Equation: One phase exponential decay.....	401
Equation: Two phase exponential decay.....	402
Equation: One phase exponential association.....	402
Equation: Two phase exponential association.....	403
Equation: Exponential growth.....	403
Equation: Power series.....	404
Equation: Sine wave.....	404
Equation: Gaussian distribution.....	405
8 Entering a user-defined model into Prism.....	405
Overview: User-defined equations	406
How to: Enter a new equation	407
How to: Clone an equation	409
How to: Manage your list of equations	410
Syntax of user-defined equations	411
Multiline models	413
Limitations when entering equations	414
Entering a differential equation	415
Entering an implicit equation	416
Available functions for user-defined equations	418
Fitting different segments of the data to different models	421
Fitting different models to different data sets	423
Column constants	424
Defining equation with two (or more) independent variables	426
Reparameterizing an equation	429
Rules for initial values	433
Default constraints	437
Reporting transforms of parameters	438
9 Plotting a function.....	442
How to: Plot a function	442
Plotting t, z, F or chi-square distributions	444
Plotting a binomial or Poisson distribution	445
10 Fitting a curve without a model.....	446
Spline and Lowess curves	447
Using nonlinear regression with an empirical model	448

Index

1 PRINCIPLES OF CURVE FITTING

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

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

- If your goal is to fit a model to your data in order to obtain best-fit values of the parameters, and want to learn the principles first, then read this principles section before trying to fit curves.
- If you already understand the principles of nonlinear regression, and want to see how to fit curves with Prism, [jump right to the tutorials](#)^[125].
- If your goal is to simply fit a smooth curve in order to interpolate values from the curve, there is no need to learn much theory. Jump right to an [explanation of interpolation with Prism](#).^[104]
- If your goal is to create a spline (a curve that goes through every data point) or a loess curve (shows the general trend with a curve that can be quite jagged), you can jump right to [the instructions for that analysis](#)^[447].

1.1 Understanding mathematical models

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

- [What exactly is a mathematical model?](#)^[9]
- Learn from [three example models](#)^[9].
- It sure would be nice if models could be chosen by Prism. [Why is this not possible?](#)^[12]
- When trying to understand a model, [here are some tips for figuring out what it means](#)^[12].
- In addition to fitting a model to data, Prism can [plot a function](#)^[442], or a family of functions. You choose a function and values for the

parameters, and Prism will plot predicted Y values for a range of X values.

1.1.1 What is a model?

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

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

A mathematical model is neither a hypothesis nor a theory. Unlike scientific hypotheses, a model is not verifiable directly by an experiment. For all models are both true and false.... The validation of a model is not that it is "true" but that it generates good testable hypotheses relevant to important problems.

R. Levins, Am. Scientist 54:421-31, 1966

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

1.1.2 Three example models

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

Optical density as a function of concentration

Background

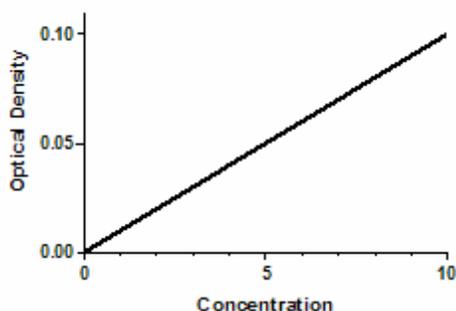
Colorimetric chemical assays are based on a simple principle. Add appropriate

reactants to your samples to initiate a chemical reaction whose product is colored. When you terminate the reaction, the concentration of colored product is proportional to the initial concentration of the substance you want to assay.

Model

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

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



Reality check

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

- Negative X values are meaningless, as concentrations cannot be negative.
- The model may fail at high concentrations of substance where the reaction is no longer limited by the concentration of substance.
- The model may also fail at high concentrations if the solution becomes so dark (the optical density is so high) that little light reaches the detector. At that point, the noise of the instrument may exceed the signal.

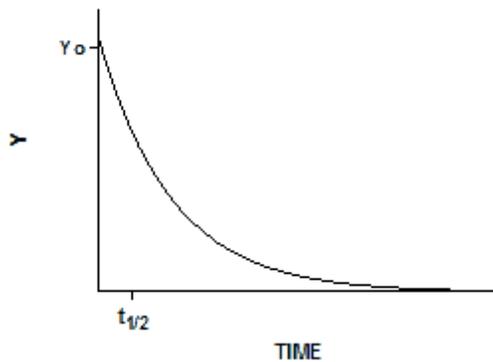
It is not unusual for a model to work only for a certain range of values. You just have to be aware of the limitations, and not try to use the model outside of its useful range.

Exponential decay

Exponential equations whenever the rate at which something happens is proportional to the amount which is left. Examples include ligands dissociating

from receptors, decay of radioactive isotopes, and metabolism of drugs. Expressed as a differential equation:

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

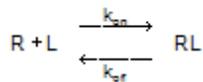


Converting the differential equation into a model that defines Y at various times requires some calculus. There is only one function whose derivative is proportional to Y, the exponential function. Integrate both sides of the equation to obtain a new exponential equation that defines Y as a function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

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

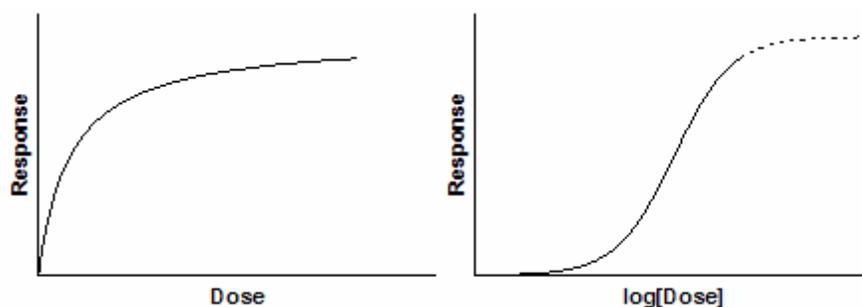
Equilibrium binding

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



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:

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



1.1.3 The problem with choosing models automatically

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

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

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

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

1.1.4 Advice: How to understand a model

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

function of substrate concentration:

$$Y = V_{\max} * X / (K_m + X)$$

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 K_m is added to X. It only makes sense to add things that are expressed in the same units, so K_m must be expressed in the same concentration units as X. This means that the units cancel in the term $X/(K_m + X)$, so V_{\max} 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 K_m , the denominator $(X+K_m)$ has a value very similar to X. So the ratio $X/(X+K_m)$ approaches 1.0, and Y approaches V_{\max} . So the graph of the model must level off at $Y=V_{\max}$ as X gets very large.

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

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

Tip 5. Graph the model with various parameter values

Graphing a family of curves with various values for the parameters can help you visualize what the parameters mean. To do this with Prism, use the analysis "[Create a family of theoretical curves](#)"⁴⁴².

1.2 Principles of linear regression

Linear regression fits a straight line through your data to find

the best-fit value of the slope and intercept.

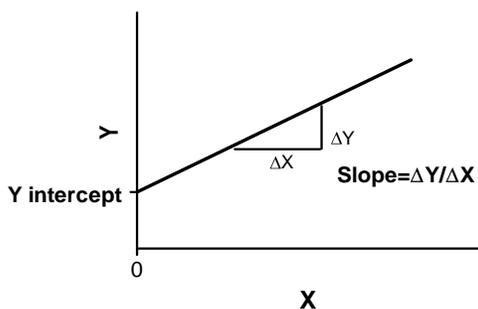
- [What is the goal of linear regression?](#)^[14]
- [How does it work?](#)^[15]
- Linear regression and correlation are often confused. [How are they distinct?](#)^[16]
- Linear regression is just a special case of nonlinear regression. [How do they differ?](#)^[17]
- Linear regression is sometimes used on transformed data to analyze Scatchard, Lineweaver-Burke and similar plots. [Why is this is not a good way to analyze data?](#)^[19]

1.2.1 The goal of linear regression

What is linear regression?

Linear regression fits this model to your data:

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



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

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

Correlation and linear regression are not the same. [Review the differences.](#)^[16]

1.2.2 How linear regression works

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

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

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

1.2.3 Comparing linear regression to correlation

Linear regression is distinct from correlation.

What is the goal?

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

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

What kind of data?

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

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

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

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

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

Assumptions

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

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

Relationship between results

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

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

1.2.4 Comparing linear regression to nonlinear regression

The goal of linear and nonlinear regression

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

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

How linear and nonlinear regression work

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

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

Nonlinear regression uses a computationally intensive, [iterative approach](#)^[62] 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.

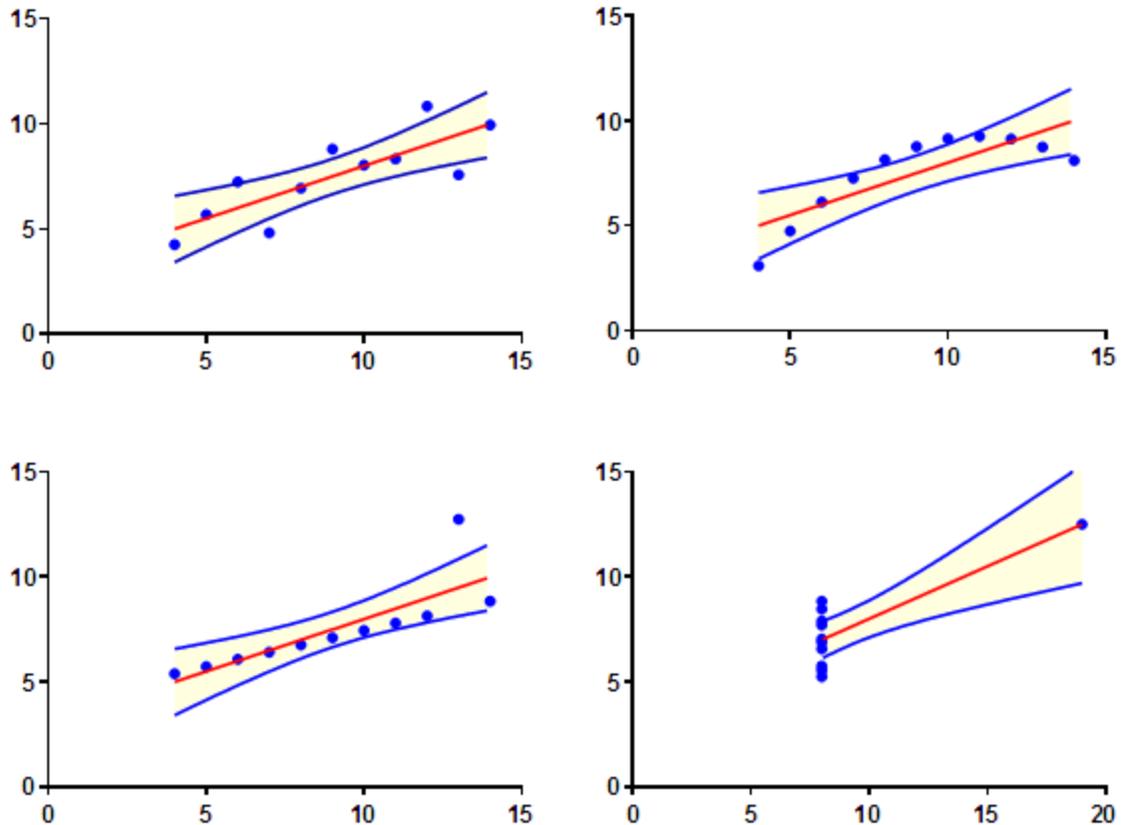
Even if your goal is to fit a straight line through your data, there are [many situations](#)^[82] where it makes sense to choose nonlinear regression rather than linear regression.

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](#)^[19] such as Scatchard or Lineweaver-Burke transforms whose only goal is to linearize your data.

1.2.5 Advice: Look at the graph

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

- The mean values of X and Y
- The slopes and intercepts
- r^2
- The standard errors and confidence intervals of the slope and intercept
- The standard deviation of the residuals ($sy.x$)
- The F ratio and P value when testing the null hypothesis that the slope is really 0.0 (horizontal).
- The [covariance](#)^[197] between slope and intercept (computed by Prism's nonlinear regression analysis)



Differences seen in linear regression are:

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

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

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

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

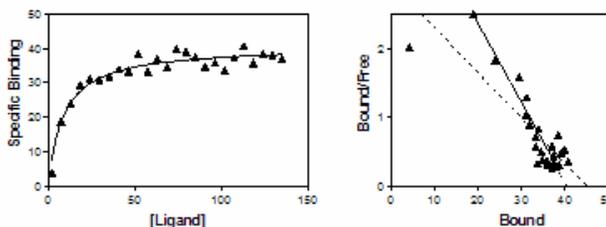
Before nonlinear regression was readily available, the best way to analyze nonlinear data was to transform the data to create a linear graph, and then analyze the transformed data with linear regression. Examples include Lineweaver-Burke plots of enzyme kinetic data, Scatchard plots of binding data,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

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

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

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

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



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

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

1.3 Getting started with nonlinear regression

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

- [Distinguishing nonlinear regression from other kinds of regression](#)^[21]
- [The goal of nonlinear regression](#)^[22]
- [The six steps of nonlinear regression](#)^[24]
- [Preparing data for nonlinear regression](#)^[25]
- [Don't fit a model to smoothed data](#)^[26]

1.3.1 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](#)^[17].

Polynomial regression

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

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 6 does not perform multiple regression. But by using [column constants](#)⁴²⁴, 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 6 does not perform logistic regression.

Proportional hazards regression

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

GraphPad Prism does not perform proportional hazards regression.

1.3.2 The goal of nonlinear regression

Nonlinear regression is used for two purposes

Scientists use nonlinear regression with one of two distinct goals:

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

perhaps to draw a graph with a smooth curve. If this is your goal, you can assess it purely by looking at the graph of data and curve. There is no need to learn much theory. Jump right to an [explanation of interpolation with Prism](#).^[104]

The general idea of regression

Linear regression fits a straight-line model to your data. Nonlinear regression extends this idea to fit any model to your data. Distinguish nonlinear regression [from linear regression](#)^[17], and [from other types of regression](#)^[21].

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

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

Least-squares regression

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

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

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

1.3.3 The six steps of nonlinear regression

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

Step 1. Choose a model

Nonlinear regression fits a model to your data. You must, therefore, choose a model or enter a new model. This is a scientific decision that must be made by someone who understands the scientific goals of the experiment. [Why a computer program should not pick a model for you](#)¹².

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

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

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

Step 3. Choose (or review) initial values

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

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

When fitting a simple model to clean data, it won't matter much if the initial

values are fairly far from the correct values. You'll get the same best-fit curve no matter what initial values you use, unless the initial values are extremely far from correct. Initial values matter more when your data have a lot of scatter or your model has many variables.

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

If you enter data into two or more data set columns, Prism will fit them all in one analysis. But each fit will be independent of the others unless you specify that one or more parameters are shared. When you share parameters, the analysis is called a [global nonlinear regression](#)^[41].

Step 5. Decide on a weighting scheme

Nonlinear regression programs generally weight each point equally. But there are many ways to [differentially weight the points](#)^[160].

Step 6. Choose which results you want reported

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

1.3.4 Preparing data for nonlinear regression

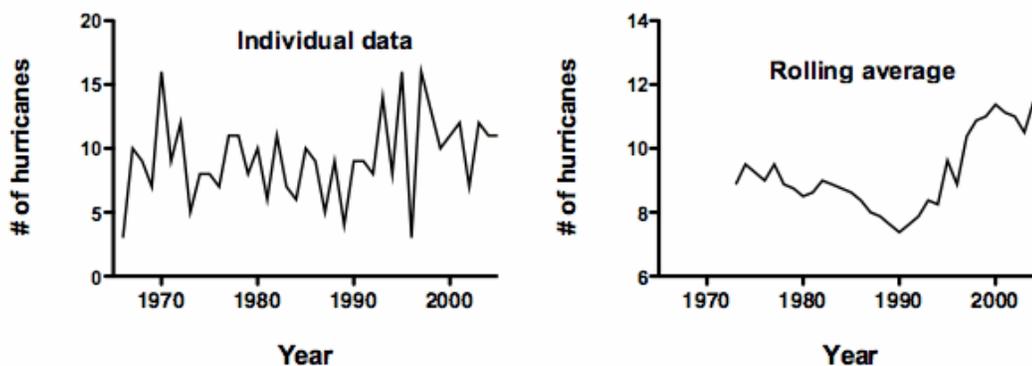
You must create an XY data table in Prism, for use with nonlinear regression.

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

- [Avoid linearizing transforms](#)^[19] such as Scatchard and Lineweaver-Burke plots. Such plots are useful for displaying data but are obsolete for data analysis.
- Transforming X values can be convenient, and will not change the results of regression (so long as the model is adjusted accordingly). Use Prism's [Transform analysis](#) to do this.
- [Don't smooth your data](#)^[26]. You will get invalid nonlinear regression results. Fit the raw data.

- Transforming Y values to change units or to subtract a baseline can be convenient, and will not substantially affect nonlinear regression. Use Prism's [Transform analysis](#) to do this.
- Avoid nonlinear Y transforms (reciprocals, logs) unless you have a very good reason. Such a transform can be useful if it equalizes the variances (so the scatter at all points along the curve is about the same), but should not be done just to linearize the data.
- Enter raw replicates when possible, and not just mean and SD or SEM. Prism will fit the same curve either way, but there are two advantages of entering the raw data. First, you can plot every individual replicate so you see the actual data. Second, entering the individual replicates lets you choose robust nonlinear regression or automatic outlier elimination
- If you have entered replicates, first graph the individual replicates rather than mean and error bar. You may want to later plot mean and error bar, but look at a graph of the raw data first.

1.3.5 Don't fit a model to smoothed data



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

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

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

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

This example is adapted from [Briggs \(2008\)](#).

1.3.6 Reparameterizing an equation can help

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

Reparameterizing an equation can make the uncertainty more symmetrical, making the SE and CI more helpful.

What is reparameterizing?

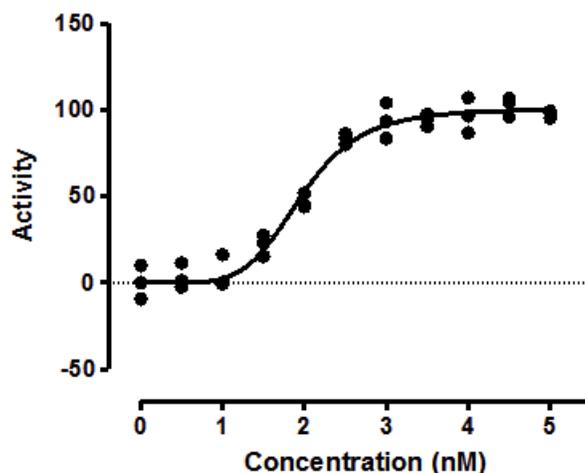
There are two forms of the model used to fit the sigmoidal enzyme kinetics data to a standard model:

$$Y = V_{\max} * X^h / (K_{\text{half}}^h + X^h)$$

$$Y = V_{\max} * X^h / (K_{\text{prime}} + X^h)$$

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

Which model is best? The two are equivalent, with $K_{\text{prime}} = K_{\text{half}}^h$, so they will generate exactly the same curve.

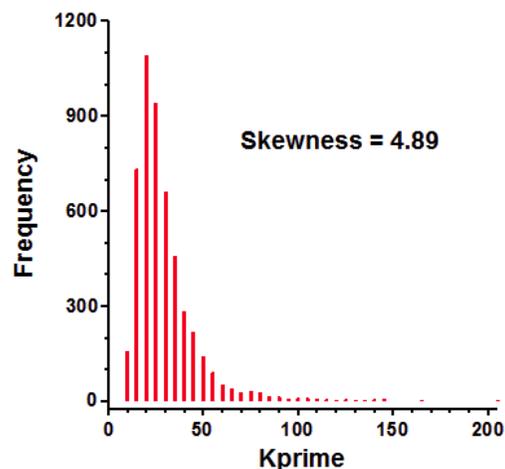
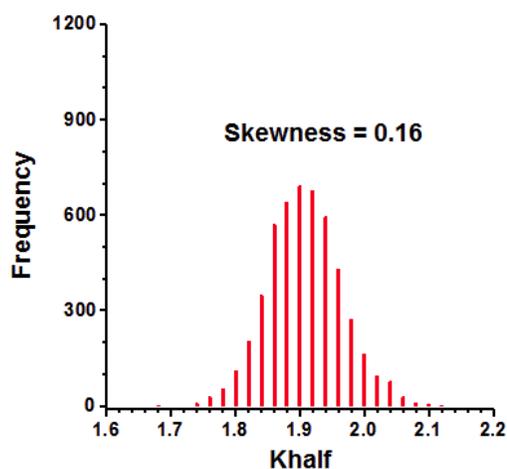


Concentration	Activity		
	Y1	Y2	Y3
0.000	10.124	0.160	-9.304
0.500	1.298	11.597	-2.417
1.000	-0.540	0.375	16.386
1.500	23.121	15.475	27.664
2.000	52.127	44.305	45.351
2.500	80.278	86.510	84.134
3.000	93.655	83.826	104.409
3.500	97.797	94.480	90.766
4.000	87.006	96.873	107.355
4.500	96.122	104.984	106.868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

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

Distribution of parameters are not always symmetrical

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



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

Consequence of asymmetrical parameters

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

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

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

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

These simulations show the advantage of choosing the equation that fits Khalf, rather than the one that fits Kprime. Khalf has a symmetrical distribution so the confidence intervals computed from these fits can be interpreted at face value.

In contrast, K_{prime} has an asymmetrical distribution and its confidence intervals cannot be interpreted at face value.

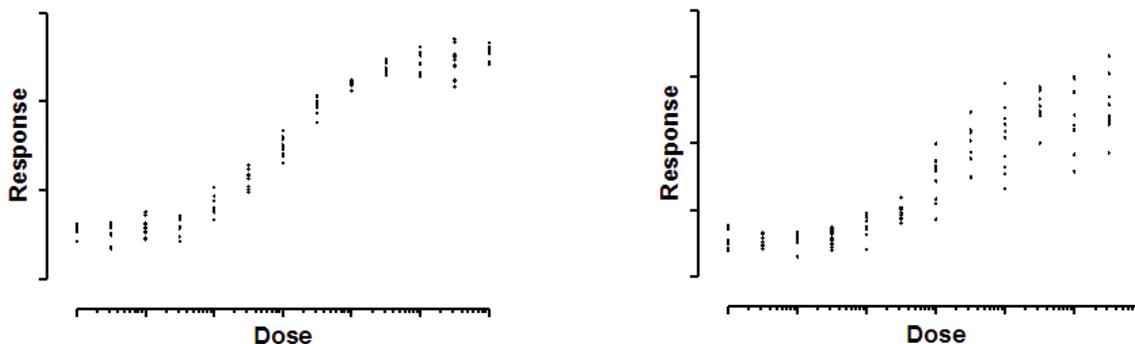
One way to choose is to match other text books and papers, so your results can easily be compared to others. Another approach is to choose the form that fits the way you think. If you prefer to think graphically, choose the K_{half} . If you think mechanistically, choose K_{prime} .

But the choice is more than a matter of convenience and convention. The rest of this article explains how the choice of model determines the accuracy of the confidence intervals.

1.4 Weighted nonlinear regression

Enter topic text here.

1.4.1 The need for unequal weighting in nonlinear regression



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

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

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

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

Prism offers six choices on the [Weights tab](#)¹⁶⁰ of nonlinear regression.

[Prism file with graphs](#)

1.4.2 Math theory of weighting

Relative weighting (weighting by $1/Y^2$)

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

If the SD of residuals is related to the value of Y, then the points with larger Y values will tend to be further from the curve (have larger residuals). These

points will therefore contribute more to the sum-of-squares, and thus dominate the calculations.

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

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

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

Poisson weighting

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

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

General weighting

The first three equations below shows how absolute, Poisson and relative weighting are related. Note that taking anything to the zero power results in

1.0, so the denominator of equation on the left always equals 1.0.

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

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

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

1. Collect data with lots (over a dozen; maybe several dozen) replicates at many points along the curve.
2. Plot the data the usual way to make sure the data seem correct.
3. Create a second graph that ignores the X values (time or concentration...). Instead, in this new graph, X is the logarithm of mean of the replicate Y values for each point on the first graph, and Y on this new graph is the logarithm of variance (square of the standard deviation) among the replicate values. You can use either natural logarithms or logarithms base 10. It doesn't matter, so long as both logarithms uses the same base.
4. Fit a straight line to this graph using linear regression. Since the assumption of a Gaussian variation around this line is dubious, use nonlinear regression and choose a robust fit.
5. The slope of this regression line is K. If K is close to 0.0, then the SD does not vary with Y so no weighting is needed. If K is close to 2.0, then the standard deviation is proportional to Y, and relative weighting is appropriate. If K is close to 1.0, choose Poisson weighting. If K has some other value, use general weighting and enter this value of K as a constant.

Note that if Y_{curve} is negative, Prism actually takes the the absolute value of Y_{curve} to the K power.

Rarely used weighting choices in Prism

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

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

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

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

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

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

Reference

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

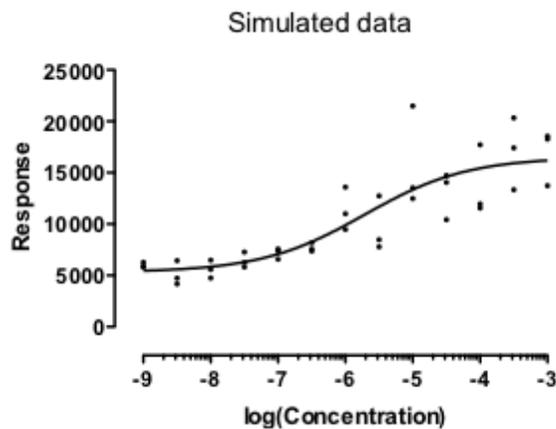
[Word file with equations](#)

1.4.3 Don't use weighted regression with normalized data

Before fitting a log(dose) vs. response curve, it is common to [first normalize the data](#)^[243]. This page explains why one should not use weighted regression in this situation.

Weighted fits to nonnormalized data -- works great

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



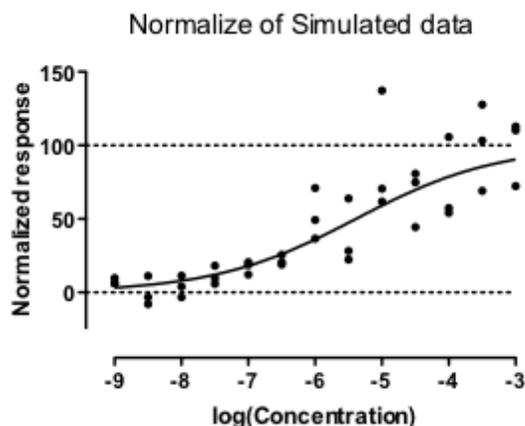
These data fit nicely using relative weighting. This minimizes the sum of the squares of the relative distance between the points and the curve. In other words it minimizes:

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

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

Weighted fits to normalized data -- ugh

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



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

What's going on?

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

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

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

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

Bottom line

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

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

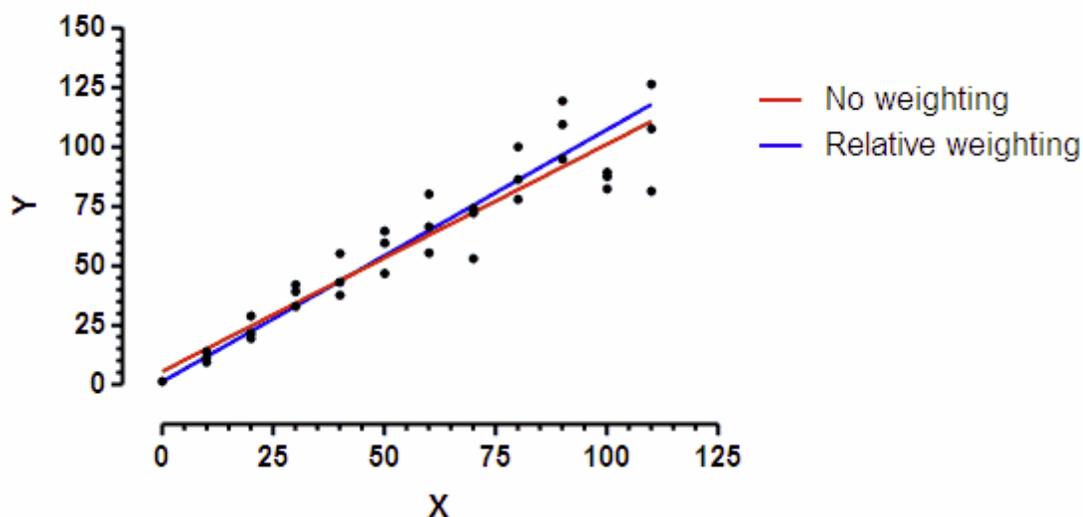
[Details and link to Prism file](#)

1.4.4 What are the consequences of choosing the wrong weighting method

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

Simulations

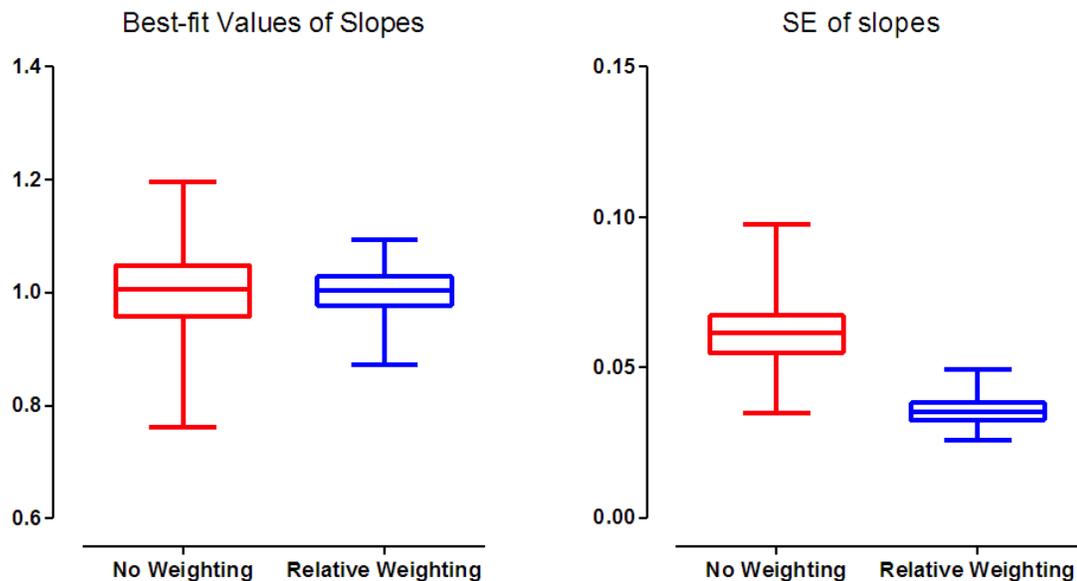
I picked a very simple model -- a straight line. I simulated the data so the SD of scatter at any point along the line is proportional to the Y value of that point. The graph below shows a simulated data set. You can clearly see that the scatter among replicates increases as the line goes up.



The line was fit to the data by "nonlinear" regression. Prism does not offer differential weighting as part of its linear regression analysis, but "nonlinear" regression can fit a straight line [with many options not available in the linear regression analysis](#)^[82].

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

I simulated 5000 such data sets using the Monte Carlo analysis of Prism 6. Each of the 5000 simulated data sets was fit with both unweighted and weighted (relative weighting) regression. I recorded both the best-fit value of the slope and its standard error (SE) for both analyses of each of the 5000 data sets.



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

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

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

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

Summary

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

- The best-fit value of the slope was less precise.
- The SE of the slope was larger, and thus the confidence intervals were wider.
- Even though the 95% confidence intervals were wider, they weren't wide enough. The "95%" confidence interval only included the true value in fewer than 95% of the simulations.

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

[Download the Prism file for this example.](#)

1.5 The many uses of global nonlinear regression

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

- [What is global nonlinear regression?](#)^[41]
- [Using global regression to fit incomplete datasets](#)^[42]
- [Fitting models where the parameters are defined by multiple data sets](#)^[43]

- [Advice: Don't use global regression if datasets use different units](#)^[45]

1.5.1 What is global nonlinear regression?

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

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

Prism makes it very easy to perform global nonlinear regression. Enter your data on one data table, click analyze, choose nonlinear regression and choose a model. On the [Constrain tab](#)^[158] of the Nonlinear regression dialog, choose which parameter(s) to share among data sets.

1.5.2 The uses of global nonlinear regression

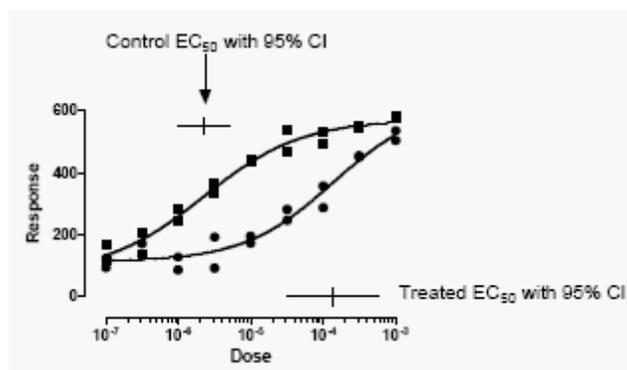
Global nonlinear regression is useful in (at least) three situations:

- Test whether a parameter value differs significantly between data sets. Compare the sum-of-squares (to assess goodness-of-fit) when the parameter is shared, with the sum of the sum-of-squares when the parameter is fit individually to each dataset. In Prism, set up this kind of comparison in the [Compare tab](#)^[158].
- Fit families of data where each dataset is incomplete, but the entire family of datasets defines the parameters. For example, one data set may do a great job of defining the bottom plateau of a dose-response curve, while another data set defines the top. Fit the two data sets separately, and the results may be ambiguous (very wide confidence intervals). Globally fit both curves, and the results might be very tight. [See an example](#)^[42].
- 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](#)^[43]. Another example is [fitting enzyme inhibition data](#)^[345].

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.

1.5.3 Using global regression to fit incomplete datasets

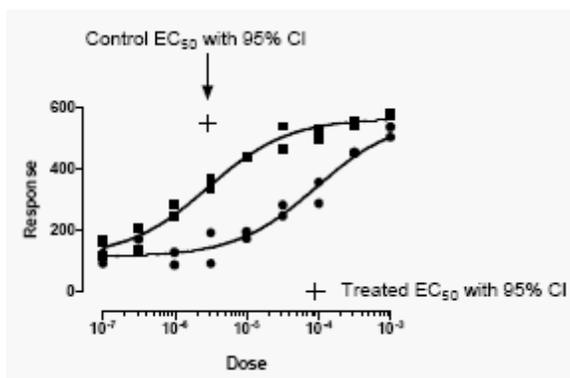
The graph below shows two dose-response curves. The goal of the experiment is to determine the two EC_{50} values. The EC_{50} is the concentration (dose) that gives a response half-way between the minimum and maximum responses. Each curve in the graph below was fit individually to one of the data sets.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

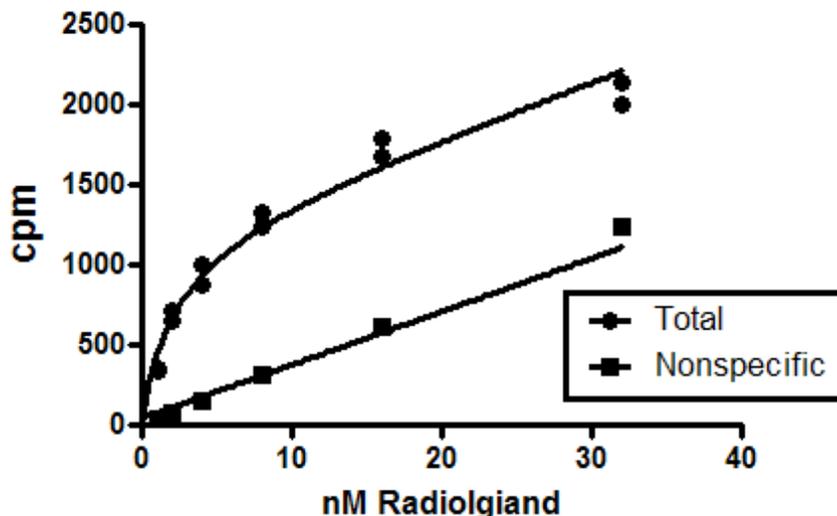
1.5.4 Fitting models where the parameters are defined by multiple data sets

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 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.

1.5.5 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.

1.6 Comparing fits of nonlinear models

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

- Comparing models can answer [four distinct](#)

[questions](#)^[46]:

- ✓ For each data set, which of two equations (models) fits best?
- ✓ Do the best-fit values of selected unshared parameters differ between data sets?
- ✓ For each dataset, does the best-fit value of a parameter differ from a theoretical value?
- ✓ Does one curve adequately fit all the data
- Prism lets you choose from [two approaches to comparing models](#)^[48]: the extra sum-of-squares F tests and the AICc approach based on information theory.
- [How the F test works to compare models](#)^[50]
- [How the AICc computations work](#)^[51]
- The idea of comparing models extends way beyond nonlinear regression. In fact, much of statistics can be viewed as comparing models

1.6.1 Questions that can be answered by comparing models

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

On the [Compare tab](#)^[157] of the Nonlinear regression dialog, Prism offers four choices:

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

Compare the fit of two models, taking into account differences in the number of

parameters to be fit. Most often, you will want to compare two related equations. Comparing the fits of two unrelated equations is rarely helpful.

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

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

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

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

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

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

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

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

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

Does one curve adequately fit all data sets?

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

This choice is new in Prism 6 and is identical to choosing "Do the best-fit values of selected unshared parameters differ between data sets?" and then selecting all the parameters.

1.6.2 Approaches to comparing models

Approach to comparing models

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

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

Two statistical approaches to comparing models

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

Extra sum-of-squares F test

The [Extra sum-of-squares F test](#)^[50] 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](#)^[51].

Which approach to choose?

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

If the two models are nested, you may use either the F test or 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.

How do these methods work to compare data sets?

The [Compare tab](#)^[157] of Prism lets you ask "Do the best-fit values of selected unshared parameters differ between data sets?" or "Does one curve adequately fit all data sets?". Applying the F test or Akaike's method to answering these questions is straightforward. Prism compares the sum-of-squares of two fits.

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

These two fits are nested (the second is a simpler case of the first, with fewer parameters to fit) so the sums-of-squares (actually the sum of sum of squares for the first fits) can be compared using either the [F test](#)^[50] or [Akaike's method](#)^[51].

[Interpreting comparison of models](#)^[212]

1.6.3 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 model (model 1).

1.6.4 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.

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

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.

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

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

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

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

From the difference in AICc values, Prism calculates and reports the probability that each model is correct, with the probabilities summing to 100%. If one model is much more likely to be correct than the other (say, 1% vs. 99%), you will want to choose it. If the difference in likelihood is not very big (say, 40% vs. 60%), you will know that either model might be correct, so will want to collect more data.

Note that this method simply compares the fits of the two models you chose.

It is possible that a third model, one you didn't choose, fits far better than either model you chose.

1.7 Outlier elimination and robust nonlinear regression

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

- [What are the advantages of removing outliers. But isn't it cheating?](#)^[53]
- Outlier removal has to be used carefully. It must be [avoided](#)^[54] in many situations.
- Once you've identified an outlier, beware of removing it from graphs and analyses. [An outlier is not always a 'bad' point.](#)^[56]
- Prism uses an outlier removal method created by GraphPad Software --the ROUT method. [Learn how it works.](#)^[57]
- When identifying outliers, the step is to fit a curve using a robust method designed so outliers won't affect the curve much. [Learn about robust nonlinear regression.](#)^[59]

1.7.1 When to use automatic outlier removal

The problem with outliers

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

Is it 'cheating' to remove outliers?

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

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

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

Which possibility is more likely?

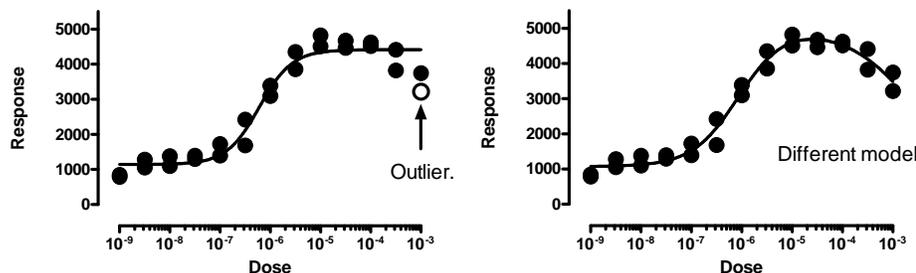
It depends on your experimental system.

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

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

1.7.2 When to avoid automatic outlier removal

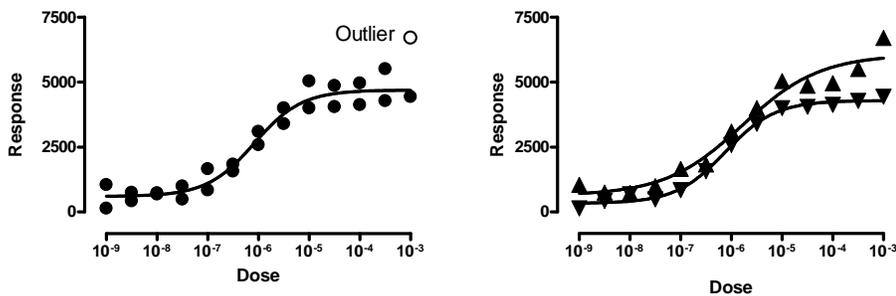
Outlier elimination is misleading when you are fitting the wrong model



The left panel above shows the data fit to a [dose response curve](#)^[256]. 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](#)^[271], 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).

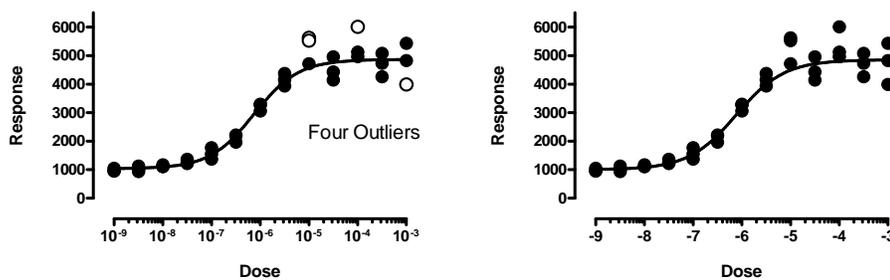
This example points out that outlier elimination is only appropriate when you are sure that you are fitting the correct model.

Outlier elimination is misleading when data points are not independent



The left panel above shows 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](#)^[30]. 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.

1.7.3 Outliers aren't always 'bad' points

Definition of an 'outlier'

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

Of course, there is some possibility that an outlier really comes from the same Gaussian population as the others, and just happens to be very high or low. You can [set the value of Q](#)^[160] 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.

1.7.4 The ROUT method of identifying outliers

How the ROUT method of removing outliers works

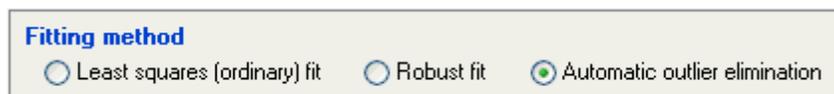
Prism offers a unique approach to identifying and removing outliers, detailed in reference 1. Because this method combines **Robust** regression and **Outlier** removal, we call it the **ROUT** method.

The ROUT method of regression follows these steps.

1. Our [robust nonlinear regression](#)^[59] 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:



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](#) 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](#).

1.7.5 Robust nonlinear regression

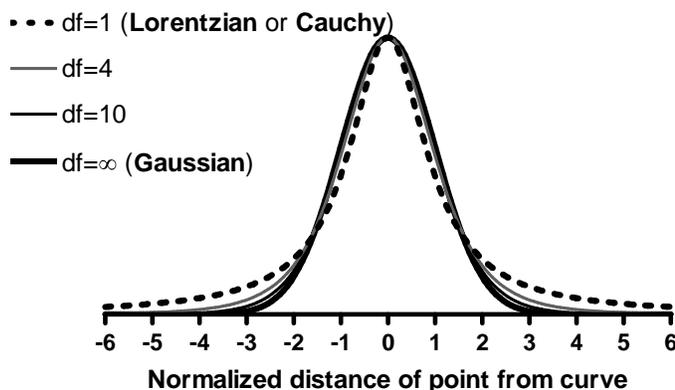
The need for robust regression

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

Experimental mistakes can lead to erroneous values whose values are way too high or too low – outliers. Even a single outlier can dominate the sum-of-the-squares calculation, and lead to misleading results. One way to cope with this problem is to perform a **robust fit** using a method that is not very sensitive to violations of the Gaussian assumption. Another approach is to use [automatic outlier elimination](#)^[53] 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](#)^[62] 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](#)^[53]. 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, Vetterling 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](#).

1.8 How nonlinear regression works

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

- [Why minimize the sum-of-squares?](#)^[61]

- [How nonlinear regression works](#)^[62]
- [The need for unequal weighting in nonlinear regression](#)^[30]
- [How standard errors and confidence intervals are computed](#)^[66]
- [How confidence and prediction bands are computed](#)^[67]
- [Replicates](#)^[68]
- [How dependency is calculated](#)^[70]

1.8.1 Why minimize the sum-of-squares?

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

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

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

1.8.2 How nonlinear regression works

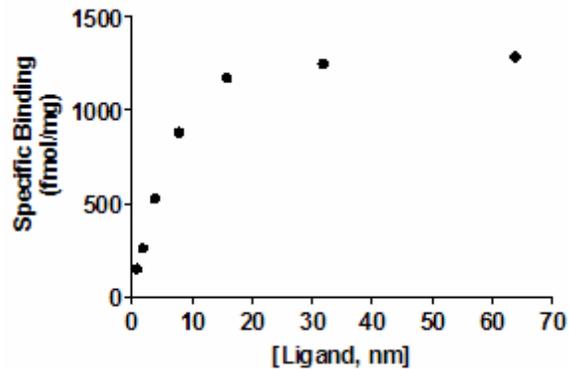
The basic idea of nonlinear regression

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

1. Start with initial estimated values for each parameter in the equation.
2. Generate the curve defined by the initial values. Calculate the sum-of-squares -- the sum of the squares of the vertical distances of the points from the curve. (Or compute the weighted sum-of-squares if you are including weighting factors.)
3. Adjust the parameters to make the curve come closer to the data points -- to reduce the sum-of-squares. There are several algorithms for adjusting the parameters, as explained below.
4. Adjust the parameters again so that the curve comes even closer to the points. Repeat.
5. Stop the calculations when the adjustments make virtually no difference in the sum-of-squares.
6. Report the best-fit results. The precise values you obtain will depend in part on the initial values chosen in step 1 and the stopping criteria of step 5. This means that repeat analyses of the same data will not always give exactly the same results.

The Marquardt method

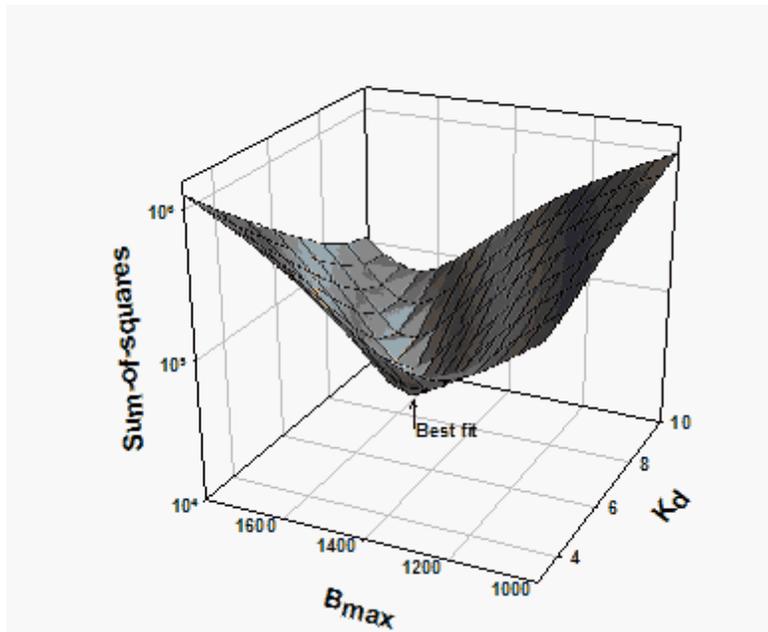
Step 3 is the only difficult one. Prism (and most other nonlinear regression programs) uses the method of Marquardt and Levenberg, which blends two other methods, the method of linear descent and the method of Gauss-Newton. The best way to understand these methods is to follow an example. Here are some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



You want to fit a binding curve to determine B_{\max} and K_d using the equation

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

How can you find the values of B_{\max} and K_d that fit the data best? You can generate an infinite number of curves by varying B_{\max} and K_d . 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 (B_{\max} and K_d 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 B_{max} and K_d 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 B_{max} and K_d 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.

1.8.3 Nonlinear regression with unequal weights

How Prism implements weighting

Prism 5 and 6 weight by the Y value of the curve. Prism 4 and earlier weighted by the Y value of the data. The distinction is subtle and rarely matters much, but our simulations show that the results are sometimes more accurate when weights are based on the value of the curve rather than the data.

The situation is a bit tricky. The goal is to adjust the values of the parameters to minimize the weighted sum-of-squares. But the values of the weights depend on the values of those parameters. Here is how Prism 6 resolves this issue:

1. Perform the first iteration without any weights. If the initial parameter values are not very accurate, weighted regression can cause problems on the first iteration. This step was introduced in Prism 5.02 and 5.0b. Without it, some fits with bad initial values simply won't converge.
2. Compute Y_{curve} at all values of X based on the curve generated by this first iteration of nonlinear regression, and use those values to compute weights for each points.
3. Keeping those weights constant, use nonlinear regression to minimize the weighted sum-of-squares.
4. Recompute Y_{curve} based on the results of that curve fit and generate a new set of weights.
5. Keeping those new weights constant, use nonlinear regression again to minimize the weighted sum-of-squares.
6. Repeat steps 4 and 5 until the weighted sum-of-squares no longer changes.

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.](#)

1.8.4 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.

How Prism calculates the standard error of parameters

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

$$SE(P_i) = \text{sqrt}[(SS/DF) * \text{Cov}(i,i)]$$

where:

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

How Prism computes the confidence intervals of parameters

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.

The 95% confidence intervals are computed by this equation:

$$\text{From } [\text{BestFit}(P_i) - t(95\%,DF) * SE(P_i)] \text{ TO } [\text{BestFit}(P_i) + t(95\%,DF) * SE(P_i)]$$

where $\text{BestFit}(P_i)$ is the best fit value for the i -th parameter, and t is the value from the t distribution for 95% confidence and DF degrees of freedom.

How Prism computes the confidence interval for the difference or ratio of two parameters

[See this document.](#)

1.8.5 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](#)^[197]. 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:

$$= \sqrt{c} * \sqrt{SS/DF} * \text{CriticalT}(\text{Confidence}\%, DF)$$

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

$$= \sqrt{c+1} * \sqrt{SS/DF} * \text{CriticalT}(\text{Confidence}\%, DF)$$

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

1.8.6 Replicates

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

- Consider each replicate as an individual point.
- Fit the means of each set of replicates.

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

Independent replicates

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

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

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

Replicates that are not independent

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

- You performed a binding experiment with a single tube at each concentration, but measured the radioactivity in each tube three times. Those three values are not independent. Any experimental error while conducting the experiment would affect all the replicates.
- You performed a dose-response experiment, using a different animal at each dose with triplicate measurements. The three measurements are not independent. If one animal happens to respond more than the others, that will affect all the replicates. The replicates are not independent.

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

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



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).

1.8.7 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.

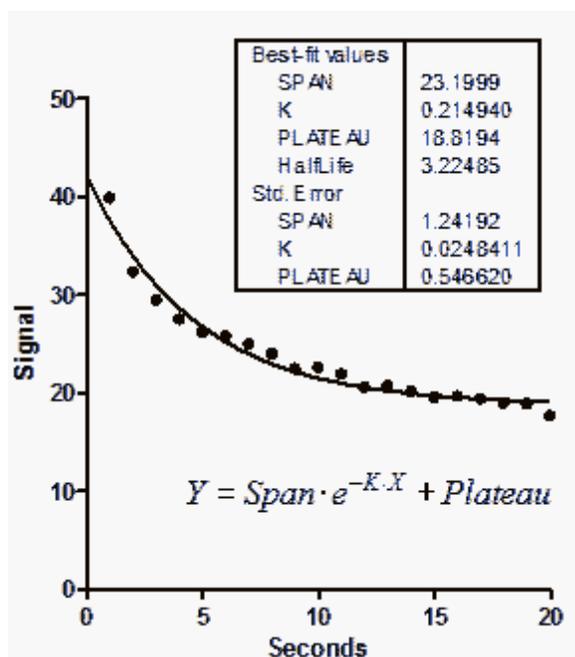
Interpreting dependency

You can [interpret dependency](#)¹⁹⁷ 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



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

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

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

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

Before we can compare the two SE values, we have to correct for a minor problem. When computing the SE, the program divides by the square root of the number of degrees of freedom (df). For each fit, df equals the number of

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

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

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

The origin of the idea of dependency

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

1.8.8 How confidence and prediction bands are computed

The difference between confidence and prediction bands

Note the difference between confidence and prediction bands:

- The 95% confidence bands enclose the area that you can be 95% sure contains the true curve. If you have many data points, the confidence bands will be near the line or curve, and most of your data will lie outside the confidence bands.
- The 95% prediction bands enclose the area that you expect to enclose 95% of future data points. They are wider than confidence bands -- much wider with large data sets.

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

How confidence and prediction bands are computed

First, let's 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 predicted 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 (inversed Hessian from last iteration). It is a square matrix with the number of rows and columns equal to the number of parameters. Each item in the matrix is the normalized covariance between two parameters (which always has a value between -1 and +1).

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:

$$= \sqrt{c} * \sqrt{SS/DF} * \text{CriticalT}(\text{Confidence}\%, DF)$$

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

$$= \sqrt{c+1} * \sqrt{SS/DF} * \text{CriticalT}(\text{Confidence}\%, DF)$$

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 fit parameters). CriticalT is a constant from the t distribution based on the level 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.

[Read a more mathematical explanation.](#)

Computing confidence and prediction bands [is easier for linear regression](#).

1.8.9 Who developed nonlinear regression?

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

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

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

2 CURVE FITTING WITH PRISM 6

Prism offers several analyses that can fit lines and curves.

- [Linear regression](#)^[76]
- [Deming linear regression](#)^[100]
- [Nonlinear regression](#)^[162] (which also does polynomial regression)
- [Spline and Lowess](#)^[447]
- [Interpolating from a standard curve](#)^[104] (linear or nonlinear)

2.1 What's new in Prism 6 (regression)?

- **Easier interpolating.** Many people use nonlinear regression for the purpose of interpolating values from standard curves. With that purpose, most of the options provided in the nonlinear regression dialog are not needed or helpful. Prism 6 has a [new analysis just for interpolating curves](#). It requires very few options, so is super easy to use.
- **Implicit and differential equations.** It is no fun to struggle with algebra and calculus to express a model as an equation that defines Y as a function of X and parameters. It can even be impossible. Instead express your model as a

[differential equation](#) (enter the equation that defines dY/dX) as or an [implicit equation](#) (an equation where Y appears on both sides of the equals sign).

- **Equation syntax.** Prism 5 let's you define different models for different data sets by designating certain lines in the equation to only apply to a certain data set. For example, a line in the equation preceded with applies only to data set B. [Prism 6 extends this syntax](#) to allow more complex equations. Precede a line with <A:D> for it to apply only to columns A-D, or with <A:K,2> to make it apply to every second data set between A and K (A, C, E, G, I, K).
- **Hougaard's skewness.** When you write a user-defined equation to fit to data with nonlinear regression, you can choose to express the parameters in multiple ways. For example choose between a rate constant or a half life. [Hougaard's measure of skewness](#) can help you choose the best parameterization.
- **New weighting choice.** When fitting a model to data with nonlinear regression, it is important to weight the data appropriately. This choice is often ignored, but it can affect the regression results. Prism always offered several choices for weighting. Prism 6 offers a new weighting choice, weighting by $1/Y^K$. where K is a constant you enter on the weighting tab of the nonlinear regression dialog.
- **Adjusted R².** Prism reports the adjusted R² which takes into account the number of degrees of freedom, and so can be compared between models with different numbers of parameters.
- **Convert from linear to nonlinear regression.** Linear regression can be viewed a special case of nonlinear regression. Prism's nonlinear regression analysis can be used to fit a straight line, and there are [more than a dozen reasons why you might want to do so](#). But few people think of using the nonlinear regression analysis to fit a straight line. Prism 6 makes it easier to switch – simply click the “More choices” button on the linear regression dialog.
- **Copy the linear regression equation.** Prism always reported the best-fit values for the slope and intercept, but not in the form of an equation. Prism 6 now reports the equation for the line, ready to copy and paste onto a graph or into a manuscript.

2.2 Linear regression with Prism

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

[How to: Linear regression](#)  76

[Finding the best-fit slope and intercept](#)^[76]

[Interpolating from a linear standard curve](#)^[79]

[Advice: When to fit a line with nonlinear regression](#)^[82]

[Results of linear regression](#)^[84]

[Slope and intercept](#)^[84]

[r², a measure of goodness-of-fit of linear regression](#)^[86]

[Is the slope significantly different than zero?](#)^[88]

[Comparing slopes and intercepts](#)^[89]

[Runs test following linear regression](#)^[90]

[Analysis checklist: Linear regression](#)^[91]

[Graphing tips: Linear regression](#)^[92]

[Questions and answers](#)^[95]

[Deming regression](#)^[100]

[Key concepts: Deming regression](#)^[100]

[How to: Deming regression](#)^[100]

[Q&A: Deming Regression](#)^[102]

[Analysis checklist: Deming regression](#)^[103]

2.2.1 How to: Linear regression

2.2.1.1 Finding the best-fit slope and intercept

1. Create a data table

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

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

If you are entering your own data, choose the subcolumn format. Choose replicate values if you have replicates to enter. Prism can plot error bars automatically. You can also choose to enter data where the mean and SD (or SEM) have already been calculated. In this case, if you want to take into account variations in the SD from point to point, [use nonlinear regression to fit the line](#)^[82].

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](#)^[90]

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](#)^[89].

Confidence and prediction bands

Learn about [confidence and prediction bands](#)^[92].

2.2.1.2 Interpolating from a linear standard curve

1. Create a data table

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

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

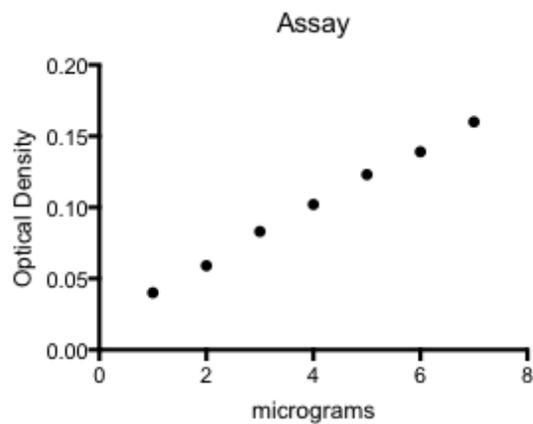
If you are entering your own data, choose the subcolumn format. Choose replicate values if you have replicates to enter. Prism can plot error bars automatically. You can also choose to enter data where the mean and SD (or SEM) have already been calculated. In this case, if you want to take into account variations in the SD from point to point, [use nonlinear regression to fit the line](#)^[82].

2. Enter data

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

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

3. View the graph



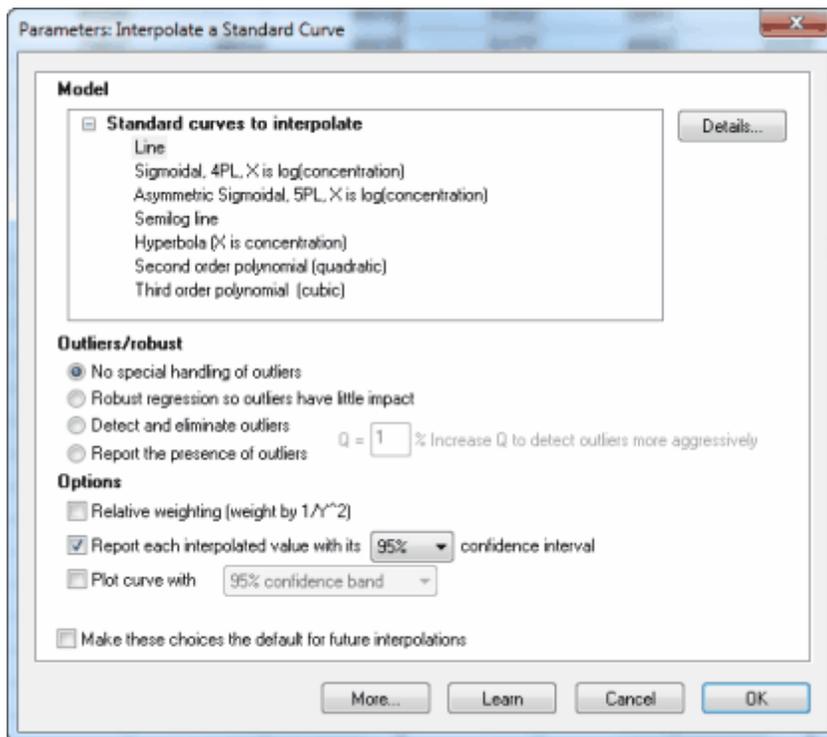
4. Analysis choices

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

Alternatively, you can click the "Interpolate a standard curve" button right on

top of the Analyze button.

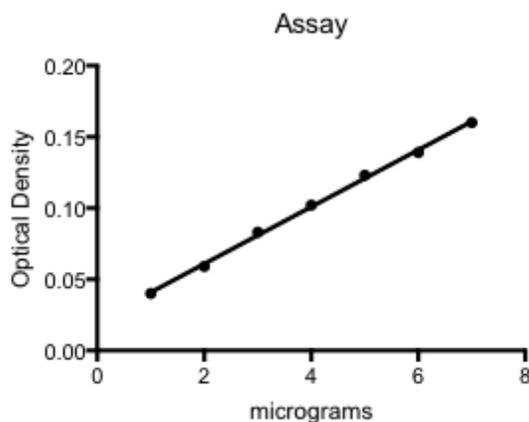




Choose Line in the list of Standard curves to interpolate. Unless you have a good reason, you can leave the other choices on that dialog set to their default values.

5. Inspect the results

View the graph with the superimposed line.



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

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

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

2.2.1.3 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.

How to convert a linear regression analysis to nonlinear regression

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

Nonlinear regression offers more options

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

- Fit to both a linear and nonlinear model, and [compare the two models](#)¹⁵⁷.
- Apply differential [weighting](#)³⁰.
- Automatically [exclude outliers](#)⁵⁹.
- Use a [robust](#)⁵⁹ fitting method.

- Perform a [normality test](#)^[185] on the residuals.
- Inspect the [correlation matrix or dependencies](#)^[197].
- Compare the scatter of points from the line with the scatter among replicates with a [replicates test](#)^[194].
- Report the best-fit values with 90% confidence limits (or any others). Prism's linear regression analysis only reports 95% CI. Nonlinear regression lets you choose the confidence level you want.
- Report the results of [interpolation](#)^[105] from the line/curve along with 95% confidence intervals of the predicted values. Prism's linear regression analysis does not include those confidence intervals.
- With linear regression, the SE of the slope is always reported with the slope as a plus minus value. With nonlinear regression, the SE values are a separate block of results that can be copy and pasted elsewhere.
- Use [global nonlinear regression](#)^[41] to fit one line to several data sets. Or share the intercept or slope among several data sets, while fitting the other parameter individually to each data set.
- Run a [Monte Carlo analysis](#).
- When you enter data with multiple replicates at each X value, Prism's nonlinear regression can perform the [replicates test](#)^[194] to ask whether the data deviate systematically from the straight line model. Prism does not offer the replicates test with linear regression.
- Test whether the slope (or intercept) significantly differs from some proposed value. For example, test whether the slope differs from a hypothetical value of 1.0, or whether an intercept differs significantly from 0.0.

Situations where Prism can give different results with linear vs. nonlinear regression

There are two situations where linear regression will give different results than fitting a straight line with nonlinear regression:

- If you enter the Y values 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](#)^{160]}) of fitting just the mean, or of accounting for scatter and sample size. With the latter choice, which is the default, 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.

- If you enter replicate Y values and choose the runs test. With linear regression, Prism averages the replicates and computes the runs test on those mean Y values. With nonlinear regression, Prism won't compute the runs test when you enter replicate Y values. Instead,

Lines that must be fit with nonlinear regression

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

- If your Y axis uses a logarithmic or probability scale, then a straight line on the graph is created by a nonlinear model. In this case, although the line on the graph is straight, the model is not actually linear. You need to [fit the 'line'](#)^{370]} 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](#)^{374]}.

2.2.2 Results of linear regression

2.2.2.1 Slope and intercept

How Prism reports the slope and intercept

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

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

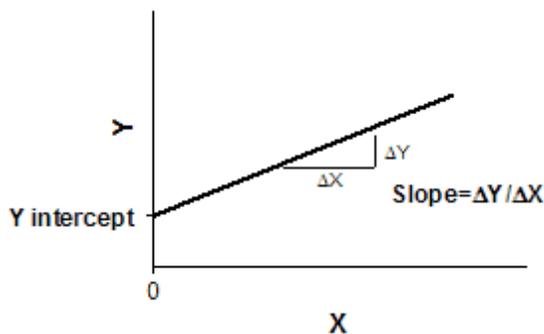
At the bottom of the results page, the slope and intercept are reported again in the form of the equation that defines the best-fit line. You can copy this equation and paste onto a graph, or into a manuscript.

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

Interpreting the slope and intercept

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

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



Interpreting the standard errors and confidence intervals

The standard error values of the slope and intercept can be hard to interpret,

but their main purpose is to compute the 95% confidence intervals.

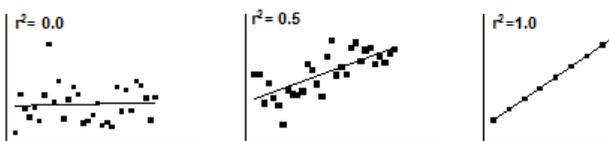
If you accept the [assumptions of linear regression](#)^[91], there is a 95% chance that the 95% confidence interval of the slope contains the true value of the slope, and that the 95% confidence interval for the intercept contains the true value of the intercept. The width of the confidence intervals is determined by the number of data points, their distances from the line, and the spacing of the X values.

Prism can also plot either the [confidence or prediction bands](#)^[171].

2.2.2.2 r^2 , a measure of goodness-of-fit of linear regression

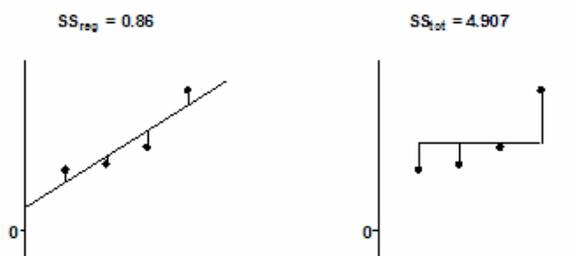
The meaning of r^2

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 .



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

The left panel shows the best-fit linear regression line. This line 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 (SS_{reg}) 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 (SS_{tot}) 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 SS_{reg} is much smaller than SS_{tot}, 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 (SS_{tot}).

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 $Y = \text{slope} * X$. 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.

2.2.2.3 Is the slope significantly different than zero?

Interpreting the P value when Prism fits both slope and intercept

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

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

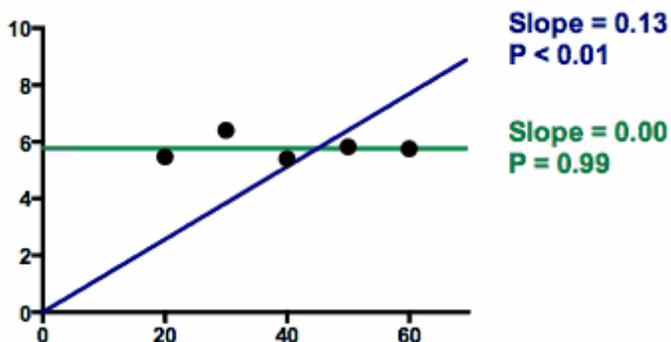
Equivalently:

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

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

Interpreting the P value when Prism fits only the slope

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



The data points almost form a horizontal line.

When fit with linear regression the usual way (fit both slope and intercept;

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

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

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

[Download the Prism file.](#)

2.2.2.4 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](#)^[41] 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.

2.2.2.5 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 N_a points above the curve and N_b points below the curve, the number of runs you expect to see equals $[(2N_aN_b)/(N_a+N_b)]+1$. 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.

2.2.2.6 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 of the total course grade, linear regression is not valid for these data.

2.2.2.7 Graphing tips: Linear regression

Graphing the regression line

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

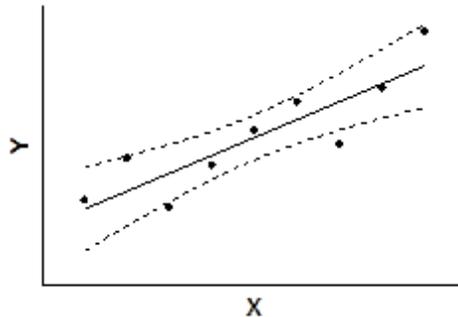
If you need to create additional graphs, or change which line is plotted on which graph, keep in mind that the line generated by linear regression is seen by Prism as a data set. You can add lines to a graph or remove lines from a graph on the 'Data sets on graph' tab of the Format Graph dialog.

Confidence and prediction bands

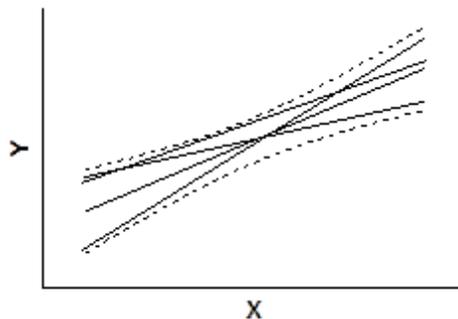
If you check the option box, Prism will calculate and graph either the 95% confidence band or 95% prediction band of the regression line.

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).



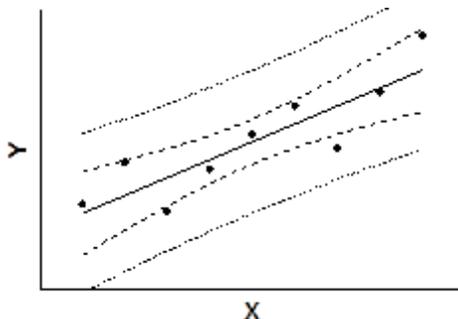
Given the assumptions of linear regression, you can be 95% confident that the two curved confidence bands enclose the true best-fit linear regression line, leaving a 5% chance that the true line is outside those boundaries.

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.



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](#)^[179] of residuals from nonlinear regression.

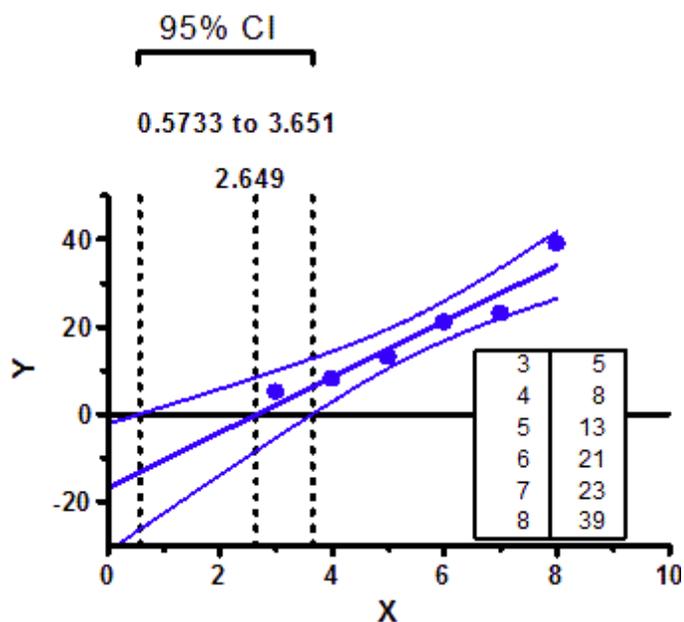
2.2.2.8 Questions and answers

- ❑ What is the difference between correlation and linear regression?

[The answer is in the Correlation section.](#)^[16]

- ❑ Why is the 95% confidence interval of the X intercept asymmetrical?

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



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

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

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

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

$$Y = \text{slope} * (X - X_{\text{intercept}})$$

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

- ❑ How to test whether the slope from linear regression differs from 1.0 or some other value?

Prism tests whether the best-fit slope from linear regression line differs significantly from zero. But you need to use extra steps to test whether the slope differs from some other value.

Since Prism's linear regression analysis can't answer the question, use nonlinear regression instead. Nonlinear regression can fit a straight line and [offers many advantages](#)^[82].

In the nonlinear regression parameters dialog, go to the Compare tab, check the option to ask compare the best-fit value of a parameter to a theoretical value, and then choose Slope and enter the theoretical value (1.0).

- ❑ Can Prism do weighted linear regression?

Prism's linear regression analysis cannot handle differential weighting. However, Prism's nonlinear regression analysis can do weighted regression, and you can [fit a straight line using "nonlinear" regression](#)^[82].

- ❑ How can Prism fit linear regression with the slope constrained to equal 1.0 (or some other value)?

Prism's linear regression analysis cannot constrain the slope to a particular value. However, Prism's nonlinear regression analysis can constrain any parameter to any value, and you can [fit a straight line using "nonlinear" regression](#)^[82].

- ❑ How to fit one linear regression line to multiple data sets?

Prism usually fits one line through each data set. But you can get it to fit one line through all the data. Use the [nonlinear regression analysis to fit the line](#)^[82]. Then go to the Constrain tab, and share both the slope and the intercept. "Sharing" means that Prism fits one best-fit value for all the data sets, rather than one for each data set. Since you share all the parameters, only one line is fit to all the data.

- ❑ Why doesn't Prism report R² for linear regression when I force the line through the origin (or some other point)?

The problem is that when you constrain a line to go through a point, there would be two ways to compute R²:

- Compare the fit of the best-fit line with the fit of a horizontal line at the mean Y value. But that null hypothesis (horizontal line through the Ymean)

doesn't obey the constraint that it go through the origin.

- Compare the best-fit line with a horizontal line at $Y=0$. This obeys the constraint, but often fits the data really badly, pushing up the R^2 value.

Since R^2 is ambiguous when you constrain linear regression, Prism simply does not report it when you force the line through the origin (or other point).

If you fit the line with nonlinear regression, Prism will report R^2 using the first definition above.

- ▣ Fitting the crossing point of two intersecting linear regression lines .

Enter your data onto one data table, with Y values in columns A and B next to corresponding X values in the X column. If the X values don't match for the two data sets, just leave some Y values blank.

Use nonlinear regression, and enter this user-defined equation.

$$Y = Y_{\text{cross}} + (X - X_{\text{cross}}) * \text{Slope}$$

This equation defines the Y coordinate of any point along either line to equal the Y coordinate of the crossing point plus the slope times the X distance from the crossing point to that point.

Set these rules for initial values:

- Ycross: $1 * Y_{\text{mid}}$ (the average of the highest and lowest Y values)
- Xcross: $1 * X_{\text{mid}}$
- Slope: $1 * (Y_{\text{max}} - Y_{\text{min}}) / (X_{\text{max}} - X_{\text{min}})$.

Define Xcross and Ycross to both be shared parameters, so Prism will find one global best-fit value for those parameters that applies to both data sets. Do not share the slope parameter, as you want Prism to fit separate slopes for each data set. Getting these sharing settings correct is essential.

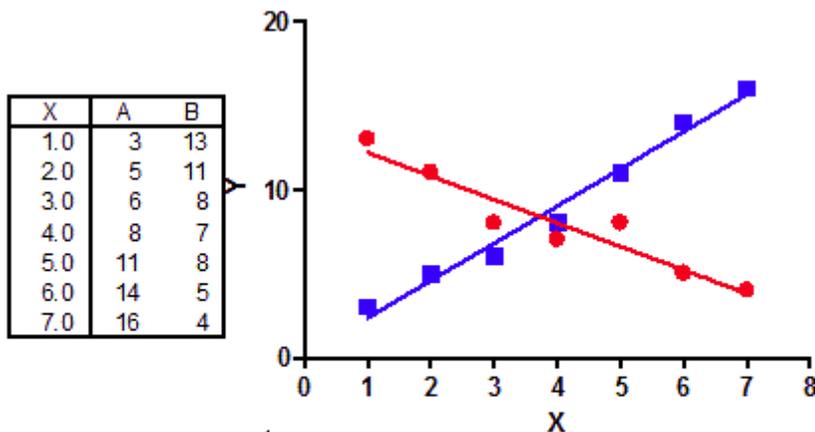
Prism will fit four parameters:

- Slope for the first data set
- Slope for the second data set

- Xcross (shared for both data sets)
- Ycross (shared for both data sets).

That makes sense. You have two lines, and it takes two parameters (usually a slope and intercept) to describe each. Here, the equations were rearranged so the program doesn't fit two separate Y intercepts, but rather fits the X and Y values of the crossing point.

Here is an example, with the key results circled. [Prism file.](#)



	A	B	Global (shared)
Best-fit values			
YCROSS	8.386	8.386	8.386
XCROSS	3.723	3.723	3.723
SLOPE	2.214	-1.393	
Std. Error			
YCROSS	0.2535	0.2535	0.2535
XCROSS	0.1371	0.1371	0.1371
SLOPE	0.09846	0.1424	
95% Confidence intervals			
YCROSS	7.821 to 8.951	7.821 to 8.951	7.821 to 8.951
XCROSS	3.417 to 4.028	3.417 to 4.028	3.417 to 4.028
SLOPE	1.995 to 2.434	-1.710 to -1.076	
Goodness of Fit			
Degrees of Freedom			10
R ²	0.9806	0.9054	0.9588
Absolute Sum of Squares	2.714	5.679	8.393
Sy.x			0.9161
Constraints			
YCROSS	YCROSS is shared	YCROSS is shared	
XCROSS	XCROSS is shared	XCROSS is shared	
Data			
Number of X values	8	8	
Number of Y replicates	1	1	
Total number of values	7	7	
Number of missing values	1	1	

2.2.3 Deming regression

2.2.3.1 Key concepts: Deming regression

Standard [linear regression](#)^[13] assumes that you know the X values perfectly, and all the uncertainty is in Y. It minimizes the sum of squares of the vertical distance of the points from the line.

If both X and Y variables are subject to error, fit linear regression using a method known as **Deming**, or Model II, regression.

If your goal is to compare two analysis methods, consider [using a Bland-Altman plot instead](#).

2.2.3.2 How to: Deming regression

1. Create a data table

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

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

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

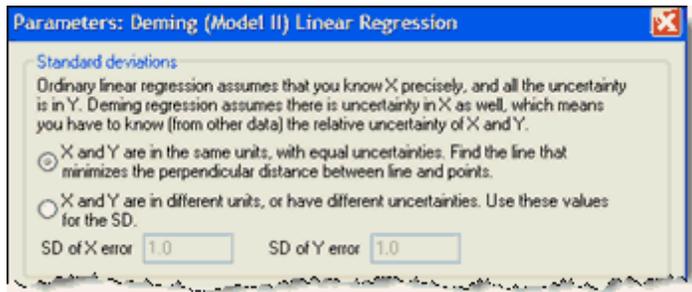
2. Enter data

If you enter Y values for several data sets (column A, B and C), Prism will report regression results for X vs. YA, for X vs. YB, and for X vs. YC.

3. Analysis choices

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

Most of the dialog is self-explanatory. The choices on top are the hardest to understand, but also are the most important as the calculations depend on knowing relative magnitude of the errors in X and Y.



X and Y have equal uncertainty

If you are comparing two analytical methods, X and Y, assessed in the same units, you will probably find it reasonable to assume that both share the same standard deviation. In this case, check the first option. There is no need to estimate that standard deviation. It is enough to declare that X and Y have equal standard deviations. In this case, Deming regression minimizes the sum of the squares of the perpendicular distances of the points from the line. This is also called *orthogonal regression*.

X and Y have different uncertainties

If you are not willing to assume that X and Y have the same amount of uncertainty, then you need to enter the SD of each. Enter the SD of X values in the same units as X values are entered, and the SD of Y values in the same units as Y values are entered. Don't enter the SD of all X (or Y) values you enter. Enter the average SD (determined separately) of repeated measurements of X and Y.

How do you know what values to enter? To assess the uncertainty (error) of a method, collect duplicate measurements from a number of samples using that method. Calculate the standard deviation of the error using the equation below(1), where each d_i is the difference between two measurements of the same sample (or subject), and N is the number of measurements you made (N equals twice the number of samples, since each sample is measured twice).

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

Compute the SD of the error for each method or variable (X and Y), enter the two SD_{error} values into the Deming regression analysis dialog, and Prism will fit the line for you. If the X variable has a much smaller SD than the Y value, the results will be almost identical to standard linear regression.

If you try to compare Prism's results with those of another program or book,

you may encounter the variable λ (lambda), which quantifies the inequality between X and Y errors.

$$= \left(\frac{SD_{X \text{ error}}}{SD_{Y \text{ error}}} \right)^2$$

Prism requires you to enter individual SD values, but uses these values only to calculate λ , which is then used in the Deming regression calculations. If you know λ , but not the individual SD values, enter the square root of λ as the SD of the X values, and enter 1.0 as the SD of the Y error. The calculations will be correct, since Prism uses those two values only to compute λ .

Reference

1. PW Strike, Statistical Methods in Laboratory Medicine, ISBN:0750613459. Equation 8.15.

2.2.3.3 Q&A: Deming Regression

Why doesn't Prism report R^2 with Deming Regression results?

When Prism performs Deming regression, it reports the slope and intercepts with confidence intervals, and reports a P value testing the null hypothesis that the slope is really zero. However, Prism does not report any measure of goodness-of-fit with Deming regression, and so does not report R^2 . The reason is that we have been unable to find any paper or text that would explain how to compute or interpret such a value. In ordinary linear or nonlinear regression, R^2 is the fraction of the variation that is accounted for by the model. But with Deming regression, this definition doesn't really make sense, and it isn't obvious to us how to extend it.

Is Deming regression the same as orthogonal linear regression?

In Prism's Deming dialog, you specify whether X and Y are in the same units with equal uncertainties (variation). If you choose this option, Deming regression minimizes the sum of the square of the perpendicular distances of the points from the line. This is also called *orthogonal linear regression*. If you specify different SD values for X and Y, then Deming regression is not the same as orthogonal linear regression.

How was the equation to compute the SD from a set of duplicate measurements derived?

The previous page showed the equation for assessing the uncertainty (error) of a method from duplicate measurements from a number of samples using that

method:

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

It is easy to derive that equation. To compute the variance, you find the average square of the difference between each value and the mean. For pairs, the mean is half the difference. So take half that distance squared, but also compute it for the other value of the pair. For both values, therefore, the contribution is d_i squared. Add that up for all the measurements and divide by the number of measurements, and you have the variance. Take the square root to get the SD.

Why N, rather than N-1? I suspect it should be N-1, but that won't matter much. It will make the SD a bit smaller than it should be, but that will be true for both the X and Y SD, so the ratio will hardly be changed.

2.2.3.4 Analysis checklist: Deming regression

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

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

✓ Are the data points independent?

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

✓ Are the X and Y values intertwined?

If the value of X is used to calculate Y (or the value of Y is used to calculate X) then linear regression calculations are invalid. One example is a Scatchard plot, where the Y value (bound/free) is calculated in part from the X value (bound). Another example would be a graph of midterm exam scores (X) vs. total course grades (Y), since Y is in part computed from X.

✓ Do you know the relative uncertainty of X and Y?

Ordinary linear regression assumes that you know the X values perfectly, and all the uncertainty is in Y. Deming regression assumes there is uncertainty in both variables. You have to specify the relative uncertainties, either by specifying that X and Y are equally uncertain or by entering the SD of each. If these values are incorrect, the

Deming results won't be useful.

2.3 Interpolating from a standard curve

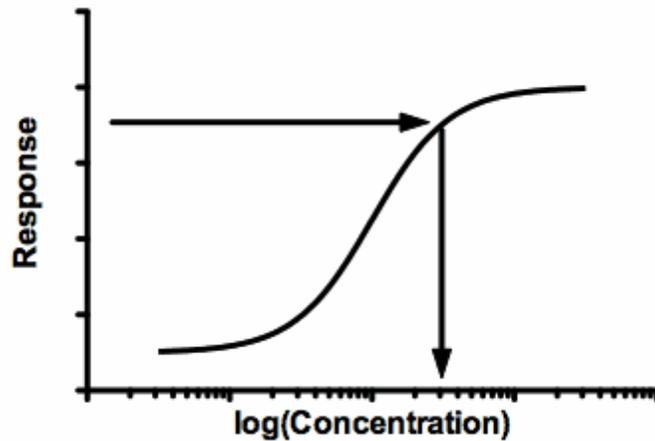
Prism makes it very easy to interpolate unknown values from a standard curve. Enter the standards with both X and Y values, fit a line or curve, and Prism will tell you which X values correspond to Y values you entered on the same data table.

- [Key concept: Interpolating](#)¹⁰⁴
- [How to interpolate](#)¹⁰⁵
- [Equations used for interpolating](#)¹¹¹
- [Where are the interpolated results?](#)¹¹³
- [When X values are logarithms](#)¹¹⁷
- [Analysis checklist: Interpolating](#)¹¹⁸
- [Q&A: Interpolating](#)¹²¹
- [Standard Addition Method](#)¹²³

2.3.1 Key concept: Interpolating

A common use of nonlinear (and linear) regression is interpolating. The basic idea is simple:

1. Enter a standard curve. Each point consists of a concentration (or logarithm of concentration) and a response. The responses are commonly radioactivity, fluorescence, or optical density.
2. Fit a line or curve through the standards. For this purpose, you don't much care about the meaning of the model, or the interpretation of the best-fit values of the parameters (EC50, etc.).
3. For unknown samples, you know the response, but not the concentration. So read the concentration from the curve:



Notes:

- This graph shows the response going up as the concentration gets larger. In many cases, the response goes down as the concentration gets larger. Interpolation works in either case.
- Sigmoid shaped dose-response curves are commonly used with many assays. If the X values analyzed by the analysis are the logarithm of concentration, then so are the interpolated values. You need to take the [antilogarithm](#) of these values to return to concentration units.
- This example shows the most common use of interpolating. Finding an X value (concentration) for a given Y value. Prism can also interpolate the other way: Finding a Y value for an entered X value.

2.3.2 How to interpolate

Prism is not designed to analyze any particular kind of assay

Prism is a very general program, used by scientists in many fields. It is easy for you to interpolate standard curves using Prism, but you must make some decisions. Unlike some specialized programs, Prism is not set up so you can blindly enter data and get results without thinking.

Entering data for interpolation

Enter the standard curve in the top rows of a table, and then enter the

unknown values right below those standards. In most cases, as shown below, you'll enter Y values and leave X blank. Prism can also interpolate the other way. In this case, enter X values and leave Y blank.

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

Entering labels for the unknowns, as shown below, is optional.

Note that it is essential that the unknowns go below the standard curve values. If you place the unknowns above the standard curve, Prism will not interpolate. Of course, the unknown Y values must be in the same units as the Y values you entered for the standard curve.

Fitting a standard curve and interpolating

Four analyses in Prism let you interpolate values from curves. The top row of buttons in the analysis block provide shortcuts to linear regression, nonlinear regression, and the simplified analysis for interpolating a standard curve.



Interpolate a standard curve

This analysis (new to Prism 6) is a simple way to interpolate using nonlinear regression.

The choices on the dialog are simple. Choose a model to fit to your data. The most [commonly used models for standard curves](#)^[111] are shown in the dialog. Next choose how to handle outliers. You can choose to use [robust regression](#)^[59], so outliers have little impact. Or you can ask Prism to identify, and perhaps eliminate, outliers using the [ROUT](#)

[method](#)⁵⁷. Finally, if the scatter among replicates is larger when the Y values are larger, consider using [relative weighting](#)¹⁶⁰.

Note that this simplified analysis to interpolate is simply a subset of the nonlinear regression analysis, with fewer choices on the dialog. If you want to select options that aren't available, click the "More.." button at the bottom of the dialog to convert the analysis to the full nonlinear regression analysis, with many more options (six tabs worth). Once you switch to nonlinear regression analysis, it is not possible to switch back (except by starting over).

Nonlinear regression

Nonlinear regression offers the interpolation option at the bottom of the first (Fit) tab of the dialog. This is a good choice if you want to use options not available on the simpler interpolation analysis.



Linear regression

[Linear regression](#)⁷⁹ offers a checkbox to interpolate the top of the dialog.

Spline

Prism can also interpolate from a standard curve in the [spline](#)⁴⁴⁷ analysis. Splines go through every point. The advantage of a spline is that you don't have to choose a model to fit to your data. The disadvantage is that the curve may wiggle too much.

2.3.3 Example: Interpolating from a sigmoidal standard curve

1. Create the data table

From the Welcome or New Table dialog, choose to create an XY data table, and select the sample data set: RIA or ELISA. Interpolate unknowns from sigmoidal curve.

2. Inspect the data

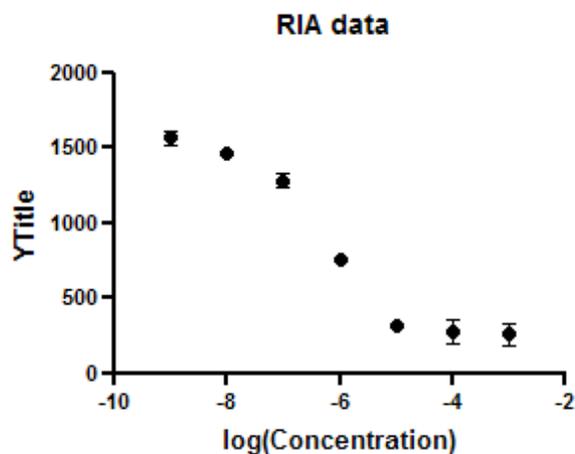
The sample data may be partly covered by a floating note explaining how to fit the data (for people who are not reading this help page). You can move the floating note out of the way, or minimize it.

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

The first seven rows contain the standard curve, in duplicate. Below that are three unknown values. These have a Y values that you measured, but no X. The goal of this analysis is to interpolate the corresponding X values (concentrations) for these unknowns. Note that three of the four unknowns are labeled, so you can later match up the results with the labels.

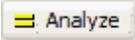
Why are X values negative? Because in this example, the X values are the logarithm of concentrations expressed in molar. So a concentration of 1 micromolar (10^{-6} Molar) would be entered as -6.

3. View the graph



The graph Prism makes automatically is fairly complete. You can customize the symbols, colors, axis labels, etc. You can also choose to plot the individual duplicates rather than plot the means. Since the unknowns have no X value, they are not included on the graph.

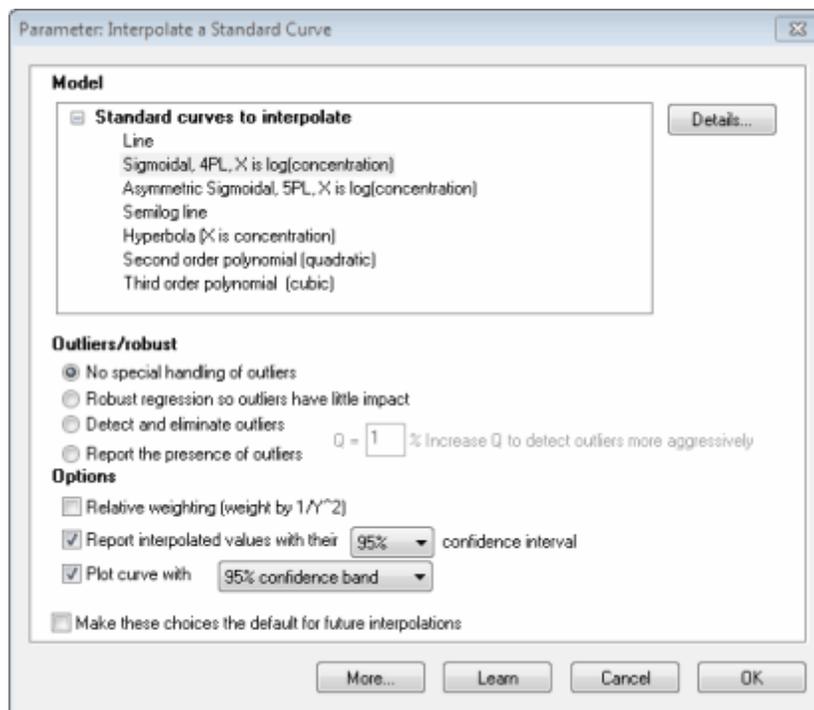
4. Choose the standard curve analysis

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

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

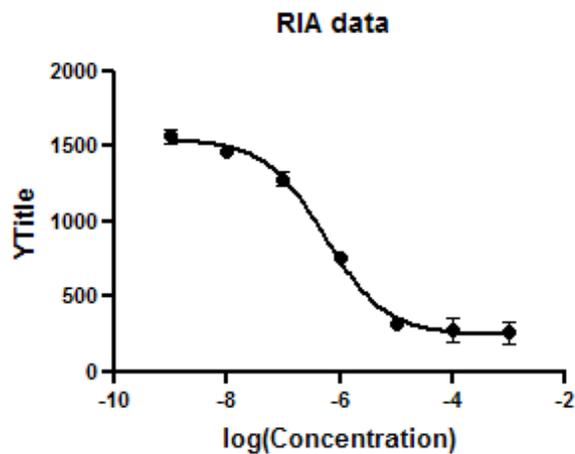
5. Choose a model

Choose the equation: Sigmoidal, 4PL, X is log(Concentration). Note that 4PL means four parameter logistic, which is another name for this kind of equation.



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

Click OK to see the curves superimposed on the graph.



6. Inspect the results

The results appear on several pages.

The first page shows you the interpolated values.

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

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

7. Transform the results

The X column of the results table has the interpolated values we want. These are in the same units as the X values, so are the logarithm of concentration. Prism can transform these values to concentration units.

Click  **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)".

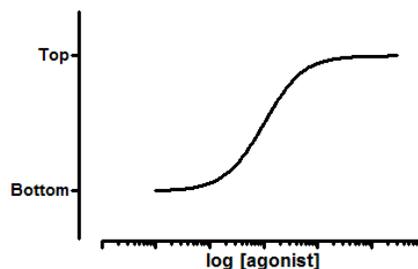
2.3.4 Equations used for interpolating

You can interpolate from any curve fit through a set of standards. However, the equations listed below are used most commonly. These equations are instantly available from the interpolation analysis. You'll also find this set of equations in the equation tree used in nonlinear regression.

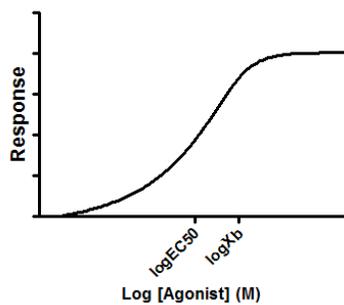
Notes:

- These graphs show the curve going up (increasing Y) as X increases. But all will work for downward sloping curves too.
- All of these equations are also present in other equation folders in the nonlinear regression dialog.

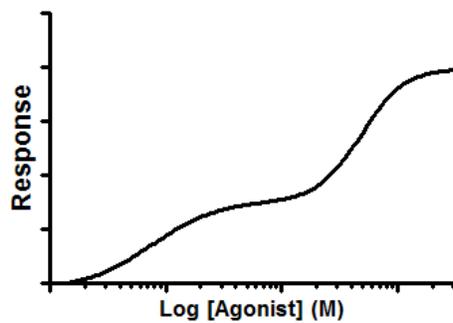
[Sigmoidal, 4PL, X is log\(concentration\)](#)



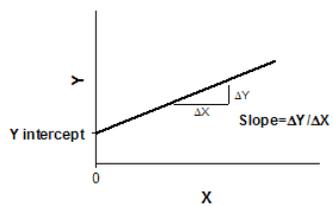
[Asymmetric Sigmoidal, 5PL, X is log\(concentration\)](#) 267



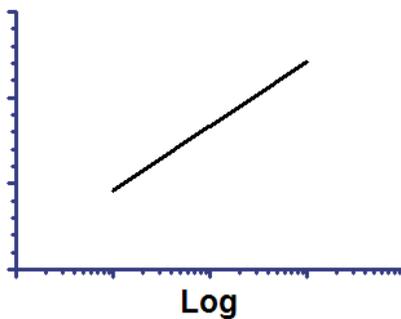
[Biphasic](#) 269



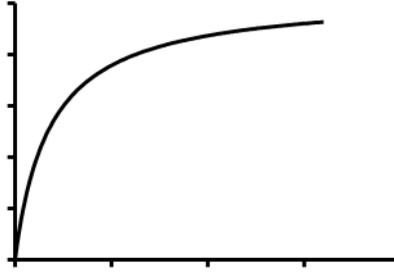
[Straight line](#) 371



[Semilog line -- X is log, Y is linear](#) 376



[Hyperbola \(X is concentration\)](#) 297



[Second order polynomial \(quadratic\)](#) 384

$$Y = B_0 + B_1 * X + B_2 * X^2$$

[Third order polynomial \(cubic\)](#) 384

$$Y = B_0 + B_1 * X + B_2 * X^2 + B_3 * X^3$$

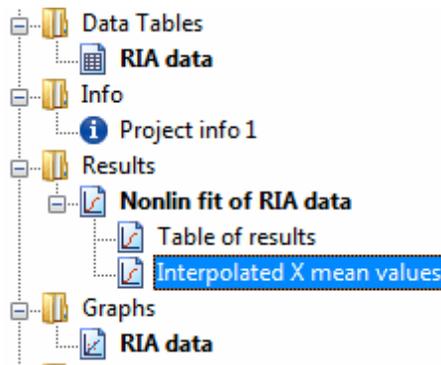
2.3.5 The results of interpolation

Where are the interpolated results?

Prism places the interpolation results into its an additional results page (or two additional pages).

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

To view this table, click the interpolated results page of the results. In some cases you may first need to click the + in front of the results main page to see all the pages within.



What units are the results in?

The interpolated results are expressed in the same units as the standard curves. If you entered a Y value, and Prism interpolated an X value, then that X value is in the same units as the standards you entered.

If the X values in the table are the logarithm of concentration (common), then the interpolated results are also logarithms of concentration.

In some assays, different samples are diluted differently. Prism does not correct for dilution, so you'll need to make those corrections manually.

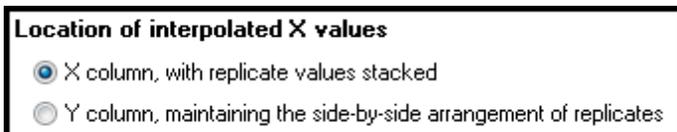
2.3.6 Interpolating with replicates in side-by-side subcolumns

Choices when entering data with replicates in subcolumns

It is common to enter replicate values for the unknowns, as shown below.

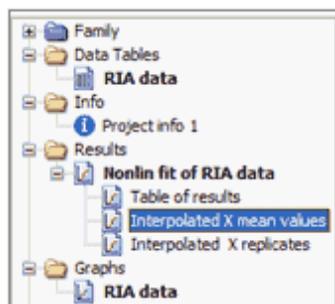
Table format:		X	Group A	
XY		log(Concentration)	Data Set-A	
		X	A:Y1	A:Y2
1	Title	-9.0	1597	1531
2	Title	-8.0	1453	1471
3	Title	-7.0	1314	1245
4	Title	-6.0	751	771
5	Title	-5.0	336	306
6	Title	-4.0	328	212
7	Title	-3.0	207	307
8	Unknown 1		1123	1211
9	Unknown 2		1345	1298
10	Unknown 3		1456	1523
11	Title		987	1111

Since Prism's results tables cannot be configured to put side-by-side replicates in the X column, you have two choices for how to organize the results. These choices are available at the bottom of the Output tab of the nonlinear regression dialog.



Place computed unknown values into the X column, with replicate values stacked

In this case, Prism creates two pages of results.



The interpolation is the same for both tables. Prism interpolates each replicate value individually.

The first page, Interpolated Mean X Values, shows the mean of the Y values you entered and the mean of the interpolated X values. The X column shows the concentrations (or logarithm of concentrations) to match the data you entered.

		X	A
		RIA data:log(Concentration):(Interpolated)	RIA data:Data Set-A:(Entered)
	X		Y
1	Unknown 1	-6.735	1167.000
2	Unknown 2	-7.103	1321.500
3	Unknown 3	-7.954	1489.500
4		-6.515	1049.000

The second page, Interpolated X replicates, shows individual replicate values for the interpolated results. The X column maintains the same meaning as it does in the data table you entered. To do so, with only a single X column, requires that the replicates be stacked.

		X	A
		RIA data:log(Concentration):(Interpolated)	RIA data:Data Set-A:(Entered)
	x	X	Y
1	Unknown 1	-6.649	1123.000
2	Unknown 1	-6.827	1211.000
3	Unknown 2	-7.175	1345.000
4	Unknown 2	-7.036	1298.000
5	Unknown 3	-7.673	1456.000
6	Unknown 3	-8.525	1523.000
7		-6.408	987.000
8		-6.627	1111.000

Place computed unknown values into a Y data set, maintaining the side-by-side arrangement of replicates.

When you make this choice, the interpolated values appear in the Y column, and the X column is blank. You can view the interpolated values, with their name (if you entered row titles). But the values you entered are not shown.

The first page shows the mean values. The second page shows the interpolated values in side-by-side replicates, to match the format you entered them. Even though these values are concentrations (or logarithm of concentrations), note that the results are in the Y column.

		X	A	
			RIA data:Data Set-A:(Interpolated)	
	x	X	A:Y1	A:Y2
1	Unknown 1		-6.649	-6.827
2	Unknown 2		-7.175	-7.036
3	Unknown 3		-7.673	-8.525
4			-6.408	-6.627

2.3.7 Interpolating several data sets at once

Prism can interpolate from several data sets at once. To do this, put all the unknowns on lower rows on the data table than all the knowns (left panel). Prism won't be able to interpolate past the first data set if you put the second data set below the first set of knowns (right panel).

Yes! All unknowns at the bottom.

No! Unknowns everywhere.

X	Group A	Group B	Group C
X Title	Assay 1a	Assay 2a	Data Set-C
X	Y	Y	Y
1.0000	165747.700		
0.5000	121733.900		
0.2500	75864.330		
0.1250	32976.810		
0.0625	14347.100		
1.0000		153566.900	
0.5000		166835.100	
0.2500		94793.330	
0.1250		46907.410	
	33599.090		
	13731.150		
	14745.600		
	7936.755		
	4967.436		
	16199.060		
		13168.850	
		12158.780	
		16889.140	
		6008.342	
		7112.103	
		9290.377	

X	Group A	Group B
X Title	Assay 1a	Assay 2a
X	Y	Y
1.0000	165747.700	
0.5000	121733.900	
0.2500	75864.330	
0.1250	32976.810	
0.0625	14347.100	
	33599.090	
	13731.150	
	14745.600	
	7936.755	
	4967.436	
	16199.060	
1.0000		153566.900
0.5000		166835.100
0.2500		94793.330
0.1250		46907.410
		13168.850
		12158.780
		16889.140
		6008.342
		7112.103
		9290.377

Why can't Prism handle the kind of data shown in the right panel above? Because then it might get mixed up about which values are unknowns to be interpolated, and which are simply adjacent to values left blank because they are missing.

2.3.8 When X values are logarithms

In many situations, you'll fit the standard curve to a log(dose) vs. response curve. In these cases, you'll enter the X values for the standard curve as the logarithm of concentration or you'll transform the data to logarithms before running nonlinear regression.

The interpolated X values are in the same units as the X values the analysis sees. If you are analyzing data entered or transformed so X is the logarithm of concentration, then the interpolated values will also be logarithms.

If you want to transform these results back to concentrations, that is easy to do:

1. From the interpolated results page of results, click Analyze.
2. Choose Transform at the very top of the Analyze dialog.
3. Choose the standard list of transforms.
4. Depending on whether your [interpolated values are in the X or Y column](#),¹¹³ choose to transform that column.
5. From the list of transforms, choose $Y=10^Y$ or $X=10^X$. These choices are about one third of the way down the long list of transforms.
6. The results sheets will have the interpolated values as concentrations, rather than logs.

Why 10^Y or 10^X ? Those are abbreviations for 10^Y or 10^X , which are the antilogarithm functions. [Review logarithms and antilogarithms](#).

2.3.9 Analysis checklist: Interpolating

Your approach in evaluating nonlinear regression depends on your goal.

In many cases, your goal is to learn from the best-fit values. If that is your goal, [view a different checklist](#).²¹⁸

If your goal is to create a standard curve from which to interpolate unknown values, your approach depends on whether this is a new or established assay.

Established assay

If the assay is well established, then you know you are fitting the right model and know what kind of results to expect. In this case, evaluating a fit is pretty easy.

- ✓ **Does the curve go near the points?**
- ✓ **Is the R^2 'too low' compared to prior runs of this assay?**

If so, look for outliers, or use Prism's automatic outlier detection.

✓ **Are the confidence bands too wide?**

The [confidence bands](#)^[171] 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 you are running an established assay, you know how wide you expect the prediction bands to be.

New assay

With a new assay, you also have to wonder about whether you picked an appropriate model.

✓ **Does the curve go near the points?**

Look at the graph. Does it look like the curve goes near the points.

✓ **Are the confidence bands too wide?**

How wide is too wide? The prediction bands show you how precise interpolations will be. Draw a horizontal line somewhere along the curve, and look at the two places where that line intercepts the confidence bands. This will be the confidence interval for the interpolation.

✓ **Does the scatter of points around the best-fit curve follow a Gaussian distribution?**

Least squares regression is based on the assumption that the scatter of points around the curve follows a Gaussian distribution. Prism offers three normality tests (in the Diagnostics tab) that can test this assumption (we recommend the D'Agostino test). If the P value for a normality test is low, you conclude that the scatter is not Gaussian.

✓ **Could outliers be impacting your results?**

Nonlinear regression is based on the assumption that the scatter of data around the ideal curve follows a Gaussian distribution. This assumption leads to the goal of minimizing the sum of squares of distances of the curve from the points. The presence of one or a few outliers (points much further from the curve than the rest) can overwhelm the least-squares calculations and lead to misleading results.

You can spot outliers by examining a graph (so long as you plot individual replicates, and not mean and error bar). But outliers can also be detected

automatically. GraphPad has developed a new method for identifying outliers we call the ROUT method. You can either ask Prism to simply identify outliers or to eliminate them. These choices are adjacent on the Interpolate a Standard Curve dialog. But they are in separate tabs in the Nonlinear Regression dialog. The option to count outliers is on the Diagnostics tab, and the the option to exclude outliers is on the Fit tab.

Does the curve deviate systematically from the data?

If either the runs test or the replicates test yields a low P value, then you can conclude that the curve doesn't really describe the data very well. You may have picked the wrong model.

2.3.10 Reasons for blank (missing) results

Interpolated value is blank

When you ask Prism to interpolate X values from Y values you enter, it will leave the result blank when it is unable to find a result. This can happen when:

- The input value for Y falls out of the vertical range of the standard curve. This happens when you fit a sigmoidal log(dose) vs response curve. If you enter an unpaired Y value that is larger than the best-fit value of Top, or smaller than the best-fit value of Bottom, there simply is no corresponding X value, and Prism will simply leave that part of the Results sheet blank. Note that "top" and "bottom" apply to the best-fit values for those fitted parameters, not the maximum and minimum values of the input data. It simply is impossible to interpolate a X value if the Y value is too high, or too low, to be on the curve.
- The interpolated X value falls way outside the output range of X for the fitted curve. Prism will try to extrapolate a bit beyond the range of X values for which it calculates and draws the curve. But it won't extrapolate far from the ends of the curve. You can extend this range. From the Nonlinear regression Parameters dialog, choose the Range tab and enter a lower value for the minimum X and/or a higher value for the maximum X.
- You entered the unknowns in rows above the standard curve values. The unknowns must be entered in rows below the standards.
- You mistakenly entered the unknowns in the wrong units. The unknown values must be in the same units as the standards. If you enter unknown Y

values and expect Prism to interpolate an X value, then the unknown Y values must be entered in the same units as the Y values for the standards.

Confidence interval of interpolated value is blank

Prism can also report the 95% confidence interval for the interpolated values. It only does this when the interpolated X value is within the range of the standard curve. If the interpolated value is extrapolated beyond that range, Prism leaves the confidence interval blank.

2.3.11 Q&A: Interpolating

Expand all

Collapse all

Can Prism interpolate several data sets at once?

Yes, Prism can interpolate X values from Y values entered into several data sets. Note the following:

- Prism has to figure out which values are unknowns. It uses this rule: X (or Y) values are considered as unpaired (and are used for calculation of unknowns) when they are lower in the table than all XY pairs in the data table. It is not enough that the unpaired Y values (from which you want Prism to interpolate X) is lower on the table than the paired XY values for that particular data set. They must be on lower rows in the table than all the paired XY values of all data sets in the table.
- Prism [offers an option in the Output tab](#)¹¹³⁾ of nonlinear regression that let you choose where the interpolated X values should go (into the X column staggered, or onto Y columns). If you have replicates and have unknowns in more than one data set, choose the option to show the unknowns in Y columns.

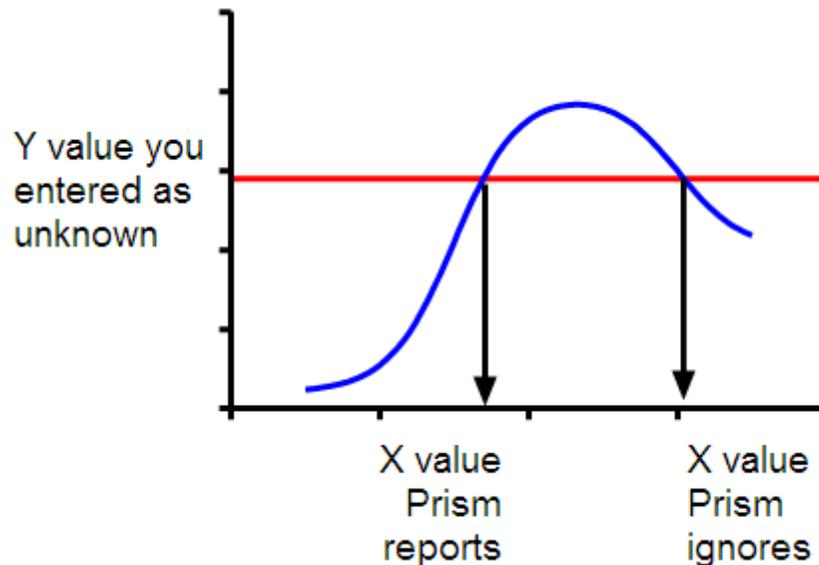
What units are the interpolated values expressed in?

When you enter a Y value, and ask Prism to interpolate the X value, that X value is in the same units as the X values you entered. Usually this is concentration. If it is the logarithm of concentration, you can [transform it back to a concentration scale](#).¹¹⁷⁾

What happens when there are two X values corresponding to the entered Y value?

In some cases, there are two X values that correspond to a Y value you enter. To

understand how Prism handles this situation, you need to know how it interpolates. It starts at the minimum X value and works its way to the largest X value. It stops when it finds the Y value you entered. If it doesn't find a Y value matching the one you entered, it then looks at X values lower than the minimum shown on the graph and to X values larger than the maximum. If there are two points on the curve that have the Y value you entered as an unknown, Prism will find the lower of the two corresponding X values.



Can Prism plot the interpolated values?

Sure. The interpolated values are tabulated on a results table. If you want to plot only these values, start from this table, click the New button, and choose New Graph of Existing data. If you want to add these interpolated values to the graph of the unknowns and its fit, go to that graph, double-click to go to Format Graph, go to the middle (Data on graph) tab, and choose to add the table of interpolated values.

2.3.12 How Prism interpolates

If you ask Prism to compute Y for a given X, it does so by evaluating the equation that defines the curve. That is very straightforward.

If you ask Prism to compute X for a given Y, it is more complicated. Prism does not try to solve the equation algebraically but rather does the interpolation numerically. The results are accurate to at least 6-7 decimal places.

Prism decides on the range of X values to consider. To allow for extrapolation a bit beyond the range of the data, Prism creates an interpolation/extrapolation range that includes the range of the data, and extends in each direction by a distance equal to half the difference between Xmax and Xmin. There are two special cases. When all the data are positive (or zero), that range is clipped to exclude negative numbers. Similarly, when the data are all negative (or zero), the interpolation/extrapolation range is clipped to exclude positive numbers.

Prism then divides that interpolation/extrapolation range into 1000 line segments.

For each value to be interpolated or extrapolated, Prism first tries to interpolate within the range of the X values of the data.

1. It starts with the lowest X value (scans from left to right on the graph). If more than one line segment includes the Y value, Prism only finds the first (lowest X value).
2. Prism then interpolates within that line segment to determine X as accurately as possible. In most cases, it does this by binary bisection. It divides the segment in half and figures out which contains Y. Then it divides that half in half again. And again. This continues until X is determined as accurately as possible given the numerical precision of the computer. In rare cases, it is possible that Y is not monotonic within the range determined in step 1. In this case, Prism uses linear interpolation rather than binary bisection.

If Prism is not able to interpolate a Y value that corresponds to the entered X value within the X range of the data, it will try to extrapolate a value to the extended range mentioned above. It first looks at X values below the minimum X value in the data, and then at X values above the maximum X value in the data.

When Prism interpolates a value (the X value is in the range of the data), it will also compute a confidence interval if you request it to do so. It does this by determining where the two confidence bands intersect the Y value you entered. When Prism extrapolates a value from the curve, it will not compute a confidence interval for that extrapolated value.

2.3.13 Standard Addition Method

Prism can easily interpolate from a linear or nonlinear standard curve. You

perform the assay at a number of known concentrations, fit a line or curve, and interpolate the unknown values.

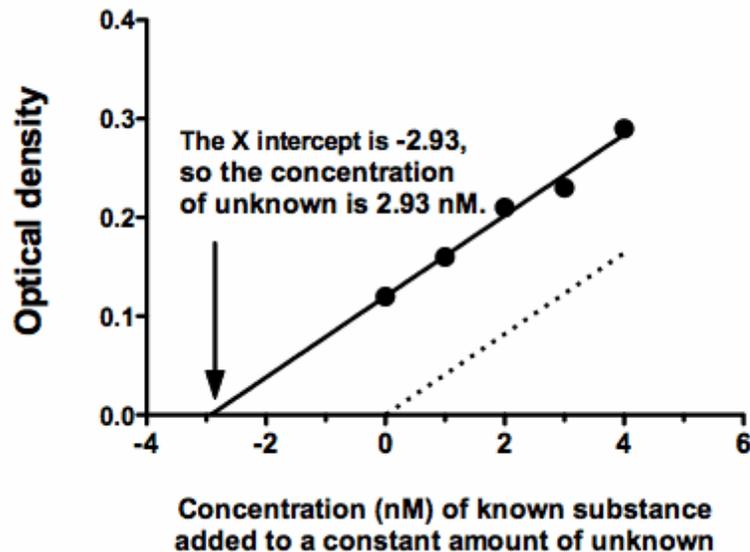
But there is a problem with interpolating from a standard curve. The results can be incorrect when the unknown sample are contaminated with other substances that alter the assay. This is known as the *matrix effect problem*.

The *Standard Addition Method* is a way to bypass this problem. Add various known concentrations (including zero) of known substance to a constant amount of the unknown, and perform the assay. This ensures that all the samples have the same amount of unknown, including any substances that interfere with the assay.

Fit the data with linear regression. The Standard Addition Method only works when the assay measurement (Y) is proportional to the concentration of substance you are measuring.

The dotted line in the figure below shows how the assay results (usually optical density) varies as you change the concentration of the substance you are assaying. The circles and solid line show the assay results in the presence of an unknown concentration of the substance you are assaying. This pushes the line up and to the left.

The value you want to know is how far the solid line is shifted to the left of the dotted line. There is an easy way to find out: Extrapolate the line down to $Y=0$. One of the parameters that Prism reports is the X intercept, which will be negative. Take the absolute value, and that is the concentration of the unknown substance. The confidence interval for the X intercept gives you the confidence interval for the concentration of the unknown. Simply multiply both confidence limits by -1.



To plot the data in Prism, you'll want to extend the linear regression line to start at an X value equal to the X intercept (a choice in the Linear regression parameters dialog). You may also want to move the origin to the lower left, a choice on the first tab of the Format Axis dialog. [Prism file.](#)

2.4 Nonlinear regression tutorials

Prism makes it quite easy to fit a model to your data. If you are new to Prism, choose from the sample XY data sets. These not only show you how to use Prism, but also review the principles of nonlinear regression, including comparing models, identifying outliers, global fitting, and more.

- [Example: Fitting an enzyme kinetics curve](#) ¹²⁶
- [Example: Comparing two enzyme kinetics models](#) ¹³¹
- [Example: Automatic outlier elimination \(exponential decay\)](#) ¹³⁵
- [Example: Global nonlinear regression \(dose-](#)

[response curves](#) 

- [Example: Ambiguous fit \(dose-response\)](#) 

Prism 6 now has a new, separate, analysis for interpolating from a standard curve, with its own example:

[Example: Interpolating from a sigmoidal standard curve](#) 

2.4.1 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, and select the sample data "Enzyme kinetics -- Michaelis-Menten".

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.

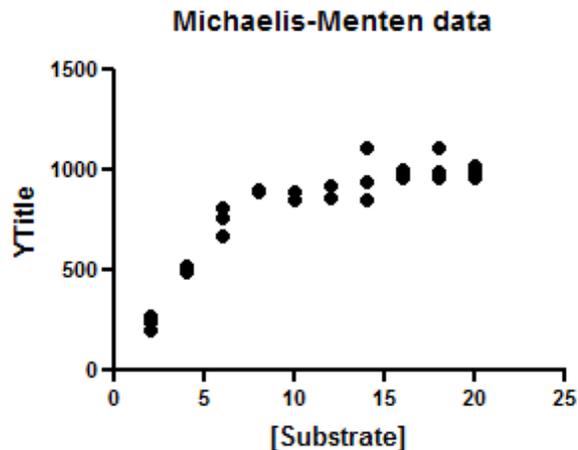
Table format:		X	A		
XY		[Substrate]	Enzyme Activity		
	x	X	A:Y1	A:Y2	A:Y3
1	Title	2	265	241	195
2	Title	4	521	487	505
3	Title	6	662	805	754
4	Title	8	885	901	898
5	Title	10	884	850	
6	Title	12	852		914
7	Title	14	932	1110	851
8	Title	16	987	954	999
9	Title	18	984	961	1105
10	Title	20	954	1021	987

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

3. View the graph

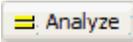
Prism automatically created a graph and gave it the same name as the data table. Click on the Michaelis-Menten graph in the graphs section.

Since this is the first time you are viewing the graph, Prism will pop up the Change Graph Type dialog. Select the fifth choice, to plot individual replicates rather than mean and error bars.

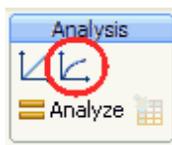


The graph Prism makes automatically is fairly complete. You can customize the symbols, colors, axis labels, position of legend, etc.

4. Choose nonlinear regression

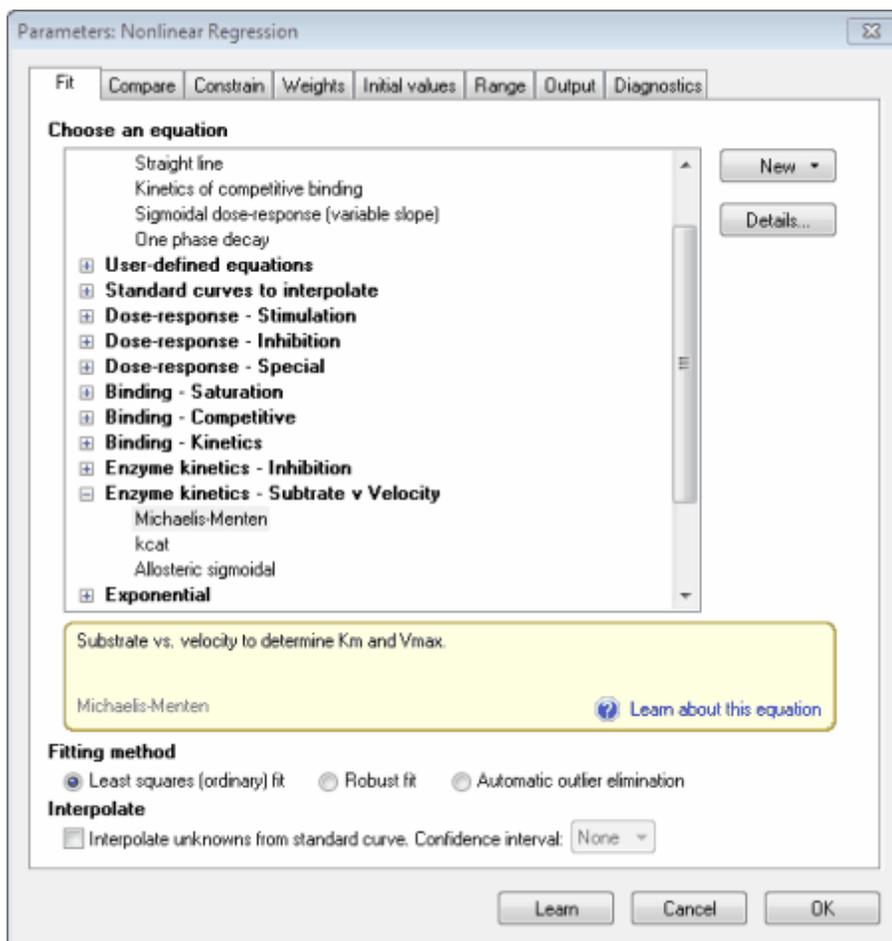
Click  and choose Nonlinear regression from the list of XY analyses.

Even faster, click the shortcut button for nonlinear regression.



5. Choose a model

On the Fit tab of the nonlinear regression dialog, open the equation folder, Enzyme Kinetics - Substrate vs. Velocity. Then choose the Michaelis-Menten equation.

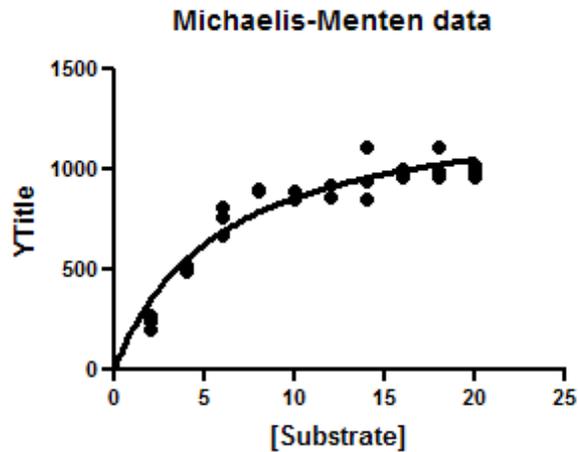


Learn more about the [principles of enzyme kinetics](#)³³⁵ and about [fitting Michaelis-Menten curves](#)³³⁸.

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

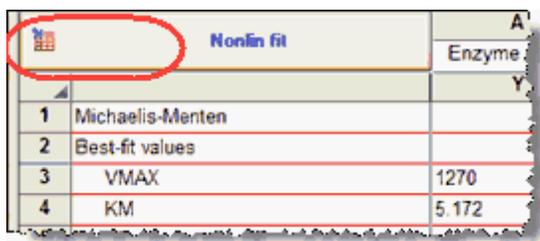
Nonlin fit		A
		Enzyme Activity
		Y
1	Michaelis-Menten	
2	Best-fit values	
3	VMAX	1353
4	KM	5.886
5	Std. Error	
6	VMAX	75.93
7	KM	0.9498
8	95% Confidence Intervals	
9	VMAX	1197 to 1509
10	KM	3.933 to 7.839
11	Goodness of Fit	
12	Degrees of Freedom	26
13	R ²	0.9041
14	Absolute Sum of Squares	170343
15	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](#)¹⁸³.

8. Go back and perform the replicates test

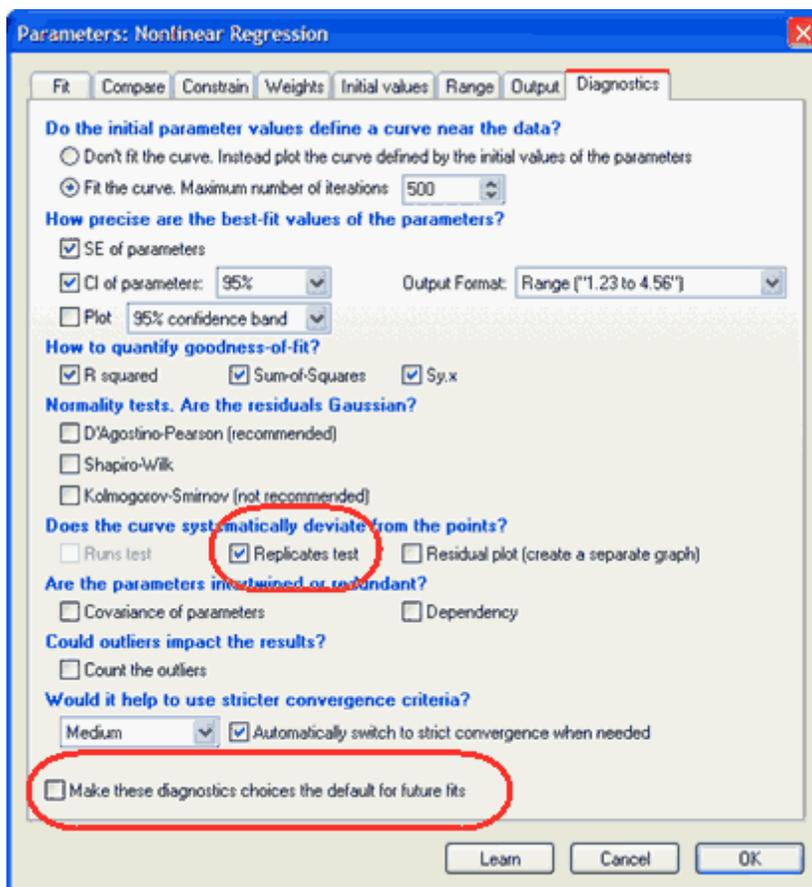
The replicates test assesses the adequacy of the fit by comparing the scatter among the triplicates with the scatter of points around the curve. It is not calculated by default, so the results do not appear in the results of step 7.

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](#)¹⁶⁵, and check the option to perform the [replicates test](#)¹⁹⁴. Note that you can also check an option to make your settings here become the default for future fits.



Parameters: Nonlinear Regression

Fit Compare Constrain Weights Initial values Range Output **Diagnostics**

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 interrelated 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

Learn Cancel OK

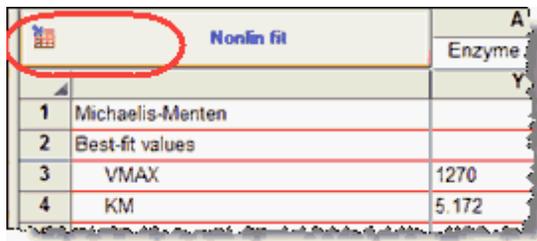
The P value is small (0.013). This means that the scatter of the data from the curve is greater than you'd expect from the variation among triplicates. This suggests that you might want to consider fitting an alternative model, which we do in the [next example](#)¹³¹.

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

2.4.2 Example: Comparing two enzyme kinetics models

1. Open analysis parameters dialog

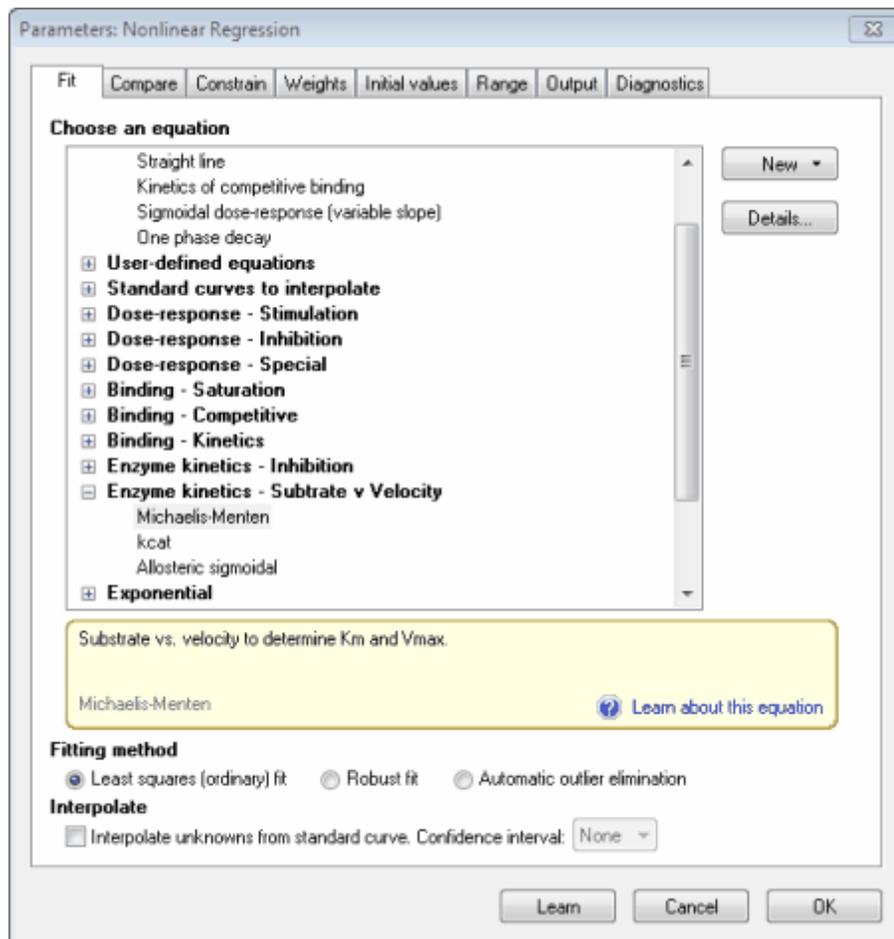
This example continues from the [previous one](#)¹²⁶. Click the analysis parameter button again.



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

2. Choose the first model

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



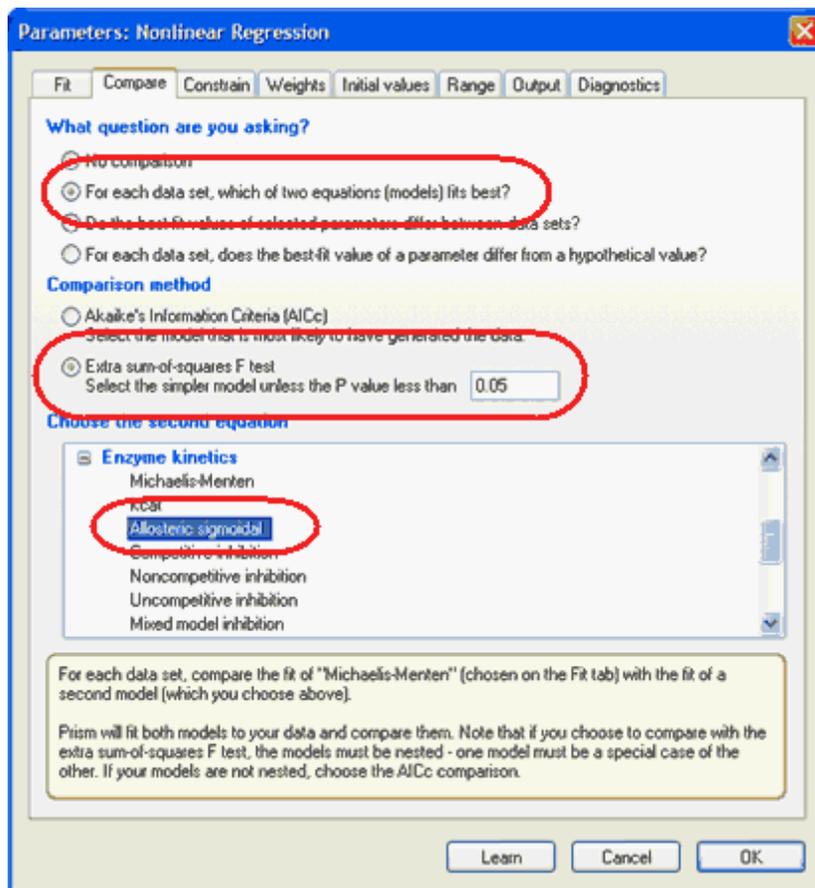
3. Choose the other model

Go to the [Compare tab](#)¹⁵⁷.

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](#)⁵⁰.

For the second equation, choose "[Allosteric sigmoidal](#)"³⁴² from the Enzyme kinetics section.



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	

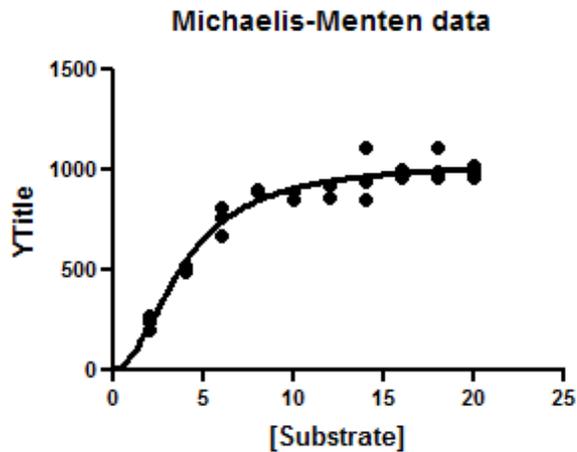
The P value of the [replicates test](#)¹⁹⁴ is high, which means the scatter of points around the curve is consistent with variability of replicates from each other.

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.



2.4.3 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](#)^[57], [when it is useful](#)^[53], and [when it should be avoided](#)^[54].

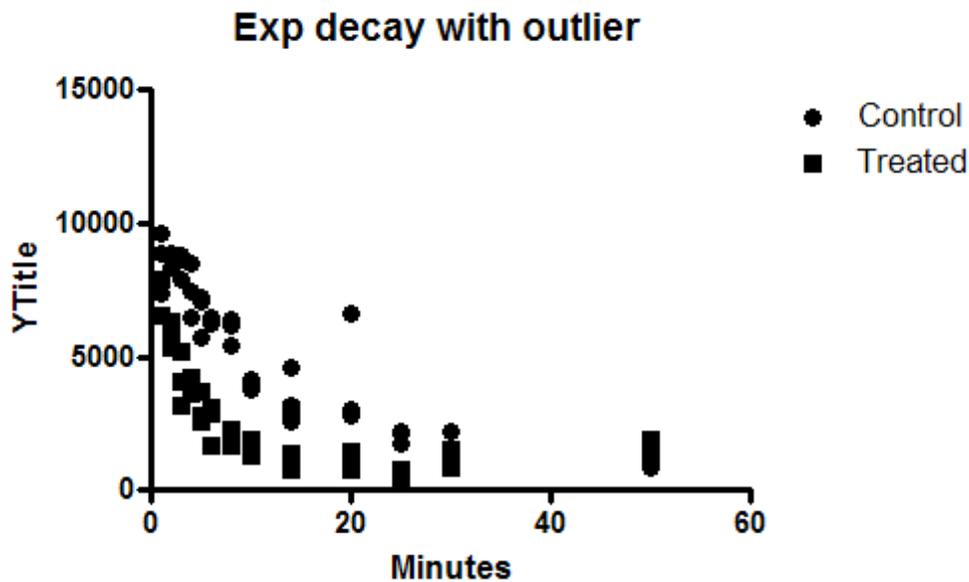
1. Create the data table

From the Welcome or New Table dialog, choose to create XY data table, and select the sample data "Eliminating outliers during nonlinear regression".

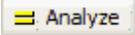
2. Inspect the data table and graph

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

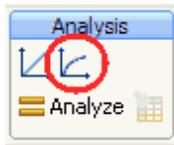
Since this is the first time you are viewing the graph, Prism will pop up the Change Graph Type dialog. Select the fifth choice, to plot individual replicates rather than mean and error bars.



3. Choose nonlinear regression

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

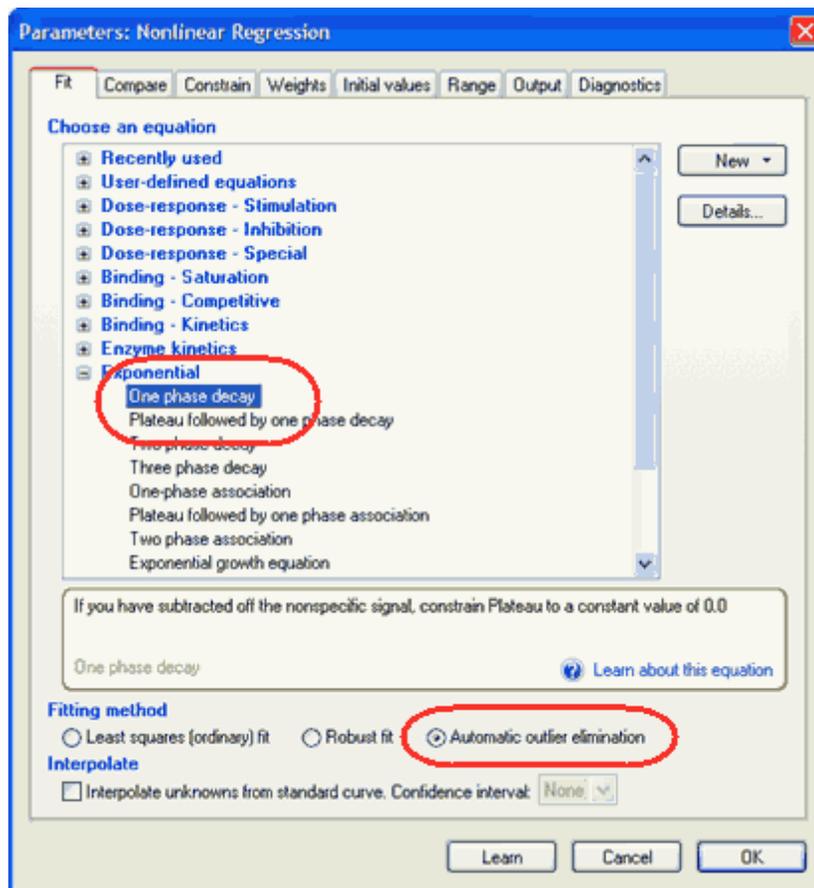
Alternatively, click the shortcut button for nonlinear regression.



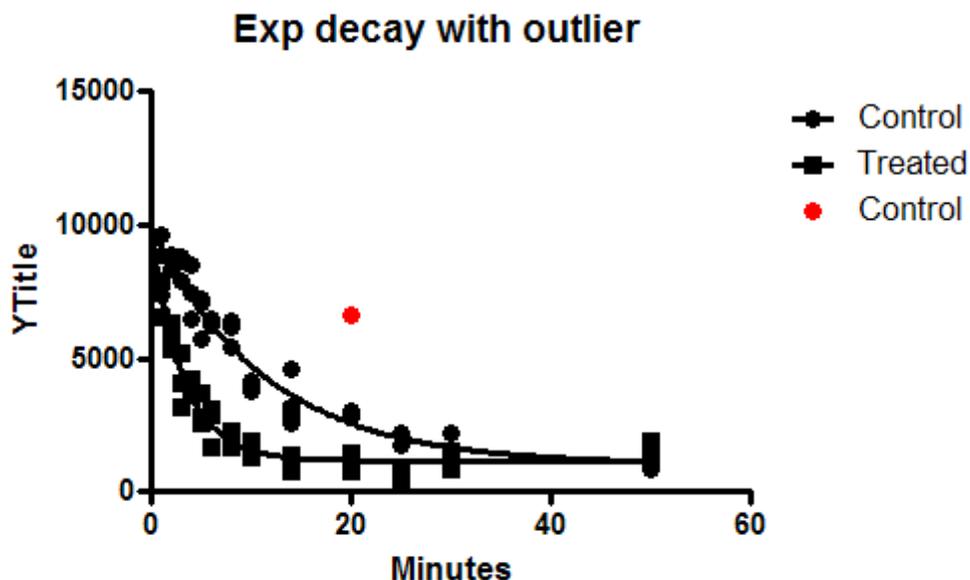
4. Choose a model, and choose automatic outlier detection

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](#)^[53], when it should be [avoided](#)^[54], and [how it works](#)^[57].



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](#)^[56].

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](#)^[176].

2.4.4 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 "Dose-response: EC50 shift by global fitting". Note that this choice is in a drop down menu of sample data sets.

2. Inspect the data

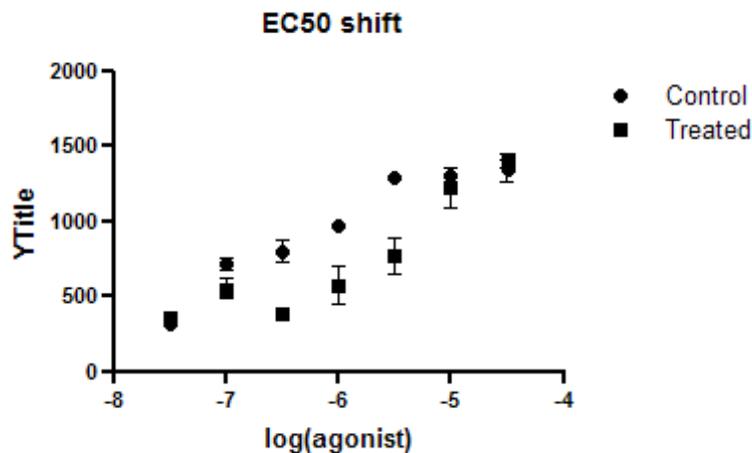
The sample data may be partly covered by a floating note explaining how to fit the data (for people who are not reading this help page). You can move the floating note out of the way, or minimize it.

	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

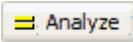
The X values are the logarithm of the concentration of agonist. The Y values are responses, in duplicate, in two conditions.

3. View the graph

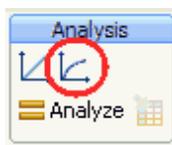
Since this is the first time you are viewing the graph, Prism will pop up the Change Graph Type dialog. Select the first choice, to plot mean and error bars, and choose the kind of error bars you prefer.



4. Choose nonlinear regression

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

Alternatively, click the shortcut button for nonlinear regression.

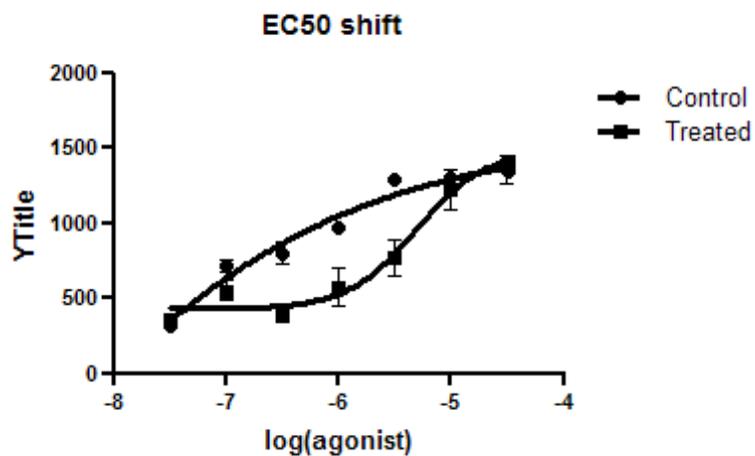


5. Choose a model

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	R ²	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.

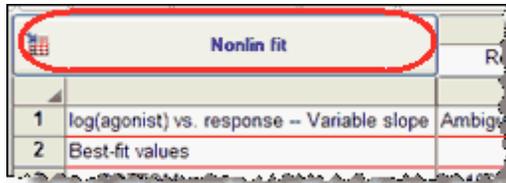
The treated curve is not labeled 'ambiguous', but the confidence intervals are wider than you'd like.

7. Go back to the dialog, and share three parameters

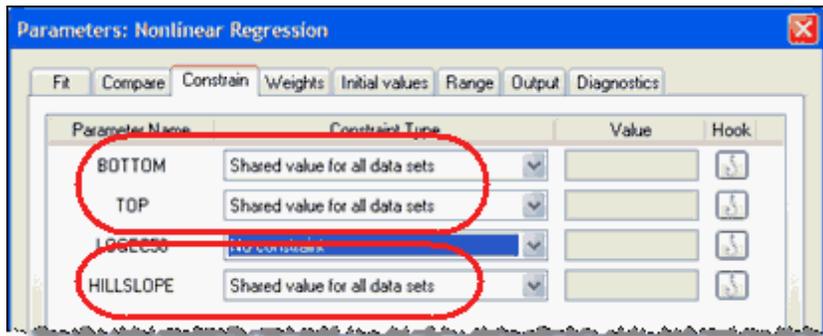
You can get much better results from this data set if you are willing to assume

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.

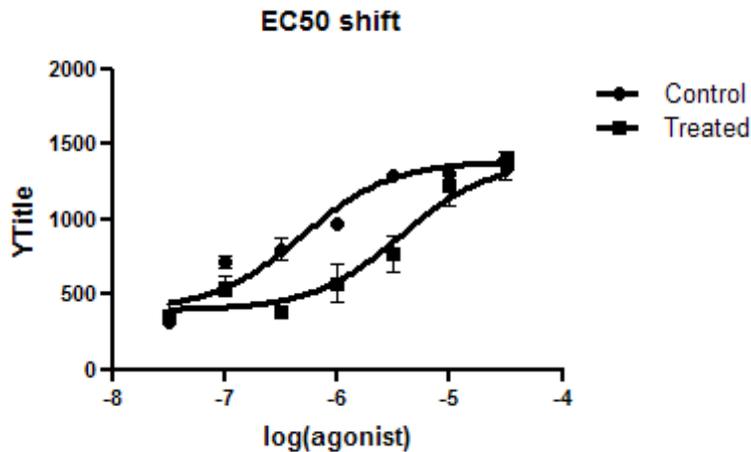
Return to the nonlinear regression dialog by clicking the button in the upper left of the results table.



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.



8. View the revised graph and results

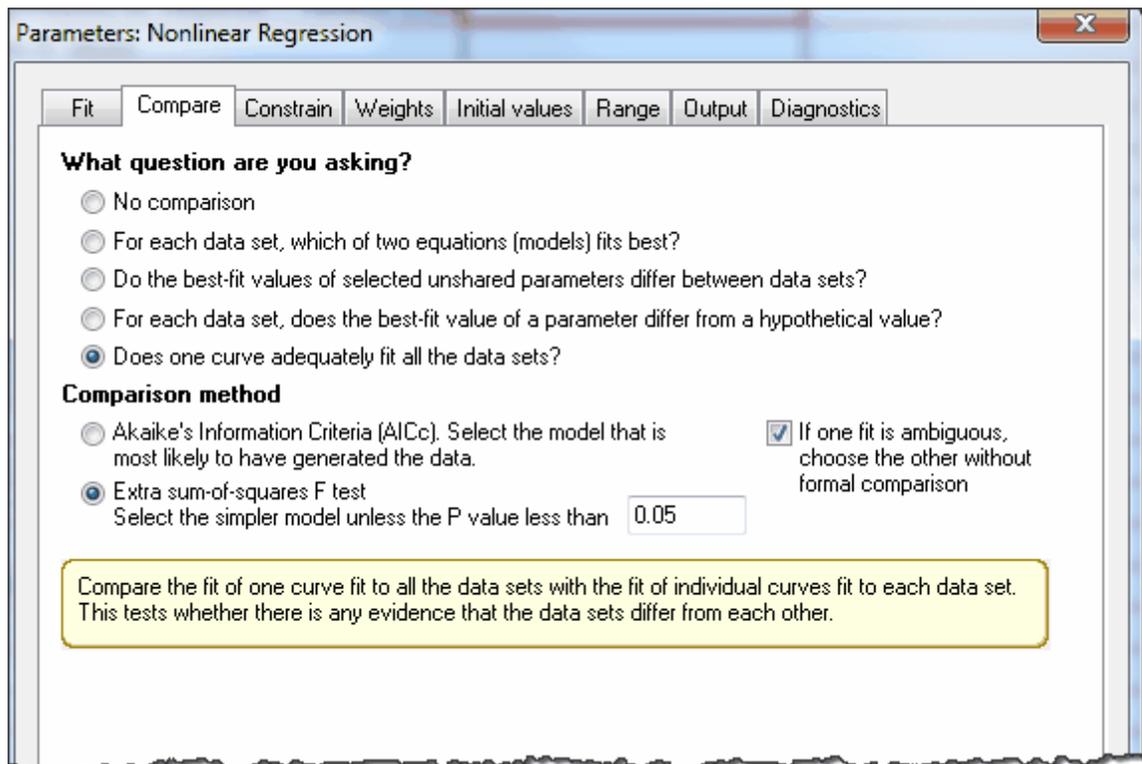


Nonlin fit		A	B	C
		Control	Treated	Global (shared)
	Y	Y	Y	
1	log(agonist) vs. response -- Variable slope			
2	Best-fit values			
3	BOTTOM	395.4	395.4	395.4
4	TOP	1388	1388	1388
5	LOGEC50	-6.304	-5.430	
6	HILLSLOPE	1.099	1.099	1.099
7	EC50	4.960e-007	3.719e-006	
8	Span	992.2	992.2	992.2
9	Std. Error			
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. Statistically compare the two logEC50 values

Go back to the parameters dialog for nonlinear regression and go to the Compare tab. Check the option to test whether one curve adequately fits all the data sets.



Prism will now fit the data two ways. The first is the same as before, fitting a separate IC50 for each data set. The second fit shares all the parameters. In this case, three parameters were already shared but one wasn't. So in this second fit, all four parameters are shared, so Prism fits one curve through all the data, ignoring which treatment group they are in. It compares the sum-of-squares (really the sum-of-sum-of squares since there are two data sets fit in each case), using the extra sum-of-squares F test. The results are shown at the top of the results sheet.

Nonlin fit		A	B	C
		Control	Treated	Global (shared)
		Y	Y	Y
1	Comparison of Fits			
2	Null hypothesis			LogIC50 same for all data sets
3	Alternative hypothesis			LogIC50 different for each data set
4	P value			< 0.0001
5	Conclusion (alpha = 0.05)			Reject null hypothesis
6	Preferred model			LogIC50 different for each data set
7	F (DFn, DFd)			33.57 (1,23)

The P value is tiny, so we reject the null hypothesis that the two IC50 values are identical in the population, and instead conclude that the two IC50 values are different.

10. Fit the ratio of two IC50 values directly

From the results in step 8, you can compute what you want to know -- the ratio of the two EC50 values.

But Prism can calculate this value directly.

Go back to the analysis parameters dialog, and on the Fit tab, change the equation to "EC50 shift" from the "Dose-response - Special" group of equations. Accept all defaults and click OK. The graph will look identical, as the model is equivalent. But now, rather than fitting two logEC50 values, Prism fits one and also fits the ratio.

Nonlin fit		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](#)^[279], so you can construct your own equations when necessary.

2.4.5 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". Note that this choice is in a drop down menu of sample data sets.

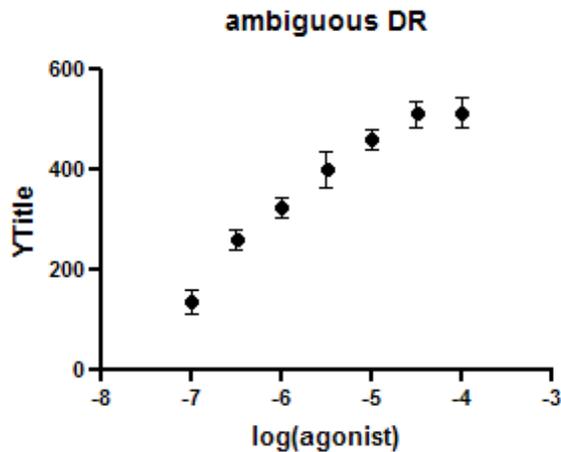
2. Inspect the data table

The X values are the logarithms of molar concentration. The Y values are responses, entered as mean and SD. With Prism, you can either enter replicate values or enter error values (here SD) computed elsewhere.

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

3. View the graph

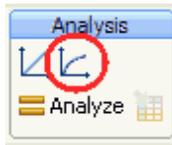
Since this is the first time you are viewing the graph, Prism will pop up the Change Graph Type dialog. Select the first choice, to plot mean and error bars, and choose the kind of error bars you prefer.



4. Choose nonlinear regression

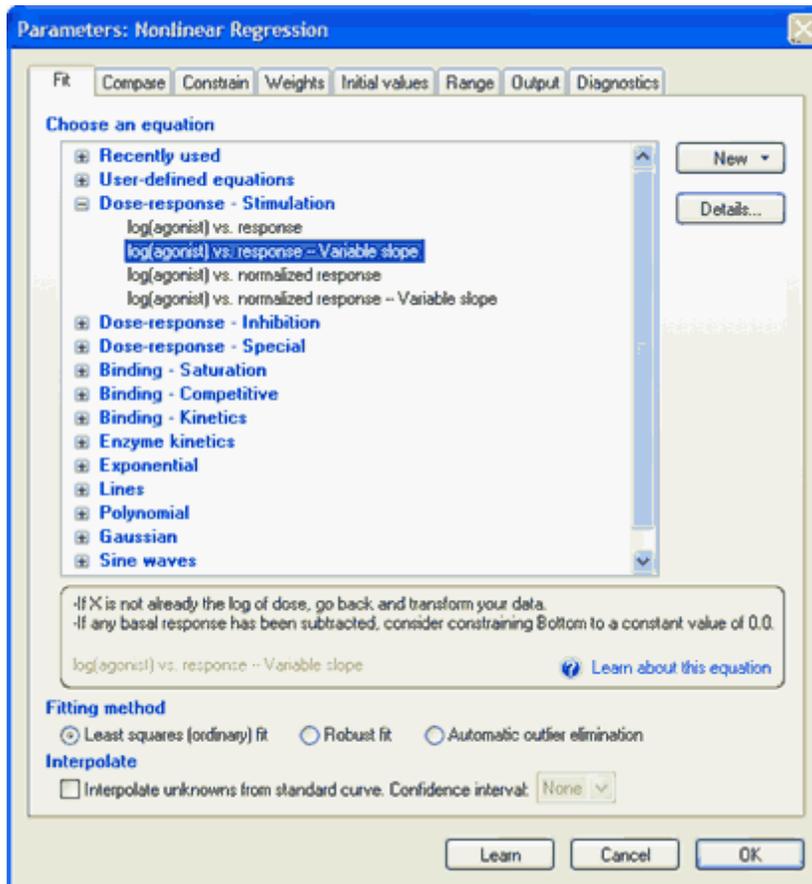
Click  Analyze and choose from the list of XY analyses.

Even faster, click the shortcut button for nonlinear regression.



5. Choose a model

On the Fit tab of the nonlinear regression dialog, open the panel of Stimulatory dose-response equations and choose: log(agonist) vs. response -- Variable slope.



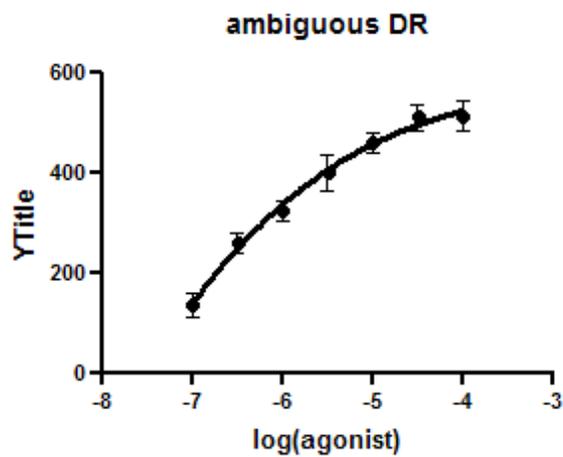
Learn more about [fitting dose-response curves](#)²⁵⁶.

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](#)²²⁸.

Prism does not report confidence intervals for the logEC50 or the Bottom of the curve, but instead simply says the intervals are 'very wide'. That tells you it was impossible to fit those parameters precisely.

While the Y values of the data run 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 R² is 0.9669), the best-fit parameter values are useless.

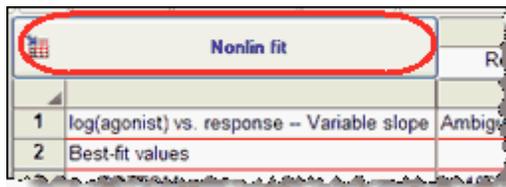
8. Constrain the Bottom of the curve

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

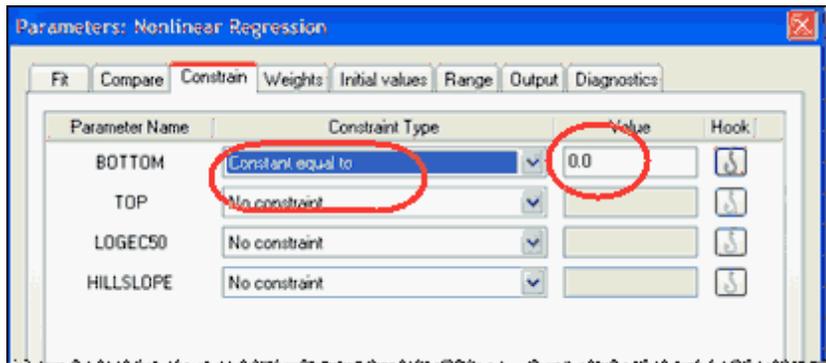
the mid point and steepness. But the data simply don't define the bottom of the curve. In fact, Prism the best fit value of Bottom is very very far from the data.

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.



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



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	R ²	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

2.5 Nonlinear regression with Prism

Prism's nonlinear regression dialog has seven tabs, explained here. It can seem overwhelming at first. But you don't have to learn about all the choices when you first fit a curve. Start by focussing on the choices in the first (Fit) tab, and then learn about the other choices later.

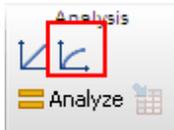
- Before learning all the details, first [understand the big picture](#)¹⁵² -- the basic steps required to fit a model to data.
- So many choices! [Which ones are essential?](#)¹⁵⁴

2.5.1 How to fit a model with Prism

Step 1. Enter data

Create an XY table, and enter data. If you have replicate Y values at each X value, format the table for entry of replicates.

From an XY table or graph, click the shortcut button to fit a model with nonlinear regression. Or click Analyze and select from the analyze dialog.



Step 2. Choose a model

Nonlinear regression fits a model to your data. You must, therefore, choose a model or enter a new model.

[Why a computer program cannot pick a model for you](#)^[12].

Step 3. Choose (or review) initial values

Nonlinear regression is an iterative procedure. The program must start with estimated values for each variable that are in the right "ball park" - say within a factor of five of the actual value. It then adjusts these initial values to improve the fit. See How nonlinear regression works.

GraphPad Prism provides initial values automatically if you use a built-in equation. If you enter your own equation, you can also enter rules for initial values. For example the initial value of one parameter may be twice the maximum Y value in the data, while the initial value of another parameter may equal the average of the highest and lowest X values. Once you define these rules, Prism will compute appropriate initial values based on the range of your data.

The [Initial Values](#)^[162] tab of the Nonlinear regression dialog lets you review, and override, the computed initial values.

You'll find it easy to estimate initial values if you have looked at a graph of the data, understand the model, and understand the meaning of all the parameters in the equation. Remember that you just need an estimate. It doesn't have to

be very accurate. If you are having problems estimating initial values, set aside your data and simulate curves using the model. Change the variables one at a time, and see how they influence the shape of the curve. Once you have a better feel for how the parameters influence the curve, you might find it easier to estimate initial values.

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

Step 4. Decide whether to constrain any parameters

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

Remember that nonlinear regression programs have no "common sense". You need to think about how you did the experiment, and decide whether some of the parameters should be fixed. For example, if a background signal has already been subtracted, it makes sense to fix the bottom plateau of a dose-response curve or an exponential decay curve to zero.

Prism also let you constrain parameters to a certain range of values.

Step 5. If you are fitting two or more data sets at once, decide whether to share any parameters

If you enter data into two or more data set columns, Prism will fit them all in one analysis. But each fit will be independent of the others unless you specify that one or more parameters are shared. When you share parameters, the analysis is called a [global nonlinear regression](#)^[41].

Step 6. Decide on a weighting scheme

Nonlinear regression programs generally weight each point equally. But there are many ways to [differentially weight the points](#)^[160].

Step 7. Choose other options

Read about choices on the [Range](#)^[164], [Output](#)^[164] and [Diagnostics](#)^[165] tab.

2.5.2 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

If your goal is to interpolate from a standard curve, you can ignore most of the choices that Prism offers. You do need to pick a model, but you should judge the adequacy of the model visually. If it goes through the points without wiggling too much, it is fine for interpolation. Note that Prism 6 offers a [simplified curve fitting analysis for interpolation](#)^[105].

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

If your goal is to fit a model to determine parameter values, then choosing a model is the essential first step. You should not expect a computer (or a software company's tech support) to choose a model for you. Choosing a model, and deciding which parameters should be constrained and which should be shared among data sets, is a scientific decision that is fundamental to analyzing your data. The other choices are useful, but can be put aside when you are first learning curve fitting..

2.5.3 Nonlinear regression choices

- Choose the model (equation) on the [Fit tab](#)^[155]. This is also where you choose to eliminate outliers and to interpolate values from the fit curve.
- [Compare tab](#)^[157]. Choose to compare the fit of two models to each data set, or to compare the fit of two (or more) data sets.
- [Constrain tab](#)^[158]. Fix parameter values to constant values and/or designate parameters to be shared among all data sets.
- There are many situations where you don't want to weight all points equally. Set unequal weighting on the [Weights tab](#)^[160].

- Nonlinear regression is an iterative procedure. The program must start with estimated initial values for each parameter. It then adjusts these initial values to improve the fit. View those values on the [Initial values tab](#)¹⁶².
- [Range tab](#)¹⁶⁴. Set the X range of data to fit, the starting and ending X values for the plotted curve, and whether you want to view a table of XY coordinates that define the curve.
- [Output tab](#)¹⁶⁴
- [Diagnostics tab](#)¹⁶⁵. Prism can create more detailed results tables.

2.5.3.1 Fit tab

Choose an equation

Selecting an equation is the most important step in fitting a curve. The choice [cannot be automated](#)¹².

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](#)⁴⁰⁹ to fit your needs.
- Part of choosing a model is choosing [constraints](#)¹⁵⁸. 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](#)^[158]), Prism finds one best-fit value for the parameter that applies to all the data sets. [Read more](#)^[41] about shared parameters (global fitting).

Fitting method

Prism offers three methods to fit curves.



If you aren't sure which method to choose, pick least-squares regression, as it is standard. [Robust regression](#)^[59] 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](#)^[53], so should be used with caution.

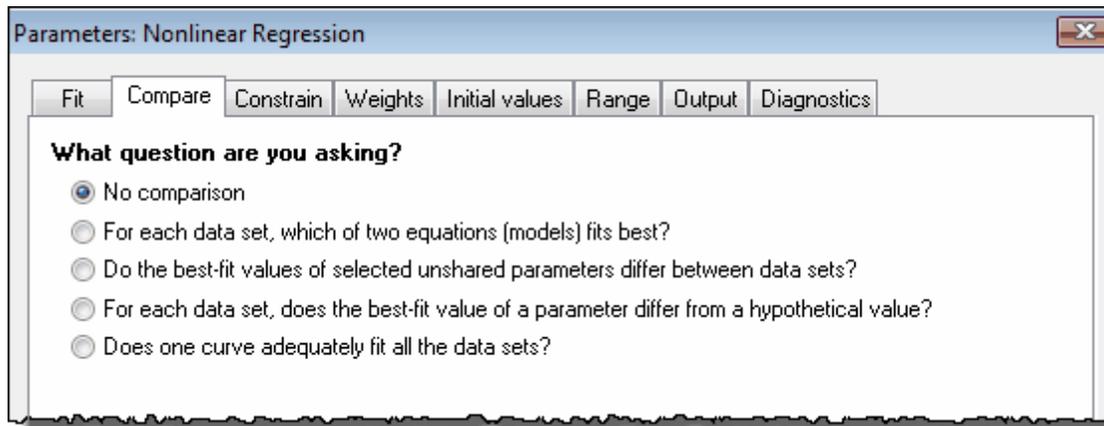
Interpolate

Check this option in order to interpolate the concentration of unknown samples from the best-fit curve. [Learn more.](#)^[107]



With this option, Prism will report the Y value for any X values you enter, and the X value for any Y values you enter (including extrapolating in each direction a distance equal to half the length of the X axis).

2.5.3.2 Compare tab



When fitting biological data with regression, your main objective is often to discriminate between different models, to ask if an experimental intervention changed a parameter, or to ask if the best-fit value of a parameter differs significantly from a theoretical value. Learn more about these [four kinds of comparisons](#)^[46]. Your choice, of course, has to be based on your experimental goals.

Prism can perform the comparison [using two alternative methods](#)^[48]: the [extra sum-of-squares F test](#)^[50], and using [Akaike's information criteria](#)^[51]. Use these guidelines to choose:

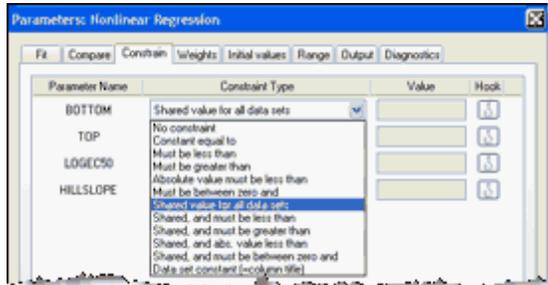
- In most cases, the two models will be 'nested'. This means that one model is a simpler case of the other. For example, a one-phase exponential model is a simpler case of a two-phase exponential model. Either the F test or the AICc method may be used with nested models. The choice is usually a matter of personal preference and tradition. Basic scientists in pharmacology and physiology tend to use the F test. Scientists in fields like ecology and population biology tend to use AICc.
- If the models are not nested, then the F test is not valid so you should choose AICc. Note that Prism does not enforce this. It will calculate the F test even if the models are not nested, but the results won't be useful.

Both these methods only make sense when the models being compared have different numbers of parameters, and so have different numbers of degrees of freedom. If you want to compare two models with the same number of parameters, there is no need to use either the F test or AIC. Simply choose the model that fits the data the best with the smallest sum-of-squares.

2.5.3.3 Constrain tab

The importance of constraints

Prism lets you constrain each parameter to a constant value, constrain to a range of values, share among data sets (global fit), or define a parameter to be a column constant. This is an important decision, and will influence the results.



Constrain to a constant value

You won't always want to fit all the parameters in a model. Instead, you can fix one or more parameters to constant values. For example, if you have normalized a dose-response curve to run from 0 to 100, constrain Top to equal 100 and Bottom to 0.0. Similarly, if you have subtracted a baseline so you know that the exponential decay curve has to plateau at $Y=0.0$, you can constrain the Bottom parameter to equal 0.0.

Deciding to constrain a parameter to a constant value can make a huge difference in the results. Remember Prism has no common sense. It doesn't know if a curve has to plateau at zero unless you tell it.

Constrain to a range of values

Constrain to a range of values to prevent Prism from letting parameters take on impossible values. For example, you should constrain rate constants to only have values greater than 0.0, and fractions (say the fraction of binding sites that are high affinity) that have a value between 0.0 and 1.0. Setting this kind of constraint can have four outcomes:

- The constraint is irrelevant, as the parameter never would have taken on a value in the forbidden range.
- The constraint helped speed up the fit. Nonlinear regression works by iteratively changing the values of the parameters. With some complicated fits, the nonlinear regression process can 'get confused' and end up spending time exploring parameter values that make no sense. Constraining

the values of one or more parameters can prevent the nonlinear regression process from being led astray. With huge numbers of data points, you might see a noticeable speeding up of the fitting process.

- The constraint helped nonlinear regression choose from several local minima. Nonlinear regression works by changing parameter values step by step until no small change affects the sum-of-squares (which quantifies goodness-of-fit). With some models, there can be two sets of parameter values that lead to local minima in sum-of-squares. Applying a constraint can ensure that nonlinear regression finds the minimum with scientifically relevant values, and ignores another minimum that fits the curve well but using parameter values that make no scientific sense (i.e. negative concentrations).
- The constraint prevents nonlinear regression from finding a minimum sum-of-squares. Instead, the best the program can do (while obeying the constraint) is set the parameter to the limit of the constrained range. Prism then reports that the fit 'hit constraint'.

In the first case, the constraint is harmless but useless.

In the next two cases, the constraint helps the nonlinear regression reach sensible results. Essentially, the constraint can give the nonlinear regression process some scientific judgment about which parameter values are simply impossible. These cases are really what constraints are for.

The last case, when the fit ends with a parameter set to one end of its constraint, it is [tricky to interpret the results](#).^[231]

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](#).^[43]
- Each dataset is incomplete, but the entire family of datasets defines the parameters. [See example](#).^[42]

Data set constant

When you fit a family of curves at once, you can set one of the parameters to be a data set constant. Prism offers two data set constants.

Column title. The value comes from the column title, which can be different for every data set. This parameter becomes almost a second independent variable. It has a constant value within any one data set, but a different value for each data set. For example, when fitting a family of enzyme progress curves in the presence of various concentrations of inhibitor, the inhibitor concentration can be entered into the column title of the data table. [View an example](#)⁴²⁴.

Mean X. The value is the mean of all the X values for which there are Y values in that data set. This is used in centered polynomial regression.

2.5.3.4 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

It is often useful to differentially weight the data points. [Learn why.](#)^[30]

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

No weighting. Regression is most often done by minimizing the sum-of-squares of the vertical distances of the data from the line or curve. Points further from the curve contribute more to the sum-of-squares. Points close to the curve contribute little. This makes sense, when you expect experimental scatter to be the same, on average, in all parts of the curve.

Weight by $1/Y^2$. In many experimental situations, you expect the average distance (or rather the average absolute value of the distance) of the points from the curve to be higher when Y is higher. The points with the larger scatter will have much larger sum-of-squares and thus dominate the calculations. If you expect the relative distance (residual divided by the height of the curve) to be consistent, then you should weight by $1/Y^2$.

Weight by $1/Y$. This choice is useful when the scatter follows a Poisson distribution -- when Y represents the number of objects in a defined space or the number of events in a defined interval.

Weight by $1/Y^k$. Also called "General weighting". [Read more.](#)^[31]

Weight by $1/X$ or $1/X^2$. These choices are used rarely. Only choose these weighting schemes when it is the standard in your field, such as a linear fit of a bioassay.

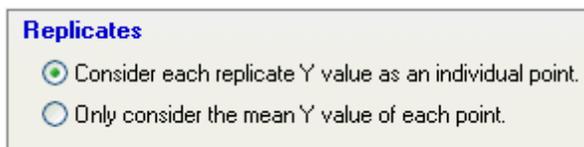
Weight by $1/SD^2$. If you enter replicate Y values at each X (say triplicates), it is tempting to weight points by the scatter of the replicates, giving a point less weight when the triplicates are far apart so the standard deviation (SD) is high. But unless you have lots of replicates, this doesn't help much. The triplicates constituting one mean could be far apart by chance, yet that mean may be as accurate as the others. Weighting needs to be based on *systematic* changes in scatter. The choice to weight by $1/SD^2$ is most useful when you want to use a weighting scheme not available in Prism. In this case, enter data as mean and SD, but enter as "SD" weighting values that you computed elsewhere for that point. In other words, the values you enter in the SD subcolumn are not actually standard deviations, but are weighting factors computed elsewhere.

Notes on weighting

- If you have normalized your data, [weighting rarely makes sense.](#)

- [Simulations can show](#) you how much difference it makes if you choose the wrong weighting scheme.
- If you choose unequal weighting, [Prism takes this into account](#)^[179] when plotting residuals.
- Prism accounts for weighting [when it computes R](#)^[187]^[187].
- Learn about the [math of weighting](#)^[31] and how [Prism does the weighting](#)^[65].

Replicates



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..

2.5.3.5 Initial values tab

Why initial values?

Nonlinear regression is an iterative procedure. The program must start with estimated initial values for each parameter. It then adjusts these initial values to improve the fit.

How does Prism provide initial values automatically?

Every equation built-in to Prism, as well as equations you define, include rules to

compute initial values. These rules use the range of the X and Y values to come up with initial values, which become the original automatic initial values. You can [change the rules](#)⁴³³ 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.

2.5.3.6 Range tab

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.

2.5.3.7 Output tab

Summary table

When analyzing several data sets, the results table is rather lengthy. To display key results on a summary table, check the option box to create a summary table and select the variable you wish to summarize. Prism creates a summary table (as an additional results view) that shows the best-fit value of that parameter for each data set, and graphs this table.

Depending on your choices in the dialog, this may be a bar graph or an XY graph. It shows the best-fit value of a selected parameter for each data set on the table. In some cases, you may analyze the summary table with linear or nonlinear regression. For example, the summary graph may show the best-fit value of a rate constant as a function of concentration (obtained from the column titles of the original data). You can fit a line or curve to that graph.

When Prism compares the fits of two equations, it shows only the results for the second equation. Since this may not be helpful, we suggest that you only

make summary tables when fitting a single equation.

Number of digits in output

Choose how many significant digits you want to see in the main results table. This is especially useful if you embed the results table on a graph or layout.

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 K_i from IC_{50}). These are available only to be compatible with earlier versions of Prism (so Prism can open files made with older versions). They are almost always unavailable, becoming available only if you select certain equations from the classic equations list.

Since Prism 5, you can calculate a [Gaddum/Schild \$EC_{50}\$ shift](#)^[277] directly, without need to separately compute dose ratios. Similarly, you can fit competitive binding curves directly to determine the K_i for [one](#)^[312] or [two](#)^[317] sites, without a separate Cheng-Prusoff calculation of K_i from $IC_{50}(1)$.

1. Cheng, Y. and Prusoff, W. H. Relationship between the inhibition constant (K_1) and the concentration of inhibitor which causes 50 per cent inhibition (I_{50}) of an enzymatic reaction. *Biochem Pharmacol*, 22: 3099-3108, 1973.

2.5.3.8 Diagnostics tab

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:

Nonlinear regression works iteratively, and begins with [initial values](#)^[162] 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". This is

often the best way to diagnose problems with nonlinear regression.

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% Output Format: Range ("1.23 to 4.56")

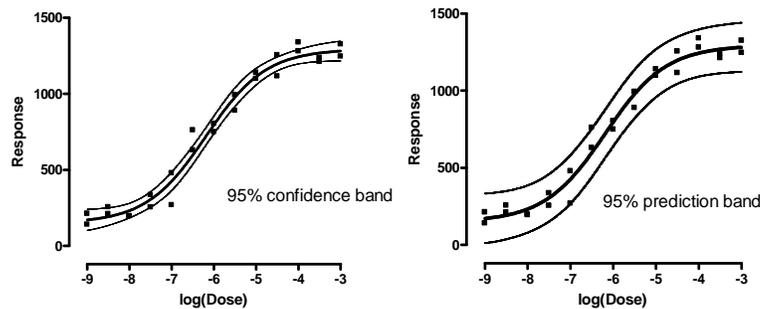
Plot 95% confidence band

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.

[Learn more about confidence and prediction bands.](#)^[200]

How to quantify goodness-of-fit

How to quantify goodness-of-fit?

R square Adjusted R squared Sum-of-Squares Sy.x

You will probably want to ask Prism to report [R²](#)^[187], simply because it is standard to do so, even though knowing R² 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.

The [adjusted R²](#)^[190] takes into account the number of parameters fit to the data, so has a lower value than R² (unless you fit only one parameter, in which case R² and adjusted R² are identical).

Prism also lets you choose to report the [AICc](#)^[51]. This would be useful only if you separately fit the same data to three or more models. You can then use the AICc to choose between them. But note that it only makes sense to compare AICc between fits, when the only difference is the model you chose. If the data or weighting are not identical between fits, then the comparison of AICc values would be meaningless.

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](#)^[185] 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](#)^[194] 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](#)^[193] 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](#)^[179], 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 or skewed?

Are the parameters intertwined, redundant or skewed?

Covariance of parameters Dependency Hougaard's measure of skewness

What does it mean for parameters to be intertwined? After fitting a model, change the value of one parameter but leave the others alone. The curve moves away from the points. Now, try to bring the curve back so it is close to the points by changing the other parameter(s). If you can bring the curve closer to the points, the parameters are *intertwined*. If you can bring the curve back to its original position, then the parameters are *redundant*. In this case, Prism will alert you by labeling the fit '[ambiguous](#)'^[228].

We suggest that you report the [dependency](#)^[197], and not bother with the [covariance matrix](#)^[197]. When you are getting started with curve fitting, it is OK to leave both options unchecked.

Even though nonlinear regression, as its name implies, is designed to fit nonlinear models, some of the inferences actually assume that some aspects of the model are close to linear, so that the distribution of each parameter is symmetrical. This means that if you analyzed many data sets sampled from the same system, the distribution of the best-fit values of the parameter would be symmetrical and Gaussian.

If the distribution of a parameter is highly skewed, the reported SE and CI for that parameter will not be very useful ways of assessing precision. [Hougaard's measure of skewness](#)^[204] quantifies how skewed each parameter is.

Could outliers impact the results?

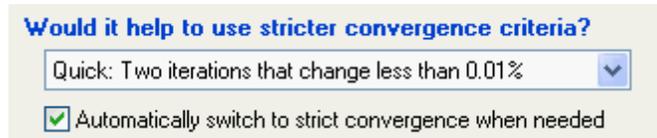


Nonlinear regression is based on the assumption that the scatter of data around the ideal curve follows a Gaussian distribution. The presence of one or a few outliers (points much further from the curve than the rest) can overwhelm the least-squares calculations and lead to misleading results.

Check this option to count the outliers, but leave them in the calculations. Choose how aggressively to define outliers by [adjusting the ROUT coefficient](#)^[160].

If you chose the option in the [Fit tab](#)^[155] 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?



Nonlinear regression is an iterative process. It starts with [initial values](#)¹⁶² 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 0.0000001%. 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%.

2.5.4 Graphing tips: Nonlinear regression

Prism automatically graphs the best-fit curves from nonlinear regression, and you'll rarely need to think about this step.

- [Tips](#)¹⁷¹ on graphing curves
- [Graphing prediction and confidence bands](#)¹⁷¹
- [Adding an equation](#)¹⁷⁵, or a table of best-fit parameters, to the graph.
- [Graphing outliers](#)¹⁷⁶.
- [Graphing residuals](#)¹⁷⁹.

2.5.4.1 Graphing best-fit curves

Automatic vs. manual plotting

When Prism performs nonlinear regression, it automatically superimposes the curves on the graph.

If you need to create additional graphs, or change which curve is plotted on which graph, keep in mind that from Prism's point of view, a curve generated by nonlinear regression is simply a data set. You can add curves to a graph or remove curves from a graph on the 'Data on graph' tab of the Format Graph dialog.

Plotting curves after comparing models

If you ask Prism to compare two models, only the selected model is graphed. If you want to graph both models, you will need to analyze the data twice, once with each model.

Plotting on logarithmic axes

With Prism 5 and 6, it is Ok to stretch an axis to a log scale, a choice on top of the Format Axis dialog. With prior versions, this didn't work well. The curve was defined as a series of line segments, and these would be far from equally spaced on a log axis. Prism 5 and 6 are smarter, so the curve will look smooth on either a linear or logarithmic axis.

Plotting only a portion of the curve

Let's say that the X values of your data range from 0 to 100, and you fit a curve with nonlinear regression. Now you want to create a graph showing only the data with X values between 0 and 10. The curve will look ragged. The problem is that Prism defines the entire curve using 150 line segments. In this example, you only plot 10% of the range, so the curve will have only 15 line segments.

To fix this problem, go to the Range tab of nonlinear regression and choose to define the curve with more line segments. In this example, increasing the value tenfold, to 1500, would be about right.

2.5.4.2 Graphing confidence and prediction bands

The difference between confidence and prediction bands

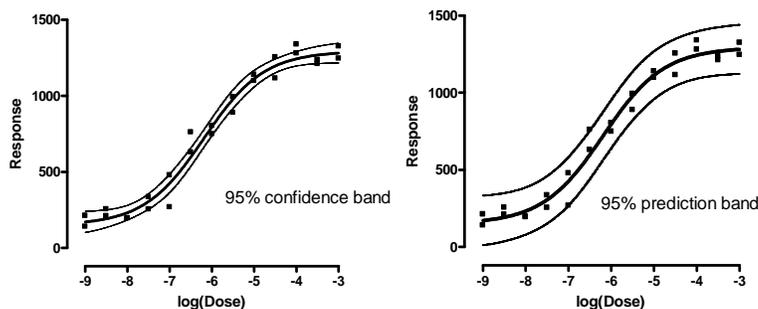
The 95% **confidence bands** enclose the area that you can be 95% sure

contains the true curve. It gives you a visual sense of how well your data define the best-fit curve.

The 95% **prediction bands** enclose the area that you expect to enclose 95% of future data points. This includes both the uncertainty in the true position of the curve (enclosed by the confidence bands), and also accounts for scatter of data around the curve.

Choose in the [Diagnostics tab](#) ¹⁶⁵ of the Nonlinear regression parameters dialog. Learn [how they are calculated](#) ⁶⁷.

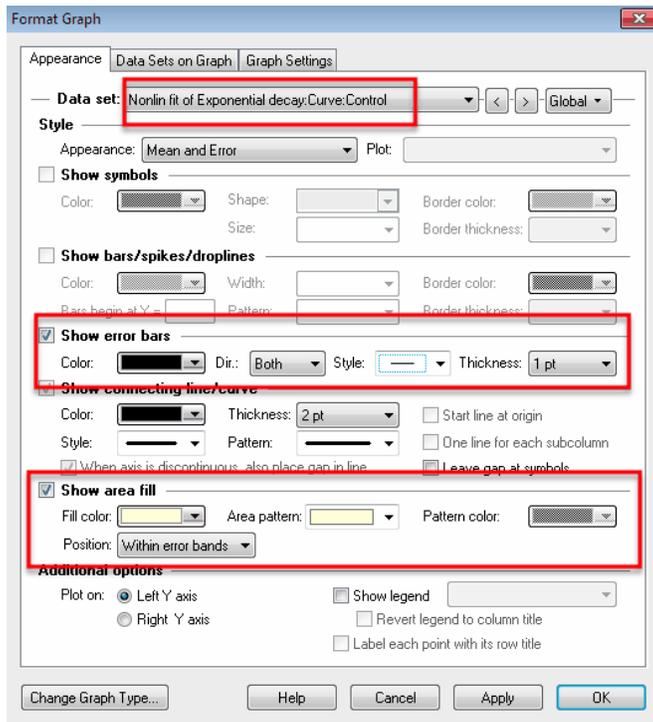
Prediction bands are always wider than confidence bands. When you have lots of data points, the discrepancy is huge.



Plotting confidence or prediction bands

After you choose to create a confidence or prediction band in the diagnostics tab, it should just appear on the graph. You can fine tune its appearance on the graph. Note the following:

- The fit curve is seen by the Format Graph dialog as one more data set to plot. be sure to select one of those nonlinear fits first (top of dialog) before adjusting its properties.
- To plot the confidence bands, choose to show error bars, and set the style to ----- or _____.
- You can also choose to show an area fill within the error bands.

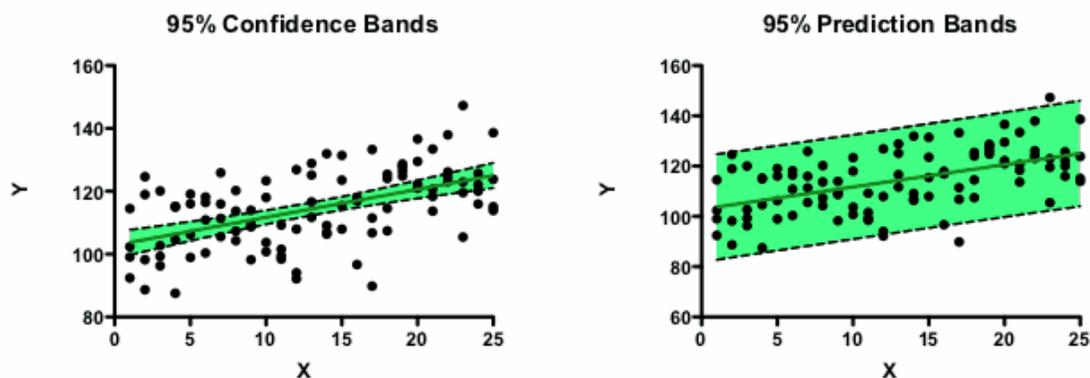


Why do so many points lie outside the confidence bands?

Confidence bands show you how well you know the location of the best fit line or curve. Given all the assumptions of the analysis, you can be 95% sure that the true curve (nonlinear regression) or line (linear regression) lies within the bands.

Prediction bands show you where you can expect the data to lie. You expect 95% of all data points to lie within the prediction bands.

With many data points, you expect a large fraction of the data points to lie outside the confidence band, but 95% to lie within the prediction bands. The confidence bands aren't supposed to show you the scatter of the data, but rather the uncertainty in the position of the line or curve. With lots of data, the line or curve is known fairly precisely, so only a small fraction of data lie within the bands. The figure below demonstrates this with linear regression.



Plotting both confidence and prediction bands

Prism lets you choose either a confidence band or a prediction band as part on the [diagnostics tab](#)¹⁶⁵. But not both. To plot both on one graph, analyze your data twice, choosing a confidence band the first time and a prediction band the second time. The curves are data sets that you can add to any graph by drag-n-drop or using the Change menu from the graph.

One-sided confidence (or prediction) bands

Confidence and prediction bands are usually two-sided. But you can get Prism to show the band on only one side.

One use for a one-sided confidence band would be on a graph that plots purity of a substance over time. You want to know, with a certain confidence, how low the purity could be at a certain time. You don't really care about quantifying how high the purity could be.

To plot a one-sided 95% confidence (or prediction) band:

1. Choose "90%" confidence bands in the Diagnostics tab of the nonlinear regression dialog. With 90% bands, there is a 5% chance that the true value is larger than the upper band, and 5% it is lower than the lower band. When you plot only one of these bands, therefore, it is really a 95% confidence band, which is probably what you want.
2. On the Format Symbols dialog, choose the best-fit line or curve and make sure that error bars are turned on with the "---" style. Then choose either Up or Down (but not Both) in the Direction drop-down menu.

Viewing the XY coordinates of confidence or prediction bands

1. Go to the Diagnostics tab and check the option to plot confidence or prediction bands.
2. Go to the Range tab of the nonlinear regression dialog.
3. Check the option to create a table of XY coordinates. This table will include the confidence or prediction bands.
4. When viewing the results, expand the results sheet in the navigator and go to the page labeled 'Curve'. For each X value, you'll see the y value and the distance the confidence or prediction curve is above and below that value.

Why won't Prism plot confidence or prediction bands?

Prism will not plot confidence or prediction bands in several situations:

- If the best-fit value of a parameter [hit a constraint](#)^[231], the fit is unlikely to be useful. Prism does not plot confidence or prediction bands, because they would almost certainly be misleading.
- If the results of nonlinear regression are [ambiguous](#)^[228], the confidence or prediction bands would be super wide, maybe infinitely wide. They would not be useful, so Prism does not plot them.
- If you choose [robust](#)^[59] nonlinear regression, Prism does not compute confidence or prediction bands, as it cannot compute standard errors or confidence intervals of the parameters.
- The [fit is perfect](#)^[233]. If the sum-of-squares is 0.0 and R2 is 1.0, it is not possible to compute or interpret confidence or prediction bands.
- If the fit is [interrupted](#)^[226], confidence and prediction bands are not computed.

2.5.4.3 Adding the equation to the graph

Prism does not automatically write equations on the graph. There are two approaches you can use to do so.

Microsoft Office comes with an equation editor. If Microsoft Equation Editor is installed on your system, you will see a quick-access button on your toolbar when you are on either a graph or a layout sheet.

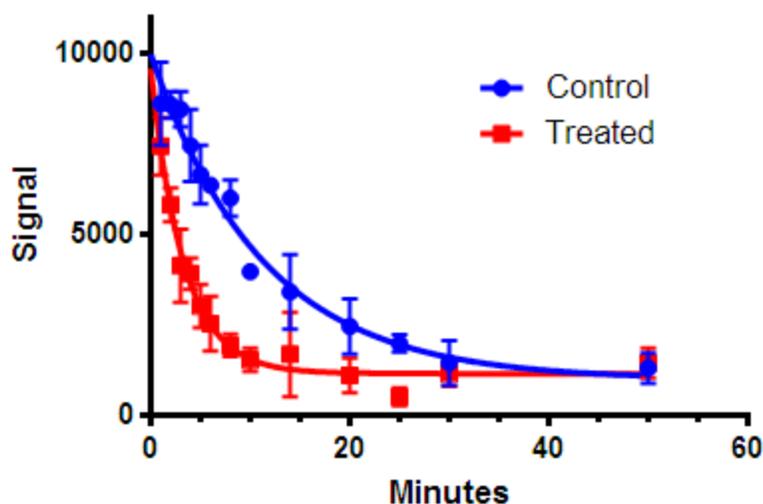


You can create either a generic equation, or an equation with the best-fit values of the parameters substituted for the variable names.

If the button doesn't appear on your toolbar, create the equation inside of Microsoft Word or some other program, and then copy and paste.

Another approach is to simply show the parameter values from the curve of best fit in tabular form. Copy selected results and then paste onto a graph. The table will be linked, so its values will change if you edit or replace the data. This doesn't show the form of the equation (exponential decay, in this example) but gives you the results of this particular experiment, which may be more important.

The example below shows use of the equation editor to write both a generic and specific equation (for the control data set) and an embedded table showing the results for both data sets.



$$Y = Plateau + (Y0 - Plateau)e^{-k \cdot X}$$

$$Y = 987 + (9992 - 987)e^{-0.08927 \cdot X}$$

	Control	Treated
Y0	9992	9593
Plateau	987.0	1154
K	0.08927	0.3042
Half Life	7.765	2.278
Tau	11.20	3.287
Span	9005	8440

2.5.4.4 Graphing outliers

How to identify or eliminate outliers

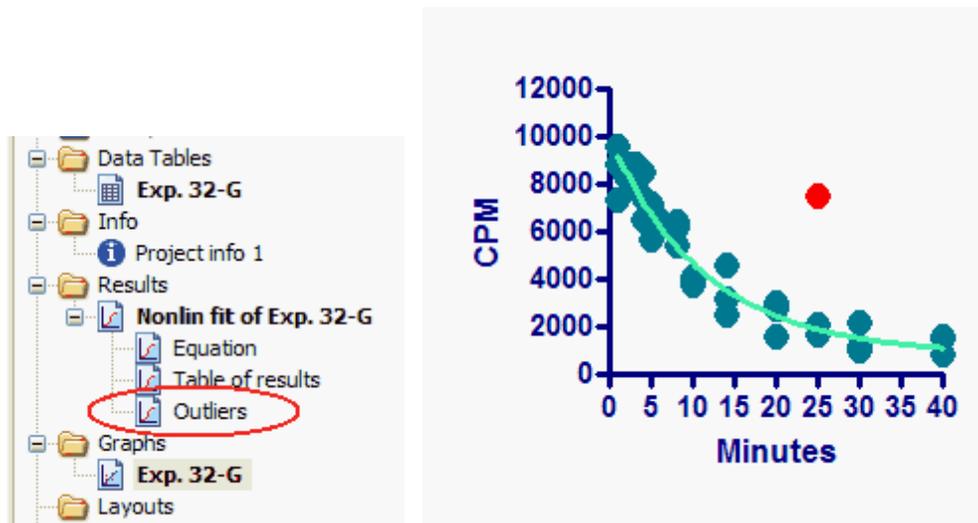
Prism can identify outliers from the curve fit, and you can choose whether

these should simply be plotted or also ignored by the curve fitting process. Choose to count the outliers on the [Diagnostics tab](#)^[165], and to eliminate outliers on the [Fit tab](#)^[59] of the nonlinear regression dialog.

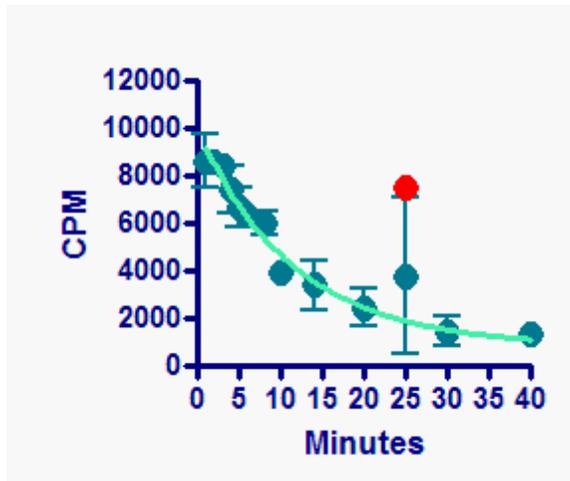
You can [adjust the value of Q](#)^[160] to redefine how aggressively Prism defines outliers.

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).

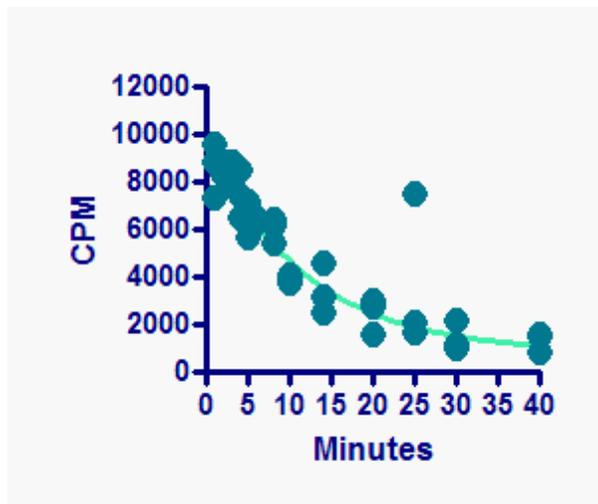
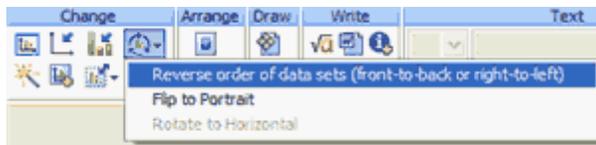


The example above shows a graph where each replicate is plotted individually. The example below shows a graph which plots mean and SD error bars. The error bars are computed from **all the data**, including any outliers. The outliers are then superimposed on the graph as well.



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).

2.5.4.5 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

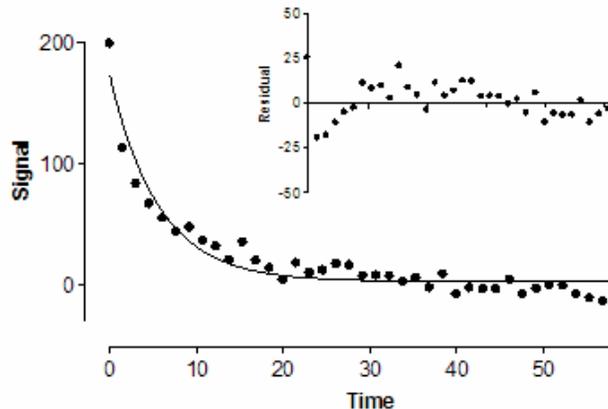
Choose to create a residual plot by checking an option on the Diagnostics tab of the nonlinear regression dialog. Prism will automatically make a new graph. The residuals are tabulated in a separate page of results, and you can use these like any other table (make additional graphs, plot on other graphs,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



Residuals with weighted fits

If you choose (or accept the default) standard weighting, then the residuals are the difference between the actual Y value you entered and the Y value predicted by the model. If the data point is above the curve, the residual is positive. If the data point is below the curve, the residual is negative. Least-squares regression works to minimize the sum of the squares of these residuals.

If you choose another weighting scheme, Prism adjusts the definition of the residuals accordingly. The residual that Prism tabulates and plots equals the residual defined in the prior paragraph, divided by the weighting factor. The most common common alternative weighting is "Weight by $1/Y^2$ (minimize relative distances squared)". In this case, the residual is defined to be the distance of the point from the curve divided by the Y value of the curve. Weighted nonlinear regression minimizes the sum of the square of these residuals.

Note the ambiguity in defining weighting. The Prism dialog gives the choice to weight by $1/Y^2$. This means that the squared residual is divided by Y^2 . The weighted residual is defined as the residual divided by Y. Prism minimizes the sum of the squares of these weighted residuals.

Earlier versions of Prism (up to Prism 4) always plotted basic unweighted residuals, even if you chose to weight the points unequally.

2.6 Interpreting nonlinear regression results

Interpreting nonlinear regression results can be confusing.

- If Prism reported error message on the top of the results column, [here are the explanations](#)^[225].
- [Help with graphing](#)^[170] best-fit curves, with their confidence or prediction bands and residual plots.
- Read detailed explanations of the results, and review analysis checklists. Where to go depends on your goal.

Goal: Determine best-fit values	Analysis checklist ^[218]	Learn more. ^[181]
Goal: Compare models	Analysis checklist ^[221]	Learn more. ^[212]
Goal: Interpolate unknowns from a standard curve	Analysis checklist ^[223]	Learn more. ^[104]

2.6.1 Interpreting results: Nonlinear regression

When evaluating nonlinear regression results, first consider what your goal is. If your goal is to interpolate unknowns from a standard curve, skip this page and go right to this [analysis checklist](#)^[118].

If your goal is to determine the values of the best-fit parameters or to compare models, answer the three questions below before looking at parameter values, R^2 , etc.

- Are your results free of error messages? Prism sometimes puts a short message at the top of the table of results. Read about the meaning of "[Bad](#)

[initial values](#)²²⁶, ["Interrupted"](#)²²⁶, ["Not converged"](#)²²⁷,
["Ambiguous"](#)²²⁸, ["Hit constraint"](#)²³¹, ["Don't fit"](#)²³²,
["Too few points"](#)²³², ["Perfect fit"](#)²³³, ["Impossible weights"](#)²³³, and ["Equation not defined"](#)²³³.

- Does the curve go near your data points? In rare cases, the curve may be far from the data points. This may happen, for example, if you picked the wrong equation. Look at the graph to make sure this didn't happen.
- Are the best-fit values of the parameters scientifically sensible? Nonlinear regression programs have no common sense and don't know the context of your experiment. The curve fitting procedure can sometimes yield results that make no scientific sense. For example with noisy or incomplete data, nonlinear regression can report a best-fit rate constant that is negative, a best-fit fraction that is greater than 1.0, or a best-fit Kd value that is negative. All these results are scientifically meaningless. Also check whether the best-fit values of the variables make sense in light of the range of the data. The results make no sense if the top plateau of a sigmoid curve is far larger than the highest data point, or an EC50 is not within the range of your X values. If the results make no scientific sense, they are unacceptable, even if the curve comes close to the points and R² is close to 1.0.

If the answer to all three questions is yes, then it makes sense to delve into the numerical results of nonlinear regression in detail.

- [Standard errors and confidence intervals of parameters](#)¹⁸³
- [Normality tests of residuals](#)¹⁸⁵
- [Goodness of fit of nonlinear regression](#)¹⁸⁷
- [Runs test](#)¹⁹³

- [Replicates test](#) ¹⁹⁴
- [Dependency and covariance matrix](#) ¹⁹⁷
- [Hougaard's measure of skewness](#) ²⁰⁴
- [Could the fit be a local minimum?](#) ²⁰⁶
- [Outliers](#) ²⁰⁷

2.6.1.1 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](#) ¹⁶⁵). The [calculation of the standard errors](#) ¹⁶⁷ 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](#)^[67] 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 Monte Carlo analyses with simulated data.

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](#)'^[228] 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](#)^[438] 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](#)^[438] between two ways to compute the confidence interval of each transformed parameter. If you pick a built-in equation, Prism always reports asymmetrical confidence intervals of transformed parameters.

Do not mix up confidence intervals and confidence bands

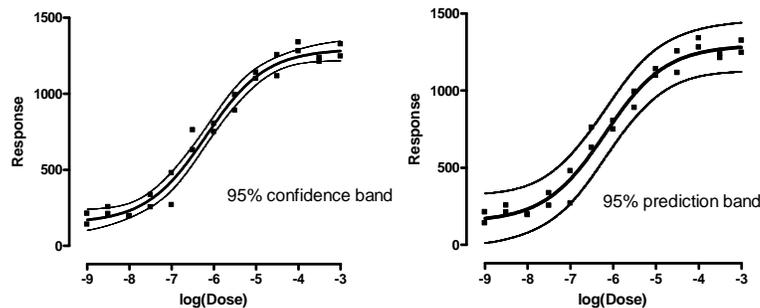
It is easy to mix up confidence intervals and confidence bands. Choose both on the [Diagnostics tab](#) ¹⁶⁵.

The **95% confidence interval** tells you how precisely Prism has found the best-fit value of a particular parameter. It is a range of values, centered on the best-fit value. Prism can display this range in two formats:

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

10	Lower 95% conf. limit	
11	SPAN	8849
12	K	0.05566
13	HalfLife	9.171
14	Upper 95% conf. limit	
15	SPAN	9959
16	K	0.07558
17	HalfLife	12.45

The 95% **confidence bands** enclose the area that you can be 95% sure contains the true curve. It gives you a visual sense of how well your data define the best-fit curve. It is closely related to the 95% **prediction bands**, which enclose the area that you expect to enclose 95% of future data points. This includes both the uncertainty in the true position of the curve (enclosed by the confidence bands), and also accounts for scatter of data around the curve. Therefore, prediction bands are always wider than confidence bands.



2.6.1.2 Normality tests of residuals

Interpreting a normality test

The result of a normality test is expressed as a P value that answers this question:

If your model is correct and all scatter around the model follows a Gaussian population, what is the probability of obtaining data whose residuals deviate from a Gaussian distribution as much (or more so) as your data does?

If the P value is large, then the residuals pass the normality test. If the P value is small, the residuals fail the normality test and you have evidence that your data don't follow one of the assumptions of the regression. Things to consider:

- Fit a different model
- [Weight](#)^[160] the data differently.
- [Exclude outliers](#)^[59].

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 statement¹: "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 D'Agostino and MA Stepenes, Macel Decker, 1986.

2.6.1.3 R squared

Meaning of R²

Key points about R²

- The value R² 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 R² 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 R²=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 R² 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.
- There is really no general rule of thumb about what values of R² are high, adequate or low. If you repeat an experiment many times, you will know what values of R² to expect, and can investigate further when R² is much lower than the expected value.
- By tradition, statisticians use uppercase (R²) for the results of nonlinear and multiple regression and lowercase (r²) for the results of linear regression, but

this is a distinction without a difference.

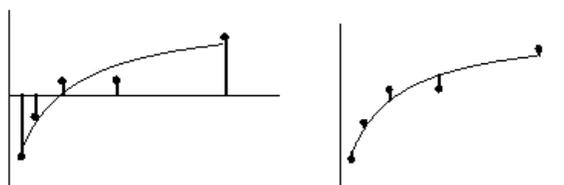
Don't overemphasize R²

A common mistake is to use R² as the main criteria for whether a fit is reasonable. A high R² tells you that the curve came very close to the points. That doesn't mean the fit is "good" in other ways. The best-fit values of the parameters may have values that make no sense (for example, negative rate constants) or the confidence intervals may be very wide. The fit may be ambiguous. You need to look at all the results to evaluate a fit, not just the R².

How R² is calculated -- unweighted fits

R² 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 R² into a fraction, the results are normalized to the sum of the square of the distances of the points from a horizontal line through the mean of all Y values. This value is called SStot. If the curve fits the data well, SSres will be much smaller than SStot.

The figure below illustrates the calculation of R². Both panels show the same data and best-fit curve. The left panel also shows a horizontal line at the mean of all Y values, and vertical lines showing how far each point is from the mean of all Y values. The sum of the square of these distances (SStot) equals 62735. The right panel shows the vertical distance of each point from the best-fit curve. The sum of squares of these distances (SSres) equals 4165.



R² is calculated using this equation:

$$R^2 = 1.0 - (SSres/SS_{tot}) = 1.0 - 4165/62735 = 0.9336$$

How R² is calculated -- weighted fits

Prism (since 5.02 and 5.0b) calculates the R² of a weighted fit as detailed below:

1. Fit the model using relative (1/Y²) or Poisson (1/Y) or general (1/X^K)

weighting to compute the weighted sum-of-squares (wSS_{model}). This is the weighted sum-of-squares of the residuals from your model.

2. Fit the data to a horizontal line model ($Y = \text{Mean} + 0 \cdot X$) using the same weighting to compute the weighted sum-of-squares ($wSS_{\text{horizontal}}$). This is the weighted sum-of-squares of the residuals from the null hypothesis model, a horizontal line.

The weighted R^2 is:

$$1.0 - (wSS_{\text{model}}/wSS_{\text{horizontal}})$$

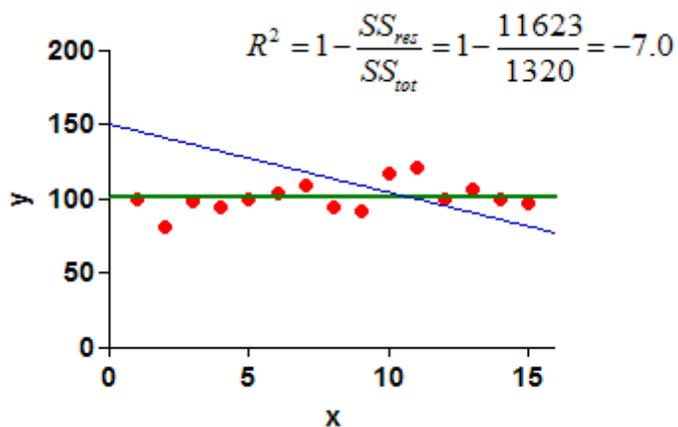
R^2 can be negative!

Appearances can be deceptive. R^2 is not really the square of anything. If SS_{res} is larger than SS_{tot} , R^2 will be negative (see equation above). 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).

How can this happen? SS_{res} is the sum of the squares of the vertical distances of the points from the best-fit curve (or line). SS_{tot} is the sum of the squares of the vertical distances of the points from a horizontal line drawn at the mean Y value. SS_{res} will exceed SS_{tot} when the best-fit line or curve fits the data even worse than does a horizontal line.

R^2 will be negative when the best-fit line or curve does an awful job of fitting the data. This can only happen when you fit a poorly chosen model (perhaps by mistake), or you apply constraints to the model that don't make any sense (perhaps you entered a positive number when you intended to enter a negative number). For example, if you constrain the Hill slope of a dose-response curve to be greater than 1.0, but the curve actually goes downhill (so the Hill slope is negative), you might end up with a negative R^2 value and nonsense values for the parameters.

Below is a simple example. The blue line is the fit of a straight line constrained to intercept the Y axis at $Y=150$ when $X=0$. SS_{res} is the sum of the squares of the distances of the red points from this blue line. SS_{tot} is the sum of the squares of the distances of the red points from the green horizontal line. Since SS_{res} is much larger than SS_{tot} , the R^2 (for the fit of the blue line) is negative.



[Prism file](#)

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

Adjusted R^2

The R^2 quantifies how well a model fits the data. When you compare models, the one with more parameters can bend and twist more to come nearer the points, and so almost always has a higher R^2 . This is a bit misleading.

The adjusted R^2 accounts for the number of parameters fit by the regression, and so can be compared between models with different numbers of parameters. The equations for the regular and adjusted R^2 are compared here (SSresiduals is the sum-of-squares of the discrepancy between the Y value of the curve and the data; SStotal is the sum-of-squares of the differences between the overall Y mean and each Y value; n is the number of data points, and K is the number of parameters fit):

$$R^2 = 1 - \frac{SS_{residuals}}{SS_{total}}$$

$$\text{Adjusted } R^2 = 1 - \frac{SS_{residuals} / (n - K)}{SS_{total} / (n - 1)}$$

The adjusted R^2 is smaller than the ordinary R^2 whenever K is greater than 1.

2.6.1.4 Sum-of-squares

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 $s_{y.x}$ is the standard deviation of the residuals, expressed in the same units as Y .

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

$$s_{y.x} = \sqrt{\frac{SS}{df}}$$

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).

2.6.1.5 Why Prism doesn't report the chi-square of the fit

Some programs report a chi-square value with the results of nonlinear regression. Prism doesn't. This page explains how the chi-square value could be useful and why we don't report it.

Is the sum-of-squares too high?

Nonlinear regression minimizes the sum of the squared vertical distances between the data point and the curve. But how can you interpret the sum-of-squares, abbreviated SS? You can't really, as it depends on the number of data points you collected and the units you used to express Y.

The chi-square approach is to compare the observed scatter of the points around the curve (SS) with the amount of experimental scatter you expect to see based theory. This is done by computing chi-square using this equation:

$$\chi^2 = \sum_{i=1}^{i=n} \left(\frac{Y_i - Y_{curve}}{\sigma_i} \right)^2$$

Chi-square is the sum of the square of the ratio of the distance of a point from the curve divided by the predicted standard deviation at that value of X. Note that the denominator is the predicted standard deviation, not the actual standard deviation computed in this particular experiment.

If you know that the SD is the same for all values of X, this simplifies to:

$$\chi^2 = \frac{\sum_{i=1}^{i=n} (Y_i - Y_{curve})^2}{SD^2}$$

The standard deviation value must be computed from lots of data so the SD is very accurate. Or, better, the SD can come from theory.

If you assume that replicates are scattered according to a Gaussian distribution with the SD you entered, and that you fit the data to the correct model, then the value of chi-square computed from that equation will follow a known chi-square distribution. This distribution depends on the number of degrees of freedom, which equals the number of data points minus the number of parameters. Knowing the value of chi-square and the number of degrees of freedom, a P value can be computed.

How can you interpret a small P value? If you are quite sure the scatter really is Gaussian, and that predicted SD is correct, a small P value tells you that your model is not right -- that the curve really doesn't follow the data very well. You should seek a better model.

But often a low P value just tells you that you don't predict the SD as well as you thought you would. It is hard to determine the SD values precisely, so hard

to interpret the chi-square value. For this reason, Prism doesn't attempt the chi-square computation. We fear it would be more misleading than helpful.

Alternatives

Several approaches have been devised to answer the question of whether the SS is too high:

- The value of sum-of-squares can be used to compute R^2 . This value is computed by comparing the sum-of-squares (a measure of scatter of points around the curve) with the total variation in Y values (ignoring X, ignoring the model). What values of R^2 do you expect? How low a value is too low? You can't really answer that in general, as the answer depends on your experimental system.
- If you have collected replicate Y values at each value of X, you can compare the SS with a value predicted from the scatter among replicates. Prism calls this the [replicates test](#)^[194]. This is very useful, but only if you have collected replicate Y measurements at each X.
- You can propose alternative models, and compare their fit to the data

Summary

Chi-square compares the actual discrepancies between the data and the curve with the expected discrepancies (assuming you selected the right model) based on the known SD among replicates. If the discrepancy is high, then you have some evidence that you've picked the wrong model. The advantage of the chi-square calculation is that it tests the appropriateness of a single model, without having to propose an alternative model and without having to have replicate values. The disadvantage is that the calculation depends on knowing the SD values with sufficient precision, which is often not the case.

Recommended alternatives are to [compare the fits of two models](#)^[46], or use the [replicates test](#)^[194].

2.6.1.6 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](#)^[194] 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 N_a points above the curve and N_b points below the curve, the number of runs you expect to see equals $[(2N_aN_b)/(N_a+N_b)]+1$.

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.

2.6.1.7 Replicates test

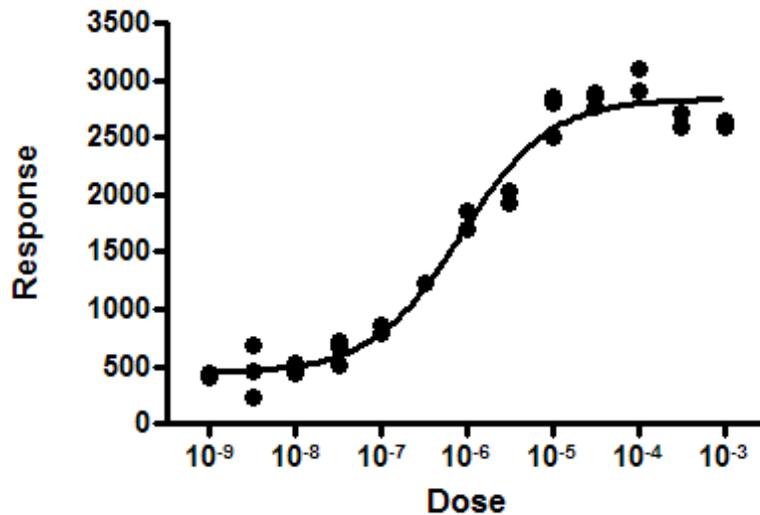
Goal of the replicates test

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 to the mean of the replicates, the value will be low. If the curve is far from some of those means, the value will be high.

The third row (F) is the square of the ratio of those two SD values.

If the model is correct, and all the scatter is Gaussian random variation around the model, then the two SD values will probably be similar, so F should be near 1. In this example, F is greater than 1 because the SD for lack of fit is so much greater than the SD of replicates. The P value answers this question:

If the model was chosen correctly and all scatter is Gaussian, what is the chance of finding a F ratio so much greater than 1.0?

A small P value is evidence that the data actually follow a model that is different than the model that you chose. In this example, this suggests that maybe some sort of biphasic dose-response model is correct -- that the dip of those last few points is not just coincidence.

How the replicates test calculations are done

This test for lack of fit is discussed in detail in advanced texts of linear regression (1,2) and briefly mentioned in texts of nonlinear regression (3, 4). Here is a brief explanation of the method:

The SD of replicates is computed by summing the square of the distance of each replicate from the mean of that set of replicates. Divide that sum by its degrees of freedom (the total number of points minus the number of X values), and take the square root. This value is based only on variation among

replicates. It can be computed without any curve fitting.

The SD lack of fit is a bit trickier to understand. Replace each replicate with the mean of its set of replicates. Then compute the sum of square of the distance of those points from the curve. If there are triplicate values, then each of the three replicate values is replaced by the mean of those three values, the distance between that mean and the curve is squared, and that square is entered into the sum three times (one for each of the replicates). Now divide that sum of squares by its degree of freedom (number of points minus number of parameters minus number of X values), and take the square root.

The F ratio is the square of the ratio of the two SDs.

The P value is computed from the F ratio and the two degree of freedom values, defined above.

References

1. *Applied Regression Analysis*, N Draper and H Smith, Wiley Interscience, 3rd edition, 1998 page 47-56.
2. *Applied Linear Statistical Models* by M Kutner, C Nachtsheim, J Neter, W Li, Irwin/McGraw-Hill; 5th edition (September 26, 2004), pages 119-127
3. *Nonlinear Regression Analysis & its applications*, DM Bates and DG Watts, Wiley Interscience, 1988, pages 29-30.
4. *Nonlinear Regression*, CAF Seber and CJ Wild, Wiley Interscience, 2003, pages 30-32.

2.6.1.8 Dependency of each parameter

Intertwined parameters

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

What does it mean for parameters to be intertwined? After fitting a model, change the value of one parameter but leave the others alone. this will move the curve away from the points. Now change the other parameter(s) in an attempt to to move the curve close to the data points. If you can bring the curve closer to the points, the parameters are intertwined. If you can bring the curve back to its original position, then the parameters are completely redundant.

Prism can quantify the relationships between parameters in two ways. If you are in doubt, we suggest that you focus on the dependency values and not bother with the [covariance matrix](#).¹⁹⁹



Dependency

What is dependency and how do I ask Prism to compute it?

Dependency is reported for each parameter, and quantifies the degree to which that parameter is intertwined with others. Check a check box on the Diagnostics tab of nonlinear regression to view dependencies for each parameter.

Interpreting dependency

The value of dependency always ranges from 0.0 to 1.0.

A dependency of 0.0 is an ideal case when the parameters are entirely independent (mathematicians would say *orthogonal*). In this case the increase in sum-of-squares caused by changing the value of one parameter cannot be reduced at all by also changing the values of other parameters. This is a very rare case.

A dependency of 1.0 means the parameters are redundant. After changing the value of one parameter, you can change the values of other parameters to reconstruct exactly the same curve. If any dependency is greater than 0.9999, GraphPad labels the fit ['ambiguous'](#).²²⁸

With experimental data, of course, the value will almost always lie between these extremes. Clearly a low dependency value is better. But how high is too high? Obviously, any rule-of-thumb is arbitrary. But dependency values up to 0.90 and even 0.95 are not uncommon, and are not really a sign that anything is wrong.

A dependency greater than 0.99 is really high, and suggests that something is wrong. This means that you can create essentially the same curve, over the range of X values for which you collected data, with multiple sets of parameter values. Your data simply do not define all the parameters in your model. If your dependency is really high, ask yourself these questions:

- Can you fit to a simpler model?

- Would it help to collect data over a wider range of X, or at closer spaced X values? It depends on the meaning of the parameters.
- Can you collect data from two kinds of experiments, and fit the two data sets together using global fitting?
- Can you constrain one of the parameters to have a constant value based on results from another experiment?

If the dependency is high, and you are not sure why, look at the covariance matrix (see below). While the dependency is a single value for each parameter, the covariance matrix reports the normalized covariance for each pair of parameters. If the dependency is high, then the covariance with at least one other parameter will also be high. Figuring out which parameter that is may help you figure out where to collect more data, or how to set a constraint.

How dependency is calculated

[This example](#)^[70] will help you understand how Prism computes dependency.

2.6.1.9 Covariance matrix

What is the covariance matrix and how do I ask Prism to compute it?

The normalized covariance is reported for each pair of parameters, and quantifies the degree to which those two parameters are intertwined. Check a check box on the Diagnostics tab of nonlinear regression to view this covariance matrix.

Interpreting the normalized covariance matrix

Each value in the normalized covariance matrix ranges from -1.0 to 1.0. A value equal to -1.0 or 1.0 means the two parameters are redundant. A value of 0.0 means the parameters are completely independent or orthogonal -- if you change the value of one parameter you will make the fit worse and changing the value of the other parameter can't make it better. You can interpret a normalized covariance much as you interpret a correlation coefficient.

Note the difference between covariance and [dependency](#)^[197]. Each value in the covariance matrix tells you how much two parameters are intertwined. In contrast, each dependency value tells you how much that parameter is intertwined with all other parameters.

How to convert to the nonnormalized variance/covariance matrix

Some other programs report the actual (not normalized) variance-covariance matrix. Compute the actual covariance -- $\text{cov}(i,j)$ -- of any two parameters (so i does not equal j) from the normalized matrix Prism reports -- $\text{NormCov}(i,j)$ -- and the standard errors of the parameters using this equation:

$$\text{Cov}(i, j) = \text{NormCov}(i, j) * \text{SE}(i) * \text{SE}(j)$$

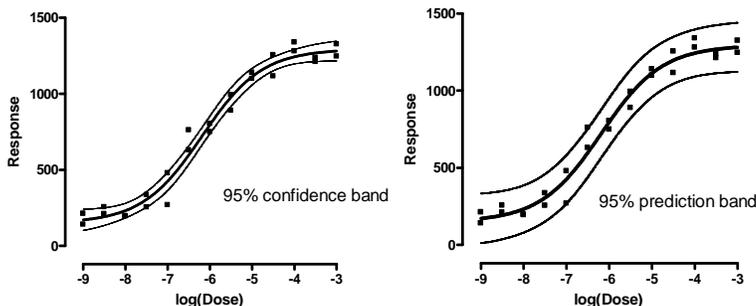
Prism does not report the normalized covariance matrix for a parameter with itself, because the normalized covariance of any parameter with itself equals, by definition, 1.0. The covariance of any parameter with itself is better called its variance. You can calculate the variance of any parameter (a diagonal value in the variance-covariance matrix) using this equation:

$$\text{Cov}(i, i) = \text{SE}(i)^2$$

2.6.1.10 Confidence and prediction bands

What are confidence and prediction bands?

The 95% **confidence bands** enclose the area that you can be 95% sure contains the true curve. It gives you a visual sense of how well your data define the best-fit curve. It is closely related to the 95% **prediction bands**, which enclose the area that you expect to enclose 95% of future data points. This includes both the uncertainty in the true position of the curve (enclosed by the confidence bands), and also accounts for scatter of data around the curve. Therefore, prediction bands are always wider than confidence bands.



How to plot confidence bands, prediction bands or both

Choose to plot confidence or prediction bands by an checking option on the [Diagnostics tab](#)¹⁶⁵.

Prism lets you choose either a confidence band or a prediction band, but not both. To plot both on one graph, you need to analyze your data twice, choosing a confidence band the first time and a prediction band the second time.

The regression lines or curves are data sets that you can add to any graph by drag-n-drop or using the Change menu from the graph.

How to view the XY coordinates of the confidence or prediction bands

Prism can also show you the XY coordinates that define the curve and confidence bands (or prediction bands). But it only shows this table if you ask for it:

1. Go to the Diagnostics tab and check the option to plot confidence or prediction bands.
2. Go to the Range tab of the nonlinear regression dialog.
3. Check the option to create a table of XY coordinates. This table will include the confidence or prediction bands.
4. Enter the number of XY pairs you want to define the curve. 150 is a reasonable starting value, but of course it all depends on why you want these values.
5. When viewing the results, expand the results sheet in the navigator and go to the page labeled 'Curve'. For each X value, you'll see the Y value and the distance the confidence or prediction curve is above and below that value.

How to plot a one-sided confidence or prediction band

In some situations, it makes sense to plot only one side of the confidence or prediction band. One use for a one-sided confidence band would be on a graph that plots purity of a substance over time. You want to know, with a certain confidence, how low the purity could be at a certain time. But you don't really care about quantifying how high the purity could be.

To plot a one sided 95% confidence or prediction band:

1. Check the option (on the Diagnostics tab) to plot "90%" confidence (or prediction) bands. When you plot in only one direction, this is really a 95% confidence band.
2. On the Format Graph dialog, choose the data set that defines regression curve and make sure that error bars are turned on with the "---" style.
3. Choose to plot those error bands in one direction.

Show error bars

Color: Dir.: Above Style: ----- Thickness: 1 pt

Do not mix up confidence intervals and confidence bands

It is easy to mix up confidence intervals and confidence bands. Choose both on the [Diagnostics tab](#) ¹⁶⁵.

The **95% confidence interval** tells you how precisely Prism has found the best-fit value of a particular parameter. It is a range of values, centered on the best-fit value. Prism can display this range in two formats:

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

10	Lower 95% conf. limit	
11	SPAN	8849
12	K	0.05566
13	HalfLife	9.171
14	Upper 95% conf. limit	
15	SPAN	9959
16	K	0.07558
17	HalfLife	12.45

Don't mix up confidence bands and prediction bands

Note the difference between confidence and prediction bands:

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

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

Situations where Prism won't plot confidence or prediction bands

Prism will not plot confidence or prediction bands in several situations:

- If the best-fit value of a parameter [hits a constraint](#)^[231], the fit is unlikely to be useful. Prism does not plot confidence or prediction bands, because they would almost certainly be misleading.
- If the results of nonlinear regression are [ambiguous](#)^[228], the confidence or prediction bands would be super wide, maybe infinitely wide. They would not be useful, so Prism does not plot them.
- If you choose robust nonlinear regression, Prism does not compute confidence or prediction bands, as it cannot compute standard errors or confidence intervals of the parameters.
- The fit is [perfect](#)^[233]. If the sum-of-squares is 0.0 and R2 is 1.0, it is not possible to compute or interpret confidence or prediction bands.
- If the fit is [interrupted](#)^[226], confidence and prediction bands are not computed.

How are confidence and prediction bands computed?

First, let's 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 (inversed Hessian from last iteration). It is a square matrix with the number of rows and columns equal to the number of parameters. Each item in the matrix is the covariance between two parameters. We use Cov to refer to the normalized covariance matrix, where each value is between -1 and 1. [Details and comparison to nonnormalized covariance matrix.](#)

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:

$$= \text{sqrt}(c) * \text{sqrt}(SS/DF) * \text{CriticalT}(\text{Confidence}\%, DF)$$

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

$$= \sqrt{c+1} * \sqrt{SS/DF} * \text{CriticalT}(\text{Confidence}\%, \text{DF})$$

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.

[Read a more mathematical explanation.](#)

2.6.1.11 Hougard's measure of skewness

Consequences of an asymmetrical parameter

Even though nonlinear regression, as its name implies, is designed to fit nonlinear models, some of the inferences actually assume that some aspects of the model are close to linear, so that the uncertainty about each parameter's value is symmetrical. This means that if you analyzed many data sets sampled from the same system, the distribution of the best-fit values of the parameter would be symmetrical and Gaussian.

If the distribution of a parameter is highly skewed, [there are two consequences](#)

 429]:

- The SE of that parameter will not be a very useful measure of uncertainty. The SE is interpreted as a plus-minus assessment of how sure you are of the parameter value. But if the parameter is very asymmetrical, then a single SE cannot really describe the uncertainty.
- The confidence interval for that parameter cannot be interpreted at face value. Prism always computes the CI from the SE of that parameter (and the number of degrees of freedom), and the CI is always symmetrical around the best-fit value. If the parameter is very asymmetrical, then that symmetrical confidence interval does not give an accurate picture of the uncertainty.

Hougaard's measure of skewness

Hougaard (1) developed a way to assess the skewness of a parameter used in nonlinear regression without doing any simulations. Prism will compute this value for each parameter when you check the option the Diagnostics tab of nonlinear regression in the section labeled "Are the parameters intertwined, redundant or skewed?". The results are tabulated along with the other results of nonlinear regression.

[Ratkowsky](#) has proposed the following interpretation:

Absolute value	Interpretation
<0.10	Ideal. Almost linear. Confidence intervals can be interpreted at face value
0.10 - 0.25	Adequate
0.25 - 1.00	Noticeable skewness. Consider alternative parameterizations of the equation
> 1.00	Considerable skewness. Strongly consider alternatives

Note that these values are for the absolute value of the Hougaard's measure.

For the simulated data set for the example above, Hougaard's skewness is 0.09 for Khalf and 1.83 for Kprime. This one value (no simulations needed) tells you to choose the form of the model that fits Khalf rather than the form that fits Kprime.

Notes

- Hougaard's measure of skewness is measured for each parameter in the equation (omitting parameters fixed to constant values).
- Prism does not compute Hougaard's skewness if you chose unequal weighting, or a robust fit, because the method is not defined for these situations.
- The values depend on the equation, the number of data points, the spacing of the X values, and the values of the parameters.
- Hougaard's measure of skewness has no units.

- The SAS documentation does a great job of explaining Hougaard's measure (3).

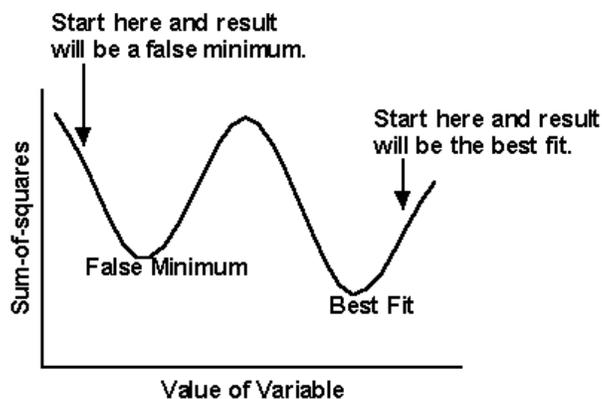
References

1. P. Hougaard. [The appropriateness of the asymptotic distribution in a nonlinear regression model in relation to curvature](#). *Journal of the Royal Statistical Society*. Series B (Methodological) (1985) pp. 103-114
2. David A. Ratkowsky, [Nonlinear Regression Modeling: A Unified Practical Approach](#) (Statistics: a Series of Textbooks and Monographs). ISBN:0824719077
3. [SAS documentation about Hougaard's measure](#).

2.6.1.12 Could the fit be a local minimum?

What is a false minimum?

The nonlinear regression procedure adjusts the variables in small steps in order to improve the goodness-of-fit. If Prism converges on an answer, you can be sure that altering any of the variables a little bit will make the fit worse. But it is theoretically possible that large changes in the variables might lead to a much better goodness-of-fit. Thus the curve that Prism decides is the "best" may really not be the best.



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:

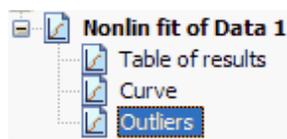
1. Note the values of the variables and the sum-of-squares from the first fit.
2. Make a large change to the initial values of one or more parameters and run the fit again.
3. Repeat step 2 several times.

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.

2.6.1.13 Outliers

The outliers table

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



Keep in mind an important distinction:

- If you chose to count outliers (on the [Diagnostics tab](#)¹⁶⁵), then the outliers are still included in the fit.
- If you chose to exclude outliers (on the [Fit tab](#)¹⁵⁵), then the outliers are ignored by the fit, but are still [included on the graph](#)¹⁷⁶.

In both cases, the aggressiveness of the outlier hunt is determined by the ROUT

coefficient Q that you enter on the [Weights tab](#)^[160].

What does it mean to be an outlier

Prism uses the [ROUT algorithm](#)^[53], using the value of Q that you can adjust in the [Weights tab](#)^[160], 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](#)^[54].
- Be alert to biological variation. In some experimental situations, outliers can only arise from lab mistakes. These should be excluded. In other situations, an outlier can be the result of biological variation. You may have discovered a new polymorphism in some gene. It could be missing a big opportunity if you exclude and ignore those values.

What to do once you have identified an outlier

After you have identified one or more outliers, you have several choices.

- While outliers are ignored by the nonlinear regression calculation, they are still plotted on the graph. If you want the outlier to be entirely removed from the graph, including error bars on the graph, go back to the data table and exclude the outlier(s). Prism will then refit the curve, and replot the graph without any outliers.
- If you want the outlier to remain on the graph, simply polish the graph Prism provides. Keep in mind that the outliers are a dataset, so you can separately adjust the size, color and symbol of outliers and all the data points. Use the "Datasets on graph" tab of Format Graph to adjust the back-to-front order of data sets to keep the outlier in front of the other data sets. The dialog shows the datasets in back to front order, so you want to keep the outlier data sets lower in the list.
- If you want to remove the special formatting of outliers on the graph, use the "Datasets on graph" tab of Format Graph. Select the outlier dataset(s) and click Remove. The outlier will still be plotted as part of the full data set, but will no longer be plotted separately as an outlier data set.
- If you asked Prism to count the outliers (Diagnostics tab) but now wish to

fit the data with the outlier ignored, go back to the nonlinear regression dialog and choose "Automatic outlier elimination" (on the Fit tab).

2.6.1.14 Troubleshooting nonlinear regression

The #1 tip for troubleshooting

Nonlinear regression starts with initial values for each parameter. If these initial values are very far from the correct values, nonlinear regression might go astray. To check this, go to the [Diagnostics tab of the nonlinear regression dialog](#)^[165] and choose the option at the top of the dialog: Don't fit a curve. Instead plot the curve defined by the initial values of the parameters.

If the curve defined by the initial values does not generally follow the shape of the data, and go near the data points, you should change the initial values ([Initial Values tab of nonlinear regression](#)^[162]) until they do.

If Prism reports an error code

Prism reports error codes as a single phrase that appears over the column of results, for example "Bad initial values" or "Impossible weights". [Learn more about each error message.](#)^[225]

Common problems in nonlinear regression

Here is a short list of potential curve fitting problems with suggested solutions.

The equation simply does not describe the data.

Try a different equation.

The initial values are too far from their correct values.

Enter different initial (estimated) values for the parameters. If you entered your own equation, check that you entered sensible rules for generating initial values.

The range of X values is too narrow to define the curve completely.

If possible, collect more data. Otherwise, hold one of the parameters to a constant value.

You have not collected enough data in a critical range of X values.

Collect more data in the important regions.

Your data are very scattered and don't really define a curve.

Try to collect less scattered data. If you are combining several experiments, consider normalizing the data for each experiment to an internal control.

The equation includes more than one component, but your data don't follow a multicomponent model.

Use a simpler equation.

Your numbers are too large or too small.

If your X or Y values are huge, divide by a constant to change the units. Avoid values greater than, say, 100,000.

If your X or Y values are tiny, multiply by a constant change the units. Avoid values less than about 0.00001.

Changing the units is unlikely to solve the problem, but it is worth a try.

You've set a parameter to an inappropriate constant value.

Check that you haven't made a simple mistake like setting a maximum plateau to 1.0 when it should be 100, or a Hill slope to +1.0 when it should be -1.0.

Consult an analysis check list

Prism has three different analysis checklists for nonlinear regression. Use the one that matches your goal.

[Analysis checklist: Fitting a model](#)²¹⁸

[Analysis checklist: Comparing nonlinear fits](#)²²¹

[Analysis checklist: Interpolating from a standard curve](#)²²³

2.6.1.15 Why results in Prism 5 and 6 can differ from Prism 4

Prism 5 and 6 use the same algorithms, so should always report identical results.

However, Prism 4 used slightly different algorithms, so curve fitting results can differ from results with Prism 5 or 6 in these cases:

- If your fit is labeled "Ambiguous" by Prism 5 or 6, you know that some of the parameters are not determined precisely. Prism 4 presented a full set of results in this case, but the results are not useful when the fit is ambiguous.
- If you chose no weighting, check the sum-of-squares from the two programs. The goal of regression is to minimize that sum of squares, so see which version of Prism found a fit with the smaller sum-of-squares. Prism 5 and 6 have a few improvements in the fitting algorithm, so occasionally it can find a better fit than did Prism 4. The differences, if any, are usually trivial.
- If you chose to weight by the Y values (or the Y values squared), Prism 5 and 6 handle weighting differently than did Prism 4. Prism now weights by the Y value of the curve, while Prism 4 (and earlier releases) weighted by the Y value of the data. Weighting by the Y value of the curve is better, so the results of Prism 5 and 6 are more correct. Since the weighting is computed differently, you can't directly compare the weighted sum-of-square values reported by the two versions of Prism.
- When you compare two models, Prism now does an extra step. If one of the models is ambiguous, then Prism chooses the other model, without doing the F test or AIC comparison.
- Prism now offers more rules for defining initial parameter values. If your equation uses one of these new rules, Prism 4 might not be able to find a reasonable fit until you tweak those initial values. In particular, Prism now has smarter rules for fitting sigmoidal log(dose) vs. response curves.
- If you entered data as mean, SD (or SEM) and N, then Prism 4 (by default) fits the means and weights by the sample size (N). This is one of the two options on the weighting tab (the other option is to fit means only, ignoring N). Prism is now smarter by default (although you can choose to just fit the means and ignore N and SD values). It accounts not only for sample size N but also for the SD (or SEM) values you enter. With Prism 5 or 6 (but not Prism 4), you'll get exactly the same results from data entered as mean, SD and N as you would have by entering raw data. Prism 4 only accounts for differences in N, but not SD. The best fit values of the parameters, and thus the appearance of the curve, is the same with Prism 4 and 5. But Prism 5

does a smarter job with standard errors, confidence intervals, and comparisons of models.

2.6.2 Interpreting results: Comparing models

In many cases, your primary goal in fitting curves is to compare models, or compare fits of two (or more) data sets.

- First [review the general principles](#)^[212] of comparing models.
- Next review the method you choose to use: [Extra sum-of-squares](#)^[213] or [AIC model comparison](#)^[215].
- Review [how Prism compares models when it detects outliers](#)^[215].

2.6.2.1 Interpreting comparison of models

Reality check before statistics

Apply a common sense reality check before looking at any statistical approach to comparing fits. If one of the fits has results that are scientifically invalid, then accept the other model. Only when both fits make scientific sense should you use statistical method to compare the two fits .

Prism partially automates this 'reality check' approach. If the fit of either model is [ambiguous](#)^[228], 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](#)^[213] or using information theory and computation of [AIC](#)^[215].

Both these methods only make sense when the models being compared have different numbers of parameters, and so have different numbers of degrees of

freedom. If you want to compare two models with the same number of parameters, there is no need to use either the F test or AIC. Simply choose the model that fits the data the best with the smallest sum-of-squares.

How do these methods work to compare data sets?

The [Compare tab](#)^[157] of Prism lets you ask "Do the best-fit values of selected unshared parameters differ between data sets?" or "Does one curve adequately fit all data sets?". Applying the F test or Akaike's method to answering these questions is straightforward. Prism compares the sum-of-squares of two fits.

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

These two fits are nested (the second is a simpler case of the first, with fewer parameters to fit) so the sums-of-squares (actually the sum of sum of squares for the first fits) can be compared using either the [F test](#)^[213] or [Akaike's method](#)^[215].

2.6.2.2 Interpreting the extra sum-of-squares F test

The extra-sum-of-squares F test compares nested models

The extra-sum-of-squares F test compares the goodness-of-fit of two alternative nested models. "Nested" means that one model is a simpler case of the other. Let's consider what this means in different contexts:

- If you asked Prism to test whether parameters are different between treatments, then the models are nested. You are comparing a model where Prism finds separate best-fit values for some parameters vs. a model where those parameters are shared among data sets. The second case (sharing) is a simpler version (fewer parameters) than the first case (individual parameters).
- If you asked Prism to test whether a parameter value is different than a hypothetical value, then the models are nested. You are comparing the fit of a

model where a parameter is fixed to a hypothetical value to the fit of a model where Prism finds the best-fit value of that parameter. The first case (fixed value) is a simpler version (fewer parameters to fit) than the second.

- if you are comparing the fits of two equations you chose, and both models have the same number of parameters, then the two models cannot be nested. With nest models, one model has fewer parameters than the other. When the two models have the same number of parameters, Prism reports that it cannot compute the F test because the two models have the same number of degrees of freedom. In this case, Prism does not report a P value, and chooses to plot the model whose fit has the lower sum-of-squares. Prism reports the error message, "Models have the same DF."
- if you are comparing the fits of two equations you chose with different numbers of parameters, the models may or may not be nested. Prism does not attempt to do the algebra necessary to make this determination. If you chose two models that are not nested, Prism will report results for the extra-sum-of-squares F test, and these results will not be useful.

Interpreting the P value

The extra-sum-of-squares F test is based on traditional statistical hypothesis testing. The F test compares the improvement of SS with the more complicated model vs. the loss of degrees of freedom.

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

The P value answers this question:

If the null hypothesis is really correct, in what fraction of experiments (the size of yours) will the difference in sum-of-squares be as large as you observed, or even larger?

If the P value is small, conclude that the simple model (the null hypothesis) is wrong, and accept the more complicated model. Usually the threshold P value is set at its traditional value of 0.05.

If the P value is high, conclude that the data do not present a compelling reason to reject the simpler model.

Prism names the null and alternative hypotheses, and reports the P value. You set the threshold P value in the Compare tab of the nonlinear regression dialog. If the P value is less than that threshold, Prism chooses (and plots) the alternative (more complicated) model. It also reports the value of F and the numbers of degrees of freedom, but these will be useful only if you want to compare Prism's results with those of another program or hand calculations.

2.6.2.3 Interpreting AIC model comparison

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

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

Of course, these probabilities are meaningful only in the context of comparing those two models. It is possible a third model you didn't test fits much better so is much more likely to be correct.

Prism names the null and alternative hypotheses, and reports the likelihood that each is correct. It also reports the difference between the AICc values (as the AICc of the simple model minus the AICc of the more complicated model), but this will be useful only if you want to compare Prism's results with those of another program or hand calculations. Prism chooses and plots the model that is more likely to be correct, even if the difference in likelihoods is small.

2.6.2.4 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](#)^[155]), 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.

2.6.2.5 Interpreting the adjusted R²

Why you should not use R² to compare models

R² quantifies how well a model fits the data, so it seems as though it would be an easy way to compare models. It sure sounds easy -- pick the model with the larger R². The problem with this approach is that there is no penalty for adding more parameters. So the model with more parameters will bend and twist more to come nearer the points, and so almost always has a higher R². If you use R² as the criteria for picking the best model, you'd almost always pick the model with the most parameters.

The adjusted R² accounts for the number of parameters fit

The adjusted R² always has a lower value than R² (unless you are fitting only one parameter). The equations below show why.

$$R^2 = 1 - \frac{SS_{residuals}}{SS_{total}}$$

$$\text{Adjusted } R^2 = 1 - \frac{SS_{residuals} / (n - K)}{SS_{total} / (n - 1)}$$

The equations above show how the adjusted R² is computed. The sum-of-squares of the residuals from the regression line or curve have n-K degrees of freedom, where n is the number of data points and K is the number of parameters fit by the regression. The total sum-of-squares is the sum of the squares of the distances from a horizontal line through the mean of all Y values. Since it only has one parameter (the mean), the degrees of freedom equals n-

1. The adjusted R^2 is larger than the ordinary R^2 whenever K is greater than 1.

Using adjusted R^2 and a quick and dirty way to compare models

A quick and easy way to compare models is to choose the one with the smaller adjusted R^2 . Choose to report this value on the [Diagnostics tab](#)^[165].

Comparing models with adjusted R^2 is not a standard method for comparing nonlinear models (it is standard for multiple linear regression), and we suggest that you use the [extra-sum-of-square F test](#)^[213] or [comparing AICc](#)^[215] instead. If you do compare models by comparing adjusted R^2 , make sure that identical data, weighted identically, are used for all fits.

Adjusted R^2 in linear regression

Prism doesn't report the adjusted R^2 with linear regression, but you can fit a straight line with nonlinear regression.

If X and Y are not linearly related at all, the best fit slope is expected to be 0.0. If you analyzed many randomly selected samples, half the samples would have a slope that is positive and half the samples would have a negative slope. But in all these cases, R^2 would be positive (or zero). R^2 can never be negative (unless you constrain the slope or intercept so it is forced to fit worse than a horizontal line). In contrast, the adjusted R^2 can be negative. If you analyzed many randomly selected samples, you'd expect the adjusted R^2 to be positive in half the samples and negative in the other half.

Here is a simple way to think about the distinction. The R^2 quantifies the linear relationship in the sample of data you are analyzing. Even if there is no underlying relationship, there almost certainly is some relationship in that sample. The adjusted R^2 is smaller than R^2 and is your best estimate of the degree of relationship in the underlying population.

2.6.3 Analysis checklists: Nonlinear regression

To understand your results, review the appropriate analysis checklist, depending on your experimental goal.

- [Analysis checklist: Fitting a model](#)^[218]
- [Analysis checklist: Comparing nonlinear fits](#)^[221]
- [Analysis checklist: Interpolating from a standard curve](#)^[223]

2.6.3.1 Analysis checklist: Fitting a model

Your approach in evaluating nonlinear regression depends on your goal.

In many cases, your goal is to create a standard curve from which to interpolate unknown values. We've created [a different checklist](#)^[223] 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](#)^[193] and [replicates](#)^[194] 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](#)¹⁷⁹ 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.

2.6.3.2 Analysis checklist: Comparing nonlinear fits

Approach to comparing models

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

The model with fewer parameters almost always fits the data worse. When you compare two models, the simpler model (with fewer parameters) will have a higher sum-of-squares. When you compare datasets, the global fit will have a higher sum-of-squares than the sum of the individual fits. And when you compare fits with and without a parameter constrained to a particular value, will have a higher sum-of-squares than a model where you don't constrain the parameter.

Both the F test and the AIC method take into account both the difference in goodness-of-fit but also the number of parameters. The question is whether the decrease in sum of squares (moving from simpler to more complicated model) is worth the 'cost' of having more parameters to fit. The [F test and](#)

[AICc methods](#)⁴⁸ 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](#)^[228] or [perfect](#)^[233], 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](#)^[218] 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?**

2.6.3.3 Analysis checklist: Interpolating from a standard curve

Your approach in evaluating nonlinear regression depends on your goal.

In many cases, your goal is to learn from the best-fit values. If that is your goal, [view a different checklist](#)^[218].

If your goal is to create a standard curve from which to interpolate unknown values, your approach depends on whether this is a new or established assay.

Established assay

If the assay is well established, then you know you are fitting the right model and know what kind of results to expect. In this case, evaluating a fit is pretty easy.

- ✓ **Does the curve go near the points?**
- ✓ **Is the R2 'too low' compared to prior runs of this assay?**

If so, look for outliers, or use Prism's automatic outlier detection.

- ✓ **Are the confidence bands too wide?**

The [confidence bands](#)  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 you 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.

2.6.4 Error messages from nonlinear regression

When Prism hits a problem, it will present a very brief message in the top row of the results tables. Read the pages in this section to learn what these messages mean.

["Bad initial values"](#)  226

["Interrupted"](#)  226

["Not converged"](#)  227

["Ambiguous"](#)  228

["Hit constraint"](#)  231

["Don't fit"](#)  232

["Too few points"](#) 

["Perfect fit"](#) 

["Impossible weights"](#) 

["Equation not defined"](#) 

2.6.4.1 "Bad initial values"

What does "bad initial values" mean?

Nonlinear regression works iteratively. Prism starts with initial estimated values for each parameter. It then gradually adjusts these until it converges on the best fit. In rare circumstances, Prism cannot compute a curve based on the initial values you entered. When it tries, it encounters a math problem such as division by zero or taking the logarithm of a negative number. In this case, Prism does not report any results and instead reports "bad initial values".

Checklist

- ✓ Did you enter or choose the right equation?
- ✓ Did you enter sensible values for initial values?
- ✓ Is the range of X values invalid? Focus on the first and last X value (and on X=0 if included in the range). Can the equation be evaluated at those X values with the initial values you entered?

2.6.4.2 "Interrupted"

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 1000).
- 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](#)  to plot that curve.
- ✓ If you entered constraints, were they entered correctly?
- ✓ If you didn't enter any constraints, consider whether you can constrain one or more parameters to a constant value? For example, in a dose-response curve can you constrain the bottom plateau to be zero?
- ✓ Can you share a parameter over all the data sets (global fitting)?

2.6.4.3 "Not converged"

What does "not converged" mean?

Nonlinear regression works iteratively. Prism starts with initial estimated values for each parameter. It then gradually adjusts these until it converges on the best fit. "Converged" means that any small change in parameter values creates a curve that fits worse (higher sum-of-squares). But in some cases, it simply can't converge on a best fit, and gives up with the message 'not converged'. This happens in two situations:

- The model simply doesn't fit the data very well. Perhaps you picked the wrong model, or applied the wrong constraints.
- The initial values generated a curve that didn't come close to the points. In that case, Prism may not be able to figure out how to change the parameters to make the curve fit well.

Checklist

- ✓ Did you enter the right model?

- ✓ Does the curve defined by your initial values come near your data? Check the option on the [diagnostics tab](#)¹⁶⁵ to plot that curve.
- ✓ If you entered constraints, were they entered correctly?
- ✓ If you didn't enter any constraints, consider whether you can constrain one or more parameters to a constant value? For example, in a dose-response curve can you constrain the bottom plateau to be zero?
- ✓ Can you share a parameter over all the data sets (global fitting)?

2.6.4.4 "Ambiguous"

What does "ambiguous" mean?

"Ambiguous" is a term coined by GraphPad to describe a fit that doesn't really nail down the values of all the parameters.

Changing the value of any parameter will always move the curve further from the data and increase the sum-of-squares. But when the fit is 'ambiguous', changing other parameters can move the curve so it is near the data again. In other words, many combinations of parameter values lead to curves that fit equally well.

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

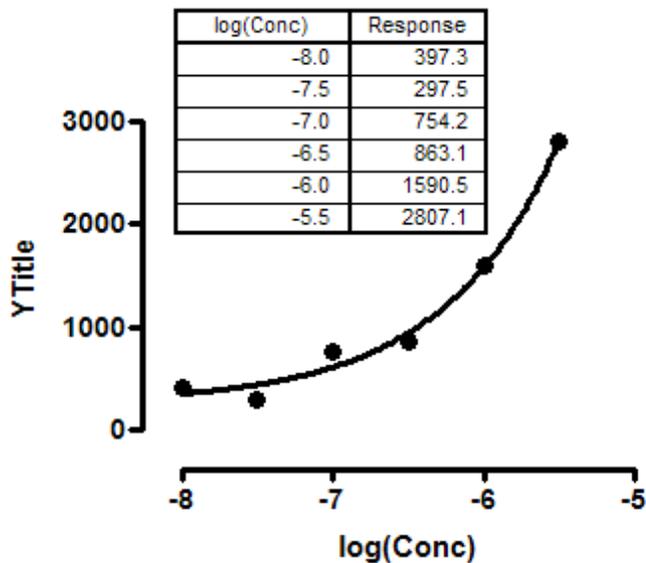
parameters, then you've got a real problem. At least one of the parameters has a best-fit value that you should not rely upon.

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.

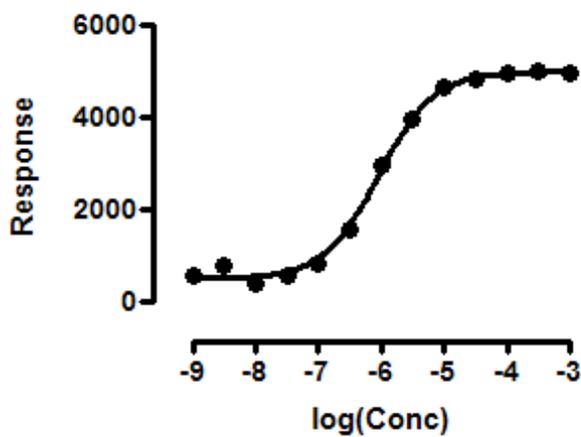
Reasons for ambiguous fits

Data not collected over a wide enough range of X values



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: $Y = A + B + C \cdot X$. This model describes a straight line with a slope equal to C and a Y intercept equal to the sum of A and B . But there is nothing in the data to fit A and B individually. There are an infinite number of pairs of values of A and B that lead to the same sum, so the same Y intercept. If Prism attempts to fit this model, it will conclude that the fit is ambiguous.

Prism cannot tell the difference between this case and the previous one. But the two cases are distinct. The model in this example has redundant parameters. It doesn't matter what the data look like, or how many data points you collect, fitting this model will always result in 'ambiguous' results, because two of the parameters are intertwined. In the previous example, the fit was ambiguous with the example data set, but would not have been ambiguous with other data sets.

Checklist

- ✓ Can you constrain any parameters to a constant value?

- ✓ Can you share one or more parameters among datasets?
- ✓ Can you collect more data points, perhaps over a wider range of X values?
- ✓ If your model has two phases or components, consider switching to a one phase or component model.

2.6.4.5 "Hit constraint"

What does "hit constraint" mean?

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

When a fit hits a constraint, the results are unlikely to provide useful information. If you had a solid reason to constrain a parameter within a range of values, it ought to end up in that range. If the fit hit the constraint limit, that means the true best-fit value is some value forbidden by the constraint.

Prism does not compute confidence and prediction bands when a parameter hit a constraint. The best-fit values are not a local minimum, so any attempt to compute confidence or prediction bands would give misleading results. Prism does compute the confidence intervals for the other parameters (the ones that didn't hit a constraint) but these need to be viewed with caution.

When a fit ends up hitting a constraint, it is likely that you set the constraint incorrectly. So the first thing to do is make sure the constraint is sensible and correctly entered. For example, if you mistakenly set a constraint that a rate constant K must be *less than or equal to zero*, Prism's nonlinear regression might converge with the best-fit value equalling zero and report that you hit the constraint. Fix the constrain to be *greater than or equal to zero*, and the fit will work fine.

Another possibility is to change the constraint from an inequality ($\text{Bottom} > 0$) to a constant constraint ($\text{Bottom} = 0$). You'll get the same parameter values, but difference confidence intervals, and you can get confidence and prediction bands.

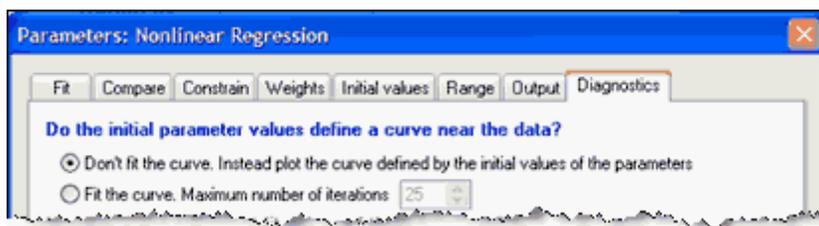
Checklist

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

- ✓ Did you enter the model correctly?
- ✓ Are you asking Prism to do the impossible -- too many parameters for your data?
- ✓ If your model has two phases or components, consider switching to a one phase or component model.

2.6.4.6 "Don't fit"

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



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.").

2.6.4.7 "Too few points"

Regression requires at least as many data points as parameters. So if you fit a dose-response model with four parameters (Bottom, Top, logEC50, and HillSlope), you must have four or more points or Prism will simply report "too few points".

Having as many points as parameters is the bare minimum. Prism will try to fit the model, and will report a [perfect fit](#)²³³, and so not be able to report standard errors or confidence intervals. If there are only a few more points than parameters, Prism is likely to end up with an [ambiguous](#)²²⁸ or [not converged](#)²²⁷ fit.

2.6.4.8 "Perfect fit"

Prism reports "perfect fit" when the curve goes through every point. The sum-of-squares is 0.0, and R^2 is 1.00.

If you are testing nonlinear regression with made up values, add some random scatter to make a better example.

If you are fitting actual data, and the fit is perfect, you either got very lucky or have very few data points. It is not possible to compute the standard errors and confidence intervals of the parameters when the fit is perfect, nor is it possible to compare models.

2.6.4.9 "Impossible weights"

Prism reports "Impossible weighting" when weighting would lead to division by zero for one or more point. This can happen when you choose to weight by the reciprocal of Y, and the Y value of the curve equals zero at some X values. Since division by zero is impossible, Prism reports "Impossible weighting". Go to the weights tab and choose a different [weighting method](#)¹⁶⁰.

Prism also reports "impossible weights" when the Y value of the curve is negative for some values of X. If you weight by $1/Y$, this would mean the weights would be negative. If you weight by $1/Y^2$, the weights would all be positive, but points with Y near zero would get lots of weight, and points with Y values far from zero (in either direction) would get low weights, and this doesn't really make sense.

2.6.4.10 "Equation not defined"

When you enter a user-defined equation, you can [write a different model for different data sets](#)⁴²³. If no model is defined for certain data sets, the message "Equation not defined" will appear in the results for that data set, with no other results.

- ["Bad initial values"](#)²²⁶
- ["Interrupted"](#)²²⁶
- ["Not converged"](#)²²⁷
- ["Ambiguous"](#)²²⁸
- ["Hit constraint"](#)²³¹

- ["Don't fit"](#)²³²
- ["Too few points"](#)²³²
- ["Perfect fit"](#)²³³
- ["Impossible weights"](#)²³³
- ["Equation not defined"](#)²³³

2.6.4.11 "Can't calculate"

If you ask Prism to use either the F test or the AICc method to compare two models, it will report "Can't calculate" if the two models have the same number of parameters. This means it can't calculate the comparison between fits, not that it can't fit the two models to the data.

Both these methods to compare models only make sense when the models being compared have different numbers of parameters, and so have different numbers of degrees of freedom. These methods are alternative methods of dealing with the tradeoff between how well the model fits and how complicated it is (assessed by the number of parameters).

If you want to compare two models with the same number of parameters, there is no need to use either the F test or AIC. Simply choose the model that fits the data the best with the smallest sum-of-squares.

2.7 Models (equations) built-in to Prism

To use nonlinear regression, you need to choose a model. In many cases, especially if you are fitting pharmacological or physiological data, you can choose one of Prism's built-in models described here.

- [Dose-response - Key concepts](#)²³⁵
- [Dose-response - Stimulation](#)²⁵⁵
- [Dose-response - Inhibition](#)²⁶¹
- [Dose-response -- Special](#)²⁶⁷
- [Receptor binding - Key concepts](#)²⁸⁵

- [Receptor binding - Saturation binding](#)  291
- [Receptor binding - Competitive binding](#)  311
- [Receptor binding - Kinetics](#)  324
- [Enzyme kinetics -- Key concepts](#)  335
- [Enzyme kinetics - Substrate vs. velocity](#)  337
- [Enzyme kinetics -- Inhibition](#)  344
- [Exponential](#)  355
- [Lines](#)  370
- [Polynomial](#)  381
- [Gaussian](#)  385
- [Sine waves](#)  393
- [Classic equations from prior versions of Prism](#)  396

2.7.1 Dose-response - Key concepts

2.7.1.1 What are dose-response curves?

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](#)^[244].

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.

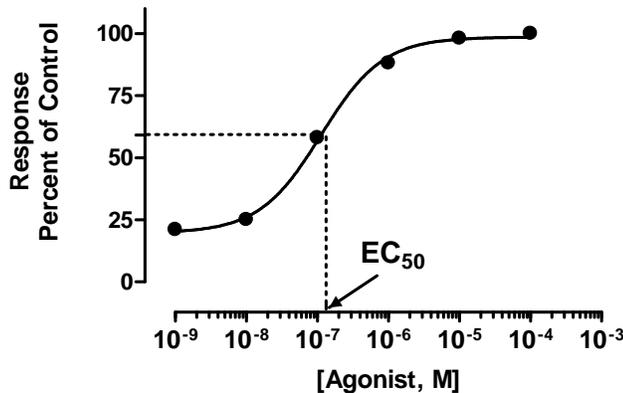
2.7.1.2 The EC50

What is the EC50?

The EC50 is defined quite simply as the concentration of agonist that provokes a response halfway between the baseline (Bottom) and maximum response (Top). It is impossible to define the EC50 until you first define the baseline and maximum response.

Depending on how you have normalized your data, the EC50 may not be the same as the concentration that provokes a response of Y=50. For instance, in

the example below, the data are normalized to percentage of maximum response, without subtracting a baseline. The baseline is about 20%, and the maximum is 100%, so the EC₅₀ is the concentration of agonist that evokes a response of about 60% (halfway between 20% and 100%). The concentration that provokes a response of 50 in this experiment is **not** the EC₅₀.



Don't over interpret the EC₅₀. It is simply the concentration of agonist required to provoke a response halfway between the baseline and maximum responses. Because the EC₅₀ defines the location of the dose-response curve for a particular drug, it is the most commonly used measure of an agonist's potency. However, the EC₅₀ is usually not the same as the K_d 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.

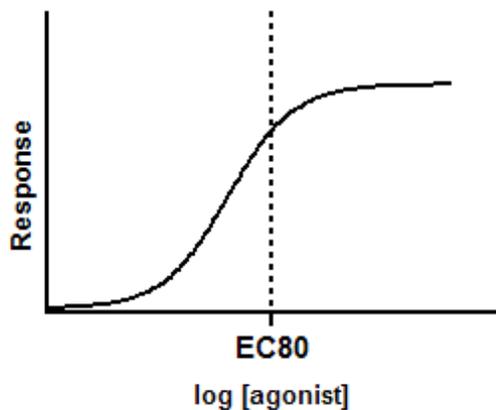
The IC₅₀

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

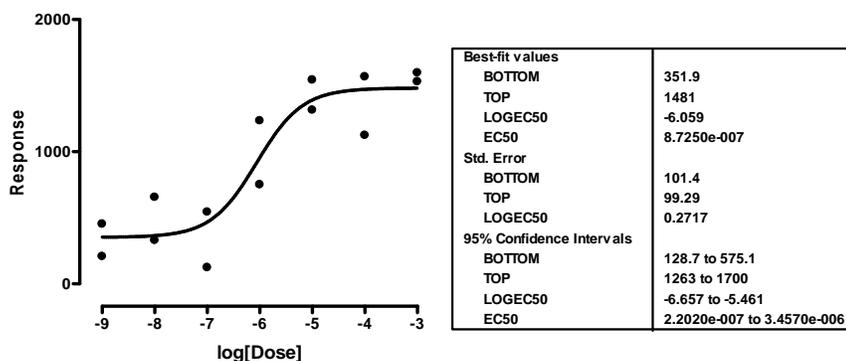
A simple rearrangement of the equation lets you fit EC₈₀ (or EC₉₀ or

ECanything) instead of the EC50. Prism [includes an equation](#)²⁸³ that fits any EC value directly.



2.7.1.3 Confidence intervals of the EC50

The 95% confidence interval of the EC50 is not symmetrical



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.

Keep in mind the fundamentals of a logarithmic 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 (assuming the X values were entered as log Molar), while increasing the logEC50 1 unit from -3 to -2 increases the EC50 by 9 mM, which equals 9,000,000 nM.

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. This value is not the SE of the EC50 and is really not a value that is helpful to compute.

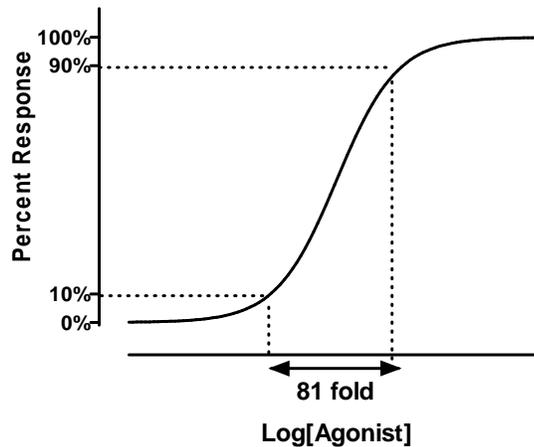
Averaging the EC50 from several experiments

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 EC50 values.

2.7.1.4 Hill slope

The usual steepness of a dose-response curve

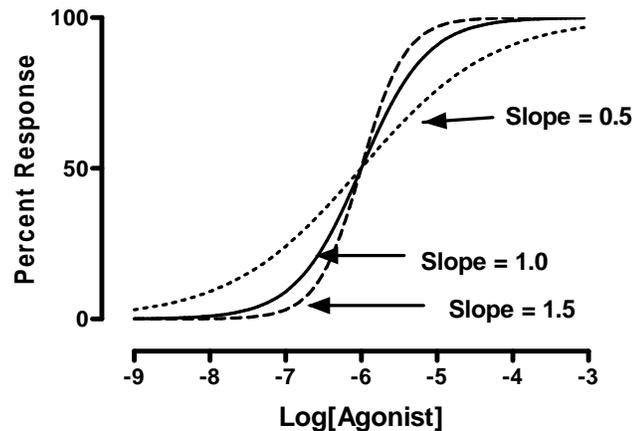
Many dose-response curves follow the shape of a receptor binding curve. As shown below, 81 times more agonist is needed to achieve 90% response than a 10% response.



Since the linkage between agonist binding and response can be very complex, any shape is possible. It seems surprising, therefore, that so many dose-response curves have shapes almost identical to receptor binding curves, even when we know there are multiple steps between binding and measured response. It turns out that no matter how many steps intervene between agonist binding and response, the dose-response curve will have the usual steepness so long as each messenger binds to a single binding site according to the law of mass action.

The slope factor or Hill slope

Some dose-response curves are steeper or shallower than the standard curve. The steepness is quantified by the Hill slope, also called a slope factor. A dose-response curve with a standard slope has a Hill slope of 1.0. A steeper curve has a higher slope factor, and a shallower curve has a lower slope factor.



If you use a single concentration of agonist and varying concentrations of antagonist, the curve goes downhill and the slope factor is negative. The steeper the downhill slope, the more negative the Hill slope.

Standard slope or variable slope?

Because this standard slope is so common, Prism comes with equations with the standard slope built in. The equations that don't have 'variable slope' in their name assume the standard slope (1.0 for stimulation, -1.0 for inhibition).

Deciding whether to fit a model with a standard slope or a variable slope is not easy.

If you have lots of data points (more than a dozen, perhaps lots more), then you can fit the slope by picking a variable slope equation. If you have fewer data points, and a standard system, it makes sense to choose an equation with a standard slope.

2.7.1.5 Choosing a dose-response equation

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](#)^[240] 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?

If your data have been normalized so the curve runs from Y= 0 to Y=100, you may wish to choose a normalized model. These models don't fit the bottom and top plateaus, but rather force the bottom plateau to equal 0 and the top plateau to equal 100. Only choose a 'normalized response' equation when you have determined the values that define 0 and 100 very precisely. Just because the data have been normalized [doesn't mean to have to constrain the curve in that way.](#)^[243]

Special dose-response models

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

[Asymmetrical \(five parameter\)](#)^[267]

[Biphasic dose-response](#)^[269]

[Bell-shaped dose-response](#)^[271]

[Operational model - Depletion](#)^[272]

[Operational model - Partial agonist](#)^[275]

[Gaddum/Schild EC50 shift](#)^[277]

[Allosteric EC50 shift](#) 

[ECanything](#) 

2.7.1.6 Pros and cons of normalizing the data

The dose-response model has four parameters: the bottom plateau, the top plateau, the EC50, and the slope factor (which is often constrained to a standard value).

The main goal of fitting the dose-response curve in many situations is to determine the best-fit value of the EC50, which is the concentration that provokes a response halfway between the top and bottom plateaus. If those plateaus are not well defined, the EC50 will be very uncertain. Think of it this way: If you have not defined "100" and "0" very precisely, you also have not defined "50" precisely, and therefore cannot determine the EC50 precisely.

One way to solve the problem is to constrain the Top or Bottom, or both, to control values. Another 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 force the bottom plateau to equal 0 and the top plateau to equal 100. Only choose a 'normalized response' equation when you have determined the values that define 0 and 100 very precisely.

Prism makes it easy to normalize the data so the values run from 0% to 100%. Simply click Analyze, choose the Normalize analysis., and define how 0% and 100% are defined. When fitting a dose-response curve, you can fit either the raw data or normalized data.

Notes:

- It is not necessary to normalize before fitting dose-response data. In many cases, it is better to show the actual data.
- You can only plot several different dose-response curves on one graph using one axis when they are comparable. If the different experiments measured different variables, normalizing puts them into comparable units. This can be useful.
- Whether or not you choose to normalize your data, you still need to choose how to fit the data. Do you want Prism to find best-fit values for the Top and Bottom plateaus? Or do you want those plateaus to be determined by control data?

- If you normalize your data, you can choose one of the normalized dose-response equations. These constrain the the curve to run from 0% to 100%. This kind of constraint only makes sense, when 0% and 100% are defined by good control data. If the definitions of 0% and 100% are ambiguous, then so is the definition of "50%", and thus the EC50 is also ambiguous.
- Just because you chose to normalize your data doesn't mean you must constrain the curve to run from 0 to 100%. You may prefer to have Prism fit those two plateaus.
- If you don't normalize your data, you can use the Constrain tab to fix Top and Bottom to values determined from control experiments. So the decision to constrain Top and Bottom is quite distinct from the decision to normalize your data before fitting.
- It is possible, and can be reasonable, to fix one of those parameters (Top or Bottom) to a constant value but not the other.
- If you normalize, [don't also choose to differentially weight the data](#)^[35].

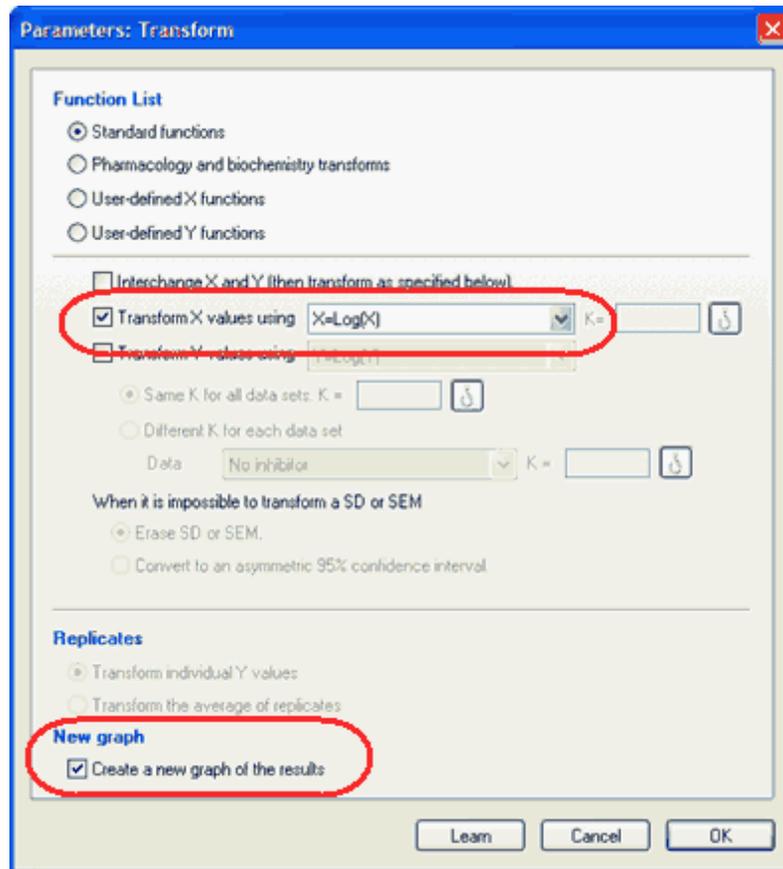
2.7.1.7 Converting concentration to log(concentration)

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.



3. Go to the newly created graph.

4. Click Analyze, choose nonlinear regression, and then choose the dose-response curve.

2.7.1.8 The term "logistic"

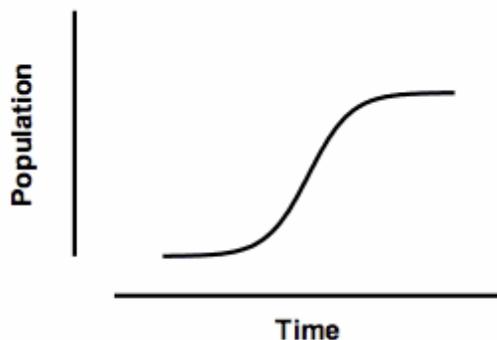
The term *logistic* has three meanings which have little relationship to each other (1).

Meaning 1: Logistic population growth

The term "logistic" was first invented in the nineteenth century to describe population growth curves. The idea is pretty simple. Population growth is limited, so can't ever exceed some value we'll call N_{max} . The rate of change of population at any time t is proportional to the number of individuals alive at that

time (Nt). But population growth slows down as it reaches the maximum, so is also proportional to $(N_{\max} - Nt)$. So the rate of change of population is proportional to $Nt(N_{\max} - Nt)$.

[Integrate that differential equation](#), and the result is called a logistic equation. It defines a sigmoidal shaped curve that defines the population at any time. The model has three parameters: the starting population, the maximum population, and the time it takes to reach half-maximal. Sometimes it is modified to add a fourth parameter to define the steepness of the curve.

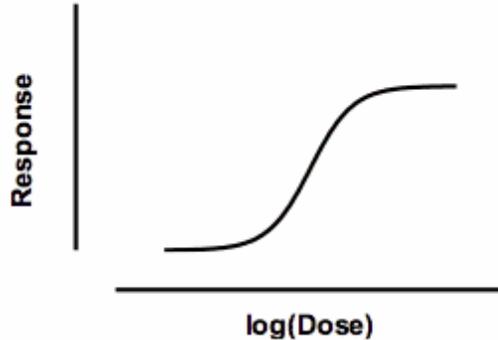


This curve was used by demographers in the past, but actually doesn't do a very good job of describing the growth of human populations. It is still used to model the growth of tumors, and to model the fraction of a population that uses a new product (like a mobile phone).

This model is also used for autocatalytic reactions, where the product of the reaction is also a catalyst for that reaction. With this kind of reaction, the rate of product accumulation is proportional to the concentration of product already produced times the concentration of remaining substrate. This has the same mathematical form as the population growth model. The graph is identical to the one above, except the Y-axis would be the concentration of the product produced by the enzyme reaction (instead of population).

Meaning 2: "Logistic" dose-response curves

Bindslev reviews the history of the many equations used to describe dose response curves(2). A common equation, based on work by Hill, defines a response as a function of the minimum response, the maximum response, the concentration required to evoke a response half-way between the minimum and maximum (the EC_{50}), and a parameter that describes how steep the curve is. In Prism, this equation is called the log(agonist) vs response curve -- Variable slope (four parameters).



Response at any dose is defined by an equation with four parameters: minimum response, maximum response, the $\log(\text{EC}_{50})$, and the slope factor (also called the Hill slope). Surprisingly, this equation has the same mathematical form as the logistic equation describing population growth, even though the two uses are entirely different. For this reason, the dose-response equation is sometimes called a "logistic" equation. Since there are four parameters, it is sometimes called the four parameter logistic equation, abbreviated 4PL.

This dose-response model describes many systems, but it is purely empirical. It does not correspond to any molecular model (except, perhaps, when the Hill slope is an integer).

Note the huge differences between the use of a logistic equation to describe population growth, and its use to describe dose-response curves.

- The X axis of the population growth curve is time. The X axis of the dose-response curve is the logarithm of dose or concentration.
- The equation for population growth comes from theory. The equation for dose-response relationships is empirical.
- Despite its name, no logarithms are used in the logistic equation for population growth. The X axis of the logistic dose-response curve is the logarithm of dose or concentration.

Meaning 3: Logistic regression

The third use of the word *logistic*, completely unrelated to the other two uses, is logistic regression. This is a multiple regression method used when the outcome (dependent variable) has two possible outcomes. The point of the equation is to predict the probability, P , of an outcome from a set of independent variables. To create a model that always predicts a value for P that

is between 0 and 1 the logistic regression model actually predicts the natural logarithm of the odds. The function that computes the natural logarithm of the odds from a fraction is called the *logit* function (pronounced with a long O and a soft G), so regression used to predict the logit of a probability from multiple independent variables is called *logistic regression*.

References

1. J.S. Cramer. [The origins and development of the logit model](#). Chapter 9 of Logit models from economics and other fields, Cambridge University Press, 2003

2. N. Bindslev, [Hill in Hell](#) (pdf), Chapter 10 of [Drug-Acceptor Interactions](#), ISBN: 978-91-977071-0-7

2.7.1.9 50% of what? Relative vs absolute IC50.

IC50 can be ambiguous

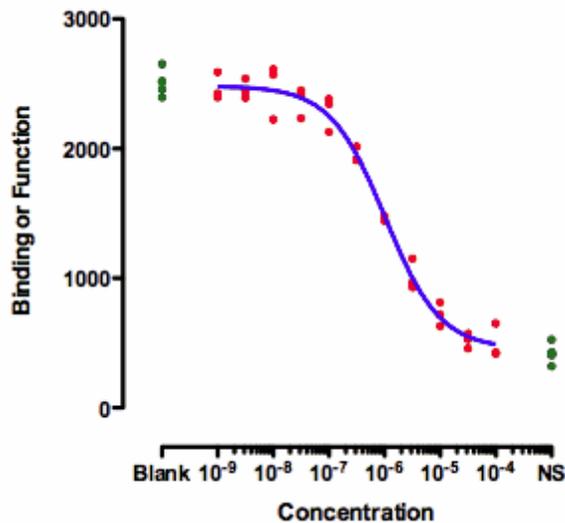
50% of what? There can be more than one answer in some situations.

The concepts of IC50 is fundamental to pharmacology. The IC50 is the concentration of an inhibitor where the response (or binding) is reduced by half. Seems simple enough. But when you actually go to fit data to determine these values, there are several complexities and ambiguities.

The rest of this article is about IC50 (I for inhibition, for downward sloping dose-response curves). All the ideas can be applied to stimulatory curves and EC50 (E for effective) as well. Just stand on your head when you view the figures!.

The ideal situation

This figure shows an ideal situation:



The green symbols show measurements made with controls. The ones on the left (Blank) have no inhibitor, so define "100%". The ones on the right are in the presence of a maximal concentration of a standard inhibitor, so define "0%". The data of the experimental dose-response curve (red dots) extend all the way between the two control values.

When fitting this curve, you need to decide how to fit the top plateau of the curve. You have three choices:

- Fit the data only, ignoring the Blank control values.
- Average the Blank control values, and set the parameter Top to be a constant value equal to the mean of the blanks.
- Enter the blank values as if they were part of the dose-response curve. Simply enter a low dose, perhaps 10⁻¹⁰ or 10⁻¹¹. You can't enter zero, because zero is not defined on a log scale.

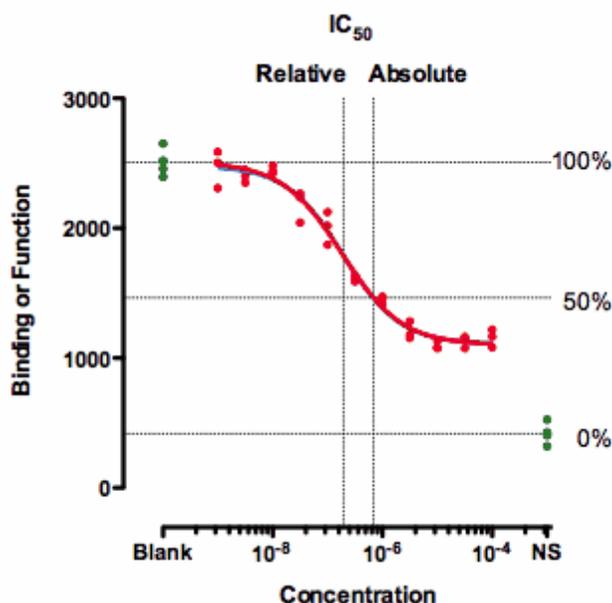
The results will be very similar with any of these methods, because the data form a complete dose-response curve with a clear top plateau that is indistinguishable from the blank. I prefer the third method, as it analyzes all the data, but that is not a strong preference.

Similarly, there are three ways to deal with the bottom plateau: Fit the data only, set Bottom to be a constant equal to the average of the NS controls, and put the NS controls into the fit as if they were a very high concentration of inhibitor.

That is the ideal situation. There is no ambiguity about what IC50 means.

A potentially ambiguous situation

The graph below shows an unusual situation where the inhibition curve plateaus well above the control values (NS) defined by a high concentration of a standard drug. This leads to alternative definitions of IC50.



Clearly, a single value cannot summarize such a curve. You'd need at least two values, one to quantify the middle of the curve (the drug's potency) and one to quantify how low it gets (the drug's maximum effect).

The graph above shows two definitions of the IC50.

The **relative IC50** is by far the most common definition, and the adjective *relative* is usually omitted. It is the concentration required to bring the curve down to point half way between the top and bottom plateaus of the curve. The NS values are totally ignored with this definition of IC50. This definition is the one upon which classical pharmacological analysis of agonist and antagonist interactions is based. With appropriate consideration of the biological system and concentrations of interacting ligands, estimated Kd values can often be derived from the IC50 value defined this way (not so for the "so-called absolute IC50" mentioned below).

The concentration that provokes a response halfway between the Blank and the NS value is sometimes called the **absolute IC50**. The horizontal dotted

lines show how 100% and 0% are defined, which then defines 50%. This term is not entirely standard. Since this value does not quantify the potency of a drug, the authors of the International Union of Pharmacology Committee on Receptor Nomenclature (1) think that the concept of absolute IC50 (and that term) is not useful (R. Neubig, personal communication). I agree.

The concept (but not the term "absolute IC50") is used to quantify drugs that slow cell growth. The [abbreviation GI50](#) is used for what we call here the absolute IC50. It is also used by the Environmental Protection Agency (EPA) in [evaluating endocrine disrupters \(Appendix A\)](#). That document uses the term IC50 to refer to the absolute IC50, and the term EC50 to refer to the relative IC50. It doesn't use the terms *relative* and *absolute*.

If you really want to use the absolute IC50, [the next page has instructions for fitting a curve to find it](#)²⁵¹.

Reference

1. R. R. Neubig et al. [International Union of Pharmacology Committee on Receptor Nomenclature and Drug Classification. XXXVIII. Update on terms and symbols in quantitative pharmacology](#). *Pharmacol Rev* (2003) vol. 55 (4) pp. 597-606

[Download the Prism file used to create all the graphs in this article.](#)

2.7.1.10 Fitting the absolute IC50

Fitting a dose-response curve to find the absolute IC50

The concept of an absolute IC50 is not standard, and many find it not to be useful. But if you do, it is not hard to fit a curve to determine it. Appendix A of [this EPA document](#), give the needed equation (which I have generalized a bit, so not require that the data already be normalized).

$$\text{Fifty} = (\text{Top} + \text{Baseline}) / 2$$
$$Y = \text{Bottom} + (\text{Top} - \text{Bottom}) / (1 + 10^{((\text{LogIC50} - X) * \text{HillSlope} + \log((\text{Top} - \text{Bottom}) /$$

Note the distinction between the parameter *Bottom* and *Baseline*. *Bottom* is the Y value of the bottom plateau of the curve itself. *Baseline* is the Y value that defines 0% -- maximal inhibition by a standard drug. You'll definitely want to constrain *Baseline* to be a constant value based on controls. You may also want to constrain *Top*.

[Download the Prism file](#) that fits that equation to make the graph shown above.

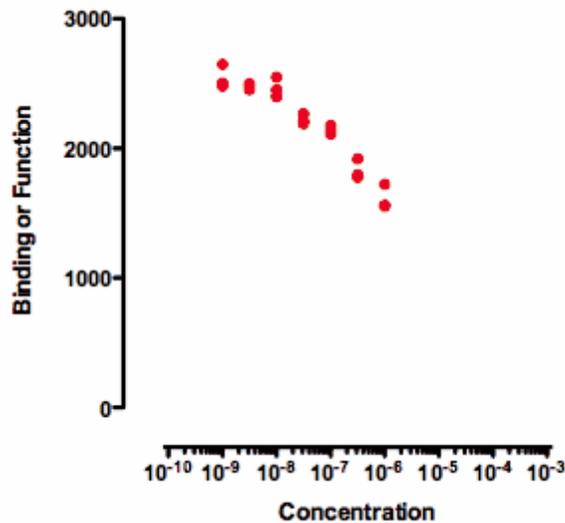
When fitting data to that equation, don't forget to constrain Baseline and Top to appropriate values determined by controls. Additionally, this file contains another graph where the data are already normalized to run from 0 to 100%. These data are fit to a simpler equation where Baseline is set to equal zero, and Top is set to equal 100. These are hard wired into the equation, so you don't have to remember to constrain those two parameters to constant values.

An alternative approach for normalized data

Here is an alternative approach you can use if your data are normalized. It does not require entering a user-defined equation.

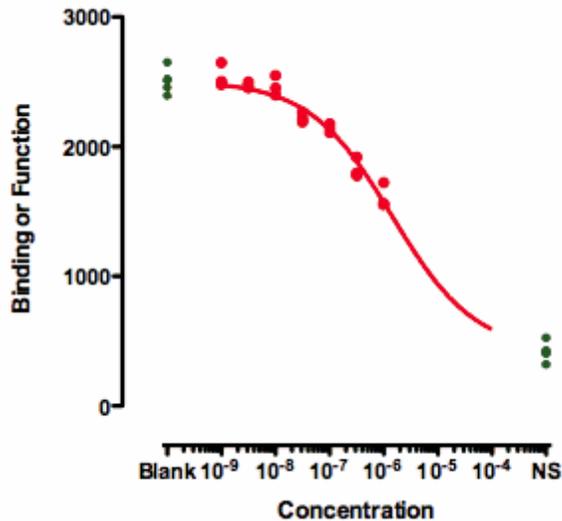
1. Make sure that your data are normalized to some controls. That means the response at the left end (low concentrations) is near 100%, and the response at higher concentrations is above 0%. If the response plateaus at 0%, then an absolute and relative IC50 are the same, and you can just fit the usual dose-response curve to find the IC50.
2. At the bottom of the data table, add a new row of data. Enter 50 into each Y column. Leave X blank for this row.
3. Use nonlinear regression to fit the data to the log(inhibitor) vs. response (variable slope) curve.
4. On the first (fit) tab of the nonlinear regression dialog, check the option: "Interpolate unknowns from standard curve." Of course, these data are not a standard curve, and there are unknowns. But this option asks Prism to interpolate the X value of the curve when Y=50.
5. On the Constrain tab, consider constraining Top to have a constant value of 100. If you have used good controls to normalize your data, then you know the top plateau of the curve has to be Y=100, so should tell Prism to use that constraint.
6. The IC50 reported as part of the main results table will be the relative IC50.
7. Look on the additional results page of interpolated results to see the value of X when Y=50. This is the absolute IC50.

2.7.1.11 Incomplete dose-response curves



Any attempt to determine an IC₅₀ by fitting a curve to the data in the graph above will be useless. Prism might, or might not, be able to fit a dose-response curve to the data. But if the curve fits, the value of the IC₅₀ is likely to be meaningless and have a very wide confidence interval. The data simply don't really define a top plateau (which would define 100) and certainly doesn't provide even a hint of a bottom plateau (which would define 0). If data haven't defined 100 or 0, then 50 is undefined too, as is the IC₅₀.

If you also have control values that define 100 and 0, then the curve can be easily fit. The curve below was created by fitting a dose response curve, but constraining the Top plateau to be a constant value equal to the mean of the Blanks values, and the Bottom plateau equal to the mean of the NS values. Note that the blank and NS values are shown in green.



The value of the IC50 fit this way only makes sense if you assume that higher concentrations of the inhibitor would eventually inhibit down to the NS values. That is an assumption that can't be tested with the data at hand.

The [distinction between relative and absolute IC50](#)^[248] doesn't really apply to these data. Because the data don't define a bottom plateau, the IC50 can only be defined relative to the NS control values.

2.7.1.12 Troubleshooting fits of dose-response curves

First consult the analysis checklists

If your goal is to interpolate from the standard curve, look at the [analysis checklist designed for that purpose](#)^[223].

If your goal is to fit a model to determine parameter values, review the general [analysis checklist for fitting a model](#)^[218].

Questions to ask about dose-response curves

- Are the X values logarithms of concentrations? If not, use the Transform analysis to make a new results table where the X values are logarithms. Note that stretching an axis to a log scale is not at all the same thing. Prism's nonlinear regression fits the values it is given. Using the Format Axis dialog to stretch the X axis to a logarithmic scale does not change the values the curve fitter sees.
- Is the top plateau defined by the data? If not, can it be constrained? If the top plateau is not defined either by the data or by a constraint, then any fit of a dose-response curve is unlikely to be useful.

- Is the bottom plateau defined by the data? If not, can it be constrained? If the bottom plateau is not defined either by the data or by a constraint, then any fit of a dose-response curve is unlikely to be useful.
- Were the data normalized? To what values? If the data were normalized, consider constraining the top and bottom plateaus to be 0 and 1 (or 100) in the Constraint tab?
- Hill slope fixed? A Hill slope of 1.0 or -1.0 is commonly seen in many systems, but not all.
- Four parameter assumes symmetry. The usual equations to fit dose response curves have four parameters (top, bottom, EC50, Hill Slope), and define symmetrical curves. You can choose an equation that [adds a fifth parameter to fit asymmetrical curves](#)²⁶⁷.

2.7.2 Dose-response - Stimulation

2.7.2.1 Equation: log(agonist) vs. response

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

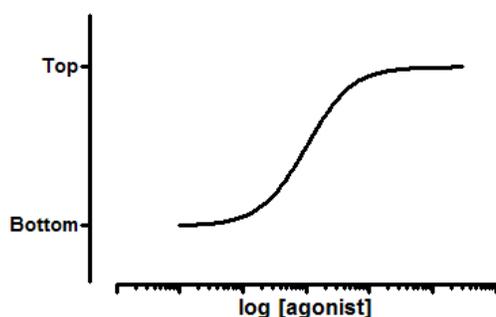
concentrations, use Prism to [transform the X values to logs](#)²⁴⁴.

From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Stimulation" and then choose the equation $\log(\text{Agonist})$ vs. response.

If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Model

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



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.

2.7.2.2 Equation: $\log(\text{agonist})$ vs. response -- Variable slope

Introduction

Many $\log(\text{dose})$ response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the EC50 of the agonist - the concentration that provokes a response half way between the basal (Bottom) response and the maximal (Top) response.

Many dose-response curves have a standard slope of 1.0. This model does not

assume a standard slope but rather fits the Hill Slope from the data, and so is called a **Variable slope** model. This is preferable when you have plenty of data points. It is also called a four-parameter dose-response curve, or four-parameter logistic curve, abbreviated 4PL.

Step by step

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

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to [transform the X values to logs](#)²⁴⁴.

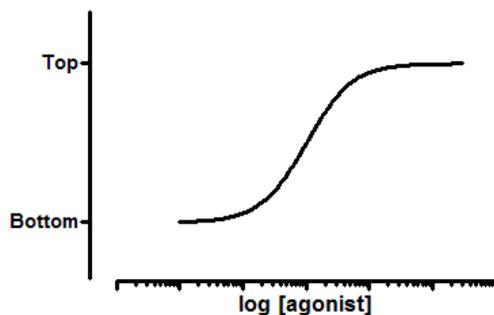
From the data table, click Analyze, choose nonlinear regression, choose the panel of equations "Dose-response curves - Stimulation" and then choose the equation "*log(Agonist) vs. response -- Variable slope*".

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

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



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.

2.7.2.3 Equation: log(agonist) vs. normalized response

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](#)²⁵⁶. 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.

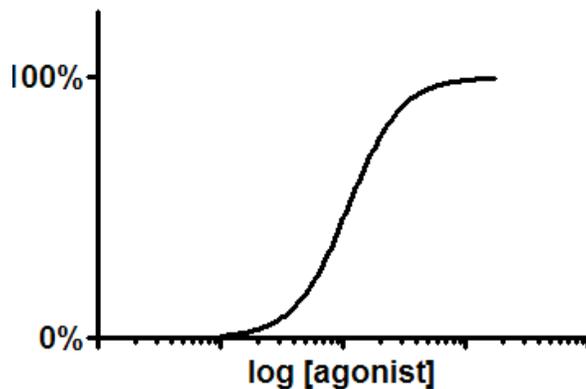
If you prefer to enter concentrations, rather than the logarithm of

concentrations, use Prism to [transform the X values to logs](#)²⁴⁴.

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

$$Y=100/(1+10^{((\text{LogEC50}-X))})$$



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.

2.7.2.4 Equation: log(agonist) vs. normalized response -- Variable slope

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](#)²⁵⁶. 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

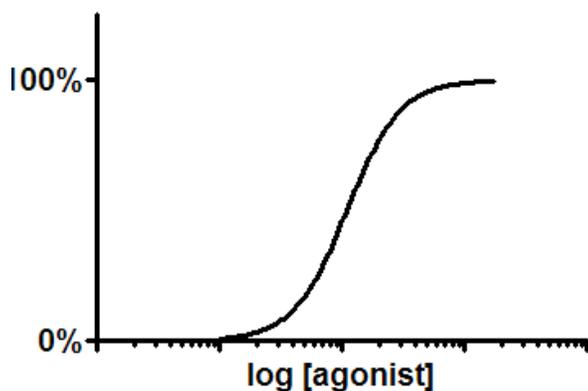
Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to [transform the X values to logs](#)²⁴⁴.

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

$$Y=100/(1+10^{((\text{LogEC50}-X)*\text{HillSlope}))}$$



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.

2.7.3 Dose-response - Inhibition

2.7.3.1 Equation: log(inhibitor) vs. response

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the IC50 of the inhibitor - the concentration that provokes a response half way between the maximal (Top) response and the maximally inhibited (Bottom) response.

This model assumes that the dose response curves has a **standard slope**, equal to a Hill slope (or slope factor) of -1.0. This is the slope expected when a ligand binds to a receptor following the law of mass action, and is the slope expected of a dose-response curve when the second messenger created by receptor stimulation binds to its receptor by the law of mass action. If you don't have many data points, consider using the standard slope model. If you have lots of data points, pick the variable slope model to determine the Hill slope from the data.

Step by step

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

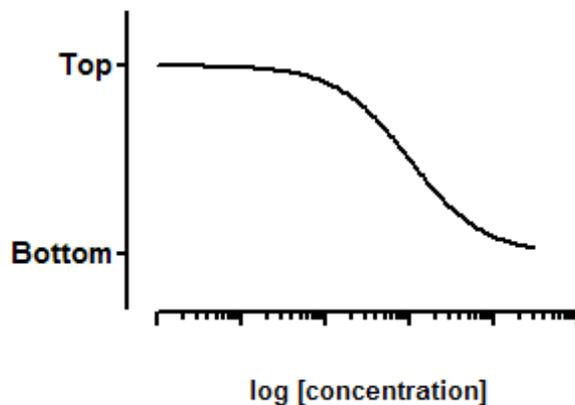
If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to [transform the X values to logs](#)²⁴⁴.

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

$$Y = \text{Bottom} + (\text{Top} - \text{Bottom}) / (1 + 10^{((X - \text{LogIC50}))})$$



Interpret the parameters

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at $Y=50$. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may give a response nowhere near "50". Prism reports both the IC50 and its log.

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

2.7.3.2 Equation: *log(inhibitor) vs. response* -- Variable slope

Introduction

Many *log(inhibitor) vs. response* curves follow the familiar symmetrical sigmoidal shape. The goal is to determine the IC50 of the inhibitor - the concentration that provokes a response half way between the maximal (Top) response and the maximally inhibited (Bottom) response.

Many inhibitory dose-response curves have a standard slope of -1.0. This model does not assume a standard slope but rather fits the Hill Slope from the data, and so is called a **Variable slope** model. This is preferable when you have plenty of data points. It is also called a four-parameter dose-response curve, or four-parameter logistic curve, abbreviated 4PL.

Step by step

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

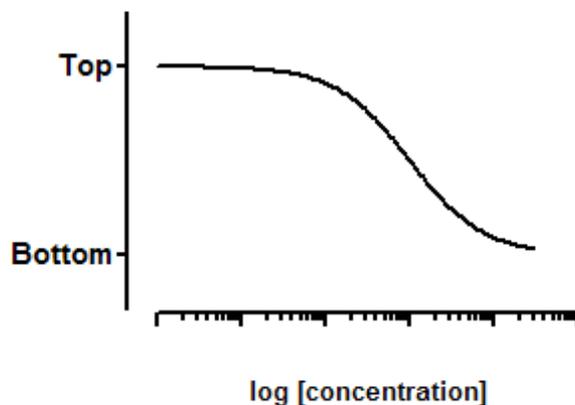
If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to [transform the X values to logs](#)²⁴⁴.

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

$$Y = \text{Bottom} + (\text{Top} - \text{Bottom}) / (1 + 10^{((\text{LogIC50} - X) * \text{HillSlope}))})$$



Interpret the parameters

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may give a response nowhere near "50". Prism reports both the IC50 and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of -1.0 is standard, and you should consider constraining the Hill Slope to a constant value of -1.0. A Hill slope more negative than -1 (say -2) is steeper.

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

2.7.3.3 Equation: log(inhibitor) vs. normalized response

Introduction

Many log(inhibitor) vs. response curves follow the familiar symmetrical sigmoidal shape.

If you have good control data, it can make sense to **normalize** the response to run between 0% and 100%. This model assumes that the data have been normalized, so forces the curve to run from 100% down to 0%. The goal is to determine the IC₅₀ of the inhibitor - the concentration that provokes a response equal to 50%.

It only makes sense to fit a normalized model when you are sure you have defined 0% and 100% quite accurately. If your data define a complete sigmoidal curve, it is best to fit the entire curve and let Prism [fit the Top and Bottom plateaus](#)²⁶¹. 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

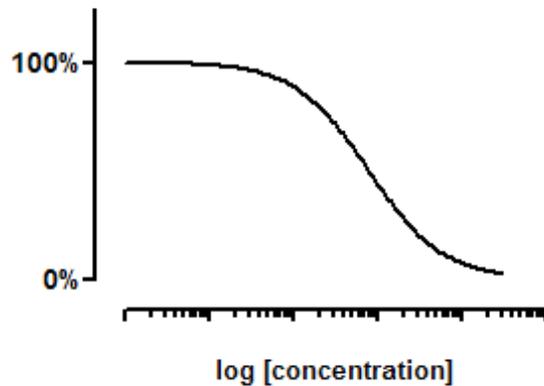
Create an XY data table. Enter the logarithm of the concentration of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to [transform the X values to logs](#)²⁴⁴.

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

$$Y=100/(1+10^{((X-\text{LogIC50}))})$$



Interpret the parameter

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at $Y=50$. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may give a response nowhere near "50". Prism reports both the IC50 and its log.

2.7.3.4 Equation: *log(inhibitor) vs. normalized response* – Variable slope

Introduction

Many *log(inhibitor) vs. response* curves follow the familiar symmetrical sigmoidal shape.

If you have good control data, it can make sense to **normalize** the response to run between 0% and 100%. This model assumes that the data have been normalized, so forces the curve to run from 100% down to 0%. The goal is to determine the IC50 of the inhibitor - the concentration that provokes a response equal to 50%.

It only makes sense to fit a normalized model when you are sure you have defined 0% and 100% quite accurately. If your data define a complete

sigmoidal curve, it is best to fit the entire curve and let Prism [fit the Top and Bottom plateaus](#)^[261]. 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

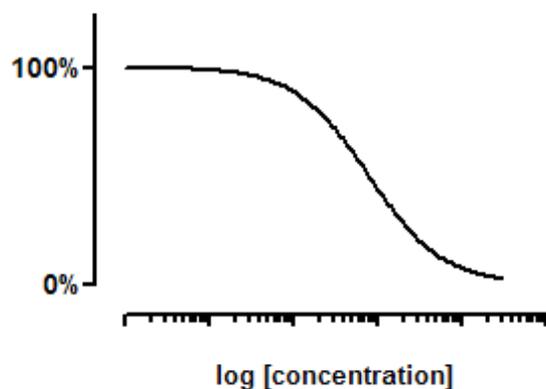
Create an XY data table. Enter the logarithm of the concentration of the inhibitor into X. Enter response into Y in any convenient units. Enter one data set into column A, and use columns B, C... for different treatments, if needed.

If you prefer to enter concentrations, rather than the logarithm of concentrations, use Prism to [transform the X values to logs](#)^[244].

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

$$Y=100/(1+10^{((\text{LogIC50}-X)*\text{HillSlope}))})$$



Interpret the parameters

IC50 is the concentration of agonist that gives a response half way between Bottom and Top. This is not the same as the response at Y=50. Depending on which units Y is expressed in, and the values of Bottom and Top, the IC50 may

give a response nowhere near "50". Prism reports both the IC50 and its log.

HillSlope describes the steepness of the family of curves. A HillSlope of -1.0 is standard, and you should consider constraining the Hill Slope to a constant value of -1.0. A Hill slope more negative than -1 (say -2) is steeper.

2.7.4 Dose-response -- Special

2.7.4.1 Asymmetrical (five parameter)

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**, abbreviated 5PL.

Step by step

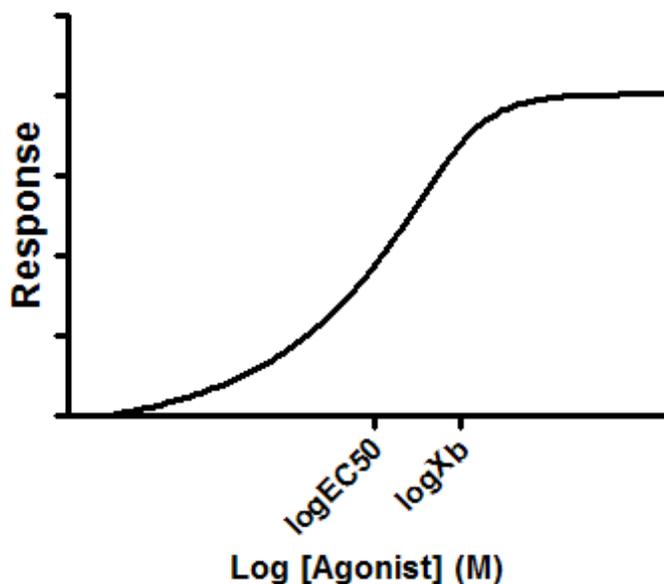
Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units.

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



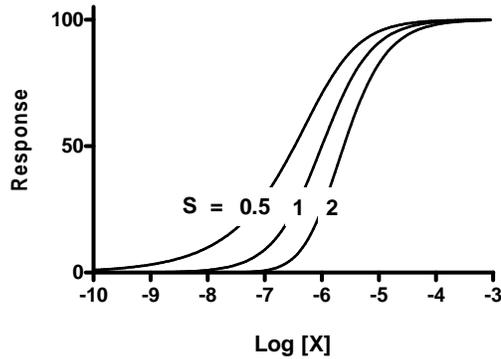
Interpret the parameters

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. Using the built in equation, Prism does not fit logXb, but you can do so using this equation:

$$\text{LogX}_b = \text{LogEC}_{50} + \left(\frac{1}{\text{HillSlope}} \right) \cdot \text{Log} \left(2^{1/S} - 1 \right)$$

You can rewrite the equation so Prism fits the logXb rather than the logEC50. [Clone](#)⁴⁰⁹ the built-in equation, and edit the copy. Remove the first line [LogXb = LogEC50 + (1/HillSlope) * Log((2^(1/S)) - 1)] from the equation. Then establish a [rule for the initial value](#)⁴³³ of the logXb to be 1 * (Value of X at Ymid).

Reference

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

2.7.4.2 Equation: Biphasic dose-response

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.

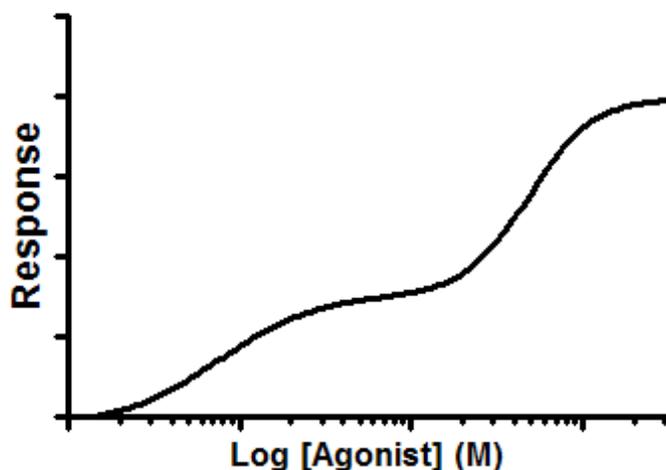
From the data table, click Analyze, choose nonlinear regression, and choose the

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.

2.7.4.3 Equation: Bell-shaped dose-response

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

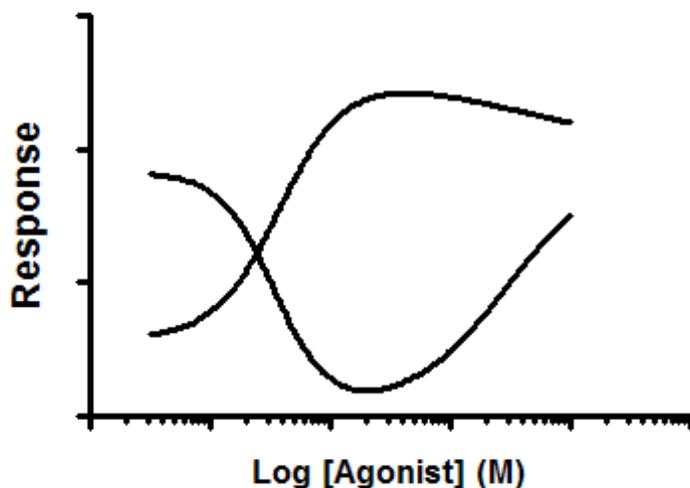
Create an XY data table. Enter the logarithm of the concentration of the agonist into X. Enter response into Y in any convenient units.

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. Note that when the curve goes up first (then down), this parameter named *Dip* would better be named *Peak*.

LogEC50_1 and **LogEC50_2** are the concentrations that give half-maximal stimulatory and inhibitory effects in the same units as X.

nH1 and **nH2** are the unitless slope factors or Hill slopes. Consider constraining these to equal 1.0 (stimulation) and -1 (inhibition).

2.7.4.4 Equation: Operational model - Depletion

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).

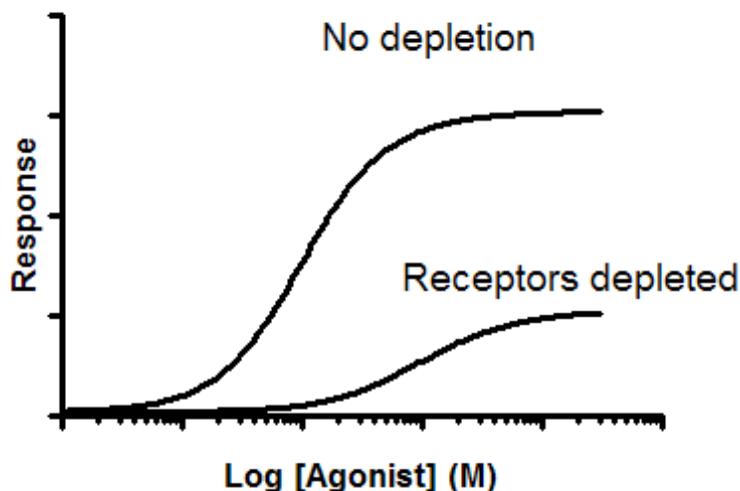
From the data table, click Analyze, choose nonlinear regression, and choose the panel of equations: Dose-Response -- Special. Then choose *Operational Model - Depletion*.

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

$$\text{operate} = \left(\frac{10^{\log KA} + 10^X}{10^{(\log \tau + X)}} \right)^n$$
$$Y = \text{Basal} + (\text{Effectmax} - \text{Basal}) / (1 + \text{operate})$$



Interpret the parameters

Effectmax is the maximum possible system response, in units of the Y axis. It is the top plateau of the dose-response curve obtained with a full agonist without receptor depletion. If your agonist isn't a full agonist, the EffectMax might be higher than the top plateau of the no depletion curve.

Basal is the response in absence of agonist, in same units as Y. If you have subtracted off any basal response, constrain basal to a constant value of zero.

KA is the agonist-receptor dissociation constant, in same units as X (usually molar). It measures the affinity of the full agonist for the receptors, which is the main goal of this kind of experiment. Prism reports both KA and its logarithm. It is not the same as the EC50.

tau is the transducer constant, a practical measure of efficacy. It is the inverse of the fraction of receptors that must be occupied by agonist to obtain the half-maximal response. If tau equals 10, that means that occupation of only 10% of the receptors leads to a half-maximal response. If tau equals 1.0, that means that it requires occupation of all the receptors to give a half-maximal response. This would happen in a tissue where the receptors had been substantially depleted. Because tau is a property of both the tissue and receptor system, it is not a direct measure of intrinsic efficacy, which is commonly defined as a property belonging only to an agonist-receptor pair, irrespective of the assay system in which it is measured. Prism reports both tau and its logarithm for each data set.

n is the Unitless transducer slope. It is similar to, but not identical to, the Hill slope. In most cases, n is constrained to a constant value of 1.0, in which case

all the dose-response curves will have Hill slopes of 1.0. If n does not equal 1.0, the Hill Slope does not equal either 1.0 or n .

Notes

Since Tau measures efficacy, Prism fits a different value of tau for each data set. Receptor depletion reduce the value of tau. The other parameters are fit globally, to find one best-fit value for all the data sets.

Reference

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

2.7.4.5 Equation: Operational model - Partial agonist

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*.

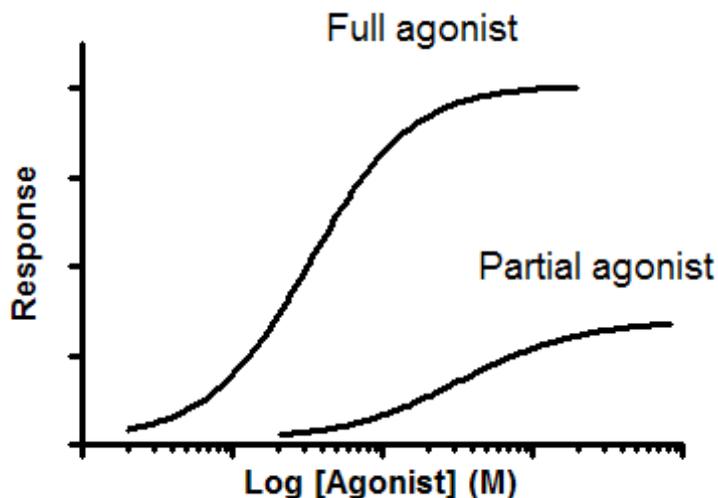
If you have subtracted off any basal response, consider constraining the parameter Basal to a constant value of zero.

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 τ equals 10, that means that occupation of only 10% of the receptors leads to a half-maximal response. If τ 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)

2.7.4.6 Equation: Gaddum/Schild EC50 shift

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.

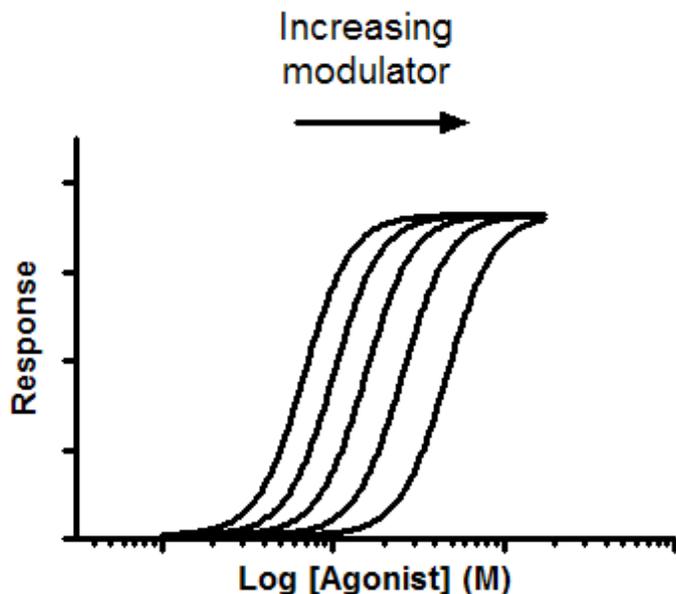
Step by step

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

$$\begin{aligned} \text{EC50} &= 10^{\text{LogEC50}} \\ \text{Antag} &= 1 + (B / (10^{(-1 * \text{pA2})}))^{\text{SchildSlope}} \\ \text{LogEC} &= \text{Log}(\text{EC50} * \text{Antag}) \\ Y &= \text{Bottom} + (\text{Top} - \text{Bottom}) / (1 + 10^{((\text{LogEC} - X) * \text{HillSlope})}) \end{aligned}$$


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.

When is the Schild model valid?

Colquhoun (1) has shown that the Schild model is valid whenever you can make these assumptions:

- The antagonist, B , is a true antagonist that, alone, does not change the conformation of the receptor.
- Binding of agonist, A , and antagonist, B , is mutually exclusive at every binding site.
- B has the same affinity for every binding site.
- The observed response is the same if the occupancy of each site by A is the same, regardless of how many sites are occupied by B .
- Measurements are made at equilibrium.

1. Colquhoun, D.. *Why the Schild method is better than Schild realised*. Trends Pharmacol Sci (2007) vol. 28 (12) pp. 608-14

2.7.4.7 Equation: EC50 shift

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

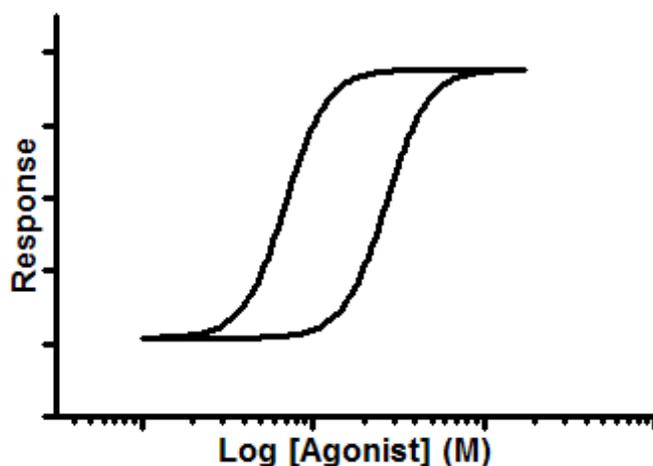
Create an XY data table. Enter the logarithm of the concentration of the agonist ligand into X. Enter response into Y in any convenient units. Enter data with no inhibitor into column A. Enter data collected with a constant concentration of inhibitor into column B.

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](#)^[277] and determine the pA2.

2.7.4.8 Equation: Allosteric EC50 shift

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

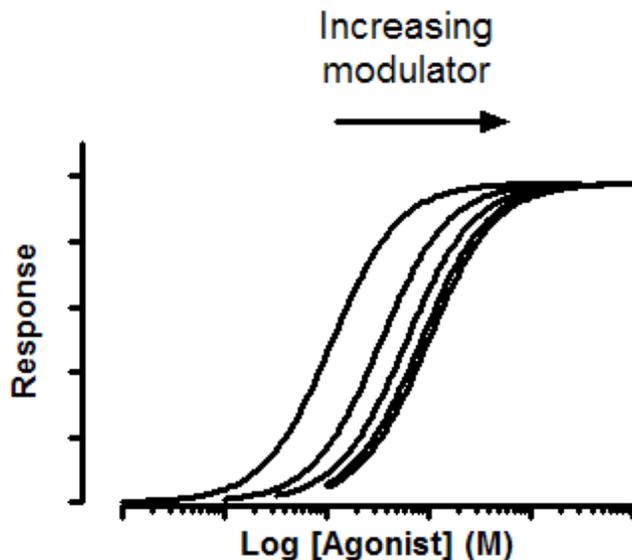
Create an XY data table. Enter the logarithm of the concentration of the agonist ligand into X. Enter response into Y in any convenient units. Enter data with no modulator into column A. Enter data collected with a constant concentration of modulator into column B. Repeat, if you have data, for column 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 α is less than 1.0, then the modulator reduces ligand binding. If α is greater than 1.0, then the modulator increases binding. In the example shown about, α 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 α , rather than α itself. This is because α 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 α and $\log(\alpha)$.

- This model assumes that the allosteric modulator is present in excess, so the concentration you added is very close to its free concentration. This model won't work when the concentration of allosteric modulator is limiting (as it is when G proteins alter agonist binding to many receptors). No explicit model can handle this situation. You need to define the model with an implicit equation (Y on both sides of the equals sign) and Prism cannot handle such equations.
- Note two important points about the progression of curves from left to right with increasing concentrations of the allosteric modulator. First, note that the maximum response doesn't change. Second, note that the effect of the modulator to right-shift the dose-response curve reaches a maximum as the modulator saturates its binding site.

Reference

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

2.7.4.9 Equation: ECanything

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.

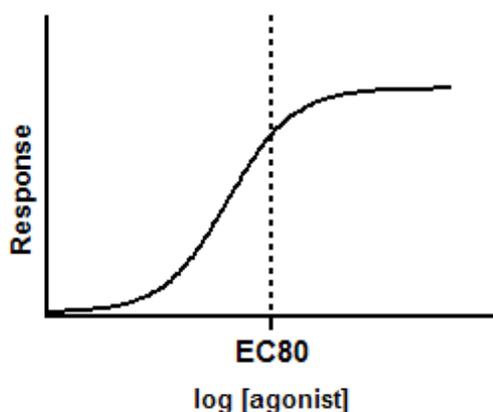
Consider constraining the parameter HillSlope to its standard value of 1.0 of -1. This is especially useful if you don't have many data points, and therefore cannot fit the slope very well.

If you have subtracted off any basal response, consider constraining Bottom to a constant value of 0.

Model

$$\log EC50 = \log ECF - (1/\text{HillSlope}) * \log(F/(100-F))$$

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



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.

IC_{anything}

This equation can also fit inhibitory data where the curve goes downhill rather than uphill. The best-fit value of the Hill Slope will be negative in this case. The result will always be reported as ECf. Let's say you set F=80. Then the ECf for inhibitory data would be the concentration (X value) required to bring the curve

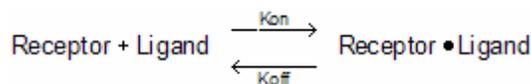
down to 80%. If you want the concentration that brings the curve down by 80%, to 20%, then you'd need to set F equal to 20.

2.7.5 Receptor binding - Key concepts

2.7.5.1 Law of mass action

What is the law of mass action?

Analysis of radioligand binding experiments is based on a simple model, called the law of mass action. This model assumes that binding is reversible.



Binding occurs when ligand and receptor collide due to diffusion, and when the collision has the correct orientation and enough energy. The rate of association is:

$$\text{Number of binding events per unit of time} = [\text{Ligand}] \cdot [\text{Receptor}] \cdot k_{\text{on}}$$

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:

$$\text{Number of dissociation events per unit time} = [\text{ligand} \times \text{receptor}] \times k_{\text{off}}$$

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:

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

Meaning of Kd

Rearrange that equation to define the equilibrium dissociation constant Kd.

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

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.

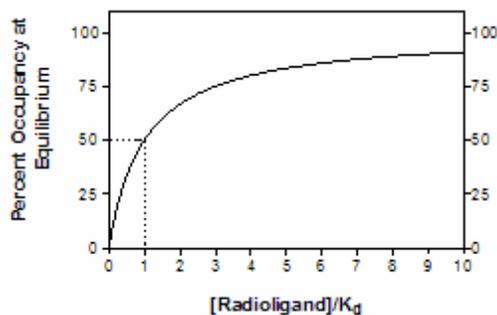
$$\text{Fractional occupancy} = \frac{[\text{Ligand} \cdot \text{Receptor}]}{[\text{Total Receptor}]} = \frac{[\text{Ligand} \cdot \text{Receptor}]}{[\text{Receptor}] + [\text{Ligand} \cdot \text{Receptor}]}$$

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]=K_D, 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.

2.7.5.2 Nonspecific binding

In addition to binding to receptors of interest, radioligands also bind to other sites. Binding to the receptor of interest is called specific binding, while binding to the other sites is called nonspecific binding. This means that nonspecific binding can represent several phenomena:

- In most cases, the bulk of nonspecific binding represents some sort of interaction of the ligand with membranes. The molecular details are unclear, but nonspecific binding depends on the charge and hydrophobicity of a ligand – but not its exact structure.
- Nonspecific binding can also be binding to receptors, transporters, or other proteins not of interest to the investigator. For example binding of the adrenoceptor agonist, epinephrine, to serotonin receptors or metabolic enzymes can be considered “nonspecific”.
- Nonspecific binding can also be binding to the filters used to separate bound from free ligand.

Nonspecific binding is usually (but not necessarily) proportional to the concentration of radioligand (within the range it is used). Add twice as much radioligand, and you will see twice as much nonspecific binding.

Nonspecific binding is detected by measuring radioligand binding in the presence of a saturating concentration of an unlabeled drug that binds to the receptors. Under those conditions, virtually all the receptors are occupied by the unlabeled drug so the radioligand can only bind to nonspecific sites. Subtract the nonspecific binding at a particular concentration of radioligand from the total binding at that concentration to calculate the specific radioligand binding to receptors.

Which unlabeled drug should you use for determining nonspecific binding? The obvious answer is to use the same compound as the radioligand, but in its unlabeled form. In many cases this is necessary, as no other drug is known to bind to the receptors. But most investigators avoid using the same compound as the hot and cold ligand and prefer to define nonspecific binding with a drug that is chemically distinct from the radioligand but which binds to the same receptor.

What concentration of unlabeled drug should you use? You want to use enough to block virtually all the specific radioligand binding, but not so much that you cause more general physical changes to the membrane that might alter binding. If you are studying a well-characterized receptor, a useful rule-of-thumb is to use the unlabeled compound at a concentration equal to 100 times its K_d 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.

2.7.5.3 Ligand depletion

In many experimental situations, you can assume that a very small fraction of the ligand binds to receptors (or to nonspecific sites). In these situations, you can also assume that the free concentration of ligand is approximately equal to the concentration you added. This assumption vastly simplifies the analysis of binding experiments, and the standard analysis methods depend on this assumption.

In other situations, a large fraction of the ligand binds to the receptors (or binds nonspecifically). This means that the concentration of ligand free in the solution does not equal the concentration you added. The discrepancy is not the same in all tubes or at all times. The free ligand concentration is depleted by binding.

Many investigators use this rule of thumb: If less than 10% of the ligand binds, don't worry about ligand depletion; if more than 10% of the ligand binds, you have three choices:

- Change the experimental conditions. Increase the reaction volume without changing the amount of tissue. The problem with this approach is that it requires more radioligand, which is usually very expensive.
- Measure the free concentration of ligand in every tube. This is possible if you use centrifugation or equilibrium dialysis, but is quite difficult if you use vacuum filtration to remove free radioligand.
- Use analysis techniques that adjust for the difference between the concentration of added ligand and the concentration of free ligand. Prism includes such models for analyzing [saturation](#)^[295] and [competition](#)^[318] 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.

2.7.5.4 The radioactivity web calculator

GraphPad Software provides a [free radioactivity calculator](#) on graphpad.com. Use it to perform seven common calculations.

Calculation	Description
Isotope decay	Calculates radioactive decay during a specified number of days. Select one of the common isotopes, or enter the half-life of another isotope.
Conc. of stock	Enter mCi/ml and Ci/mmol, which should be on the label. If you are using a molecule labeled with ^{125}I , the specific activity equals 2200 Ci/mmol 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 ^{125}I , 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.

Calculation	Description
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.

2.7.6 Receptor binding - Saturation binding

2.7.6.1 Key concepts: Saturation binding

What is saturation binding?

In a saturation binding experiment, you vary the concentration of radioligand and measure binding at equilibrium. The goal is to determine the K_d (ligand concentration that binds to half the receptor sites at equilibrium) and B_{max} (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](#)^[297].
- [Analyze the total binding only](#)^[292], inferring the amount of nonspecific binding from the shape of the total binding curve.
- Globally [fit total and nonspecific binding together](#)^[293].

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 uncomfortable 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](#)^[295] (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.

2.7.6.2 Equation: One site -- Total binding

Introduction

You don't have to measure nonspecific binding directly. Instead, you can determine Bmax and Kd by fitting only total binding by assuming that the amount of nonspecific binding is proportional to the concentration of radioligand.

Step by step

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

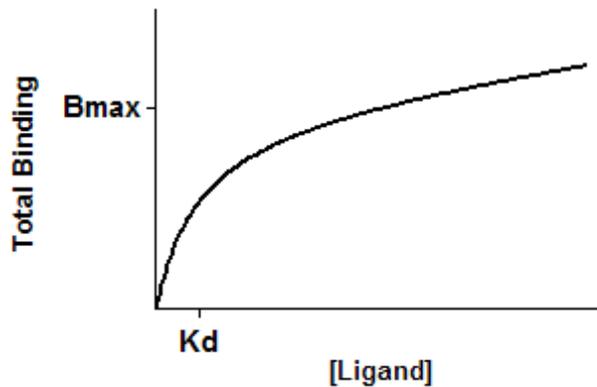
Use any convenient units for X and Y. The Kd will be reported in the same units as X, and the Bmax will be reported in the same units as Y.

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

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

Model

$$Y = B_{max} * X / (Kd + X) + NS * X + Background$$



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](#) 295.

2.7.6.3 Equation: One site -- Fit total and nonspecific binding

Introduction

In a saturation binding experiment, you vary the concentration of radioligand and measure binding. The goal is to determine the K_d (ligand concentration that binds to half the receptor sites at equilibrium) and B_{max} (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](#)²⁹⁷.
- Analyze the total binding only, inferring the amount of nonspecific binding from the shape of the total binding curve. [Learn more](#)²⁹².
- 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*.

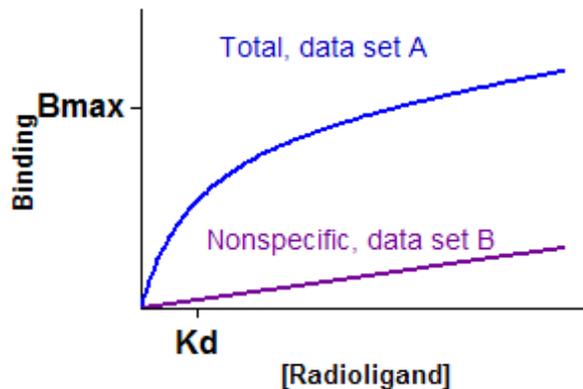
Consider constraining the parameter Background to a constant value of zero. This is the measured 'binding' when there is no radioligand binding added, so represents the counter background, if there is any.

Model

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

The <A> and 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](#)⁴¹ 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 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.

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.

2.7.6.4 Equation: One site -- Total, accounting for ligand depletion

Introduction

You don't have to measure nonspecific binding directly. Instead, you can determine Bmax and Kd by fitting only total binding by assuming that the amount of nonspecific binding is proportional to the concentration of radioligand.

If only a small fraction of radioligand binds, you can use a [simpler model](#)²⁹².

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.

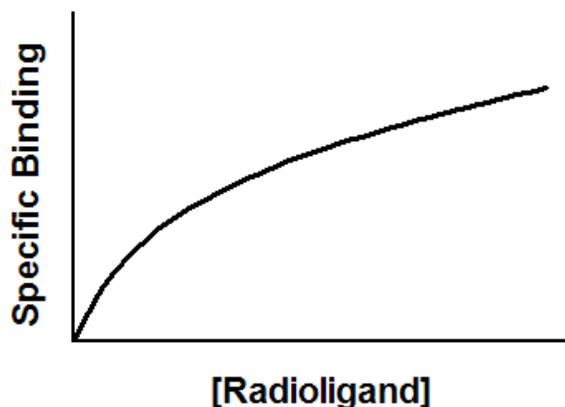
From the table of total binding, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *One site -- Total, accounting for ligand depletion*.

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

B_{max} is the maximum specific binding in cpm.

K_d 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](#)²⁹².

Reference

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

2.7.6.5 Equation: One site -- Specific binding

Introduction

In a saturation binding experiment, you vary the concentration of radioligand and measure binding. The goal is to determine the K_d (ligand concentration that binds to half the receptor sites at equilibrium) and B_{max} (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](#)²⁹².
- Globally analyze the total and nonspecific binding at one time. [Learn more.](#)²⁹³

Step by step

Create an XY data table. Enter radioligand concentration into X, and specific

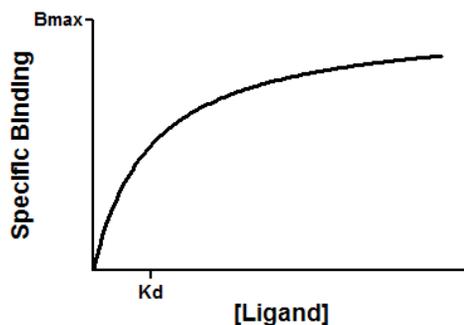
binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

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

$$Y = B_{\max} * X / (K_d + X)$$



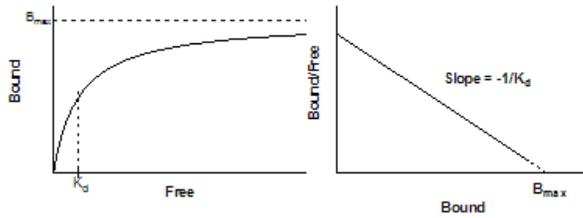
Interpret the parameters

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 Scatchard 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 B_{max} or K_d .

Don't use the slope and intercept of a linear regression line to determine values for B_{max} and K_d . If you do this, you won't get the most accurate values for B_{max} and K_d . 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 K_d and B_{max} .

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 $X=0$, $Y=B_{max}/K_d$ (previously determined by nonlinear regression). You need to do the calculation manually, and enter a number.
3. Into row 2 enter $X=B_{max}$ and $Y=0$. Again enter the number into the X column, not the text ' B_{max} '.
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](#)^[293], 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 $-1/K_d$).

2.7.6.6 Equation: One site -- Specific binding with Hill slope

Introduction

In a saturation binding experiment, you vary the concentration of radioligand and measure binding. The goal is to determine the K_d (ligand concentration that binds to half the receptor sites at equilibrium) and B_{max} (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](#)^[297].

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.

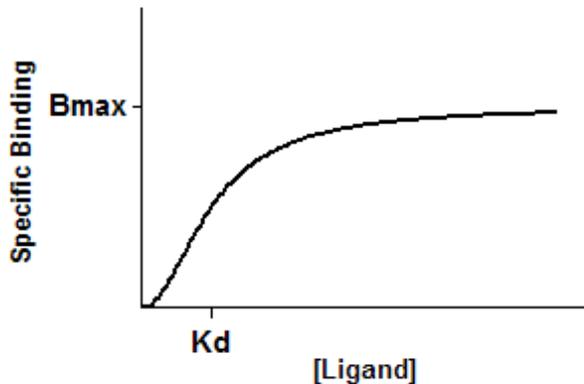
An alternative approach would be to enter total binding into column A, and nonspecific into column B. Then use the Remove Baseline analysis to subtract column B from column A, creating a new results table with the specific binding.

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

$$Y = B_{\max} * X^h / (K_d^h + X^h)$$



Note that the X axis is concentration, not log(concentration).

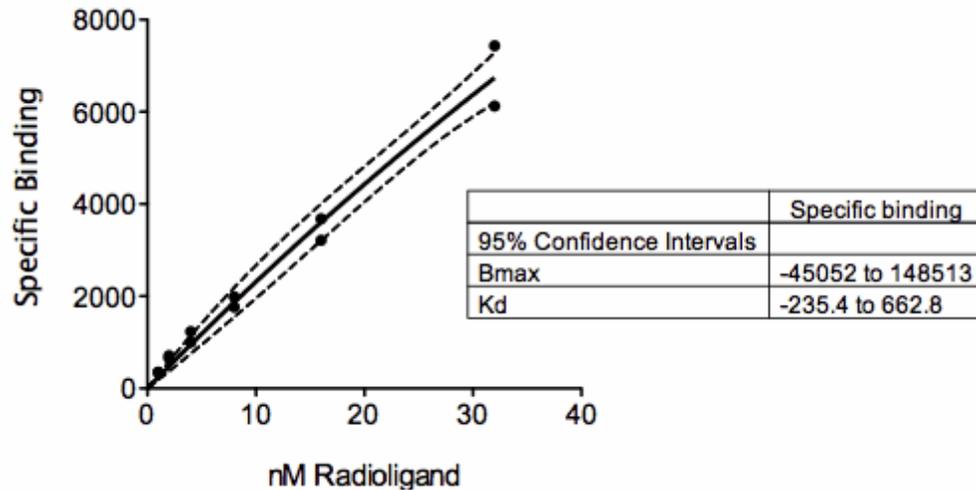
Interpret the parameters

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 K_d depends on why h is not 1.0.

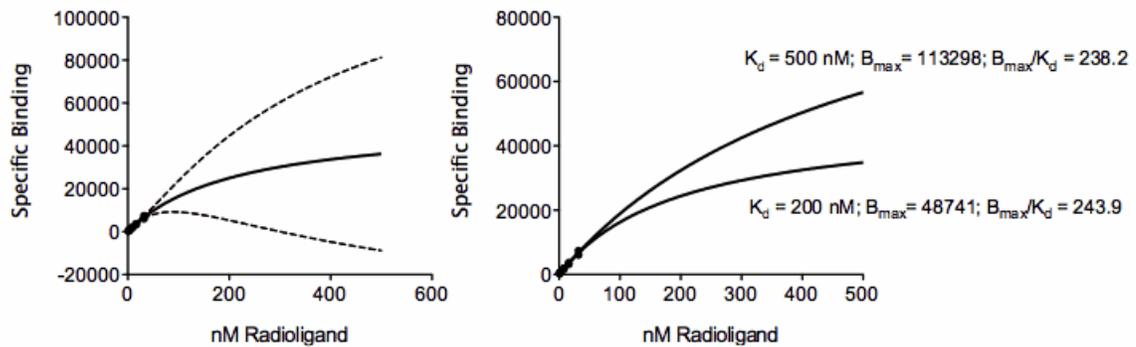
h is the Hill slope. It equals 1.0 when a monomer binds with no cooperativity to one site. When it is greater than 1.0, you see a sigmoidal look to the graph as shown above. This happens when the receptor or ligand has multiple binding sites with positive cooperativity. The Hill slope is less than zero when there are multiple binding sites with different affinities for ligand or when there is negative cooperativity.

2.7.6.7 Binding potential

The problem: Binding isotherms without going to high enough concentrations

The figure above shows a saturation binding equation fit to specific binding data. But all the concentrations are relatively low (compared to the K_d of binding) so the data almost form a straight line. The saturation binding model fits the data fine (solid curve), with a narrow 95% confidence band around the curve (dashed lines). But look at the confidence interval for B_{max} and K_d (shown in the box)! They are super wide, even descending into negative (impossible) values. With these data, the curve fit the data fine, but neither parameter in the model (B_{max} and K_d) were fit with reasonable confidence intervals. This is no surprise. The data don't show even a hint of plateauing, so the data simply don't define the B_{max} and K_d at all.

The problem is that the data are consistent with a system with a huge number of low affinity receptors (high B_{max} ; high K_d) or a smaller number of high affinity receptors (low B_{max} ; low K_d). The figure below shows two ways to visualize this. The graph on the left shows the same fit as the one above, but with the best-fit curve and its confidence bands extended out to larger concentrations. While the confidence band is tight near the points, it is super wide as it goes beyond concentrations with data. The graph on the right shows two different fits, with the K_d constrained to equal 200 nM or 500 nM. The curves are very different, yet both go near the data.



Another way to look at the problem with these data is that the B_{max} and K_d are correlated. Prism can report the covariance matrix as part of its nonlinear regression results. With only two parameters, there is only one value in that "matrix". The covariance between K_d and B_{max} (which can range from 0.0 to 1.0) is 0.9993. Prism can also report the dependency of each parameter (which also can range from 0.0 to 1.0). With only two parameters, both have the same dependency, which is 0.9986 for this example. This is not quite high enough for Prism to declare the results *ambiguous*, but the threshold for that designation (dependency > 0.9999) is arbitrary.

It is tempting to give up at this point and say that nothing can be determined without more data at higher concentrations. But in some systems, especially those using PET scanning to detect receptors, data like this are typical. It is impossible to use higher concentrations of ligand.

Fitting the binding potential

What can be learned from these data? The data are nearly linear. What does the slope of that line tell us? The slope, or rather the initial slope, is termed the *binding potential* (1). Don't be misled by the term potential. It has nothing to do with potential energy and thermodynamics. Rather it is a measure of how much binding will occur with a small amount of ligand.

If you start with the specific binding equation, take its derivative with respect to X (which is concentration), and set X equal to zero, you'll derive the initial slope of a specific binding curve. It turns out that this slope, the binding potential, equals B_{max}/K_d . B_{max} is measured in the same units as the Y values in the data. K_d is measured in the same units as the X values. So the binding potential has units equal to the Y units divided by the X units.

Prism can fit a specific binding curve, and also report the ratio of B_{max}/K_d with its confidence interval. To do this either create a new user-defined equation or (easier) clone the built in one. Then go to the last tab of the dialog that defines

the equation. “Transforms to Report”. Choose to report the ratio of two parameters, and define the numerator and denominator.

The screenshot shows the 'Transforms to Report' dialog box in GraphPad Prism. It has four tabs: 'Equation', 'Rules for Initial Values', 'Default Constraints', and 'Transforms to Report'. The 'Transforms to Report' tab is active. It is divided into two sections: 'Report transforms of best-fit parameters (with 95% CI)' and 'Report these combinations of best-fit parameters (with 95% CI)'. The first section has four rows, each with an empty text box, an equals sign, another empty text box, and a dropdown menu set to 'Asymmetrical'. The second section has four rows. The first row is highlighted with a red box and contains: 'Binding Potential' in the text box, an equals sign, 'P1/P2' in the dropdown, 'P1 = Bmax' in a dropdown, and 'P2 = Kd' in a dropdown. Below this section, there is a note: 'To use this equation, you must have already submitted off any nonspecific binding. One site -- Specific binding, no binding potential.' At the bottom, there are radio buttons for 'Least squares (ordinary) fit', 'Robust fit', and 'Automatic outlier elimination', and a 'Interpolate' checkbox. There are also 'Cancel' and 'OK' buttons.

In this example, Prism finds that the Binding Potential is 242.1, with a confidence interval ranging from 183.4 to 300.7. That confidence interval is reasonably narrow, so the result is quite useful.

Reference

1. Innis et al. Consensus nomenclature for in vivo imaging of reversibly binding radioligands. *Journal of Cerebral Blood Flow & Metabolism* (2007) vol. 27 (9) pp. 1533-1539

[Download the Prism file for this example.](#)

2.7.6.8 Equation: Two sites -- Specific binding only

Introduction

In a saturation binding experiment, you vary the concentration of radioligand and measure binding. The goal is to determine the K_d (ligand concentration that binds to half the receptor sites at equilibrium) and B_{max} (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.](#)

307

Step by step

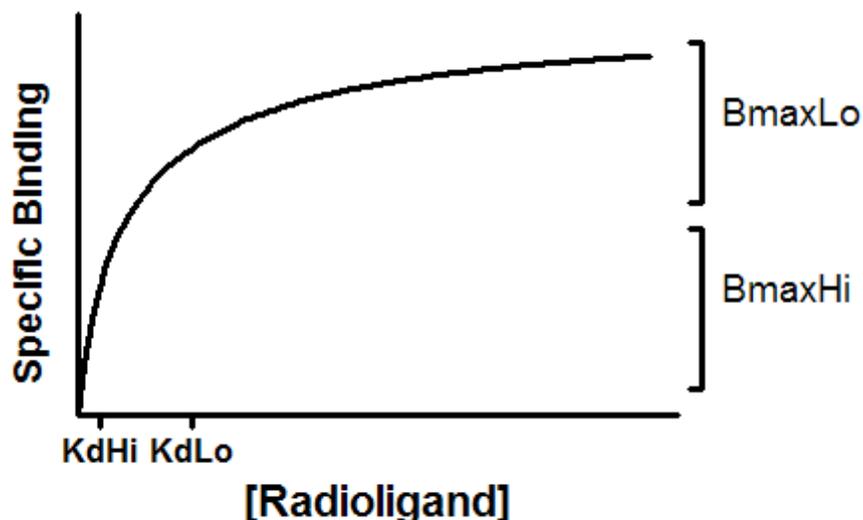
Create an XY data table. Enter radioligand concentration into X, and specific binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

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

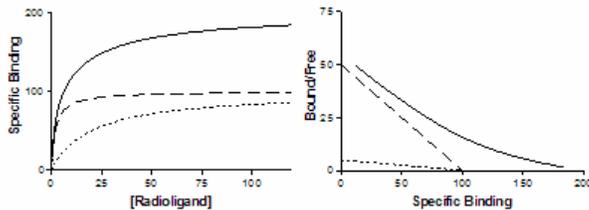
BmaxHi and **BmaxLo** are the maximum specific bindings to the two sites in the same units as Y.

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 K_d . 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.

2.7.6.9 Equation: Two sites – Fit total and nonspecific binding

Introduction

In a saturation binding experiment, you vary the concentration of radioligand and measure binding. The goal is to determine the Kd (ligand concentration that binds to half the receptor sites at equilibrium) and Bmax (maximum number of binding sites).

The ligand binds not only to receptors sites, but also to nonspecific sites. There are three approaches to dealing with nonspecific binding.

- Subtract off the nonspecific, and [analyze only the specific binding](#) ³⁰⁴.
- Analyze the total binding only, inferring the amount of nonspecific binding from the shape of the total binding curve. This approach doesn't work well when the ligand binds to two sites
- Globally analyze the total and nonspecific binding at one time. This is the best approach, and the details are explained below.

Step by step

Create an XY data table. Enter radioligand concentration into X, total binding into Y, and nonspecific binding into column B.

Use any convenient units for X. The Kd will be reported in those same concentration units. Use the same units for total and nonspecific binding. The Bmax will be reported in those same units.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *Two sites -- Total and nonspecific binding*.

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

Model

```
Specific1=BmaxHi*X/(X+KdHi)
Specific2=BmaxLo*X/(X+KdLo)
Nonspecific=NS*X + Background
<A>Y=Specific1 + Specific2 + Nonspecific
<B>Y=Nonspecific
```

The <A> and 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.

2.7.6.10 Equation: One site with allosteric modulator

Introduction

An allosteric modulator can reduce radioligand binding. This model fits experiments, where entire radioligand binding curves are measured in the absence and presence of modulator. The goal is to learn the affinity of the modulator for binding to its site, and also determine the value of alpha, the ternary complex constant that quantifies the degree to which binding of the modulator alters the affinity of the radioligand for the receptor site.

Step by step

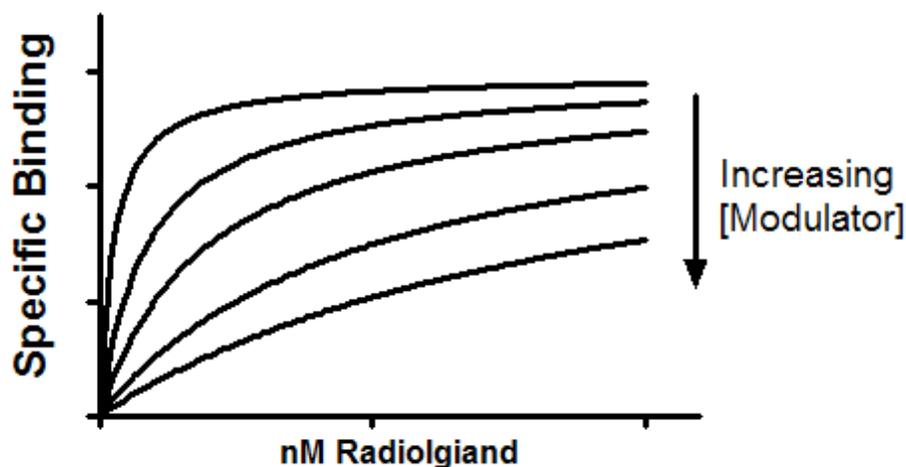
Create an XY data table. Enter the concentration of the labeled ligand into X, using any convenient units (maybe nM). Enter specific binding into Y in any convenient units. Enter data with no modulator into column A. Enter data collected with a constant concentration of modulator into column B. Repeat, if you have data, for column C, D, E, ..., each with a different concentration of modulator. Enter the modulator concentration (in nanomolar so 1nM is entered as '1') into the column titles. Don't forget to enter '0' as the column title for data set A.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Saturation Binding equations, and choose *Allosteric modulator shift*.

You don't need to constrain any parameters to constant values.

Model

```
Hot=X
Alpha=10^logalpha
KB=10^logKB
KApp=KHot*( (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 α is less than 1.0, then the modulator reduces ligand binding. If α is greater than 1.0, then the modulator increases binding. In the example shown above, α 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 α , rather than α itself. This is because α 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 α 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

2.7.7 Receptor binding - Competitive binding

2.7.7.1 Key concepts: Competitive binding

What is competitive binding?

In a competitive binding experiment, you use a single concentration of labeled (hot) ligand and vary the concentration of unlabeled (cold) drugs, and measure binding at equilibrium.

Comparing one- and two-site models

Prism offers models for fitting one or two sites. You can use choices in the Compare tab to compare the two fits. When comparing the fits to the one- and two-site models, use common sense as well as statistics. Don't accept a two site model, if one of the sites is only a tiny fraction of the total, or if its K_i is outside the range of competitor concentrations you used in the experiment.

Ligand depletion

If a large fraction of the added radioligand binds to the receptors, the ligand is depleted so the concentration you added is greater than the free concentration. You need to fit these data to a [model that accounts for ligand depletion](#)³¹⁸.

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](#)³²¹.

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](#)³²² (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.

2.7.7.2 Equation: One site - Fit Ki

Introduction

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

Step by step

Create an XY data table. Enter the logarithm of the concentration of the unlabeled compound into X, and binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc. Use subcolumns to enter replicates.

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

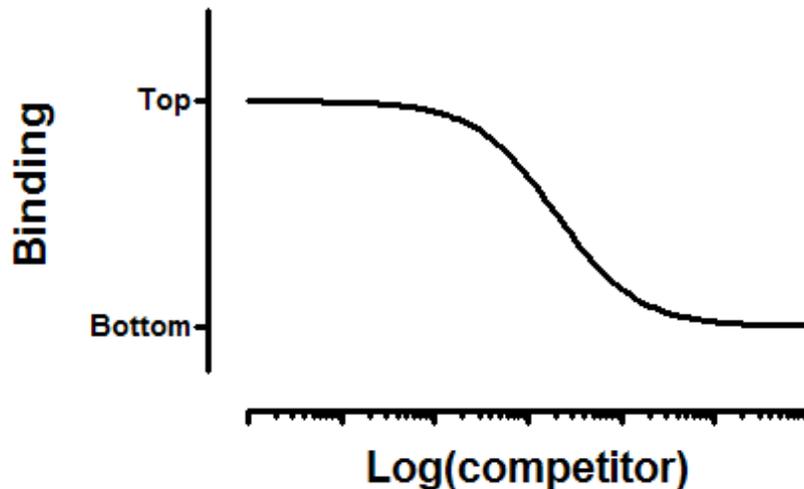
You must constrain two parameters to constant values based on your experimental design:

- RadioligandNM is the concentration of labeled ligand in nM. A single concentration of radioligand is used for the entire experiment.
- HotKdNM is the equilibrium dissociation constant of the labeled ligand in nM.

Model

$$\log EC50 = \log(10^{\log Ki} * (1 + \text{RadioligandNM} / \text{HotKdNM}))$$

$$Y = \text{Bottom} + (\text{Top} - \text{Bottom}) / (1 + 10^{(X - \log EC50)})$$



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 K_i of the unlabelled ligand directly. It does not report the EC_{50} , so you do not need to apply the Cheng and Prusoff correction(1). Instead you enter the concentration of radioligand and its K_d as constants, and Prism directly fits the K_i of your cold compound. If you want to know the IC_{50} , fit a log(dose)-response curve.

If you want to know the IC_{50} (which is not very informative in this situation) fit the data using an [alternative equation](#)³¹⁴.

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

1. Cheng, Y. and Prusoff, W. H. Relationship between the inhibition constant (K_1) and the concentration of inhibitor which causes 50 per cent inhibition (I_{50}) of an enzymatic reaction. *Biochem Pharmacol*, 22: 3099-3108, 1973.

2.7.7.3 Equation: One site - Fit logIC50

Introduction

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

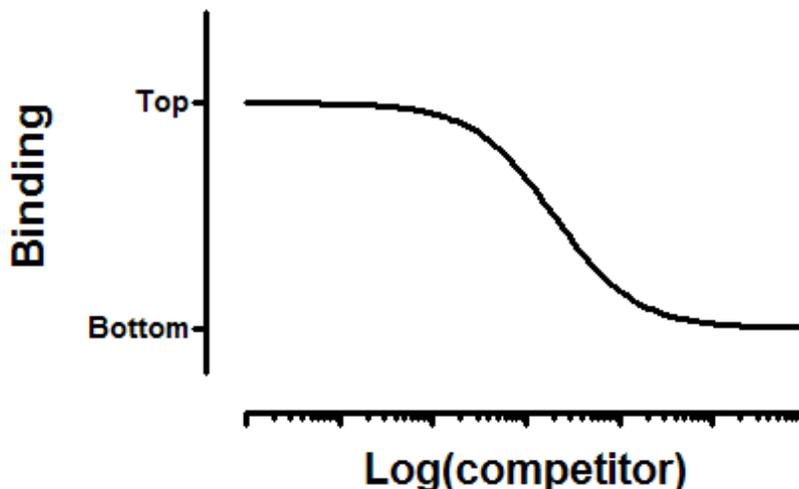
Step by step

Create an XY data table. Enter the logarithm of the concentration of the unlabeled compound into X, and binding into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc. Use subcolumns to enter replicates.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *One site - Fit logEC50*.

Model

$$Y = \text{Bottom} + (\text{Top} - \text{Bottom}) / (1 + 10^{-(X - \text{LogIC50})})$$



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 $\log IC_{50}$, which is not the same as the K_i of the unlabelled ligand for binding. The K_i depends on the IC_{50} , the concentration of radioligand, and its K_d for binding. You can fit the K_i directly using a [different equation](#)³¹².

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

2.7.7.4 Equation: Two sites - Fit K_i

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 K_i* .

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.

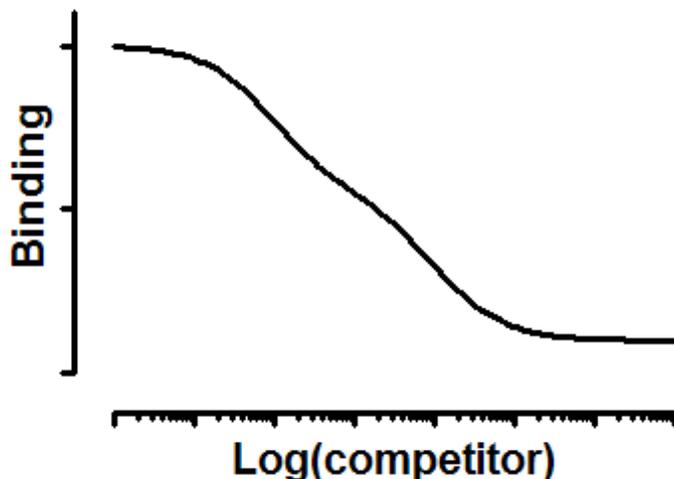
Model

$$C_{oldnM} = 10^{-(X+9)}$$

```

KIHiM = 10^(LogKI_Hi+9)
KILoM = 10^(LogKI_Lo+9)
SITE1= HotnM*(Top-Bottom)/(HotnM + KDHotNM_Ki*(1+coldnM/KiHiM))
SITE2= HotnM*(Top-Bottom)/(HotnM + KDHotNM_Lo*(1+coldnM/KiLoM))
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 $\log(K_i)$ values of the unlabelled ligand directly. It does not report the IC50s, so you do not need to apply the Cheng and Prusoff correction(1). Instead you enter the concentration of radioligand and its K_d as constants, and Prism directly fits the K_i of your cold compound. If you want to fit the two IC50 values instead of the K_i values, use a [different equation](#)^[317].

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 K_{dHi} is the K_d of the hot ligand for the receptors with the high affinity for the unlabeled ligand, and K_{dLo} is the K_d of

the hot ligand for the receptors with lower affinity for the unlabeled ligand. So K_{dHi} may be larger or smaller than K_{dLo} .

This analysis assumes that the binding is reversible and at equilibrium. It also assumes that the labeled and unlabeled ligands compete for the same binding sites.

1. Cheng, Y. and Prusoff, W. H. Relationship between the inhibition constant (K_1) and the concentration of inhibitor which causes 50 per cent inhibition (I_{50}) of an enzymatic reaction. *Biochem Pharmacol*, 22: 3099-3108, 1973.

2.7.7.5 Equation: Two sites - Fit logIC50

Introduction

You can determine the equilibrium dissociation constant of an unlabelled ligand by measuring its competition for radioligand binding. This model assumes that there are two classes of sites with identical affinity for the radioligand, but different affinities for the competitor.

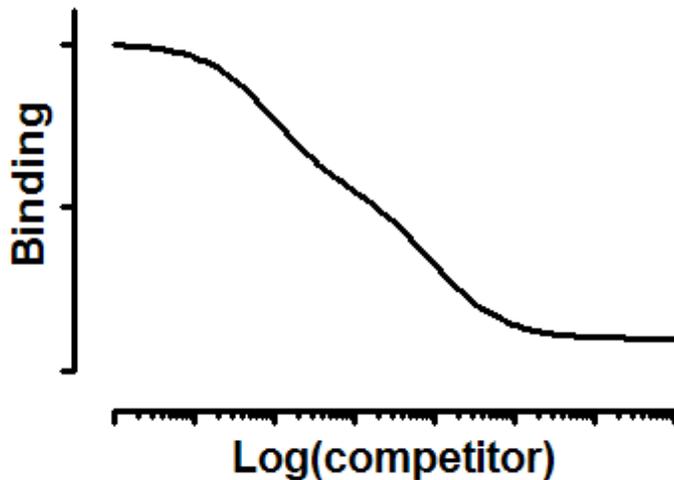
Step by step

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 K_i values. The K_i values depend on the IC50s, the concentration of radioligand, and its K_d for binding. You can fit the K_i values directly using a [different equation](#)^[315].

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.

2.7.7.6 Equation: One site - Ligand depletion

Introduction

This model for competitive binding is useful when a large fraction of the added radioligand binds to the receptors, so the concentration you added is greater than the free concentration.

These equation allows for a substantial fraction of the added ligand to bind. This only works with radioactive ligands, so the assessment of added ligand and bound ligand are in the same counts-per-minute units. This method doesn't work with fluorescent ligands.

Step by step

Create an XY data table. Enter the logarithm of the molar concentration of the unlabeled compound into X, and binding into Y. The Y values must be in cpm.

If you have several experimental conditions, place the first into column A, the second into column B, etc. Use subcolumns to enter replicates.

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

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

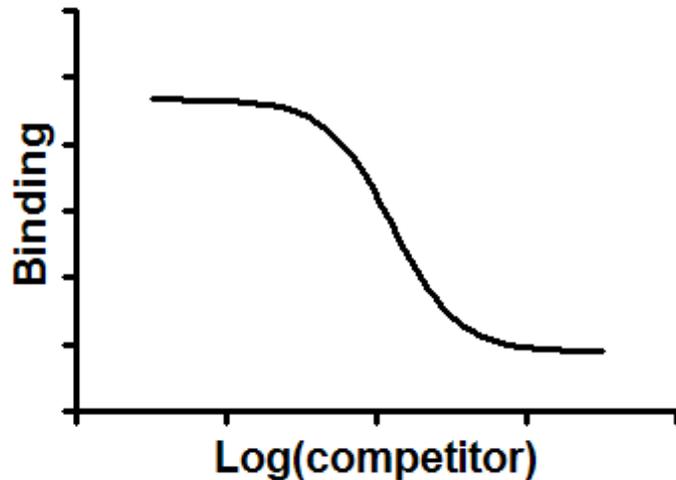
- **Hot** is the amount of labeled ligand in cpm. A single concentration of radioligand is used for the entire experiment.
- **KdNM** is the equilibrium dissociation constant of the labeled ligand in nM.
- **SpAct** is the specific radioactivity in cpm/fmol.
- **Vol** is the reaction volume in ml.

Model

```

KdCPM=KdnM*SpAct*vol*1000
; nmol/L *(cpm/fmol * ml * .001L/ml * 1000000fmol/nmol) = cpm
R=NS+1
S=[1+10^(X-LogKi)]*KdCPM+Hot
a=-1*R
b=R*S+NS*Hot + Bmax
c= -1*Hot*(S*NS + Bmax)
Y= (-1*b + sqrt(b*b-4*a*c))/(2*a)

```



Interpret the parameters

logKI is the log of the molar equilibrium dissociation constant of unlabeled ligand.

Ki is the equilibrium dissociation constant in Molar.

Bmax is the maximum binding of ligand to receptors in cpm. This represents binding to all the receptors so is higher than the top plateau of the curve.

NS is the fraction of the radioligand that binds to nonspecific sites.

Notes

This analysis accounts for the fact that a large fraction of the added radioligand binds to the receptors. If you are able to assume that only a small fraction of radioligand binds, which means that the concentration you added is virtually identical to the free concentration, use an [alternative analysis](#)³¹².

Reference

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

2.7.7.7 Equation: One site - Homologous

Introduction

An **homologous** binding experiment is one where the labeled and unlabeled ligands have identical affinities for the receptors. Generally this is because the two are chemically identical. Receptor number and affinity are determined by analyzing the competition of varying concentrations of unlabeled ligand for one (or better, two) concentrations of labeled ligand.

Step by step

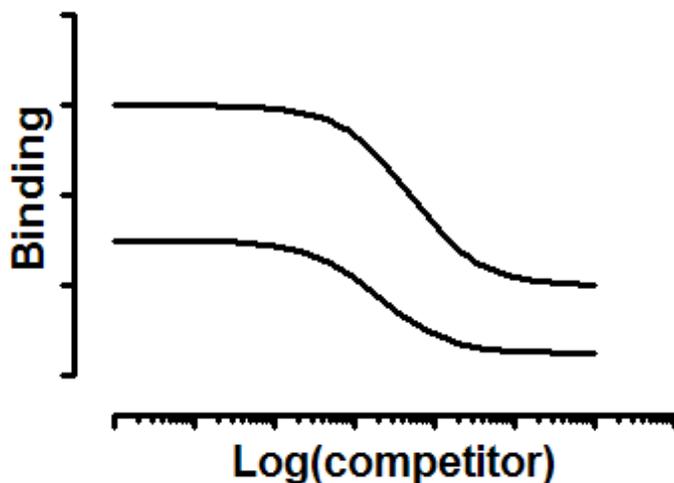
Create an XY data table. Enter the logarithm of the concentration of the unlabeled compound into X, and binding into Y. Enter the concentration of labeled ligand (in nM) as the column title. You will get better results if you use two different concentrations of labeled ligand.

If you have several experimental conditions, place the first into column A, the second into column B, etc. Use subcolumns to enter replicates.

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

Model

```
ColdNM=10^(x+9) ;Cold concentration in nM  
KdNM=10^(logKD+9) ;Kd in nM  
Y=(Bmax*HotnM)/(HotnM + ColdNM + KdNM) + Bottom
```



Interpret the parameters

logKd Log of the equilibrium 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 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.

2.7.7.8 Equation: Allosteric modulator

Introduction

Fits a curve of "competition" of binding by an allosteric modulator, based on the ternary complex model. Note that this model assumes the allosteric modulator is present in excess, so is not depleted by binding to the receptors. Since it binds to a different site than the radioligand, the term 'competition' is not apt, but we list it here because the experimental design is the same as used for competitive binding.

Step by step

Create an XY data table. Enter the logarithm of the concentration of the unlabeled modulator (in nM) into X, and specific binding into Y in any convenient units.

From the data table, click Analyze, choose nonlinear regression, choose the panel of Competition Binding equations, and choose *Allosteric modulator titration*.

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

- RadioligandNM is the concentration of labeled ligand in nM. A single concentration of radioligand is used for the entire experiment.
- HotKdNM is the equilibrium dissociation constant of the labeled ligand in nM.

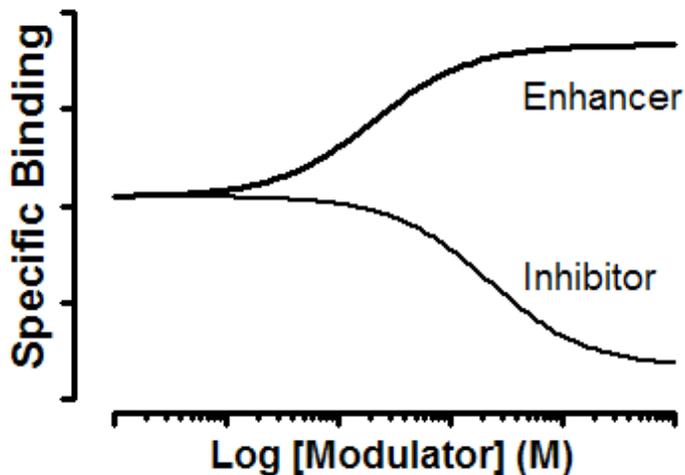
Also consider constraining Y0 (radioligand binding in the absence of modulator) to a constant value.

Model

```

AlloNM=10^(X+9)
KbNM=10^(logKb +9)
alpha=10^logAlpha
KAppNM=HotKDnm*((1+(AlloNM/KBNM))/(1+alpha*(AlloNM/KBNM)))
HotOccupancy = RadioligandNM/(RadioligandNM + HotKDnm)
Y=(Y0/HotOccupancy)*(RadioligandNM/(RadioligandNM + KAppNM))

```



Interpret the parameters

Kb is the equilibrium dissociation constant (Molar) of modulator binding.

Alpha is the ternary complex constant. When $\alpha=1.0$, the modulator does not alter binding. If α is less than 1.0, then the modulator reduces ligand binding. If α is greater than 1.0, then the modulator increases binding.

Y0 is the radioligand binding in the absence of modulator. Consider constraining

this to a constant value.

Notes

This model is designed to analyze data when the unlabeled compound works via an allosteric site. Since the labeled and unlabeled ligands are acting via different sites, it is inappropriate (and incorrect) to refer to these types of experiments as "competition binding assays". In some cases, in fact, the allosteric modulator enhances radioligand binding.

The model is written to fit the logarithm of alpha, rather than alpha itself. This is because alpha is asymmetrically (all values from 0 to 1 mean that the modulator decreases binding, while all values from 1 to infinity mean that the modulator enhances binding. On a log scale, its values are more symmetrical, so the confidence interval computed on a log scale (as Prism does) are more accurate.

The Y axis plots specific binding. Even at very high concentrations of inhibitor, the specific binding does not descend to zero. This is the nature of allosteric inhibition. If alpha is very high, then the binding is inhibited almost to zero. If alpha is not so high, then the maximum inhibition is more modest. For example, if $\alpha=3$, the maximum inhibition is down to 33%.

This model assumes that the allosteric modulator is present in excess, so the concentration you added is very close to its free concentration. This model does not apply when the concentration of allosteric modulator is limiting (as it is when G proteins alter agonist binding to many receptors). No explicit model can handle this situation. You need to define the model with an implicit equation (Y on both sides of the equals sign) and Prism cannot handle such equations.

Reference

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

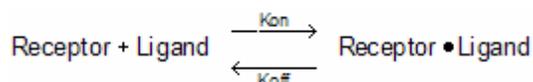
2.7.8 Receptor binding - Kinetics

Kinetics refers to changes over time. Kinetic binding experiments are used to determine the association and dissociation rate constants.

2.7.8.1 Key concepts: Kinetics of binding

Law of mass action

Kinetic binding experiments are used to determine the association and dissociation rate constants. All the analyses assume that binding follows the law of mass action:



At any given time, the rate at which receptor-ligand complexes form is proportional to the radioligand concentration and the number of receptors still unoccupied. The rate of dissociation is proportional to the concentration of receptor-ligand complexes.

Rate of dissociation

A dissociation binding experiment measures the “off rate” for radioligand dissociating from the receptor. Initially ligand and receptor are allowed to bind, perhaps to equilibrium. At that point, you need to block further binding of radioligand to receptor (by adding an unlabeled drug or by dilution) so you can measure the rate of dissociation, which follows a one-phase exponential decay with a rate constant equal to the rate of radioligand dissociation.

Rate of association

In an association experiment, you add radioligand and measure specific binding at various times thereafter.

Binding increases over time until it plateaus. This plateau is not the same as the B_{max} . The plateau in an association experiment depends on the concentration of radioligand used, while the B_{max} 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, k_{on} 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, k_{off} 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](#)³²⁸, measure association kinetics with [two or more concentrations of radioligand](#)³²⁹, or determine the [rate of association and dissociation in one experiment](#)³³¹.

2.7.8.2 Equation: Dissociation kinetics

Introduction

A dissociation binding experiment measures the “off rate” for radioligand dissociating from the receptor. Initially ligand and receptor are allowed to bind, perhaps to equilibrium. At that point, you need to block further binding of radioligand to receptor so you can measure the rate of dissociation. There are several ways to do this:

- If the tissue is attached to a surface, you can remove the buffer containing radioligand and replace with fresh buffer without radioligand.
- Spin the suspension and resuspend in fresh buffer.
- Add a very high concentration of an unlabeled ligand. If this concentration is high enough, it will instantly bind to nearly all the unoccupied receptors and thus block binding of the radioligand.
- Dilute the incubation by a large factor, at least 100 fold dilution. This will reduce the concentration of radioligand by that factor. At such a low concentration, new binding of radioligand will be negligible. This method is only practical when you use a fairly low concentration of radioligand so its concentration after dilution is far below its K_d for binding.

You then measure binding at various times after that to determine how rapidly the radioligand falls off the receptors.

Step by step

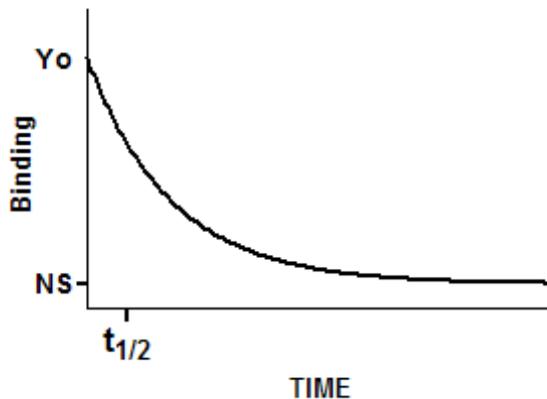
Create an XY data table. Enter time into X, and total binding into Y. If you have

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

$$Y = (Y_0 - NS) * \exp(-K * X) + NS$$



Interpret the parameters

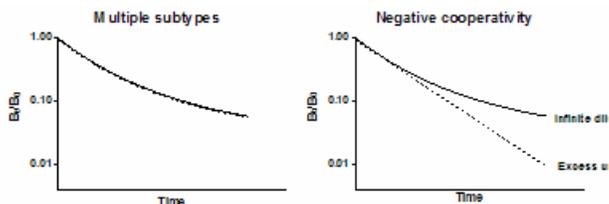
Y_0 is the binding at time zero, in the units of the Y axis.

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.

2.7.8.3 Equation: Association kinetics (one ligand concentration)

Introduction

When you measure the association rate of a radioligand, the rate at which the binding equilibrates depends not only on the association rate constant and the amount of ligand you used, but also on its dissociation rate constant. ([Why?](#)³²⁵) 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](#)³²⁹, or to analyze an experiment that measures [both association and dissociation](#)³³¹ rate sequentially.

Step by step

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

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

You must constrain Hotnm ([radioligand] in nM) and Koff (dissociation rate constant, in inverse minutes) to constant values.

Model

$$Kd = Koff / Kon$$

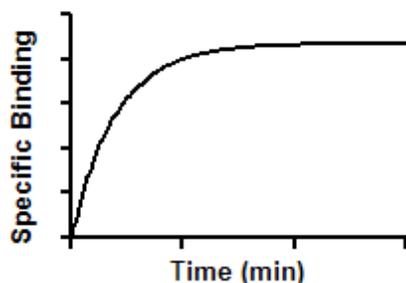
$$L = Hotnm * 1e-9$$

$$Kob = Kon * L + Koff$$

$$Occupancy = L / (L + Kd)$$

$$Ymax = Occupancy * Bmax$$

$$Y = Ymax * (1 - \exp(-1 * kob * X))$$



Interpret the parameters

Kon is the association rate constant, in units of $M^{-1} \text{ min}^{-1}$

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

Bmax is the maximal binding at equilibrium, extrapolated to maximal radioligand concentration, in the units used to enter the Y values.

Notes

2.7.8.4 Equation: Association kinetics (two ligand concentrations)

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 in minutes into X, and total binding into Y. Enter binding at one concentration of radioligand into the column A, binding at another concentration into column B, etc. Enter the concentrations, in nM, into the column titles.

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

Model

$$K_d = K_{off} / K_{on}$$

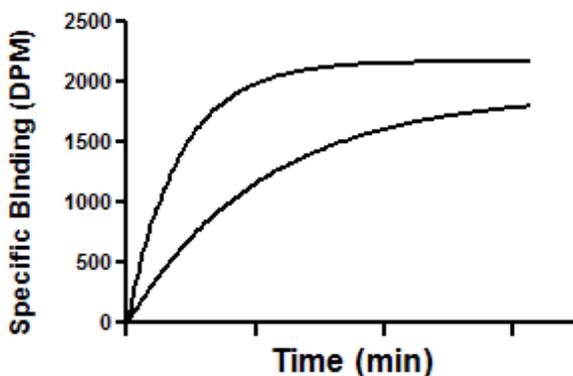
$$L = \text{Hotnm} * 1e-9$$

$$K_{ob} = K_{on} * L + K_{off}$$

$$\text{Occupancy} = L / (L + K_d)$$

$$Y_{max} = \text{Occupancy} * B_{max}$$

$$Y = Y_{max} * (1 - \exp(-1 * k_{ob} * X))$$



Interpret the parameters

Koff is the dissociation rate constant in min^{-1} .

Kon is the association rate constant, in units of $\text{M}^{-1} \text{min}^{-1}$.

Kd is the equilibrium binding constant, in Molar, computed as K_{off}/K_{on}

Bmax is the maximal binding at equilibrium, at maximal radioligand

concentration, in the units you used to enter Y.

Notes

According to the law of mass action, the ratio of koff to kon is the Kd of receptor binding:

$$K_d = \frac{k_{off}}{k_{on}}$$

Compare the Kd calculated this way (from kinetic experiments) with the Kd determined from a saturation binding curve. If binding follows the law of mass action, the two Kd values should be indistinguishable.

2.7.8.5 Equation: Association then dissociation

Introduction

You cannot determine the association rate constant by simply observing the association of a single concentration of radioligand. The rate at which a ligand reaches equilibrium is determined not only by the association rate constant and the ligand concentration, but also by the dissociation constant.

One way to determine the association rate constant is to globally fit data obtained with [two different concentrations of radioligand](#)³²⁸. An alternative approach, explained here is to measure association and dissociation in one experiment.

Add a radioligand and measure total binding at multiple time points, then at Time0 initiate dissociation (by adding an antagonist or by massive dilution) and then measure dissociation at various times.

Step by step

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

From the table of specific binding, click Analyze, choose nonlinear regression, choose the panel of Kinetics Binding equations, and choose *Association then dissociation*.

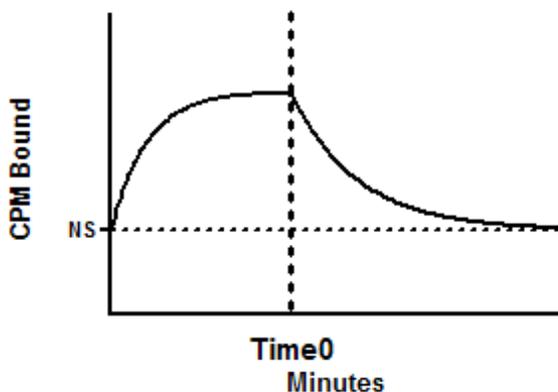
Constrain HotNM ([radioigand in nM] and Time0 (time at which dissociation was initiated) to constant values. If you entered specific binding into the Y column, also constrain NS to a constant value of zero.

Model

```

Radioligand=HotNM*1e-9
Kob=[Radioligand]*Kon+Koff
Kd=Koff/Kon
Eq=Bmax*radioligand/(radioligand + Kd)
Association=Eq*(1-exp(-1*Kob*X))
YatTime0 = Eq*(1-exp(-1*Kob*Time0))
Dissociation= YatTime0*exp(-1*Koff*(X-Time0))
Y=IF(X<Time0, Association, Dissociation) + NS

```



Interpret the parameters

Koff is the dissociation constant in min^{-1} .

Kon is the association constant in inverse minutes multiplied by inverse concentration.

KD is computed from $Koff/Kon$. Expressed in Molar units.

Bmax is the maximum binding at equilibrium with maximum concentration of radioligand, in units of Y axis. Unless you used a very high concentration of radioligand, Bmax will have a wide confidence interval as the experiment is not designed to determine Bmax.

NS is the nonspecific binding, in units of the Y axis. It is the Y value at time 0, and also the Y value at very late times after all the ligand has dissociated.

Notes

2.7.8.6 Equation: Kinetics of competitive binding

Introduction

Kinetics experiments can determine the dissociation and association rate constants (off-rate and on-rate) of an unlabeled compound. Add labeled and unlabeled ligand together and measure the binding of the labeled ligand over time. Fit to the appropriate model described below, constraining the rate constants of the labeled ligand to constant values determined from other experiments, and fit the rate constants of the unlabeled compound.

Using only a single concentration of labeled and radioligand, it is very hard to determine the rate constants with any reasonable precision. But measure the kinetics at two (or more) concentrations of the unlabeled ligand, and the results are much more precise.

Step by step

Create an XY data table. Enter time in minutes into X, and specific binding in cpm into Y. Enter the binding for one concentration of the unlabeled ligand in column A, and another concentration in column B, etc.. Enter the concentrations, in nM, into the column titles.

From the table of specific binding, click Analyze, choose nonlinear regression, choose the panel of Kinetics Binding equations, and choose *Kinetics of competitive binding*.

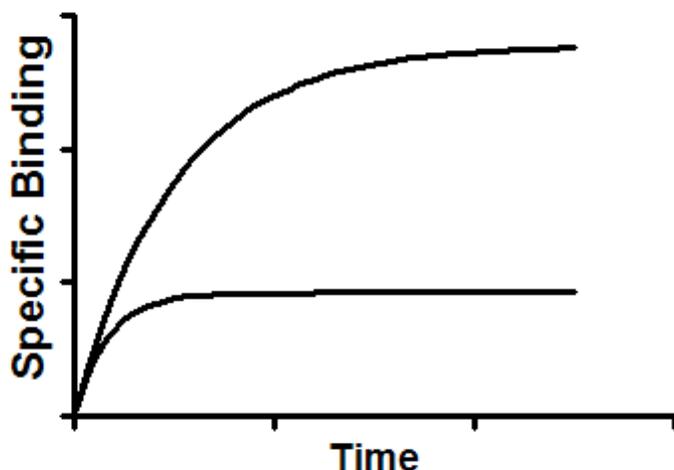
Constrain k1 and k2 to constant values determined from kinetic binding experiments. k1 is the association rate constant of the hot ligand in $M^{-1} \text{ min}^{-1}$ and k2 is its dissociation rate constant in units of min^{-1} .

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

$$\begin{aligned}
 KA &= K1 * L * 1E-9 + k2 \\
 KB &= K3 * I * 1e-9 + K4 \\
 S &= \text{SQRT}((KA - KB)^2 + 4 * K1 * K3 * L * I * 1e-18) \\
 KF &= 0.5 * (Ka + KB + S) \\
 KS &= 0.5 * (KA + KB - S) \\
 DIFF &= KF - KS \\
 Q &= Bmax * K1 * L * 1e-9 / DIFF \\
 Y &= Q * (k4 * DIFF / (KF * KS) + ((K4 - Kf) / KF) * \exp(-KF * X) - ((K4 - KS) / KS) * \exp(-KS * X))
 \end{aligned}$$



Interpret the parameters

k3 is the association rate constant of unlabeled ligand in $M^{-1} \text{ min}^{-1}$.

k4 is the dissociation rate constant of unlabeled ligand in min^{-1} .

Bmax is the total number of receptors. Either leave as a variable or set to a constant you know from other experiments. The Bmax is the maximum binding at equilibrium with a very high concentration of radioligand. It is usually much larger than any binding seen in the experiment.

All three parameters are constrained to be shared, so Prism fits one value of each of the three parameters for all the data sets.

Notes

- This equation does not account for ligand depletion. It assumes that only a small fraction of radioligand binds to receptors, so that the free concentration of radioligand is very close to the added concentration.
- This method will only give reliable results if you have plenty of data points at early time points.
- The ratio $K4/K3$ is the equilibrium dissociation constant of the cold ligand in Molar. You should compare this value (determined via kinetics) with the same value determined by equilibrium competition.

Reference

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

2.7.9 Enzyme kinetics -- Key concepts

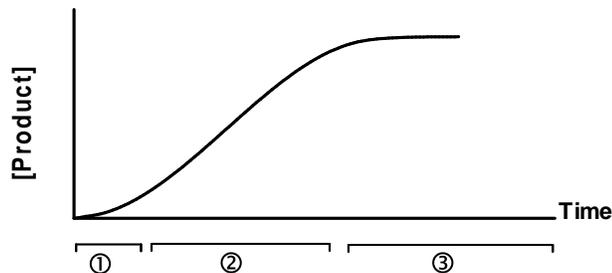
2.7.9.1 Key concepts: Terminology

What is an enzyme?

Living systems depend on chemical reactions which, on their own, would occur at extremely slow rates. Enzymes are catalysts that reduce the needed activation energy so these reactions proceed at rates that are useful to the cell. The study of enzyme kinetics can help us understand the function and regulation of enzymes.

Enzyme progress curves

In most cases, an enzyme converts one chemical (the substrate) into another (the product). A graph of product concentration vs. time follows three phases marked on the graph below.



1. At very early time points (usually less than a second), the rate of product accumulation increases over time. Special techniques, not available in Prism, are needed to study the early kinetics of enzyme action. The graph above exaggerates this first phase.
2. For an extended period of time, the product concentration increases linearly with time. All the analyses built-in to Prism use data collected during this second phase.
3. At later times, the substrate is depleted, so the curve starts to level off.

Eventually the concentration of product reaches a plateau.

It is very difficult to fit a curve to these kind of data. The model simply cannot be reduced to an equation that expresses product concentration as a function of time. To fit these kind of data (called an enzyme progress curve) you need to use a program that can fit data to a model defined by differential equations or by an implicit equation. For more details, see RG Duggleby, [Analysis of Enzyme Reaction Progress Curves by Nonlinear Regression](#), Methods in Enzymology, 249: 61-60, 1995.

Rather than fit the enzyme progress curve, most analyses of enzyme kinetics (including all those built-in to Prism) measure product at a single time point. Analyses assume that the time point you chose is on the linear (second) phase of product accumulation and ignore the nonlinear first phase (which is usually very short). Therefore, if you divide the amount of product produced by the time the reaction was allowed to proceed, you compute the amount of product formed per unit time, which is the enzyme velocity.

Terminology

The terminology can be confusing. Note these confusing points:

- As mentioned above, almost all studies of enzyme "**kinetics**" are done by collecting data at a single time point. The X axis is substrate (or inhibitor) concentration, not time.
- The second phase shown in the graph above is often called the "**initial rate**", a phrase that makes sense only if you ignore the short transient phase that precedes it.
- That second phase is also called "**steady state**", because the concentration of enzyme-substrate complex doesn't change during that phase. However, the concentration of product accumulates, so the system is not truly at steady state until, much later, the concentration of product truly doesn't change over time.

2.7.9.2 Key concepts: Assumptions

Standard analyses of enzyme kinetics (the only kind discussed here) assume:

- The production of product is linear with time during the time interval used.

- The concentration of substrate vastly exceeds the concentration of enzyme. This means that the free concentration of substrate is very close to the concentration you added, and that substrate concentration is essentially constant throughout the assay.
- A single enzyme forms the product.
- There is negligible spontaneous creation of product without enzyme.

With the exception of the built-in allosteric model, Prism's built-in equations also assume:

- No cooperativity. Binding of substrate to one enzyme binding site doesn't influence the affinity or activity of an adjacent site.
- Neither substrate nor product acts as an allosteric modulator to alter the enzyme velocity.

2.7.10 Enzyme kinetics - Substrate vs. velocity

2.7.10.1 Key concepts: Substrate vs. velocity

Fitting a substrate-velocity curve

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

The standard way to fit these data is to [fit the Michaelis-Menten model](#)^[338] to determine the V_{max} (maximum enzyme velocity) and its K_m (the concentration of substrate needed to get half-maximal velocity).

The V_{max} equals the product of the concentration of active enzyme sites times the turnover rate, k_{cat} . 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 \$k_{cat}\$ and \$K_m\$](#) ^[340]. The curve will be identical to the Michaelis-Menten fit.

If the enzyme has cooperative subunits, the graph of enzyme velocity as a function of substrate concentration will appear sigmoidal. Prism offers [one empirical equation for fitting sigmoidal substrate-velocity curves](#)^[342].

2.7.10.2 Equation: Michaelis-Menten model

Introduction

The most common kind of enzyme kinetics experiment is to vary the concentration of substrate and measure enzyme velocity. The goal is to determine the enzyme's K_m (substrate concentration that yield a half-maximal velocity) and V_{max} (maximum velocity). If your goal is to determine the turnover number k_{cat} , rather than the V_{max} , use an [alternative version](#)³⁴⁰ of the equation.

Step by step

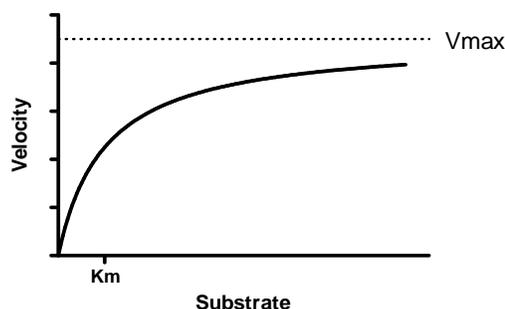
Create an XY data table. Enter substrate concentration into X, and enzyme velocity into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

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

$$Y = V_{max} * X / (K_m + X)$$



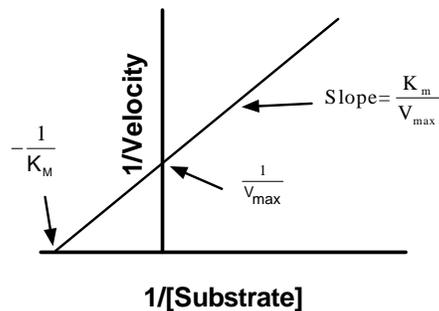
Interpret the parameters

V_{max} 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.

K_m 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 V_{max} and K_m . If you do this, you won't get the most accurate values for V_{max} and K_m . 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 K_m and V_{max} .

To create a Lineweaver-Burk plot with Prism, use the Transform analysis, then choose the panel of biochemistry and pharmacology transforms.

To create a Lineweaver-Burke line corresponding to the nonlinear regression fit, follow these steps:

1. Create a new XY data table, with no subcolumns.
2. Into row 1 enter $X = -1/K_M$, $Y = 0$ (previously determined by nonlinear regression).
3. Into row 2 enter $X = 1/S_{min}$ (S_{min} is the smallest value of [substrate] you want to include on the graph) and $Y = (1/V_{max})(1.0 + K_M/S_{min})$.
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](#)^[335] of all analyses of enzyme kinetics.
- This equation fits exactly the same curve as the [equation that fits the turnover number Kcat](#)^[340] 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](#)^[342]. 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.

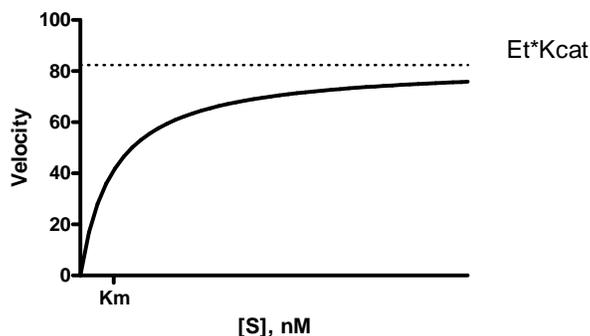
2.7.10.3 Equation: Determine kcat

Introduction

Kcat is the turnover number -- the number of substrate molecule each enzyme site converts to product per unit time. If you know the concentration of enzyme sites, you can fit Kcat instead of Vmax when analyzing a substrate vs. velocity curve.

The model

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



X is the substrate concentration.

Y is enzyme velocity.

kcat is the turnover number, the number of times each enzyme site converts substrate to product per unit time. This is expressed in the inverse of the time units of the Y axis. For example, if Y is in micromoles of substrate per minute, then kcat is the number of molecules of substrate produced per catalytic site per minute.

Km is the Michaelis-Menten constant, in the same units as X. It is the substrate concentration needed to achieve a half-maximum enzyme velocity.

Et is the concentration of enzyme catalytic sites. If the enzyme has multiple subunits, note that Et is the concentration of catalytic sites, which can be larger than the concentration of enzyme molecules. The Y values are entered in units of concentration per time, and Et must be entered in those same concentration units.

Vmax is the maximum enzyme velocity in the same units as Y. It is not directly shown in the model above. It is the velocity of the enzyme extrapolated to very high concentrations of substrate, so is almost always higher than any velocity measured in your experiment. It is computed by multiplying Et times kcat.

Relationship to the Michaelis-Menten model

The curve shown above is identical to the curve defined by the Michaelis-Menten model. You'll get identical curves when you fit either model to your data, and identical values for Km.

The Michaelis-Menten model finds the Vmax, which is the maximum enzyme velocity extrapolated out to very high concentrations of substrate. It is expressed in the same units you used to enter your Y values (enzyme activity). Usually it is straightforward to express this (or convert to) moles/minute/mg of protein. The Vmax is determined by how many enzyme sites are present (Et) and the rate at which the enzyme can convert substrate to product (kcat).

If you know the concentration of enzyme sites you've added to the assay (Et) then you can fit the catalytic constant Kcat using the model above.

When calculating Kcat, the concentration units cancel out, so Kcat is expressed in units of inverse time. It is the turnover number -- the number of substrate molecule each enzyme site converts to product per unit time.

Fitting the model with Prism

1. Create an XY data table. Enter substrate concentration into X, and enzyme velocity into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc. You can also choose Prism's sample data: Enzyme kinetics -- Michaelis-Menten.
2. After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Kcat*.
3. You must constrain **Et** to a constant value, based on other experiments. To constrain the value of Et, go to the Constrain tab of the nonlinear regression dialog, make sure that the drop down next to Et is set to "Constant equal to" and enter the value. For the sample data, enter 100 as the value of Et.

If you don't know the value of Et, you cannot fit the *kcat*, but instead should [fit the Vmax](#)^[336]. It is not possible for Prism to fit both the *kcat* and Et, as the two parameters are intertwined, and a substrate-velocity curve gives no information about their individual values.

4. With the sample data, Prism reports that $K_m = 5.886$ with a 95% confidence interval ranging from 3.933 to 7.839. The best fit value of *kcat* is 13.53 with a 95% confidence interval ranging from 11.97 to 15.09.

Notes

- See the list of [assumptions](#)^[335] of all analyses of enzyme kinetics.
- This equation fits exactly the same curve as the [equation that fits Vmax](#)^[336], 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.

2.7.10.4 Equation: Allosteric sigmoidal

Introduction

If the enzyme has cooperative subunits, the graph of enzyme velocity as a function of substrate concentration will appear sigmoidal. Prism offers one

empirical equation for fitting sigmoidal substrate-velocity curves. Read advanced books on enzyme kinetics for alternative methods based on molecular models of allosteric action.

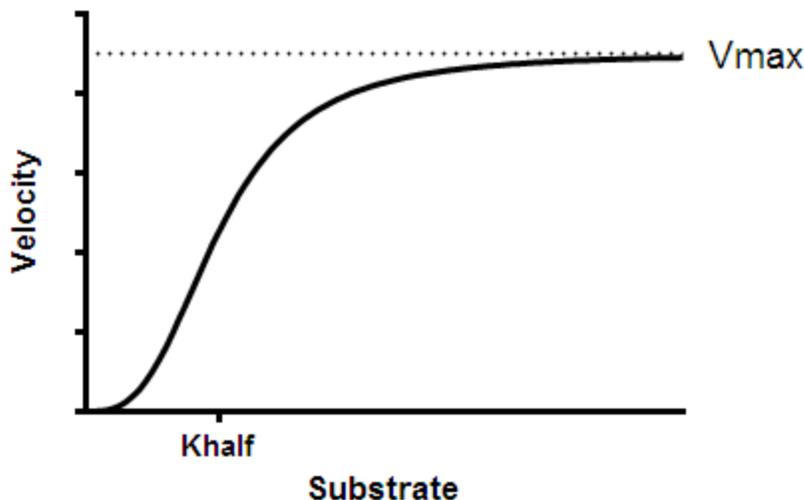
How to enter data

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

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

The model

$$Y = V_{\max} * X^h / (K_{\text{half}}^h + X^h)$$



Interpret the parameters

Vmax is the maximum enzyme velocity in the same units as Y. It is the velocity of the enzyme extrapolated to very high concentrations of substrate, and therefore is almost always higher than any velocity measured in your experiment.

Khalf is the concentration of substrate that produces a half-maximal enzyme velocity. It is the EC50.

h is the Hill slope. When $h=1$, this equation is identical to the standard [Michaelis-Menten equation](#)³³⁸. When it is greater than 1.0, the curve is sigmoidal due to

positive cooperativity. The variable h does not always equal the number of interacting binding sites, but its value can not exceed the number of interacting sites. Think of h as an empirical measure of the steepness of the curve and the presence of cooperativity.

Kprime is related to the K_m . It is computed as K_{half}^h , and is expressed in the same units as X .

An alternative form of the equation

An alternative version of the equation (equation 5.47 in the book by Dr. Copeland referenced below) fits **Kprime**.

$$Y = V_{max} * X^h / (K_{prime} + X^h)$$

Note that **Kprime** in this equation equals K_{half}^h in the equation built in to Prism. The two models generate exactly the same curve, and simply have alternative methods for reporting the parameters. Prism reports both **Kprime** and K_{half} .

Reference

Equation 5.47, in RA Copeland, *Enzymes*, 2nd edition, Wiley, 2000. In this reference, Copeland shows how to fit **Kprime**. Via personal communication, he extended the model to also fit the K_{half} .

2.7.11 Enzyme kinetics -- Inhibition

Many drugs work by inhibiting enzyme activity, either by preventing the substrate from binding to the enzyme, or by stabilizing the enzyme-substrate complex so as to slow formation of product. To distinguish between the models of enzyme inhibition and determine the K_i of the inhibitor, measure substrate-velocity curves in the presence of several concentrations of inhibitor (including one curve with no inhibitor).

2.7.11.1 Key concepts: Enzyme inhibition

Many drugs work by inhibiting enzyme activity, either by preventing the substrate from binding to the enzyme, or by stabilizing the enzyme-substrate complex so as to slow formation of product. To distinguish between the models of enzyme inhibition and determine the K_i 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](#)^[345] 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](#)^[347] inhibitor binds with equal affinity to the enzyme, and the enzyme-substrate complex. The inhibition is not surmountable by increasing substrate concentration. The substrate-velocity curve is shifted down but neither to the right or left.
- An [uncompetitive](#)^[349] inhibitor reversibly binds to the enzyme-substrate complex, but not to the enzyme itself. This reduces both the effective V_{max} and the effective K_m . The substrate-velocity curve is shifted down and to the left.
- The [mixed](#)^[350] 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

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](#)^[352].

Reference

RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. ISBN:0471686964.

2.7.11.2 Equation: Competitive inhibition

Introduction

A competitive inhibitor reversibly binds to the same site as the substrate, so its inhibition can be entirely overcome by using a very high concentration of substrate. The V_{max} doesn't change, and the effective K_m increases. You can determine the K_i 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.

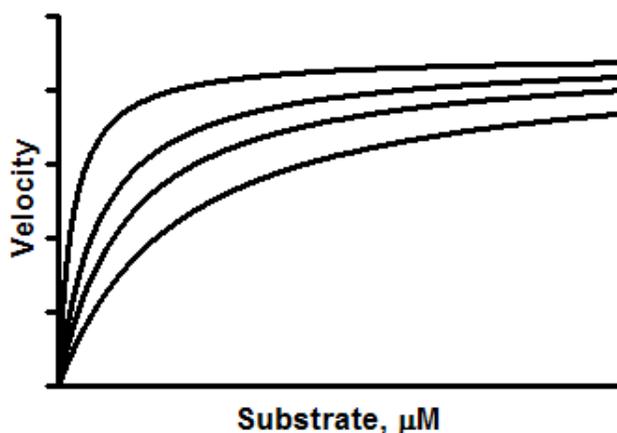
After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Competitive enzyme inhibition*.

The model

$$K_{mObs} = K_m * (1 + [I] / K_i)$$
$$Y = V_{max} * X / (K_{mObs} + X)$$

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 **V_{max}**, **K_m** and **K_i** are shared, so Prism fits one best-fit value for the entire set of data.



Interpreting the parameters

K_i is the inhibition constant, expressed in the same units as I, which you entered into the column titles.

V_{max} is the maximum enzyme velocity, in the absence of inhibitor, expressed in the same units as Y.

K_m 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](#)^[347] or [uncompetitive](#)^[349] model. Or fit to the more general equation for [mixed-model inhibition](#)^[350].

Reference

Equation 3.1 in: RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. ISBN:0471686964. RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. ISBN:0471686964.

2.7.11.3 Equation: Noncompetitive inhibition

Introduction

A noncompetitive inhibitor reversibly binds to both the enzyme-substrate complex, and the enzyme itself. This means that the effective V_{max} decreases with inhibition but the K_m does not change. You can determine the K_i 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](#)^[350].

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.

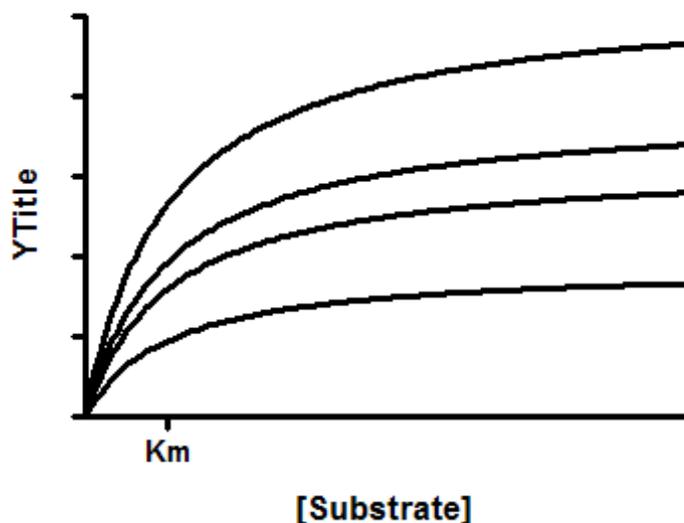
After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Noncompetitive enzyme inhibition*.

The model

$$V_{maxinh} = V_{max} / (1 + I/K_i)$$
$$Y = V_{maxinh} * X / (K_m + X)$$

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](#)³⁴⁵ or [uncompetitive](#)³⁴⁹ model. Or fit to the more general equation for [mixed-model inhibition](#)³⁵⁰.

Reference

Equation 3.4 in: RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. ISBN:0471686964. RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. ISBN:0471686964.

2.7.11.4 Equation: Uncompetitive inhibition

Introduction

An uncompetitive inhibitor binds to the enzyme-substrate complex, but not the free enzyme. This reduces both the effective V_{max} and the effective K_m . The substrate-velocity curve is shifted down and to the left.

You can determine the K_i 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.

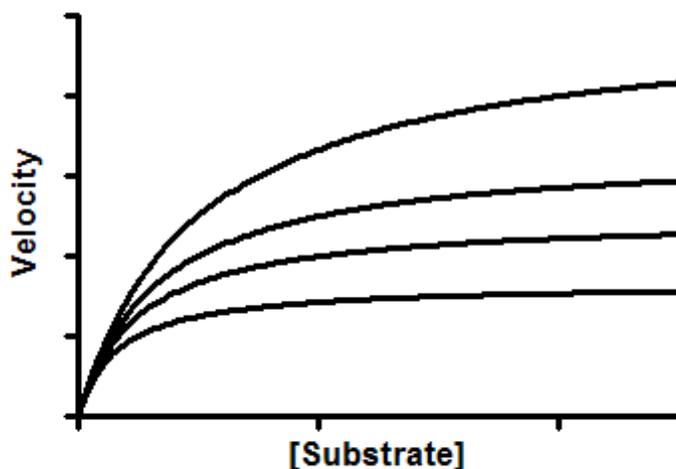
After entering data, click Analyze, choose nonlinear regression, choose the panel of enzyme kinetics equations, and choose *Uncompetitive enzyme inhibition*.

The model

$$\begin{aligned}V_{maxApp} &= V_{max} / (1 + I / \text{Alpha}K_i) \\ K_{mApp} &= K_m / (1 + I / \text{Alpha}K_i) \\ Y &= V_{maxApp} * X / (K_{mApp} + X)\end{aligned}$$

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 **V_{max}**, **K_m** and **K_i** are shared, so Prism fits one best-fit value for the entire set of data.



Interpreting the parameters

V_{max} is the maximum enzyme velocity without inhibitor, expressed in the same units as Y.

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

AlphaK_i is the inhibition constant, expressed in the same units as I, which you entered into the column titles. It is the product of K_i (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 K_i separately, but only to determine their product. Some books call this product K_i'.

If the data don't fit the model well, consider instead fitting to a [competitive](#)^[345] or [noncompetitive](#)^[347] model. Or fit to the more general equation for [mixed-model inhibition](#)^[350].

Reference

Equation 3.6 in: RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. ISBN:0471686964. RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. ISBN:0471686964.

2.7.11.5 Equation: Mixed-model inhibition

Introduction

The mixed model is a general equation that includes [competitive](#)^[345], [uncompetitive](#)^[349] and [noncompetitive](#)^[347] 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

$$\begin{aligned}V_{maxApp} &= V_{max} / (1 + I / (\text{Alpha} * K_i)) \\K_{mApp} &= K_m * (1 + I / K_i) / (1 + I / (\text{Alpha} * K_i)) \\Y &= V_{maxApp} * X / (K_{mApp} + X)\end{aligned}$$

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](#)³⁴⁷ inhibition. When alpha is very large, binding of inhibitor prevents binding of the substrate and the mixed-model becomes identical to [competitive](#)³⁴⁵ 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](#)³⁴⁹ model.

Reference

Equation 3.2 in: RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. ISBN:0471686964. RA Copeland, *Evaluation of Enzyme Inhibitors in Drug Discovery*, Wiley 2005. ISBN:0471686964.

2.7.11.6 Equation: Substrate inhibition

Introduction

At high concentrations, some substrates also inhibit the enzyme activity. Substrate inhibition occurs with about 20% of all known enzymes. It happens when two molecules of substrate can bind to the enzyme, and thus block activity.

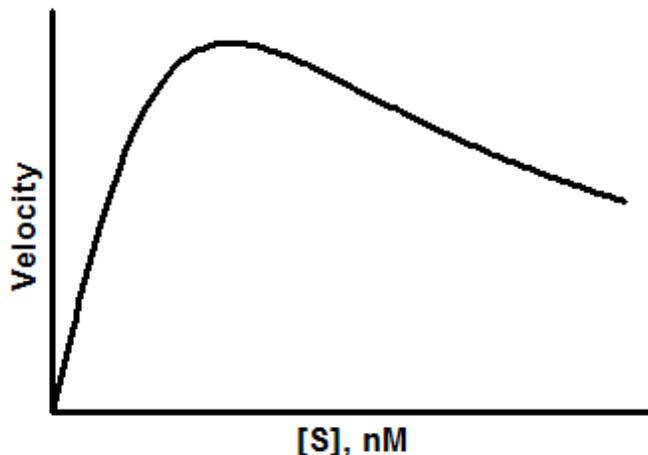
Step by step

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

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

Model

$$Y = V_{\max} * X / (K_m + X * (1 + X / K_i))$$



Parameters

V_{max} is the maximum enzyme velocity, if the substrate didn't also inhibit enzyme activity, expressed in the same units as Y.

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

K_i 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.

2.7.11.7 Equation: Tight inhibition (Morrison equation)

Introduction

This equation accounts for tight binding, so it does not assume that the free concentration of inhibitor equals the total concentration.

Step by step

Create an XY data table. Enter inhibitor concentration into the X column (usually in micromolar, but any concentration units is fine), and enzyme activity into the Y columns (any units). If you have several experimental conditions, place the first into column A, the second into column B, etc.

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

Constrain Et, S and K_m 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 K_m 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.

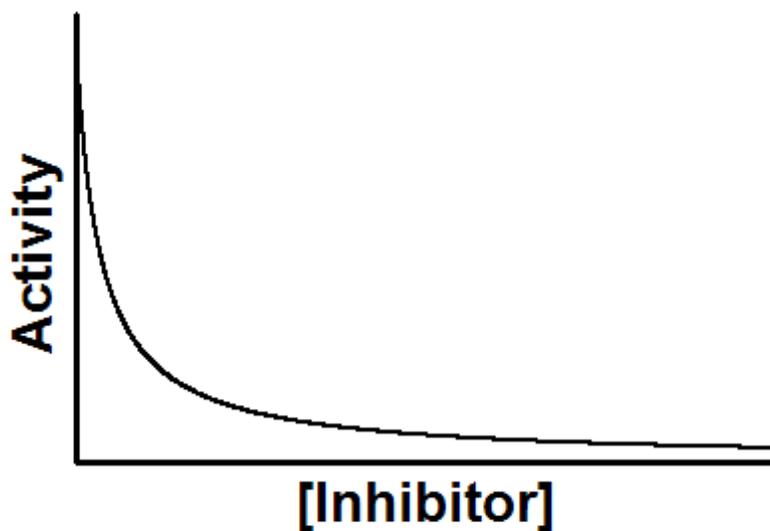
- **Km** is the Michaelis-Menten constant, expressed in the same units as X, determined in an [experiment](#)³³⁶ without competitor.

Prism cannot fit any of these parameters from the graph of activity vs inhibitor concentration. You must know S from your experimental design, determine Km and Et in other experiments, and constrain all three to constant values.

Model

$$Q = (K_i * (1 + (S/K_m)))$$

$$Y = V_o * (1 - (((Et + X + Q) - (((Et + X + Q)^2 - 4 * Et * X)^{0.5})) / (2 * Et)))$$



Interpreting parameters

V₀ is the enzyme velocity with no inhibitor, expressed in the same units as Y. This is not the same as V_{max}, which would require a maximal concentration of substrate.

K_i is the inhibition constant, expressed in the same units as X.

Reference

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

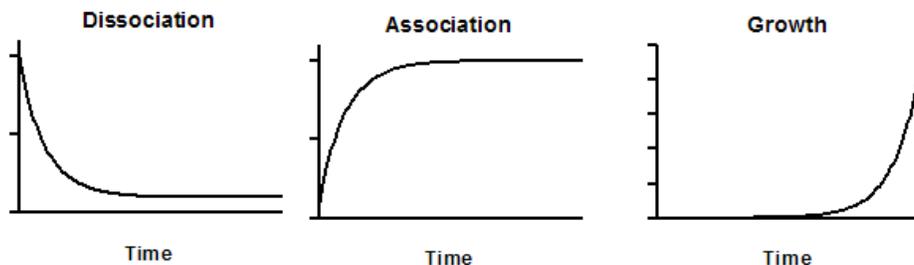
2.7.12 Exponential

2.7.12.1 Key concepts: Exponential equations

What is exponential?

Processes follow exponential models when the rate at which something is happening depends on the amount that is present.

Exponential dissociation vs. association vs. growth



The dissociation model always heads downhill gradually approaching a plateau.

The association model always heads uphill, and also approaches a plateau.

The growth model goes up and up and up, getting steeper, never reaching a plateau.

Rate constants vs. time constants vs. half-lives

In all the exponential models, one (or more in some cases) parameter describes how rapidly the process occurs. It can be expressed as a rate constant (in units of inverse time) or as a time constant (in units of time), or as a half-life, also in units of time.

The rate constant and time constants are simply reciprocals of each other. Prism always fits the rate constant (k), but computes the time constant (τ) 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.

2.7.12.2 Key concepts: Derivation of exponential decay

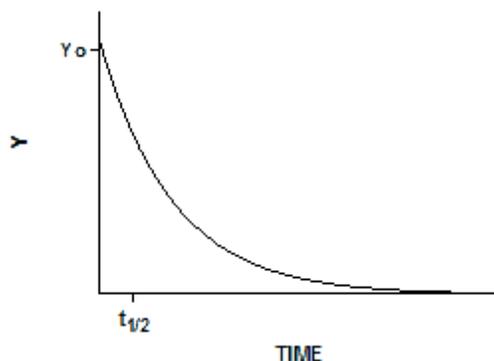
Define X to be time, and Y to be the outcome you are measuring. Three examples:

- Number of ligand-receptor complexes still present -- the ligand has not yet dissociated.
- The number of radioactive atoms that have not yet decayed.
- The concentration of drug in the plasma, not yet metabolized.

At any given time X . The rate of change of Y is proportional to Y . Expressed as a differential equation:

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

Shown as a graph:



When you integrate both sides of the equation, you get the equation for exponential decay:

$$Y=Y_0 * \exp(-k*X)$$

The function $\exp()$ takes the constant e (2.718...) to the power contained inside the parentheses.

2.7.12.3 Equation: One phase decay

Introduction

An exponential decay equation models many chemical and biological processes. It is used whenever the rate at which something happens is proportional to the amount which is left. Here are three examples:

- When ligands dissociate from receptors, the number of molecules that dissociate in any short time interval is proportional to the number that were bound at the beginning of that interval. Equivalently, each individual molecule of ligand bound to a receptor has a certain probability of dissociating from the receptor in any small time interval. That probability does not get higher as the ligand stays on the receptor longer.
- When radioactive isotopes decay, the number of atoms that decay in any short interval is proportional to the number of undecayed atoms that were present at the beginning of the interval. This means that each individual atom has a certain probability of decaying in a small time interval, and that probability is constant. The probability that any particular atom will decay does not change over time. The total decay of the sample decreases with time because there are fewer and fewer undecayed atoms.
- When drugs are metabolized by the liver or excreted by the kidney, the rate of metabolism or excretion is often proportional to the concentration of drug in the blood plasma. Each drug molecule has a certain probability of being metabolized or secreted in a small time interval. As the drug concentration goes down, the rate of its metabolism or excretion goes down as well.

Entering data

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

You can also choose a sample data set for exponential decay.

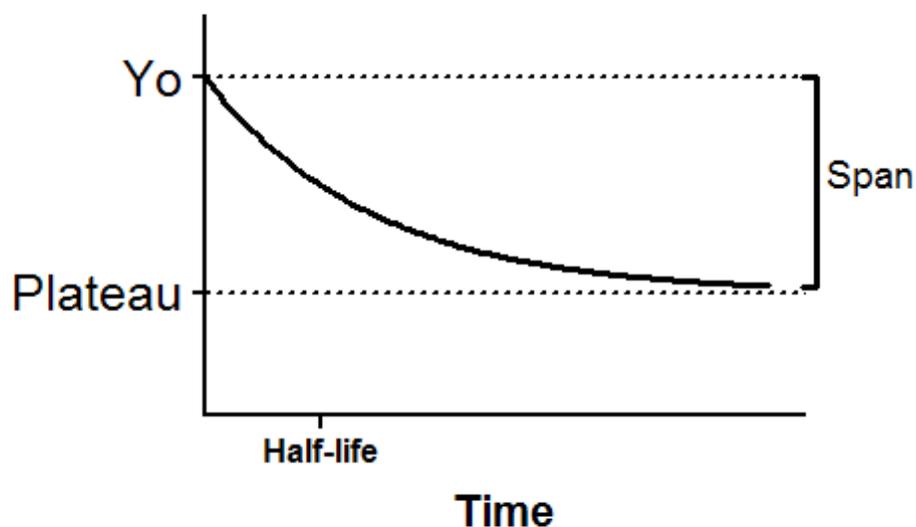
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *One phase decay*.

Consider constraining Plateau to a constant value of zero

If you have subtracted off any background signal, then you know the curve has to plateau at $Y=0$. In this case, you should constrain the parameter Plateau to be a constant value equal to zero. To do this, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to Plateau to "Constant equal to" and enter the value 0.0.

Model

$$Y = (Y_0 - \text{Plateau}) * \exp(-K * X) + \text{Plateau}$$



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 Y_0 and Plateau, expressed in the same units as your Y values.

The derivative of an exponential decay equals $-k*Y$. So the initial rate equals $-k*Y_0$.

2.7.12.4 Equation: Plateau followed by one phase decay

Introduction

In the standard [one-phase decay](#)³⁵⁷ 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 X_0 .

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 X_0 and Plateau to a constant values

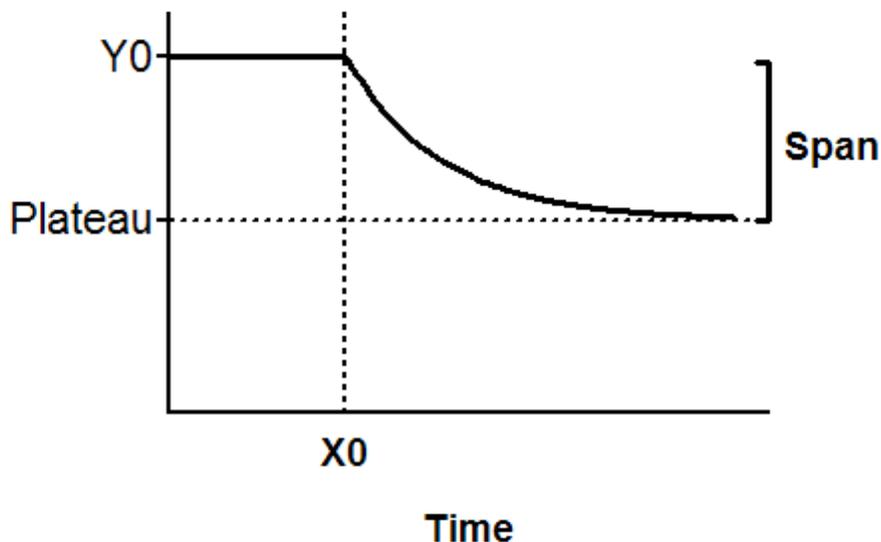
If you know the time at which you initiated the decay, you should constrain X_0 to that value.

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

$$Y = IF(X < X_0, Y_0, Plateau + (Y_0 - Plateau) * \exp(-K * (X - X_0)))$$



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.

2.7.12.5 Equation: Two phase decay

Introduction

An exponential decay equation models many chemical and biological processes. It is used whenever the rate at which something happens is proportional to the

amount which is left.

A two-phase model is used when the outcome you measure is the result of the sum of a fast and slow exponential decay. This is also called a double exponential decay.

Entering data

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

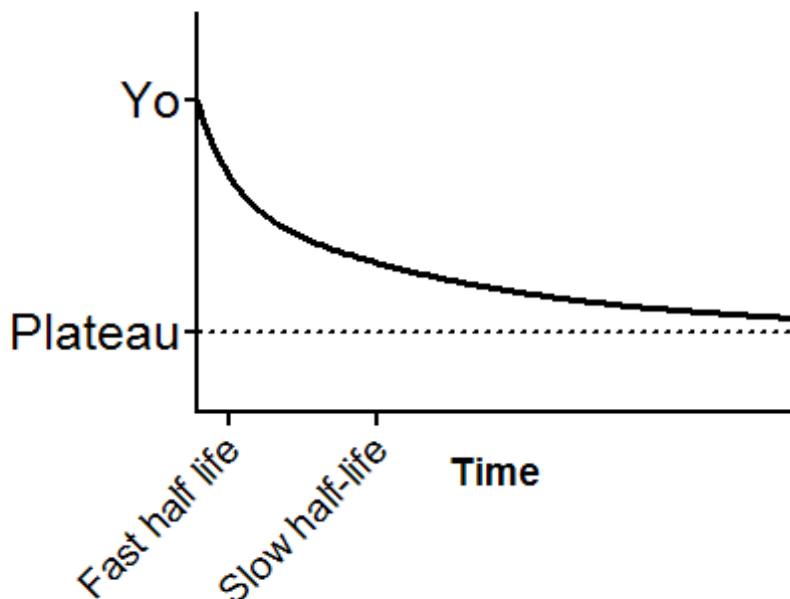
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Two phase decay*.

Consider constraining Plateau to a constant value of zero

If you have subtracted off any background signal, then you know the curve has to plateau at $Y=0$. In this case, you should constrain the parameter Plateau to be a constant value equal to zero. To do this, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to Plateau to "Constant equal to" and enter the value 0.0.

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.

2.7.12.6 Equation: Three phase decay

Introduction

An exponential decay equation models many chemical and biological processes. It is used whenever the rate at which something happens is proportional to the

amount which is left.

A three-phase model is used when the outcome you measure is the result of the sum of a fast, medium and slow exponential decay. You need lots of data with little scatter to adequately fit a three phase model.

Entering data

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

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

Consider constraining Plateau to a constant value of zero

If you have subtracted off any background signal, then you know the curve has to plateau at $Y=0$. In this case, you should constrain the parameter Plateau to be a constant value equal to zero. To do this, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to Plateau to "Constant equal to" and enter the value 0.0.

Model

```
YFast=(Y0-Plateau)*PercentFast*.01*exp(-KFast*X)
YSlow=(Y0-Plateau)*PercentSlow*.01*exp(-KSlow*X)
YMedium=(Y0-Plateau)*(100-PercentFast - PercentSlow)*.01*exp(-Kmedium*X)
Y=Plateau + YFast + YMedium +YSlow
```

Y0 is the Y value when X (time) is zero. It is expressed in the same units as Y,

Plateau is the Y value at infinite times, expressed in the same units as Y.

Kfast, **Kmedium** and **Kslow** are the rate constants, expressed in reciprocal of the X axis time units. If X is in minutes, the rate constants are expressed in inverse minutes.

Half-life (fast, medium and slow) are in the time units of the X axis. they are computed as $\ln(2)$ divided by the corresponding rate constant.

PercentFast is the percentage of the span (from Y0 to Plateau) accounted for by the fastest of the three components.

PercentSlow is the percentage of the span (from Y0 to Plateau) accounted for by the slowest of the three components.

2.7.12.7 Equation: One phase association

Introduction

This equation describes the pseudo-first order association kinetics of the interaction between a ligand and its receptor, or a substrate and an enzyme. During each time interval a certain fraction of the unoccupied receptors become occupied. But as time advances, fewer receptors are unoccupied so fewer ligand bind and the curve levels off.

Entering data

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

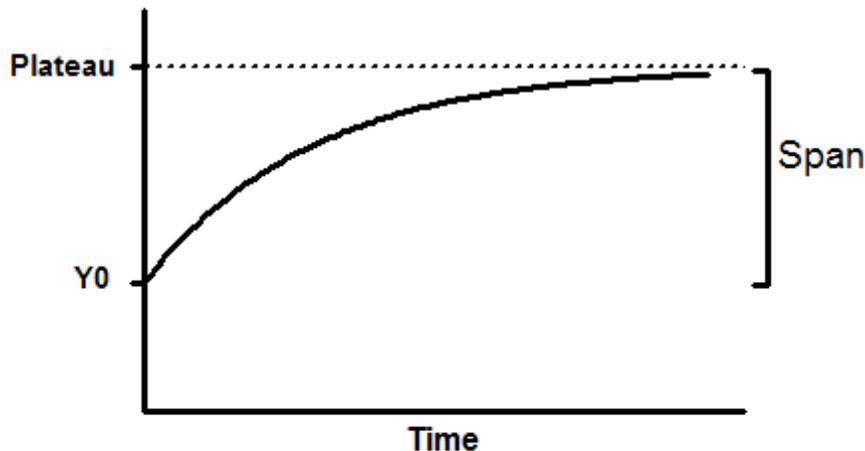
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *One phase association*.

Consider constraining 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

$$Y = Y_0 + (\text{Plateau} - Y_0) * (1 - \exp(-K * x))$$



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.

2.7.12.8 Equation: Plateau followed by one phase association

Introduction

In the standard [one-phase association](#)³⁶⁴ 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.

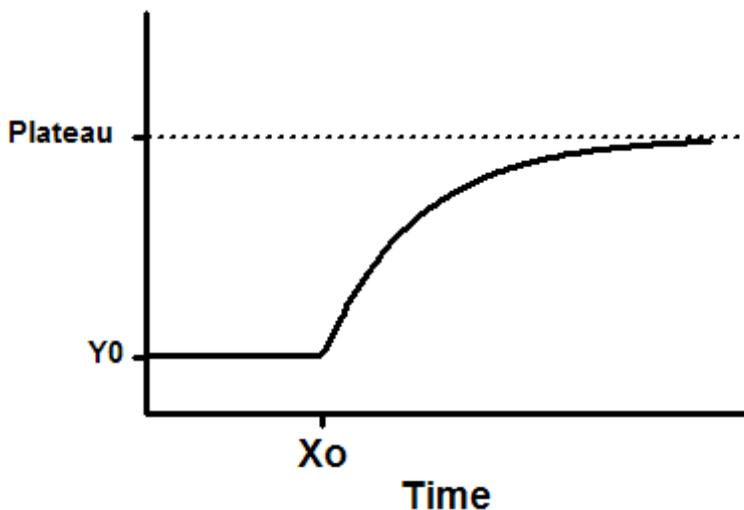
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Plateau followed by one phase association*.

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

$$Y = \text{IF}(X < X_0, Y_0, Y_0 + (\text{Plateau} - Y_0) * (1 - \exp(-K * (X - X_0))))$$



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.

2.7.12.9 Equation: Two phase association

Introduction

An exponential decay equation models many chemical and biological processes. It is used whenever the rate at which something happens is proportional to the amount which is left.

A two-phase model is used when the outcome you measure is the result of the sum of a fast and slow exponential decay.

Entering data

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

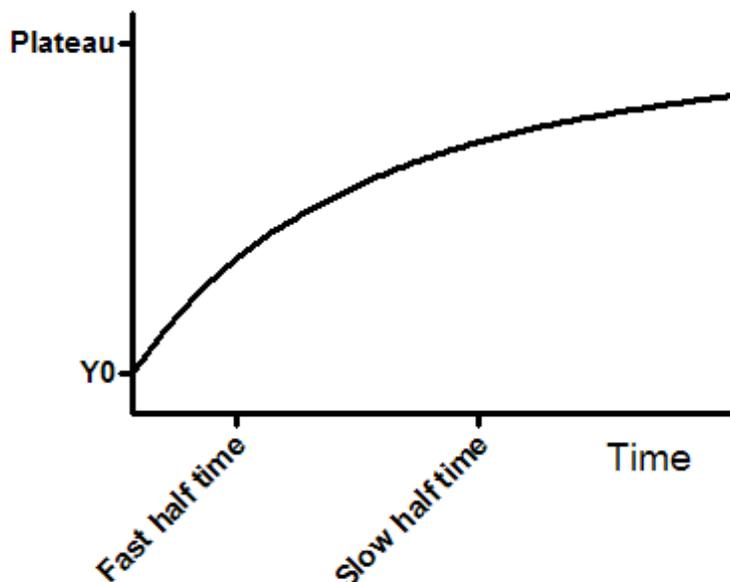
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Two phase association*.

Consider constraining Plateau to a constant value of zero

If you have subtracted off any background signal, then you know the curve has to plateau at Y=0. In this case, you should constrain the parameter Plateau to be a constant value equal to zero. To do this, go to the Constrain tab of the nonlinear regression dialog, set the drop down next to Plateau to "Constant equal to" and enter the value 0.0.

Model

```
SpanFast=(Plateau-Y0)*PercentFast*.01
SpanSlow=(Plateau-Y0)*(100-PercentFast)*.01
Y=Y0+ SpanFast*(1-exp(-KFast*X)) + SpanSlow*(1-exp(-KSlow*X))
```



Y0 is the Y value when X (time) is zero. It is expressed in the same units as Y,

Plateau is the Y value at infinite times, expressed in the same units as Y.

Kfast and **Kslow** are the two rate constant, expressed in reciprocal of the X axis time units. If X is in minutes, then K is expressed in inverse minutes.

TauFast and **TauSlow** are the two time constants, expressed in the same units as the X axis. They are computed as the reciprocals of the rate constants.

Half-time (fast) and **Half-time (slow)** are in the time units of the X axis. They are computed as $\ln(2)/K$.

PercentFast is the fraction of the span (from Y0 to Plateau) accounted for by the faster of the two components.

2.7.12.10 Equation: Exponential growth

Introduction

This equation describes the growth with a constant doubling time.

Entering data

Create an XY data table. Enter time into X, and response (cell number ..) into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

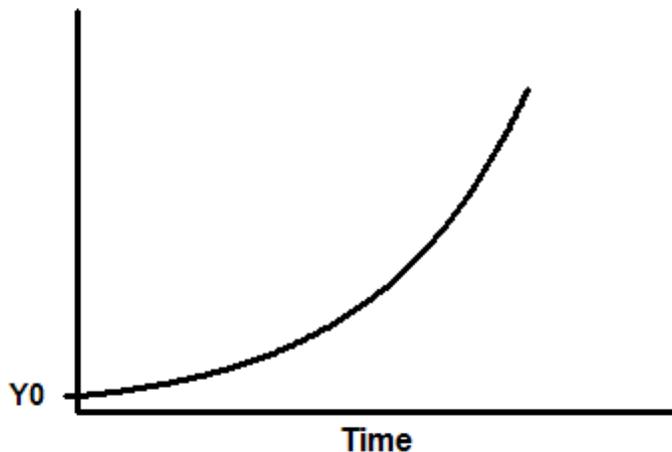
After entering data, click Analyze, choose nonlinear regression, choose the panel of exponential equations, and choose *Exponential growth*.

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

$$Y = Y_0 * \exp(k * X)$$



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 $\ln(2)/K$.

2.7.13 Lines

2.7.13.1 Key concepts: Fitting lines

Choosing nonlinear regression, rather than linear regression to fit a line

Prism offers separate analyses for linear regression and nonlinear regression. But the nonlinear regression analysis can fit a straight-line model. This is useful when you want to take advantage of features in Prism's nonlinear regression analysis that Prism does not offer in its linear regression analysis, such as the ability to [compare two models](#)^[157], apply [weighting](#)^[158], or automatically [exclude outliers](#)^[59]. See a longer discussion of [the advantages of using the nonlinear regression analysis to fit a straight line](#)^[82].

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](#)^[374] 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.

2.7.13.2 Equation: Fitting a straight line with nonlinear regression

Introduction

Linear regression fits a straight line through your data. Nonlinear regression fits any model, which includes a straight line model. Prism offers separate analyses for linear regression and nonlinear regression, so you can choose either one to fit a line.

Prism's nonlinear regression analysis offers more options than its [linear regression analysis](#)^[13], such as the ability to [compare two models](#)^[157], apply [weighting](#)^[158], automatically [exclude outliers](#)^[59] and perform [normality tests](#)^[185] on the residuals. See a longer discussion of [the advantages of using the nonlinear regression analysis to fit a straight line](#)^[82].

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

$$Y = Y_{\text{Intercept}} + \text{Slope} * X$$

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:

$$Y = \text{Mean} + 0 * X$$

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](#)"^[372] model built in:

$$Y = \text{Slope} * X$$

The only parameter is the slope.

2.7.13.3 Equation: Line through origin

Introduction

[Prism's linear regression analysis](#)^[76] fits a straight line through your data, and lets you force the line to go through the origin. This is useful when you are sure that the line must begin at the origin (X=0 and Y=0).

Prism's nonlinear regression offers the equation Line through origin. It offers more options than its [linear regression analysis](#)^[13], such as the ability to [compare two models](#)^[157], apply [weighting](#)^[158], automatically [exclude outliers](#)^[59] and perform [normality tests](#)^[185] on the residuals. See a longer discussion of [the advantages of using the nonlinear regression analysis to fit a straight line](#)^[82].

When to force the line through the origin

In many scientific situations, it just makes sense that when X=0, Y must also equal 0, so the line should be forced to go through the origin (X=0, Y=0). But even in these situations, it can make sense to fit an ordinary linear regression line that also fits the intercept. The data you are analyzing may be far from the origin, and you may get a better fit through the points (what you care about) when you don't force the line to go through the origin. This can happen when the true model is curved (beginning at the origin), so a line through the data points that is not forced to the origin may fit a whole lot better than line forced to go through the origin.

Prism makes it easy to [compare](#)^[157] the fit that goes through the origin with one that doesn't.

Step by step

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

After entering data, click Analyze, choose nonlinear regression, choose the panel equations for lines, and choose *Line Through Origin*

Model

$$Y = \text{Slope} * X$$

Slope is the slope of the line, expressed in Y units divided by X units. It estimates the ratio of Y/X in the entire population.

Weighting

In situations where linear regression through the origin is appropriate, it is common for the variation among replicate Y values increases as X (and Y) increase. Prism provides two weighting choices (in the Weights tab) that are used in this situation. Weight by $1/X^2$ when you think the variance in Y is proportional to the square of X, which means the SD among Y values is proportional to X. Weight by $1/X$ if you think the variance in Y is proportional to X.

R² when fitting a line through the origin

When you constrain a line to go through a point, there are two possible ways to compute R²:

- Compare the fit of the best-fit line with the fit of a horizontal line at the mean Y value. But that null hypothesis (horizontal line through the Ymean) doesn't obey the constraint that it go through the origin.
- Compare the best-fit line with a horizontal line at Y=0. This obeys the constraint, but often fits the data really badly, pushing up the R² value.

When you use nonlinear regression to fit a line through the origin, Prism uses the first definition above. If you use linear regression, and ask Prism to constrain the line to go through the origin, it simply will not report R² at all.

Reference

1. J. G. Eisenhauer, Regression through the Origin. *Teaching Statistics* **25**, 76–80 (2003).

2.7.13.4 Equation: Segmental linear regression

Introduction

Segmental regression fits one line to all data points with X less than some value X_0 , and another line to all points with X greater than X_0 , while ensuring that the two lines intersect at X_0 .

Segmental linear regression is helpful when X is time, and you did something at $\text{time}=X_0$ to change the slope of the line. Perhaps you injected a drug, or rapidly changed the temperature. In these cases, your model really does have two slopes with a sharp transition point.

In other cases, the true model has the slope gradually changing. The data fit a curve, not two straight lines. In this situation, fitting the data with segmental linear regression is not helpful.

Caution

Don't use segmental linear regression to analyze a biphasic Scatchard or Lineweaver-Burk plot. A biphasic Scatchard plot follows a curve, not two intersecting lines. There is no abrupt break point. You should fit the original data to a two-site binding curve instead.

Step by step

Create an XY data table. Enter time into X, and your measurements into Y. If you have several experimental conditions, place the first into column A, the second into column B, etc.

After entering data, click Analyze, choose nonlinear regression, choose the panel equations for lines, and choose *Segmental linear regression*.

Model

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

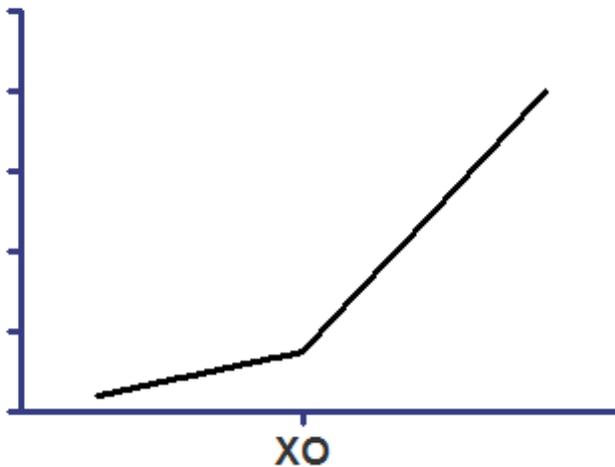
The first line of the equation defines the first line segment from its intercept and slope.

The second line of the equation computes the Y value of the first regression at

the right end of that segment, when $X=X_0$.

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 (Y_{atX_0}). The Y value at any other position along the second segment equals Y_{atX_0} plus the increase due to the second regression line. That increase equals the slope of the second segment (slope2) times the distance from X_0 to X .

The final line defines Y for all values of X. If X is less than X_0 then Y is set equal to Y_1 . Otherwise Y is set equal to Y_2 .



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.

X_0 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 a 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))

```

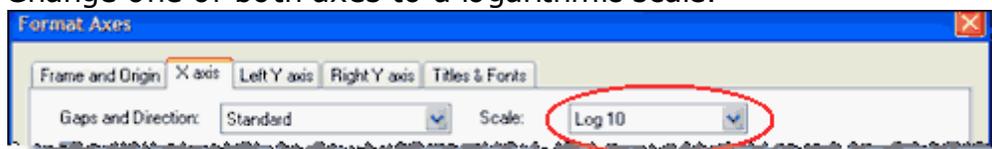
2.7.13.5 Equation: Fitting a straight line on a semi-log or log-log graph

Straight lines on graphs with logarithmic axes

The nonlinear regression analysis fits the data, not the graph. Since Prism lets you choose logarithmic axes, some graphs with data points that form a straight line follow nonlinear relationships. Prism's collection of "Lines" equations includes those that let you fit nonlinear models to graphs that appear linear when the X axis is logarithmic, the Y axis is logarithmic, or both axes are logarithmic. In these cases, linear regression will fit a straight line to the data but the graph will appear curved since an axis (or both axes) are not linear. In contrast, nonlinear regression to an appropriate nonlinear model will create a curve that appears straight on these axes.

Entering and fitting data

1. Create an XY table, and enter your X and Y values.
2. Go to the graph, double click on an axis to bring up the Format Axis dialog. Change one or both axes to a logarithmic scale.



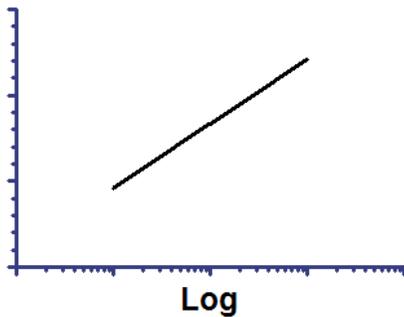
3. Click Analyze, choose Nonlinear regression (not Linear regression) and then choose one of the semi-log or log-log equations from the "Lines" section of equations.

Equations

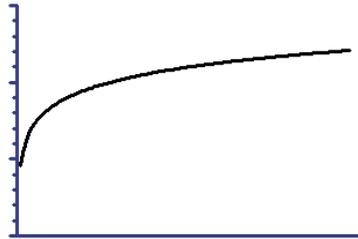
Semilog line -- X axis is logarithmic, Y axis is linear

$$Y = Y_{\text{intercept}} + \text{Slope} * \log(X)$$

On semilog axis



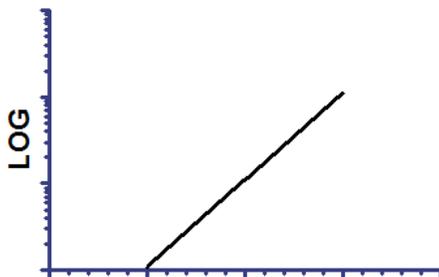
On linear axes



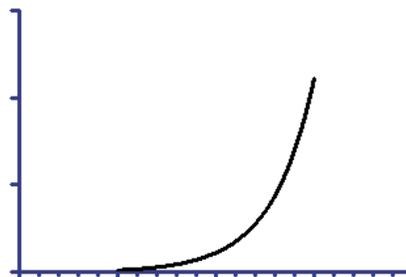
Semilog line -- X axis is linear, Y axis is logarithmic

$$Y = 10^{(\text{Slope} * X + Y_{\text{intercept}})}$$

On semilog axis

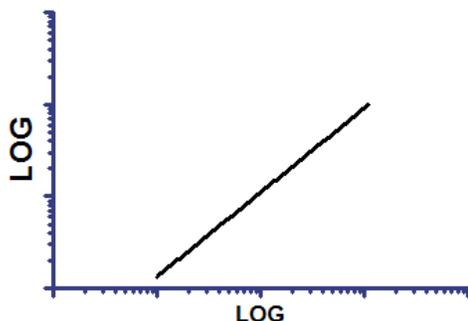


On linear axes



Log-log line -- Both X and Y axes are logarithmic

$$Y = 10^{(\text{slope} * \log(X) + Y_{\text{intercept}})}$$

On log-log axes**On linear axes**

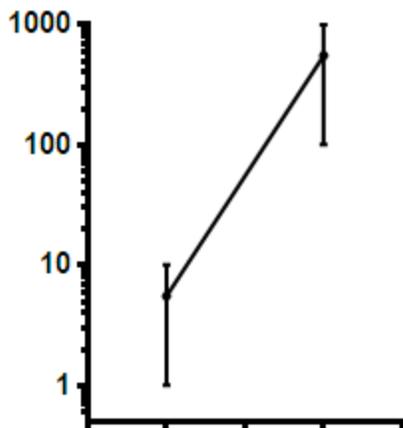
Since both axes are transformed the same way, the graph is linear on both sets of axes. But when you fit the data, the two fits will not be quite identical.

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.

An alternative way to handle these data

The nonlinear regression analysis minimizes the sum of the squares of the difference between the actual Y value and the Y value predicted by the curve. This is not the same as minimizing the sum of squares of the distances (as seen on the graph) between points and curve. In the graph below, the two vertical lines look the same distance but one represents a difference of 9 Y units, and the other a difference of 900.



An alternative approach, that might be better in some circumstances, is to use Prism's transform analysis to transform Y (and maybe also X) to logarithms. Then perform linear regression on the logarithms. The regression results will not be the same as using nonlinear regression on log axes.

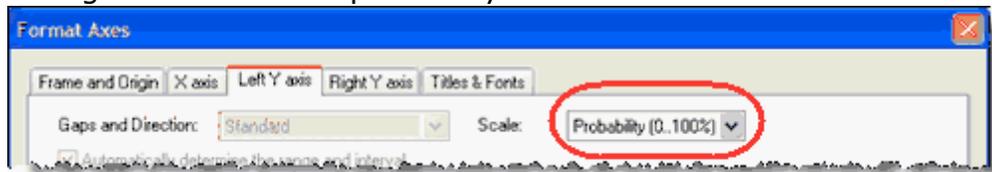
2.7.13.6 Equation: Fitting a straight line on a graph with a probability axis

Fitting straight lines on graphs with nonlinear axes

The nonlinear regression analysis fits the data, not the graph. If you choose probability Y axis, graphs with data points that form a straight line follow nonlinear relationships. Prism's collection of "Lines" equations includes those that let you fit nonlinear models to graphs that appear linear when the Y axis is a probability axis. In these cases, linear regression will fit a straight line to the data but the graph will appear curved since an axis (or both axes) are not linear. In contrast, nonlinear regression to an appropriate nonlinear model will create a curve that appears straight on these axes.

Entering and fitting data

1. Create an XY table, and enter your X and Y values.
2. Go to the graph, double click on an axis to bring up the Format Axis dialog. Change the Y axis to a probability scale.

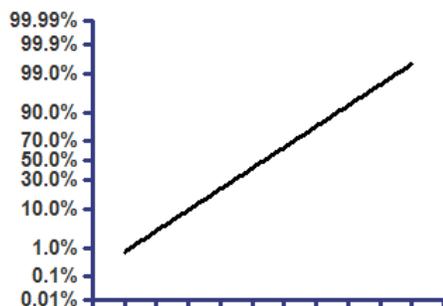
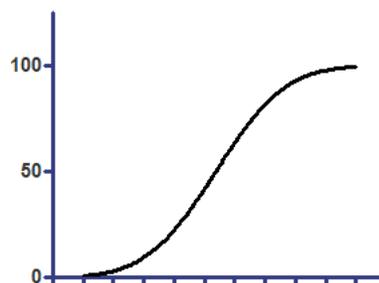


3. Click Analyze, choose Nonlinear regression (not Linear regression) and then choose one of the Cumulative Gaussian distribution equations from the "Lines" section of equations.

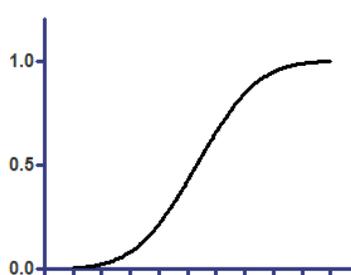
Equations

Cumulative Gaussian - Y values are percentages

$$\begin{aligned} \text{Top} &= 100 \\ z &= (X - \text{Mean}) / \text{SD} \\ Y &= \text{Top} * \text{zdist}(z) \end{aligned}$$

On probability axis**On linear axis****Cumulative Gaussian - Y values are fractions**

$$\begin{aligned} \text{Top} &= 1 \\ z &= (X - \text{Mean}) / \text{SD} \\ Y &= \text{Top} * \text{zdist}(z) \end{aligned}$$

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.

2.7.14 Polynomial

2.7.14.1 Key concepts: Polynomial

Usefulness of polynomial models

There are two situations where you might want to choose a polynomial model:

- Your scientific model is described by a polynomial equation. This is rare in biology. Few chemical or pharmacological models are described by polynomial equations.
- You don't have a scientific model, but want to fit a curve to interpolate unknown values. With this goal, you often don't care much about the details of the model. Instead, you care only about finding a model that goes near the data points. Polynomial models often work well.

Which polynomial model?

The order of a polynomial model expresses how many terms it has. Prism offers up to a sixth order equation (and it would be easier to enter higher order equations). The higher order equations have more inflection points.

Choosing the best polynomial model is often a matter of trial and error. If the curve doesn't follow the trend of your data, pick a higher order equation. If it wiggles too much, pick a lower order equation.

How are polynomial models special?

To a mathematician, polynomial models are very special. Strictly speaking, polynomial models are not 'nonlinear'. Even though a graph of X vs. Y is curved (in all but some special cases), the derivative of Y with respect to the parameters is linear.

Because polynomial models are not nonlinear, it is possible (but not with Prism) to fit polynomial models without fussing with initial values. And the fit can be in one step, rather than the iterative approach used for nonlinear models.

Since Prism treats polynomial models the same way it treats nonlinear models, it does require initial values (it chooses 1.0 for each parameter automatically). It doesn't matter what values are used -- polynomial regression cannot encounter [false minima](#)^[206].

Why you should choose a centered polynomial equation

There are two problems with polynomial fits, often solved by centering:

- When the X values are large, and start well above zero (for example, when X is a calendar year), taking the very large X values to large powers can lead to math errors.
- Even when the X values are not large, the parameters of the model are intertwined, so have high covariance and dependency. This results in large standard errors, wide confidence intervals, and huge confidence or prediction bands.

Both these problems are solved by using a centered polynomial model. The idea of centering is simple. Subtract the mean of all X values from each X value, and use those differences instead of X in the model.

Fitting the centered model leads to exactly the same curve (unless the regular approach led to math errors). Accordingly, the sum-of-squares are R^2 are the same, as are results of model comparisons. However, the parameters have different meanings, so have different best-fit values (except the first parameter which is the same), different standard errors and confidence intervals, smaller covariances and dependencies, and tighter confidence/prediction bands.

[Read more about centered polynomial models.](#) 

2.7.14.2 Centered polynomial equations

Included in Prism, are both a set of ordinary polynomial equations and also a set of centered polynomial equations. For example, when you look in the list of polynomials you'll see both 'Second order polynomial' and 'Centered second order polynomial'. We recommend always choosing one of the centered equations instead of an ordinary polynomial equation. This page explains why.

What's wrong with ordinary polynomial models?

The standard polynomial models look like this:

$$Y = B_0 + B_1 * X + B_2 * X^2$$

More terms are included with the higher order equations.

There are two problems with polynomial fits:

- When the X values are large, and start well above zero (for example, when X is a calendar year), taking the very large X values to large powers can lead to math overflow. Even if the program doesn't report any math error, the results can be inaccurate. Some coefficients will be positive and some negative, so the value of Y depends on subtracting huge numbers from other

huge numbers, leading to imprecise results.

- Even when the X values are not large, the parameters of the model are intertwined, so have high covariance and dependency. This results in large standard errors, wide confidence intervals, and very wide confidence or prediction bands. In many cases, this problem is severe enough that Prism reports that the results are 'ambiguous' and so doesn't report confidence intervals for all the parameters and can't graph confidence bands.

What are centered polynomial models?

Both problems go away when the X values are centered. The idea of centering is to subtract the mean X from all X values before fitting the model. This can be done as part of nonlinear regression, using this model:

$$\begin{aligned}XC &= X - X_{\text{mean}} \\ Y &= B_0 + B_1 * XC + B_2 * XC^2\end{aligned}$$

Here XC is the centered X value, equal to the X value minus Xmean, which is the mean of all X values. In other words, XC is the distance of any X value from the mean of all X values. Xmean is constant, and not a parameter that Prism tries to fit. Of course, you can include more terms in the definition of Y to create higher order polynomial equations.

The advantages of centered models

Fitting the centered model leads to exactly the same curve (unless the regular approach led to math errors). Accordingly, the sum-of-squares is the same, as are results of model comparisons.

However, the centered equation has reparameterized the model. The parameters have different meanings, so have different best-fit values (except the first parameter which is the same), different standard errors and confidence intervals, smaller covariances and dependencies, and tighter confidence/prediction bands.

How centered models are implemented in Prism

You can fit data to a built-in centered polynomial equation without knowing how Prism implements the model. If you are curious, read on.

The built-in set of centered polynomial equations, written as shown above, constrain the parameter XMean to equal the mean of X value by constraining it to equal a "Data set constant (= Mean X)".

If you open a file using centered polynomial regression in an version of Prism

prior to 5.02 or 5.0b, that constraint will be lost, and centered polynomial regression won't work.

2.7.14.3 Equations: Polynomial models

Step by step

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

After entering data, click Analyze, choose nonlinear regression, choose the panel equations for polynomial equations, and choose one.

The "order" of a polynomial equation tells you how many terms are in the equation. Prism offers first to sixth order polynomial equations (and you could enter higher order equations as user-defined equations if you need them). Higher order models wiggle more than do lower order models. Since the equation rarely corresponds to a scientific model, use trial and error. If it isn't close enough to the data, pick a higher order equation. If it wiggles too much, pick a lower order equation.

Polynomial models

Order	Equation
First	$Y=B_0 + B_1*X$ (straight line)
Second	$Y=B_0 + B_1*X + B_2*X^2$ (quadratic equation)
Third	$Y=B_0 + B_1*X + B_2*X^2 + B_3*X^3$
Fourth	$Y=B_0 + B_1*X + B_2*X^2 + B_3*X^3 + B_4*X^4$
Fifth	$Y=B_0 + B_1*X + B_2*X^2 + B_3*X^3 + B_4*X^4 + B_5*X^5$
Sixth	$Y=B_0 + B_1*X + B_2*X^2 + B_3*X^3 + B_4*X^4 + B_5*X^5 + B_6*X^6$

There is no general way to interpret the coefficients B_0 , B_1 , etc. In most cases, the goal of fitting a polynomial model is to make a curve that looks good, and the parameters really don't matter.

Centered polynomial models

The [Centered polynomial models](#)³⁸² are identical to the ones listed above, with one exception. Wherever X appears above, replace it with $(X - X_{\text{Mean}})$, where X_{Mean} is the mean of all X values (for rows that have Y values). Using a centered model can avoid computer math problems (overflows), and we recommend that you use them routinely.

2.7.15 Gaussian

2.7.15.1 Key concepts: Gaussian

What is a frequency distribution?

All the equations in this section fit frequency distributions. The X values represent Y values in the original data set, and the Y values are the frequency or cumulative frequency.

Prism can create a frequency distribution from column data, using an analysis created for that purpose (called, of course, Frequency Distribution).

What is a cumulative frequency distribution?

A frequency distribution plots the number of observations as a function of value. A cumulative frequency distribution plots the *cumulative* number of observations as a function of value. Each Y value is the number of observations in the original data set that have a value less than or equal to the X value.

The Y values can be expressed as the counted number of observations, as fractions or as percentages.

2.7.15.2 Equation: Gaussian distribution

Introduction

Data follow a Gaussian distribution when scatter is caused by the sum of many independent and equally weighted factors.

A frequency distribution (histogram) created from Gaussian data will look like a bell-shaped Gaussian distribution.

Step-by-step

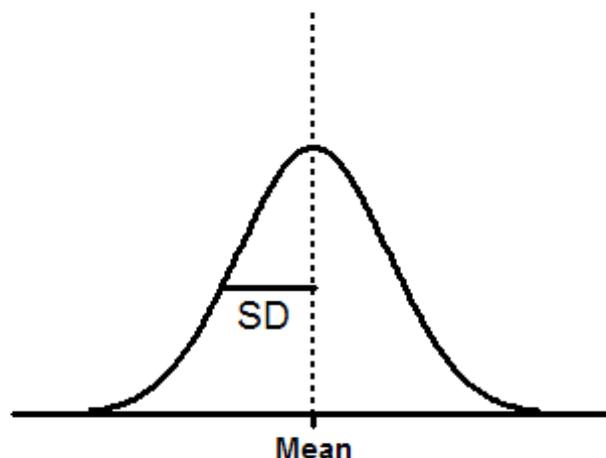
The data you fit must be in the form of a frequency distribution on an XY table. The X values are the bin center and the Y values are the number of observations.

If you start with a column of data, and use Prism to [create the frequency distribution](#), make sure that you set the graph type to "XY graph", with either points or histogram spikes. This ensures that Prism creates an XY results table with the bin centers entered as X values. If you pick a bar graph instead, Prism creates a column results table, creating row labels from the bin centers. This kind of table cannot be fit by nonlinear regression, as it has no X values.

Starting from the frequency distribution table, click Analyze, choose Nonlinear regression from the list of XY analyses, and then choose the "Gaussian" equation from the "Gaussian" family of equations.

Model (Gaussian distribution)

$$Y = \text{Amplitude} * \exp(-0.5 * ((X - \text{Mean}) / \text{SD})^2)$$



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.

$$\begin{aligned}\text{One} &= \text{Amplitude1} * \exp(-0.5 * ((X - \text{Mean1}) / \text{SD1})^2) \\ \text{Two} &= \text{Amplitude2} * \exp(-0.5 * ((X - \text{Mean2}) / \text{SD2})^2) \\ Y &= \text{One} + \text{Two}\end{aligned}$$

Amplitude1 and **Amplitude2** are the heights of the center of the distribution in Y units.

Mean1 and **Mean2** are the X values at the center of the two distributions.

SD1 and **SD2** are measures of the widths of the distributions, in the same units as X.

Prism is not very smart about assigning initial values to the parameters. If you have trouble getting this model to fit, try fussing with the initial parameter values.

Area under a Gaussian distribution

The area under a Gaussian distribution equals $\text{Amplitude} * \text{SD} / 0.3989$. That constant equals the reciprocal of the square root of two pi.

2.7.15.3 Equation: Log Gaussian distribution

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

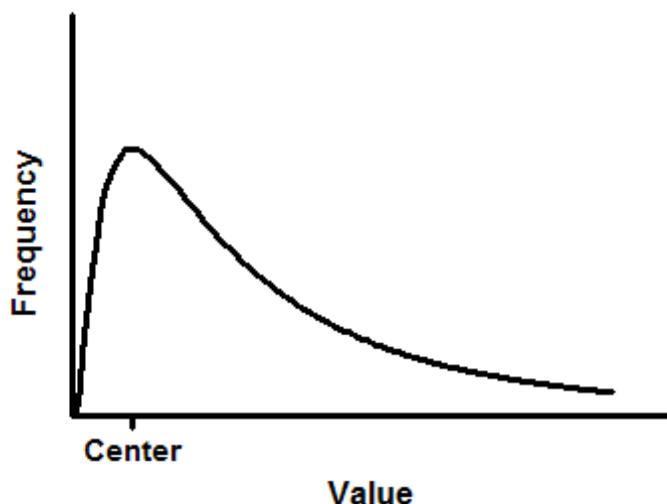
The data must be in the form of a frequency distribution on an XY table. The X values are the bin center and the Y values are the number of observations.

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

$$Y = \text{Amplitude} * \exp(-0.5 * (\ln(X / \text{Center}) / \text{Width})^2)$$



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. It is also called the [geometric standard deviation](#). If you transformed all the X values to their natural logarithms, and then fit a Gaussian distribution, the SD would equal the Width from this log Gaussian fit.

A more standard form of the model

A more standard form of the model (from [Wikipedia](#) or [MathWorld](#)) is:

$$Y = (1 / (X * S * \sqrt{2 * \pi})) * \exp(-0.5 * (\ln(X) - M)^2 / (S^2))$$

This standard form of the equation doesn't have a parameter describing the amplitude as that is computed from S (what we call Width) and M (what we call Center). The area under the curve, using the equation above, is 1.0. To allow for a variable area under the curve (and thus a variable maximum height), you'd need to multiply that equation by an Area parameter.

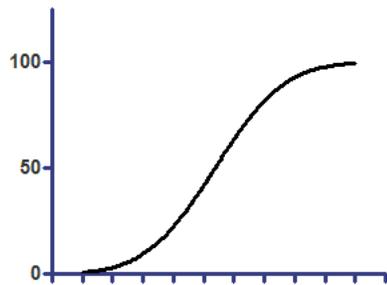
2.7.15.4 Equation: Cumulative Gaussian distribution

Introduction

A frequency distribution plots the number of observations as a function of value. A cumulative frequency distribution plots the *cumulative* number of observations as a function of value. Each Y value is the number of observations in the original data set that have a value less than or equal to the X value.

The advantage of creating a cumulative distribution is that you don't have to make any choice regarding bin width.

If your data follow a Gaussian distribution, the cumulative distribution has a sigmoidal shape.



Step-by-step

1. Create an XY table, and enter your X and Y values. The X values correspond to the value in the original data set, and the Y values are the number (or fraction or percent) of values in the original data set that are less than or equal to the X 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.

$$z = (X - \text{Mean}) / \text{SD}$$

$$Y = N * \text{zdist}(z)$$

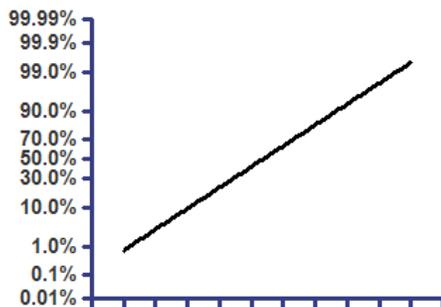
Mean is the average of the original distribution, from which the frequency distribution was created.

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.



2.7.15.5 Equation: Lorentzian

Introduction

A Lorentzian distribution is bell shaped, but has much wider tails than does a Gaussian distribution.

Step-by-step

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

If you start with a column of data, and use Prism to [create the frequency distribution](#), make sure that you set the graph type to "XY graph", with either points or histogram spikes. This ensures that Prism creates an XY results table with the bin centers 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)

$$Y = \text{Amplitude} / (1 + ((X - \text{Center}) / \text{Width})^2)$$

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)

$$\begin{aligned} \text{One} &= \text{Amplitude1} / (1 + ((X - \text{Center1}) / \text{Width1})^2) \\ \text{Two} &= \text{Amplitude2} / (1 + ((X - \text{Center2}) / \text{Width2})^2) \\ Y &= \text{One} + \text{Two} \end{aligned}$$

Amplitude1 and **Amplitude2** are the heights of the center of the distribution in Y units.

Center1 and **Center2** are the X values at the center of the two distributions.

Width1 and **Width2** are measures of the widths of the distributions, in the same units as X.

Prism is not very smart about assigning initial values to the parameters. If you have trouble getting this model to fit, try fussing with the initial parameter values.

2.7.16 Sine waves

2.7.16.1 Standard sine wave

Introduction

Sine waves describe many oscillating phenomena.

Step by step

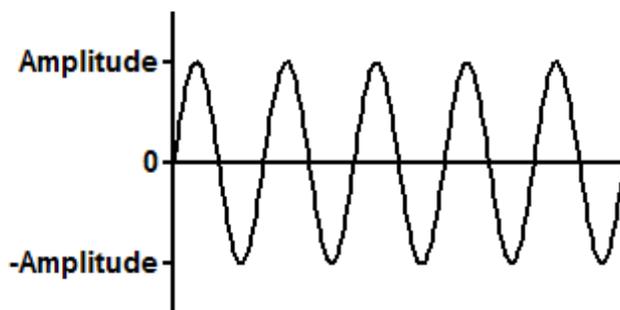
Create an XY data table. There is one X column, and many Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

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

$$Y = \text{Amplitude} * \sin((2 * \pi * X / \text{Wavelength}) + \text{PhaseShift})$$



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.

2.7.16.2 Damped sine wave

Introduction

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

Step by step

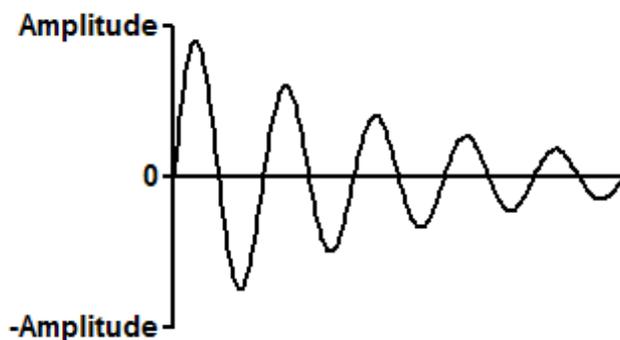
Create an XY data table. There is one X column, and many Y columns. If you have several experimental conditions, place the first into column A, the second into column B, etc.

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

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

Model

$$Y = \text{Amplitude} * \exp(-K * X) * \sin((2 * \pi * X / \text{Wavelength}) + \text{PhaseShift})$$



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.

HalfLife is the time it takes for the maximum amplitude to decrease by a factor of 2. It is computed as $0.693/K$.

2.7.16.3 Sinc wave

Introduction

The $\text{sinc}()$ 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.

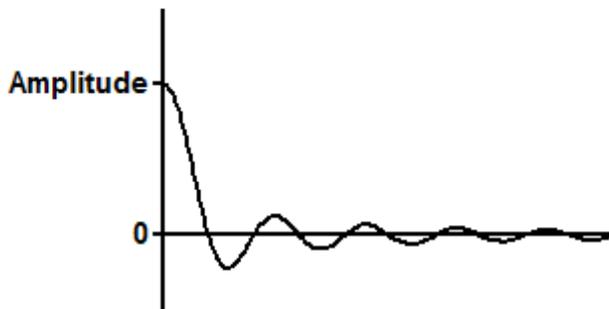
Step by step

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 $\text{Sinc}()$ function.

Model

$$Y = \text{IF}(X=0, \text{Amplitude}, \text{Amplitude} * \sin(2 * \pi * X / \text{Wavelength}) / (2 * \pi * X / \text{Wavelength}))$$



Interpret the parameters

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

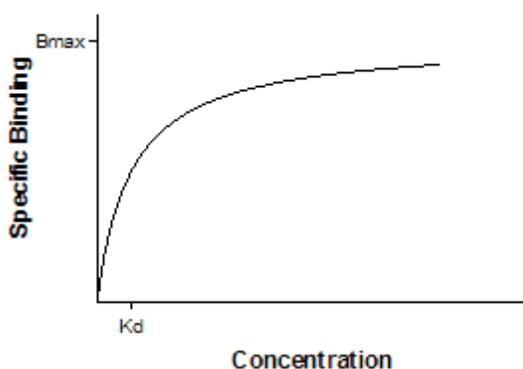
Wavelength is the time it takes for a complete cycle, in units of X

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.

2.7.17 Classic equations from prior versions of Prism

2.7.17.1 Equation: One site binding (hyperbola)

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



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

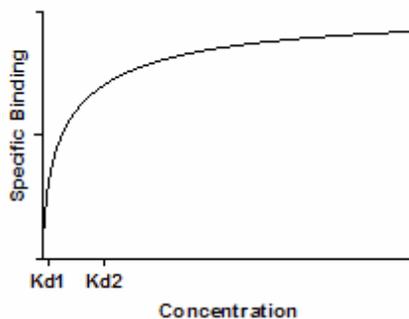
This equation describes the equilibrium binding of a ligand to a receptor as a function of increasing ligand concentration.

- X is the concentration of the ligand.
- Y is the specific binding.
- B_{\max} 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).
- K_d is the equilibrium dissociation constant, expressed in the same units as the X-axis (concentration). When the drug concentration equals K_d , 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.

2.7.17.2 Equation: Two site binding

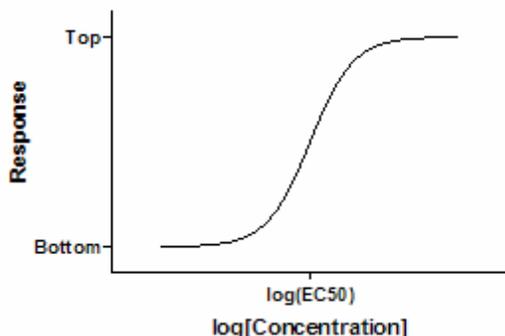
$$Y = \frac{B_{\max 1} \cdot X}{K_{d1} + X} + \frac{B_{\max 2} \cdot X}{K_{d2} + X}$$



This equation is an extension of the [one site binding curve](#)³⁹⁶. 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.

2.7.17.3 Equation: Sigmoidal dose-response

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



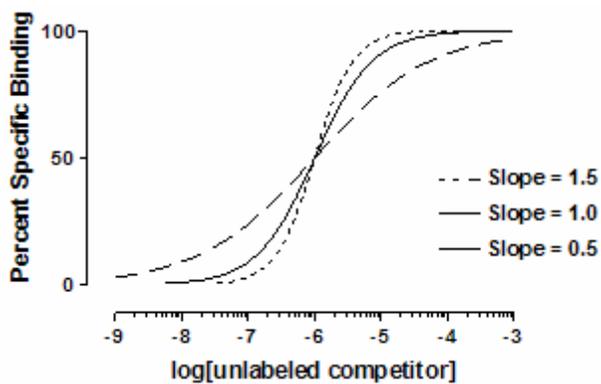
This is a general equation for a dose-response curve. It shows response as a function of the logarithm of concentration. X is the logarithm of agonist concentration and Y is the response. This equation is also called a three-parameter logistic equation.

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](#) ³⁹⁸ allows for a variable slope.

2.7.17.4 Equation: Sigmoidal dose-response (variable slope)

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



This equation extends the previous equation, but allows for a variable slope. This equation is also called a four-parameter logistic equation.

- Bottom is the Y value at the bottom plateau.
- Top is the Y value at the top plateau.
- LogEC50 is the X value when the response is halfway between Bottom and Top. With different kinds of variables, this variable is sometimes called ED50 (effective dose, 50%), or IC50 (inhibitory concentration, 50%, used when the curve goes downhill).
- HillSlope describes the steepness of the curve. This variable is called the Hill slope, the slope factor, or the Hill coefficient. If it is positive, the curve increases as X increases. If it is negative, the curve decreases as X increases. A standard sigmoid dose-response curve (previous equation) has a Hill Slope of 1.0. When HillSlope is less than 1.0, the curve is more shallow. When HillSlope is greater than 1.0, the curve is steeper. The Hill slope has no units.

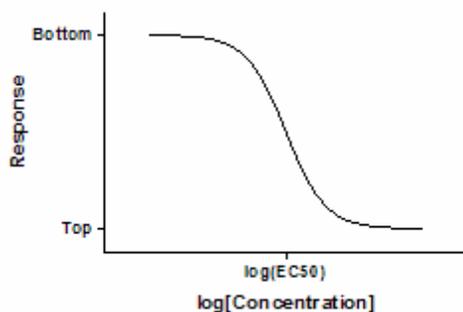
2.7.17.5 Equation: One site competition

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

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

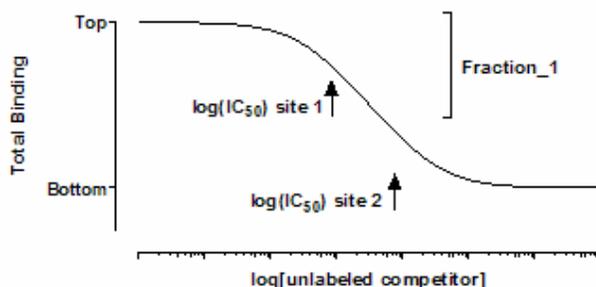
The variable LogEC50 is the concentration of the competitor required to compete for half the specific binding. We use the term EC50 to be consistent with the equations for the other sigmoid curves. The term IC50 is used more frequently ("E" stands for effective; "I" stands for inhibitory).

Usually the Y values are total binding. If you enter specific binding instead, fix BOTTOM to have a constant value of zero. If you enter percent specific binding, also set TOP to be a constant equal to 100.



2.7.17.6 Equation: Two site competition

$$Y = \text{Bottom} + (\text{Top} - \text{Bottom}) \left[\frac{\text{Fraction_1}}{1 + 10^{X - \text{LogEC50_1}}} + \frac{1 - \text{Fraction_1}}{1 + 10^{X - \text{LogEC50_2}}} \right]$$

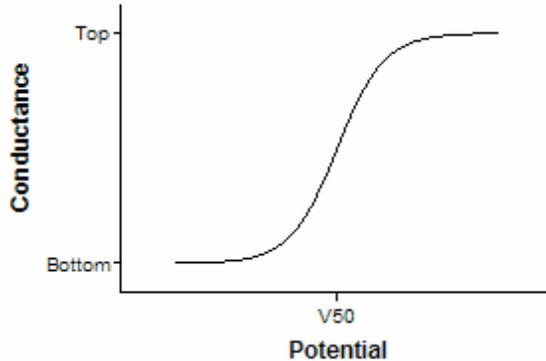


This equation describes the competition of a ligand for two types of receptors. The radioligand has identical affinities for both receptors, but the competitor has a different affinity for each.

Y is binding (total or specific) and X is the logarithm of the concentration of the unlabeled ligand. FRACTION_1 is the fraction of the receptors that have an affinity described by LogEC50_1. The remainder of the receptors have an affinity described by LogEC50_2. If LogEC50_1 is smaller than LogEC50_2, then Fraction_1 is the fraction of high affinity sites. If LogEC50_1 is larger than LogEC50_2, then Fraction_1 is the fraction of low affinity sites.

2.7.17.7 Equation: Boltzmann sigmoid

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



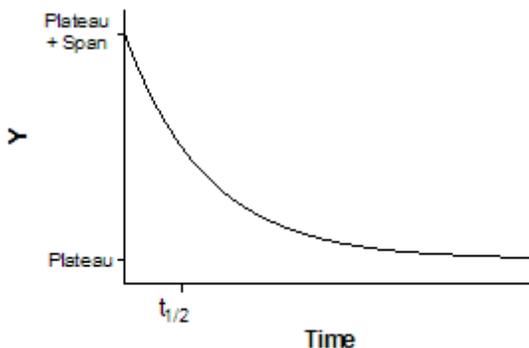
This equation describes voltage dependent activation of ion channels. It describes conductance (Y) as a function of the membrane potential (X). Conductance varies from BOTTOM to TOP. V50 is the potential at which conductance is halfway between BOTTOM and TOP. SLOPE describes the steepness of the curve, with a larger value denoting a shallow curve. Slope is expressed in units of potential, usually mV, and is positive for channels that activate upon depolarization.

Under appropriate experimental conditions, you can use SLOPE to calculate the valence (charge) of the ion moving across the channel. SLOPE equals RT/zF where R is the universal gas constant, T is temperature in °K, F is the Faraday constant, and z is the valence. Since $RT/F \approx -26$ mV at 25°C, $z = -26/\text{SLOPE}$.

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.

2.7.17.8 Equation: One phase exponential decay

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



This equation describes the kinetics such as the decay of radioactive isotopes, the elimination of drugs, and the dissociation of a ligand from a receptor.

X is time.

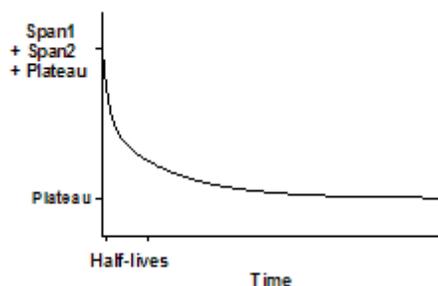
Y may be concentration, binding, or response. Y starts out equal to SPAN + PLATEAU and decreases to PLATEAU with a rate constant K.

The half-life of the decay is $0.6932/K$.

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

2.7.17.9 Equation: Two phase exponential decay

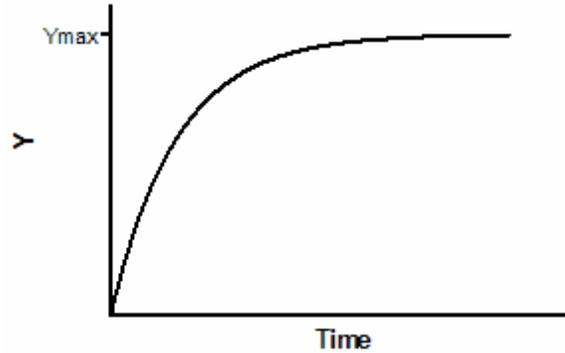
$$Y = \text{Span1} \cdot e^{-K_1 \cdot X} + \text{Span2} \cdot e^{-K_2 \cdot X} + \text{Plateau}$$



This equation describes a two phase exponential decay. Y starts out equal to $\text{Span1} + \text{Span2} + \text{PLATEAU}$ and decays to PLATEAU with fast and slow components. The two half-lives are $0.6932/K_1$ and $0.6932/K_2$. 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.

2.7.17.10 Equation: One phase exponential association

$$Y = Y_{\text{max}} \cdot (1 - e^{-K \cdot X})$$



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 $0.5 \cdot YMAX$.

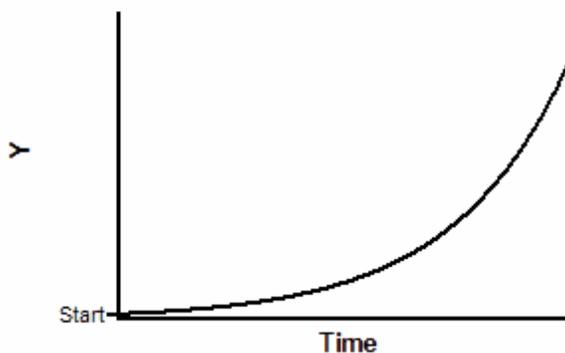
2.7.17.11 Equation: Two phase exponential association

$$Y = Y_{max_1} \cdot (1 - e^{-K_1 \cdot X}) + Y_{max_2} \cdot (1 - e^{-K_2 \cdot X})$$

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

2.7.17.12 Equation: Exponential growth

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



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.

2.7.17.13 Equation: Power series

This versatile equation has many uses.

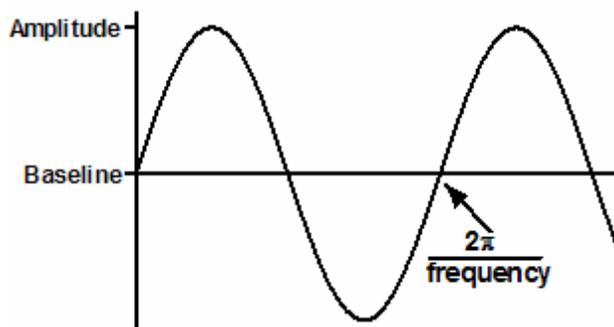
$$Y = A \cdot X^B + C \cdot X^D$$

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

The equation is not defined, and leads to a floating point error, if X equals zero and B or D are negative numbers or if X is negative and B or D are between 0.0 and 1.0.

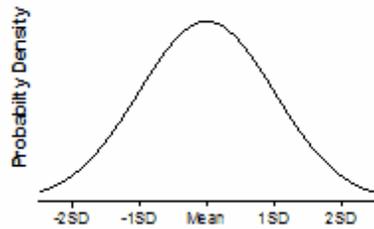
2.7.17.14 Equation: Sine wave

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



X is in radians. In most cases, you will want to fix BASELINE to a constant value of zero. AMPLITUDE is the maximum height of the curve away from the baseline. FREQUENCY is the number of complete oscillations per 1 X unit.

2.7.17.15 Equation: Gaussian distribution



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

2.8 Entering a user-defined model into Prism

Prism provides lots of built-in models, but you should not hesitate to adapt these or enter additional models. This section shows you how to do so.

- [Overview: User-defined equations](#) ⁴⁰⁶
- [How to: Enter a new equation](#) ⁴⁰⁷
- [How to: Clone an equation](#) ⁴⁰⁹
- [How to: Manage your list of equations](#) ⁴¹⁰
- [Syntax of user-defined equations](#) ⁴¹¹
- [Multiline models](#) ⁴¹³
- [Limitations when entering equations](#) ⁴¹⁴
- [Entering a differential equation](#) ⁴¹⁵
- [Entering an implicit equation](#) ⁴¹⁶
- [Available functions for user-defined equations](#) ⁴¹⁸
- [Fitting different segments of the data to different models](#) ⁴²¹

- [Fitting different models to different data sets](#)^[423]
- [Column constants](#)^[424]
- [Defining equation with two \(or more\) independent variables](#)^[426]
- [Rules for initial values](#)^[433]
- [Default constraints](#)^[437]
- [Transforms to report](#)^[438]

2.8.1 Overview: User-defined equations

Few people will need to enter user-defined equations into Prism. If you need to do this, keep these points in mind:

- Rather than starting with a blank screen, it is usually much easier to [clone an existing equation](#)^[409], and then modify it.
- The syntax Prism uses is pretty standard. But use plenty of parentheses, so there is no doubt about what the equation means. Many people have made the mistake of entering "A/B +C" expecting Prism to divide A by the sum of B plus C. Instead, Prism first divides A by B, and then adds that quotient to C. Writing the equation with parentheses as "A/(B+C)" avoids any ambiguity.
- Prism lets you use many [standard mathematical functions](#)^[418] when defining your model. Most of these are standard. But watch out for a few that are implemented differently in different languages. The log() function in Prism is common (base 10) log. If you want to use natural logs, use the ln() function. The sqr() function in Prism squares the value, while the sqrt() function computes the square root.
- Prism lets you write an equation in such a way that [different models are fit to different data sets](#)^[423].
- When you write a [model using several lines](#)^[413], Prism evaluates from the top down. This follows the conventions of all computer languages, but is opposite to how mathematical text books present complicated equations.
- After writing the equation, don't forget to [define rules for initial values](#)^[433]. If you don't do this, you'll need to enter initial estimated values for each parameter every time you fit the model to data.
- When you enter a new equation, Prism also lets you define [default constraints](#)^[437] and [transforms to report](#)^[438]. These features are optional, so you can skip them when first learning about user-defined equations.

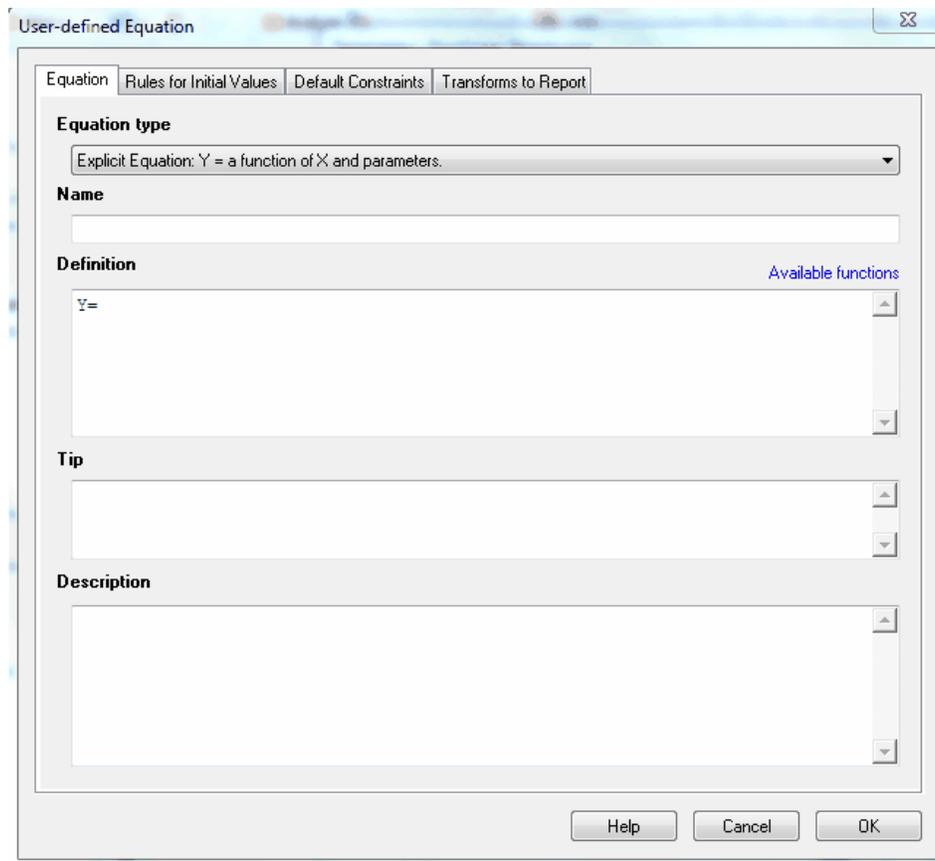
2.8.2 How to: Enter a new equation

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

Equation type

Choose whether you are entering an ordinary explicit equation (that defines Y as a function of X and parameters), an implicit equation (Y appears on both sides of the equals sign) or a differential equation (which defines the derivative of Y with respect to X).

Name

Enter a name you will recognize, so you can choose the equation again. The name also appears on the results table, so make it both descriptive and concise.

Definition

Write the math that defines Y as a function of X and one or more parameters. Read the details of equation [syntax](#)⁴¹¹, a list of available [math functions](#)⁴¹⁸ you can use, and how to fit different models to [different ranges of X values](#)⁴²¹ and to [different data sets](#)⁴²³.

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](#)⁴³³.

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](#)¹⁶² 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](#)⁴²⁴. 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](#)¹⁵⁸ 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](#)⁴³⁸ 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.

2.8.3 How to: Clone an equation

If none of the built-in equations, suit your needs, you may not need to enter a new equation. Instead, consider cloning an existing equation. You may also clone an equation if your only goal is to rename some of the parameters.

To clone an equation:

1. Click the New button on the top of the Fit tab of the Nonlinear regression dialog.



2. A drop down menu lets you choose to enter a new equation, clone an

existing one, or import an equation from a saved .PZF file. Choose to clone.

3. Edit the duplicated (cloned) copy. You can change the equation, the rules for initial values, constraints and transforms to report.
4. The default name of the cloned equation is the original name followed by a digit. Change to a more informative name.

The cloned equation is an exact duplicate of the equation you started with except for a digit appended to the equation name. Cloning does more than duplicate the math. It also includes the description, hint, rules for initial values, default constraints, and transforms to report.

2.8.4 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.

2.8.5 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. Don't make any assumptions about the order of precedence. Include enough parentheses so there is no ambiguity about how the equation is evaluated.
- Use a single equals sign to assign a value to a variable.

- You don't need any special punctuation at the end of a statement.
- To enter a long line, type a backslash (\) at the end of the first line, then press Return and continue. Prism treats the two lines as one.
- To enter a comment, type a semicolon (;) and then the text. Comments can begin anywhere on a line.
- You can use many [functions](#)⁴¹⁸, most which are similar to those built-in to Excel.
- Be careful not to use the name of a built-in function as a parameter name. For example, since *beta* is the name of a function, you cannot name a parameter *beta*.

IF-THEN relationships

Prism allows you to introduce some branching logic through use of the IF function. The syntax is:

IF (conditional expression, value if true, value if false).

You can precede a conditional expression with NOT, and can connect two conditional expressions with AND or OR. Examples of conditional expressions:

```
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](#)⁴²³.

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 $0*1+1*10=10$.

2.8.6 Multiline models

Equations can be written over several lines. Here is an example, the mixed model enzyme inhibition model built-in to Prism:

$$\begin{aligned} V_{\max App} &= V_{\max} / (1 + I / (\text{Alpha} * K_i)) \\ K_{m App} &= K_m * (1 + I / K_i) / (1 + I / (\text{Alpha} * K_i)) \\ Y &= V_{\max App} * X / (K_{m App} + X) \end{aligned}$$

Prism follows the convention of all computer languages. It starts at the top and goes down.

First it computes the intermediate variable $V_{\max App}$. It knows this is an intermediate variable, and not a parameter to fit, because it appears to the left of the equals sign.

Next Prism computes the value of $K_{m App}$.

Finally it uses those two values to compute Y.

Math textbooks tend to write equations in the opposite order. A math text might first define Y as a function of $V_{\max App}$ and $K_{m App}$, and then lower on the page define how to calculate $V_{\max App}$ and $K_{m App}$ from V_{\max} , K_m , Alpha and I . Prism (like all computer languages) requires that you define an intermediate variable before you use it.

Here is a second example:

$$\begin{aligned} \text{Specific} &= X * B_{\max} / (X + K_d) \\ \text{Nonspecific} &= NS * X \\ \text{<A>} Y &= \text{Specific} + \text{Nonspecific} \\ \text{} Y &= \text{Nonspecific} \end{aligned}$$

The first line calculates the intermediate variable Specific. The second line defines the intermediate variable Nonspecific.

The third line is preceded by <A>. This means that this line only applies to data set A. The second line is preceded by so only applies to data set B. This allows the model to fit a table of data where column A is the total binding and column B is the nonspecific binding. Read more about the syntax used to specify that a particular line [only applies only to selected data sets](#)⁴²³.

You can define constants right in the multiline equation. This makes sense for defining true constants, whose value will never change. If the constant is something like a concentration that changes from experiment to experiment, it is better to not define it in the equation itself, but rather define it in the [Constrain tab](#)¹⁵⁸. If it first appears on the right side of the equation, Prism will treat it like a parameter. You can use the Constrain tab to fix that parameter to a constant value. If the variable name appears first to the left of the equals sign, it is used only within the equation and won't appear in the constraints tab. For example, this line defines Pi:

```
Pi=3.141529
```

2.8.7 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**.

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.

[Ways to evade this limitation.](#)

Model complexity

Prism compiles your equation into an internal format it uses to calculate the math efficiently. If the compiled version of your equation won't fit in the space Prism sets aside for this purpose, it reports that the equation is "too complex" .

If you see this message, don't give up. You can usually rewrite an equation to make it less complex. Do this by defining an intermediate variable that defines combinations of variables. For example if your equation uses the term "K1+K2" four times, you reduce complexity (but keep exactly the same mathematical meaning) by defining an intermediate variable at the top of your equation (say, $K1P2=K1+K2$) and then using that intermediate later in the equation. That way Prism has fewer steps to store.

2.8.8 Entering a differential equation

An example. Exponential decay

A single phase exponential decay is defined by this equation.

$$Y=Y0*\exp(-K*X)$$

Simple calculus shows that this has the property that the derivative of Y is proportional to the value of Y.

$$dY/dX = -K*Y$$

How to fit a differential equation with Prism.

Prism can fit a model defined by a differential equation.

Choose the differential equation type at the top of the Equation dialog, and define Y' (the derivative of Y with respect to X) as a function of X and parameters. For the example, enter:

$$Y' = -K*Y$$

That's it. Putting an apostrophe after the Y on the left side of the equation tells Prism that you are defining the derivative of Y with respect to X.

Notes

- Prism does not understand the other nomenclature for differential equations. Don't try to define an equation that starts with "dY/dX = ".
- Note that X doesn't actually appear in the equation. That's ok. It is there in spirit, since Y' defines the derivative of Y with respect to X.

- When you look at that equation, there appears to be only one parameter, K. In fact, the equation has two parameters. Prism generates a parameter Y[0], which is the value of Y at X=0.
- When you go to add constraints and initial values, Y[0] appears just like the other parameter K.
- It is not possible to set this parameter to equal Y at some other X value (other than 0.0). Let us know if this limitation matters to you.
- Fitting a differential equation requires more calculations, so it takes noticeably more time than fitting the usual kind of equation.
- It is only possible to define Y'. It is not possible to use differential equations to define intermediate variables. This would be useful for fitting compartmental models, but Prism cannot (yet) fit this kind of model.

2.8.9 Entering an implicit equation

An implicit equation is one where Y appears on both sides of the equals sign. Prism 6 will (try to) fit models defined by implicit equations to your data.

Example of an implicit equation

The model for total binding at equilibrium to a binding site that follows the law of mass action is

$$Y = NS * X + B_{max} * X / (K_d + X)$$

That form of the model assumes that X is the free concentration of ligand. Since X is actually the concentration you entered, this model assumes that a tiny fraction of the ligand binds so X is

What if a substantial fraction of the ligand binds? A much more complicated model is built into Prism to handle this situation. But in other similar situations, the model may not have been derived and it may be very difficult or even impossible to derive one. Implicit equations to the rescue.

In the situation above, it is easy to write an implicit equation. There are three places in the model where X appears. The X values you enter are the total concentration of ligand. The value that enters into the model must be the free concentration, which is the total concentration minus the concentration that bound. Assuming X and Y are both in the same units, we simply replace X with

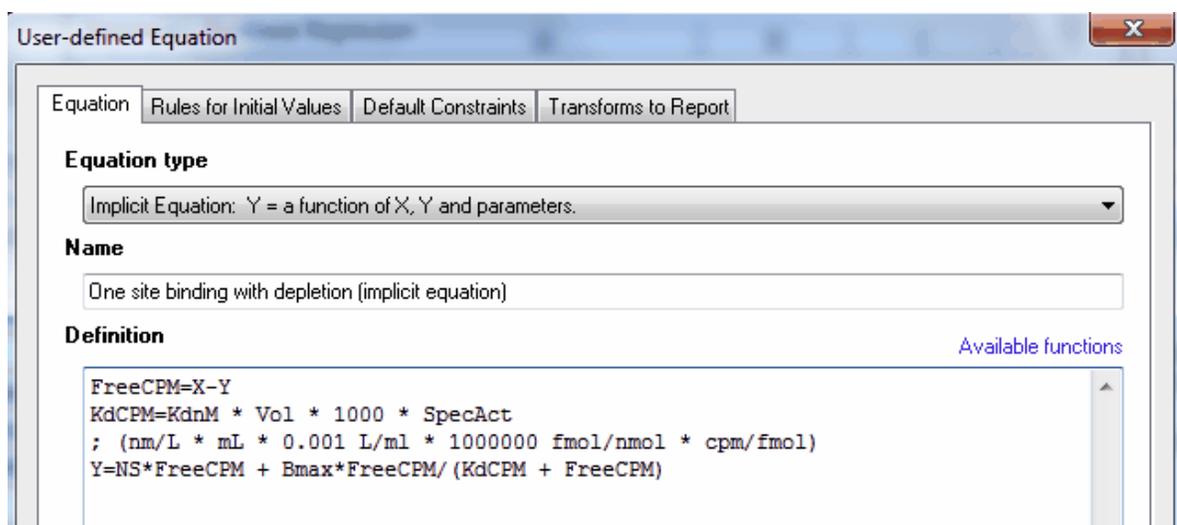
X-Y in all three places:

$$Y = NS * (X - Y) + B_{max} * (X - Y) / (K_d + (X - Y))$$

When you actually go to fit the data, you'll probably want to change the equation a bit (as shown below) to handle unit conversions so the K_d is reported in nM rather than cpm.

How to fit implicit equations with Prism

Choose the implicit equation type at the top of the Equation dialog, and define the equation with Y on both sides of the equals sign.



That's it. You'll need to define constraints and initial values as with any user-defined equation.

Notes

- In this example you don't actually see Y on both sides of the equals sign in the same equation line. But the first line puts Y on the right side of the equations sign, and the fourth line puts it on the right side. That makes the equation implicit.
- If you subtract Y from X, as in the example here, then X and Y must both be entered in the same units. Here both are entered as radioactivity counts per minute (cpm). Of course, it would make no sense to subtract Y from X if Y were in cpm and X were in nM.
- Prism finds it "harder" to fit implicit equations than ordinary ones. You may have to fuss with initial values and constraints to get it to work. The calculations take much longer, although this may not be noticeable with small data sets and fast computers.
- While Y appears on both sides of the equals sign in an explicit equation, X must appear only on the right side of the equals sign.

- In this particular example, the explicit equation has been derived and is even built into Prism. The file you can download (link below) fits the data both ways (explicit equation and implicit equation) and the results are identical. In other cases, it may be difficult or impossible to derive an explicit equation.

[Example Prism file.](#)

2.8.10 Available functions for user-defined equations

[Allowed syntax](#) 

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(gammain(j) +gammain(k) gammain(j+k))
binomial(k,n,p)	Binomial. Probability of k or more	1 -

Function	Explanation	Excel equivalent
	"successes" in n trials, when each trial has a probability p of "success".	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)
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(gammln(k))
gammln(k)	Natural log of gamma function.	gammln(k)
hypgeometricm(a,b,x)	Hypergeometric M.	(no equivalent)
hypgeometricu	Hypergeometric U.	(no equivalent)

Function	Explanation	Excel equivalent
a, b, x)		
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)
$j \bmod k$	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)
sgn(k)	Sign of k . If $k > 0$, sgn(k)=1. If $k < 0$, sgn(k)= -1.	sign(k)

Function	Explanation	Excel equivalent
	If $k=0$, $\text{sgn}(k)=0$.	
$\sin(k)$	Sine. K is in radians.	$\sin(k)$
$\sinh(k)$	Hyperbolic sine. K is in radians.	$\sinh(k)$
$\text{sqr}(k)$	Square.	$k*k$
$\text{sqrt}(k)$	Square root.	$\text{sqrt}(k)$
$\tan(k)$	Tangent. K is in radians.	$\tan(k)$
$\tanh(k)$	Hyperbolic tangent. K is n radians.	$\tanh(k)$
$\text{tdist}(t,v)$	P value (one-tailed) corresponding to specified value of t with v degrees of freedom. T distribution.	$\text{tdist}(t,v,1)$
$\text{zdist}(z)$	P value (one-tailed) corresponding to specified value of z. Gaussian distribution.	$\text{normsdist}(z)$

2.8.11 Fitting different segments of the data to different models

What is segmental regression?

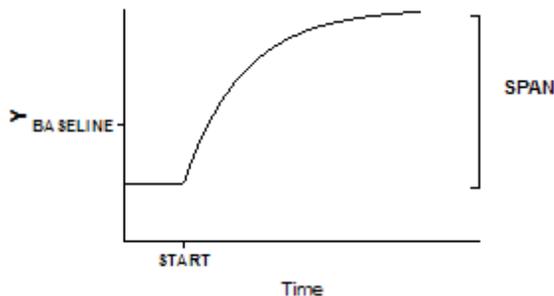
In some situations you may wish to fit different models to different portions of your data. This is called **segmental regression**. This often occurs in kinetic experiments where you add a drug or perform some sort of intervention while recording data. The values collected before the intervention follow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

If you didn't perform an intervention at a particular time along the X axis, it is unlikely that this kind of segmental regression is the analysis of choice.

How to perform segmental regression with Prism

Although Prism has no built-in way to fit different equations to different portions of the data, you can achieve that effect using a user-defined equation containing the IF function.

In this example, you collected data that established a baseline early in the experiment, up to "Start". You then added a drug, and followed the outcome (Y) as it increased towards a plateau. Prior to the injection, the data followed a horizontal line; after the injection the data formed an exponential association curve.



```

Y1=BASELINE
Y2=BASELINE + SPAN*(1-exp(-K*(X-START)))
Y=IF[(X<START), Y1, Y2]

```

It is easiest to understand this equation by reading the bottom line first. For X values less than START, Y equals Y1, which is the baseline. Otherwise, Y equals Y2, which is defined by the exponential association equation.

This equation has two intermediate variables (Y1 and Y2). Prism can fit the four true variables: START, SPAN, K, and BASELINE.

In many cases, you will make START a constant equal to the time of the experimental intervention. If you want Prism to fit START, choose an initial value carefully.

This kind of model is most appropriate when X is time, and something happens at a particular time point to change the model. In the example above, a drug was injected at time=Start.

How to fit a model to only a portion of the data

Ignore points outside of specified X range

Don't fit points when X is less than

Don't fit points when X is greater than

The Range tab of the nonlinear regression dialog lets you define an X range that determines which points are fit and which are ignored.

2.8.12 Fitting different models to different data sets

Writing a model so some lines only apply to selected data sets

Prism can only fit data from one table at a time, and all the data sets must be fit by a single equation you select or create. But you can create this equation in such a way that different data sets are fit to mathematically distinct equations.

When entering a user defined equation, you use multiple lines to define intermediate variables. Each line can have a prefix that specifies which data sets that line will be used for:

Prefix	This line applies to
<C>	Data set C only
<~B>	All data sets, except B.
<A:D>	Data sets A-D.
<~A:D>	All data sets except A-D.
<A:J,3>	Data sets A,D,G, and J (every third data set between A and J.)
<~A:J,3>	All data sets except A,D,G, and J.

Here is an example. It fits column A to a model that defines total binding and column B to a model that defines nonspecific binding only. The first two lines of the equation are evaluated for all data sets, the third line is only evaluated for data set A, while the last line is only evaluated for data set B. To fit this model, you would want to set the constraint that the parameter NS is shared between data sets.

```
Specific=X*Bmax/(X+Kd)
Nonspecific=NS*X
<A>Y=Specific + Nonspecific
<B>Y=Nonspecific
```

Ambiguity when you select noncontinuous range of data sets to analyze

```
<A>Y=1/(1+Ka*X^h)
<C>Y=(Ka*X^h)/(1+Ka*X^h)
```

The equation above seems clear at first. The first line is for data set A, and the second line is one for data set C. But what if you asked Prism to only analyze data sets A and C, skipping B? Now it is a bit ambiguous. Does <C> mean data set C, or does it mean the third data set included in the analysis? Prism uses the second definition. So you need to enter the equation like this.

$$\langle A \rangle Y = 1 / (1 + K_a * X^h)$$

$$\langle B \rangle Y = (K_a * X^h) / (1 + K_a * X^h)$$

Now the second line is for the second data set, which in this example is data set C (since only A and C were selected in the Analyze dialog).

2.8.13 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 μ M	15 μ M	50 μ M
		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.

$$\begin{aligned} \text{KmObs} &= \text{Km} (1 + [I] / \text{Ki}) \\ Y &= \text{Vmax} * X / (\text{KmObs} + X) \end{aligned}$$

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](#)³⁴⁵.

Summary. The advantage of column constants

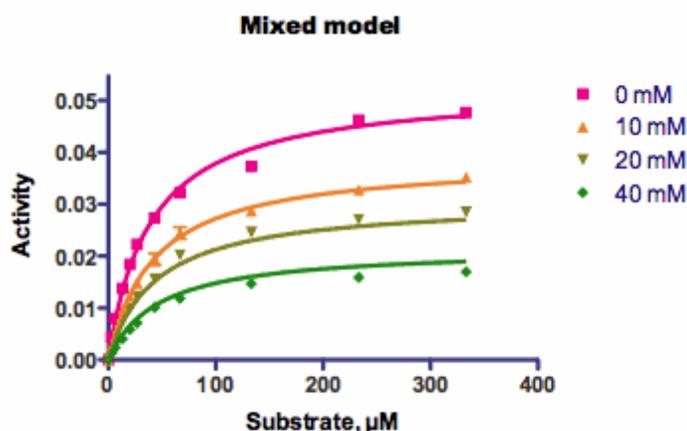
By using column constants and global fitting (shared parameters), this example determined a parameter (Ki) whose value cannot be determined from any one dataset, but can only be determined by examining the relationships between

datasets.

2.8.14 Defining equation with two (or more) independent variables

Prism is designed to perform nonlinear regression with one independent (X) variable. But, with a bit of cleverness, it is possible to also fit data with two independent variables. There are three separate cases:

Fitting a family of curves

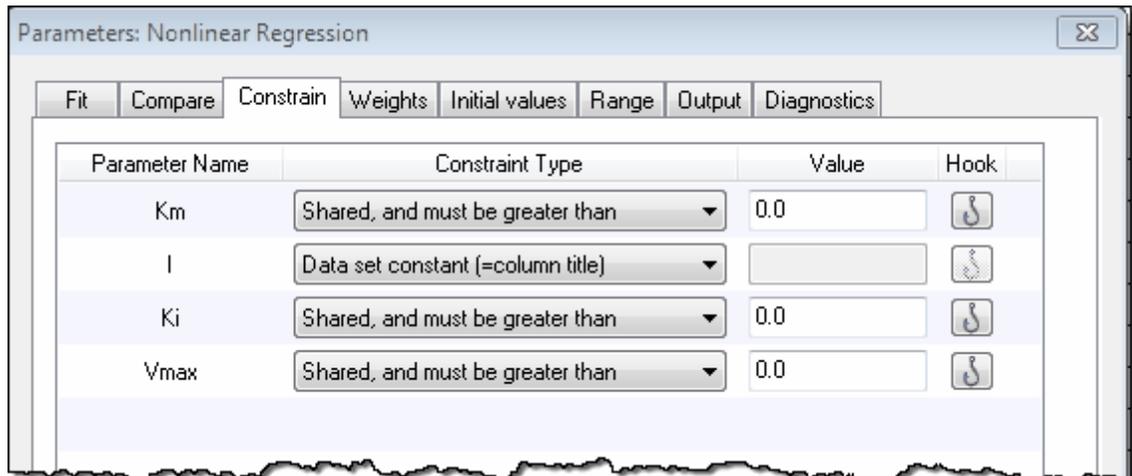


The graph above shows the sample data for competitive enzyme kinetics, and shows how to fit a family of curves. Here, each curve shows enzyme activity as a function of substrate concentration. The curves differ by the presence of various amounts of an inhibitor. There are, essentially, two independent variables: substrate concentration, and inhibitor concentration.

The substrate concentration is entered into the X column and the inhibitor concentration into the column titles.

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

When fitting the data, the Constrain tab of the nonlinear regression dialog is used to define the parameter I as dataset constant whose value comes from the column titles (so is not fit by regression), and to share the values of all the other parameters so there is one global fit of all the data.



Fitting data where each value has two independent variables

But what if each value you collect has two independent variables? You don't have a family of curves. Rather you have one set of Y values, each associated with two X variables. Follow these steps to enter the data:

1. Create an XY data table, with whatever form of subcolumns fits your data.
2. Enter the first Y value in row 1 of column A.
3. Enter the first independent variable corresponding to that Y into first row of the X column.
4. Enter the other independent variable for that first Y value as the column title of column A.
5. Enter the second Y value in row 2 of column B.
6. Enter the first independent variable for that Y into the second row of the X column.
7. Enter the other independent variable for that Y as the column title of column B.

8. Continue with the rest of the data. You'll be entering data diagonally down the table. Since Prism can only have 104 columns, you will be limited to 104 data points, each with a Y value and two X values.

9. When you are done, you'll have the same number of rows as Y columns.

X	A	B	C	D	E	F	G
X Title	0.000556	0.000542	0.000529	0.000505	0.000483	0.000462	0.000444
X	Y	Y	Y	Y	Y	Y	Y
0.000000	0.000000						
0.117000		0.015103					
0.229000			0.022470				
0.436000				0.035167			
0.626000					0.041316		
0.799000						0.046776	
0.958000							0.046327

When you fit the data, keep in mind the following:

- "X" refers to the values you enter into the X column. Use some other name ("X2" or whatever you want) for the other independent variable.
- In the constraints tab, define that parameter to be a data set constant equal to the column title. You don't want Prism to fit that parameter, but rather to get its value from the column titles.
- For all other parameters in the model, use the Constraint tab to share the value among data sets. Prism sees each column as a data set. In this data table, each data set has only one value so Prism can't possibly fit anything unless all the parameters are shared.

Prism will report the nonlinear regression results clearly. All the results you want will be in the final "global" column. Prism cannot graph the results since each data set has only one point, and Prism cannot plot 3D graphs.

Three independent variables

Prism is not really designed to fit models with three independent variables, but you can get it to do so in some circumstances.

Enter your data as above, with one independent variable as X and the second as column titles. Make sure each column of data has only one value for the third independent variable. Then define that third variable in a user-defined

equation. Let's imagine that the third variable has the name Z. Add syntax like this to your equation:

<A> Z=2.34

 Z=45.34

<C>Z= -23.4

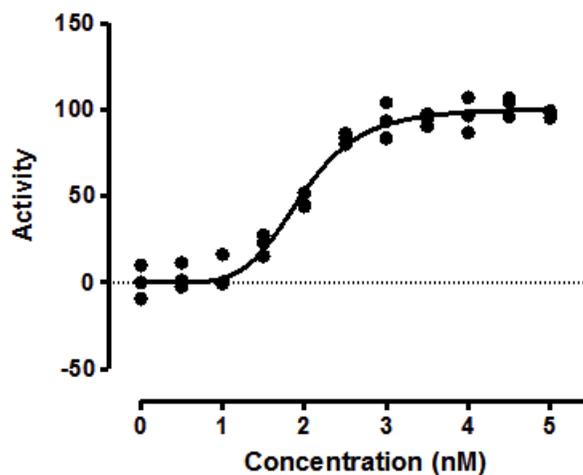
<D>Z=12.45

That tells Prism to assign Z one value for column A, a different value for column B, etc. This approach gives you three independent variables, with some constraints:

- The X column is one independent variable, with one value for each row.
- The column titles form a second independent variable with one value per data set column.
- The Z values defined within the equation for a third independent variable, also with one value per data set.

2.8.15 Reparameterizing an equation

Reparameterizing an equation does not change the best-fit curve



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

You want to fit the sigmoidal enzyme kinetics data to a standard model. But there are two forms of that model that are commonly used:

$$Y = V_{\max} \cdot X^h / (K_{\text{half}}^h + X^h)$$

$$Y = V_{\max} \cdot X^h / (K_{\text{prime}} + X^h)$$

The two are equivalent, with K_{prime} equal to K_{half}^h , so the two fits will generate exactly the same curve, with the same sum-of-squares, the same R^2 , and the same numbers of degrees of freedom. Even though the two equations express the same model, they are written differently. The fancy term is that they are *parameterized* differently.

They both fit V_{\max} (the maximum activity extrapolated to very high concentrations of substrate) and h (Hill slope, describing the steepness of the curve). But one model fits K_{half} (the concentration needed to obtain a velocity half of maximal) and the other fits K_{prime} (a more abstract measure of substrate action).

Which model is preferred?

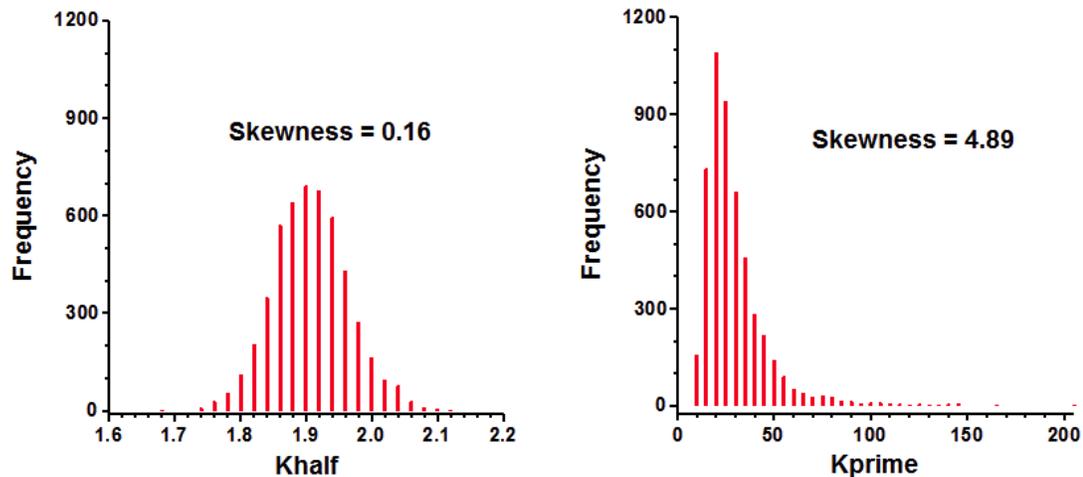
One way to choose between the two models is to match other text books and papers, so your results can easily be compared to others. Another approach is to choose the form that fits the way you think. For this example, if you prefer to think graphically, choose the K_{half} . If you think mechanistically, choose K_{prime} .

But the choice is more than a matter of convenience and convention. The choice of model determines the accuracy of the confidence intervals. Read on to see why.

Distribution of parameters are not always symmetrical

GraphPad Prism, like almost all nonlinear regression programs, computes the confidence interval of a best-fit parameters so the interval is symmetrical around the best fit values. If the uncertainty of a parameter is truly symmetrical, then these confidence intervals can be interpreted at face value. If the uncertainty is not symmetrical, then the confidence interval will not be accurate.

[Simulations](#) can determine parameter symmetry. I simulated sigmoidal enzyme kinetics using $V_{\max}=100$, $h=5$, $K_{\text{prime}}=25$, and Gaussian scatter with a SD equal to 7.5. The X values matched those in the figure above, with triplicate Y values at each X. Prism can do such simulations easily. Using Prism's [Monte Carlo analysis](#), I repeated the simulations 5000 times, fit each curve to both forms of the models, and tabulated the best-fit values of K_{prime} and K_{half} , and computed the skewness of each.



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

Coverage of the confidence intervals

Ideally, a confidence interval is easy to interpret. A 95% CI has a 95% chance of including the true value of the parameter, and a 5% chance of missing it. When analyzing real data, we never know the value of the true parameter, so never know if the interval includes it or not. But when you simulate data, you know the true values of the parameters, so can quantify the coverage of the confidence intervals. I set up the same simulations mentioned above, fit each data set to both equations, and tabulated whether each confidence interval included the true parameter value or not. This table shows the fraction of 5,000 simulations where the confidence interval did not include the true parameter value (25 for Kprime, and 1.9037 for Khalf).

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

These results show that Khalf is well behaved, as expected given its symmetry

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

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

Hougaard's skewness

The results above were obtained by running numerous simulations. There is an easier way to figure out how symmetrical a parameter is. [Hougaards skewness](#) ^[204] (new in Prism 6) quantifies the asymmetry of each parameter, computed from the equation, the number of data points, the spacing of the X values, and the values of the parameters.

For the simulated data set, Hougaard's skewness is 0.09 for Khalf and 1.83 for Kprime. A rule of thumb is to expect problems from asymmetry when the absolute value of the Hougaard's skewness is greater than 0.25, and big problems when the value is greater than 1.0. So Hougaard's skewness tells you that the confidence intervals will be accurate when you fit Khalf, but not be so accurate when you fit Kprime.

Note that Hougaard's skewness can be reported as part of the results of nonlinear regression (choose in the Diagnostics tab). No simulations are required.

Bottom line

Models can often be parameterized in multiple ways. You'll get the same curve either way, but choosing an optimum parameterization ensures that the confidence intervals for the parameters are believable. The best way to assess various parameterizations is to ask Prism to report the value of [Hougaards](#) ^[204] measure of skewness for every parameter. Simulations take a bit more work, but let you see how symmetrical a parameter is.

[File used for this example](#)

2.8.16 Rules for initial values

Why initial values?

Before it can perform nonlinear regression, Prism must have initial (estimated) values for each parameter in the equation. It then iteratively modifies these values to improve the fit of the model to the data. The initial values must be reasonable, so that the curve defined by those initial values has generally the shape of the data, and doesn't go awfully far from the data points. If the initial values are really bad, Prism may not be able to fit the model.

The need for initial values is not unique to Prism, but is intrinsic to nonlinear regression.

Why rules for initial values?

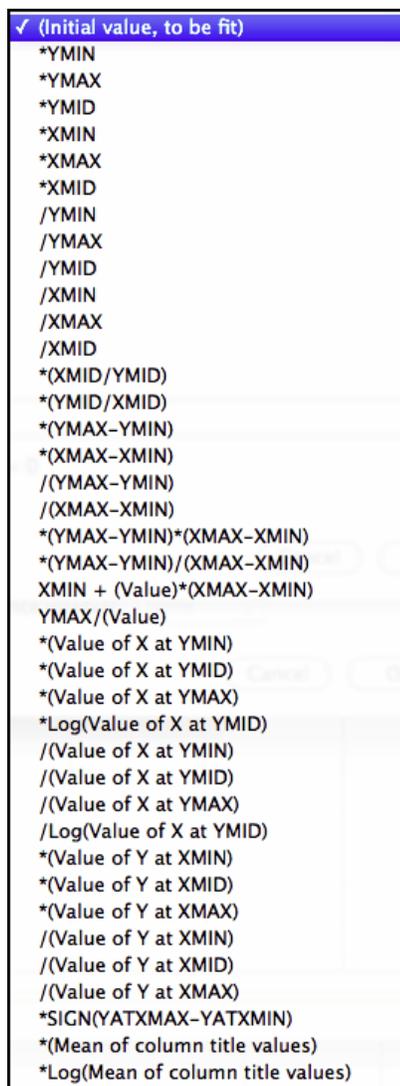
Prism can store rules (algorithms) to calculate initial values from the range of your data. If you choose a built-in equation, the rules are taken care of. If you enter your own (user-defined) equation, you can also enter rules to define the initial values of each parameter. If you enter a good set of rules, then your equation will work in the future with no need to think about initial values.

If you don't enter rules for initial values, you will need to enter the initial values for every variable, for every data set, every time you fit data.

If you enter a poor set of rules, then the initial values will not generate a curve that goes near the data points, and it is likely that Prism will not be able to fit the model to the data.

How to enter a rule

While entering or editing a user-defined equation, click on the tab labeled "Rules for initial values". For each parameter, enter a number in the first column and select a multiplier from the drop-down list in the second column.



Initial value, to be fit

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

Initial value based on the range of the data

Most of the rules are based on the range of the X and Y values.

The abbreviation YMIN is the minimum value of Y; YMAX is the maximum value, and YMID is the average of YMIN and YMAX. For example, if you enter "0.5" in the first column and select "YMAX" in the second column, Prism sets the initial

value to half of YMAX (which differs for each data set).

The phrase "X at YMID" refers to the X value that corresponds to the mean of the highest and lowest Y values. [Calculation details.](#)

SIGN(YatXmax - YatXmin)

It equals +1 if the curve generally goes up as it goes from left to right: /

It equals -1 if the curve generally goes down as it goes from left to right: \

It is used as the initial value for the Hill Slope in dose response curves.

Here is an explanation of the math. The SIGN() function equals +1 when given a positive number, and -1 when given a negative number. YatXMax is the Y value at the largest X values. YatXmin is the Y value at the smallest X value. (YatXmax - YatXmin) will be positive when the curve goes up, and negative when the curve goes down.

Mean of column title values (or the log of that mean)

With some kinds of data, it makes sense to label the data sets numerically. Each column titles has a value.

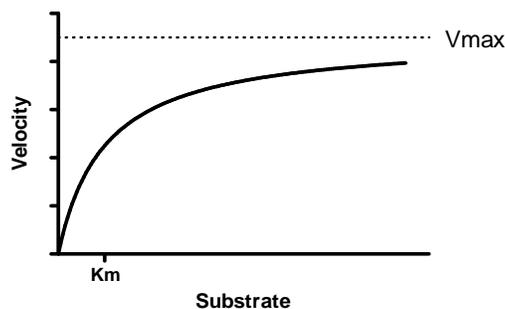
With these kinds of data, you can access the column title values in two ways:

- Use the Constrain tab and choose "Data set constant (=column title)". That parameter in the model is now a constant, but the value of that constant will differ for each data set column, and gets it value from the column title.
- On the Rules for initial values tab in the dialog that defines an equation, set the initial value of a parameter to equal the mean of those column titles. Or to the logarithm of that mean.

The best way to understand this is with an example. Bring up the sample XY data for [competitive enzyme inhibition](#)^[345], and follow the instructions, and inspect the dialog settings.

Example

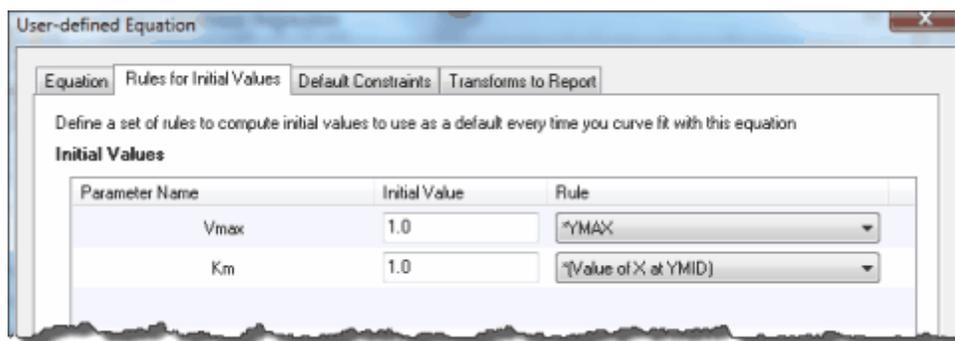
$$Y = V_{max} * X / (K_m + X)$$



The equation above is the Michaelis-Menten enzyme kinetics model built in to Prism. If you pick this equation as a built-in equation, the rules for initial values are provided automatically. But if you were entering this as a new user-defined equation, it would be easy to come up with rules.

Vmax is the Y value (enzyme velocity) at very high values of X (substrate). So a reasonable initial value would be to set Vmax equal to the largest Y value in the data set. To do this, enter the value 1.0 and choose the multiplier *Ymax. This means Prism will compute the initial value by multiplying the largest Y value by 1.0. You could argue that Vmax is usually higher than the largest Y value, so should compute the initial value by multiplying 1.25 times Ymax. There are often multiple rules one could justify. These rules only compute the initial value, so there is a lot of room for variation.

Km is the X value (substrate concentration) corresponding to a Y value (enzyme velocity) that is half Vmax. Compute its initial value by multiplying 1.0 times "Value of X at Ymid". Prism then averages the largest and smallest Y values to compute Ymid, and then interpolates from straight connecting lines to find the corresponding X value.



What happens if you don't enter any rules?

If you don't enter rules for every parameter, Prism will insist that you enter initial values each time you fit data to the equation.

How to override the initial values defined by rule

The rules are stored with each equation, and the rules can be edited as part of the dialog that defines and edits equations.

The values defined by the rules are shown in the Initial Value tab of the nonlinear regression dialog. To override the initial values:

1. Select one or several data sets on top of the dialog. Or click "Select all" to select all data sets.
2. Uncheck the "Choose automatically" option.
3. Enter the initial value (as a value, a number, not a rule) .

To check how good the initial values are, go to the Diagnostics tab and check the option:

(*) Don't fit the curve. Instead plot the curve defined by the initial values of the parameters.

If the resulting curve doesn't go near the data, revise the initial values (or the rules that generate those values) before asking Prism to fit the model to the data.

2.8.17 Default constraints

What is a constraint?

Constraining parameters is often essential to getting useful results. Constraints can be used to fix parameters to [constant](#)^[158] values, to share parameters among data sets ([global](#)^[40] fitting), and to define one parameter to be a [column constant](#)^[424] (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.

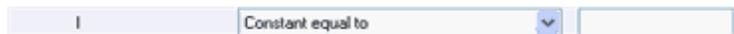
Constraints can be for one parameter (Kfast must be greater than zero) or for the relationship between two parameters (Kfast must be greater than Kslow). But note that you cannot invoke a constraint between two parameters if both parameters also are themselves constrained. In a two phase exponential 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 ($K_{fast} > K_{slow}$). Prism won't let you do that. What you have to do is define one constraint that K_{fast} is greater than zero, and another that K_{fast} is greater than K_{slow} . But don't put in the constraint that K_{slow} is greater than zero. That is implied by being larger than K_{fast} .

Defining experimental constants

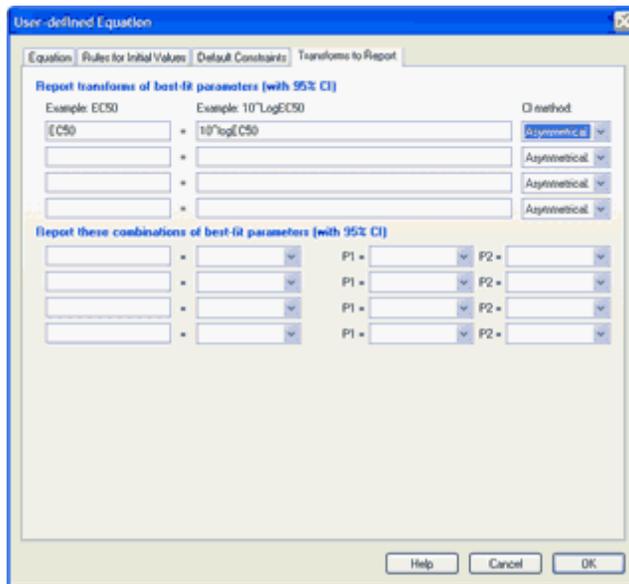
In some cases, one of the parameters in the equation is an experimental constant. It needs to be set to a constant value, but that value differs each time you run the experiment. Do that by going to the constraints tab of the nonlinear regression dialog every time that equation is used.

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).



2.8.18 Reporting transforms of parameters

When defining an equation, you can also ask Prism to report transforms of the best-fit values on the "Transforms to report" tab.



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 $\log EC_{50}$ that is the logarithm of an EC_{50} , but you want to also report the EC_{50} . Enter the label 'EC50' on the left and ' $10^{\log EC_{50}}$ ' 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 $a \cdot K + b$

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.

2.9 Plotting a function

Graphing a family of theoretical curves, often called plotting a function, is useful in three contexts:

- To understand the properties of a function. After looking at the graph, go back and change the parameters to see how the graph changes. Or plot a family of curves, where one parameter varies from curve to curve.
- To create graphs for teaching theory.
- To understand what initial values would make sense when fitting a model to data with nonlinear regression.

2.9.1 How to: Plot a function

Prism offers a *Plot a function* analysis, but it doesn't in fact analyze any data. Rather it generates curves from an equation you choose and parameters you enter.

How to: Plot a function

1. Start from any data table or graph, click Analyze, open the **Generate Curve** folder, and then select **Plot a function**.
2. On the first tab (Function), choose the equation, the starting and ending values of X, and the number of curves you want to plot.
3. On the second tab (Options), choose whether you also want to plot the first derivative, second derivative or integral of the function. The "curve" is actually a set of X and Y coordinates that define a series of points that are connected to form the curve. You can choose the number of line segments that will define the curve. There is little reason to change the default (150), unless you want to plot only a portion of the curve on some graphs, in which case you should increase that value.
4. On the third tab (Parameter values), enter the parameter values (or click the fish hook icon to hook analysis or info constants).

Tips for plotting a function

Plotting a family of curves

If you choose to plot more than one curve (a choice on the first tab), the rest of the dialog works a bit differently.

On the bottom of the second (Options) tab, choose if you want to label each curve manually (enter the labels the top of the third tab) or using the value of one of the parameters.

The top of the third tab lists all the curves you will generate. Select one or more of these curves (or click "select all") and then enter the parameter values below. Often you'll want to first click "select all" and enter most of the parameters. Then click on one curve at a time, and enter the value for the parameter that varies between curves.

Plotting each curve on a separate graph

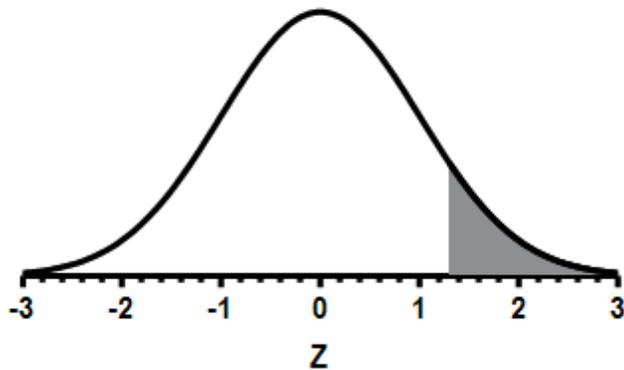
By default, Prism will make graph that contains all of the curves on one graph. If you want each curve to be on its own graph, go to the results table, click New and choose Graph of existing data. On the dialog that pops up, choose to make one graph for each data set (in this context, a data set is a curve).

If you plan to zoom in and plot only part of the curve

The curve is defined, by default, as 150 line segments. That creates a smooth looking curve. But if you then change the range of X values shown on the graph, only a fraction of those line segments will be visible, and the curve may seem coarse. To fix this problem, go back to the parameters dialog, to the Options tab, and increase the number of line segments to a much larger value.

Combining two curves on one graph

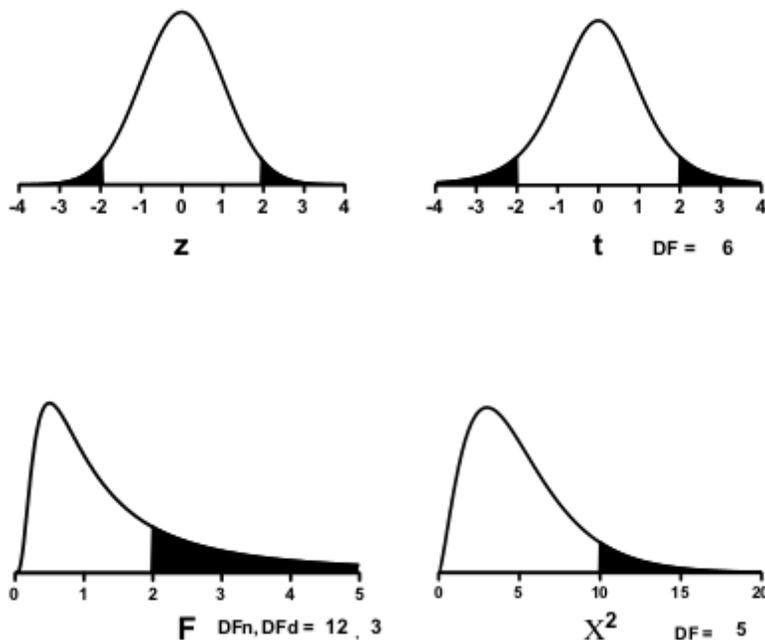
The graph below combines two plotted functions on one graph. The first time I plotted a function, I chose a Gaussian distribution, with the X ranging from -3 to 3. I set the mean to 0.0, the SD to 1.0, and the amplitude to 100.0 (arbitrary, since I hid that axis). I then repeated that analysis, but this time set the X range from 1.3 to 3.0. I put both curves on one graph (Change.. Add data sets -- remember that a curve generated by this analysis is a "data set" to Prism). For the shorter curve, I chose to create an area fill.



2.9.2 Plotting t, z, F or chi-square distributions

GraphPad Prism can generate probability distributions. This demonstrates Prism's ability to plot functions from user-defined functions, and also the use of hooking info constants to analyses.

[Download this Prism 5 file](#) to generate and plot the graphs shown below.



In each case, the simulation generates two (or three) data sets. The first (A) data set plots the entire curve. The equation is written so the second curve (data set B) only plots values where X is less than a specified cutoff value, and the third curve (data set C) only plots values when X is greater than the cutoff value. The second and third data set are plotted with area fill to shade the tails of the distributions. Remove data set B or C from the graph if you only want to shade one tail. For example, here is the equation used for the first graph (z distribution):

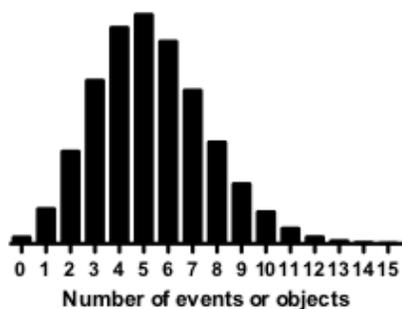
```
G=exp(-0.5*X^2)/sqrt(2*3.1415926)
<A>Y=G
<B>Y=IF(X<-z, G, 0)
<C>Y=IF(X>z, G, 0)
```

Change the numbers of degrees of freedom and the cutoff values (for shading) in the Info sheet. This demonstrates how values entered into an info sheet can be 'hooked' to constants used in analyses.

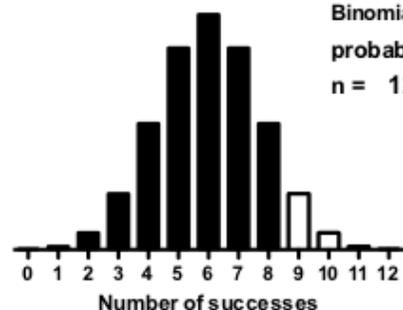
2.9.3 Plotting a binomial or Poisson distribution

How to plot a binomial or Poisson distribution

Poisson. True average number (lamda) = 5.3



Binomial
probability = 0.5
n = 12



[Download the Prism file.](#)

To modify this file, change the value of lamda (for Poisson) or the probability, n, and cutoff (Binomial) in the Info sheet. Enter new values there, and the graph updates. This is a good example of the usefulness of hooking an info constant to an analysis.

If you want to recreate graphs like these, keep in mind these points:

- As its name suggests, the analysis 'Create a Family of Theoretical Curves' is usually used to create curves, not bar graphs. When you choose the range of

X values, specify the appropriate number of 'line segments' (points) so that the X interval equals 1.0. The binomial example on the left created 16 'line segments' starting at X=0 and ending at X=15. The Poisson on the right created 13 'segments' with X starting at 0 and ending at 13.

- The analysis will create a set of line segments (an attempt to create a curve). Click the change type of graph button, or drop the Change menu and choose Graph Type. Then choose the Grouped tab, and then choose interleaved bars.
- The binomial example on the right has two data sets. You don't want them plotted interleaved, as selected in the previous step. Double click to bring up Format Graph, then go to the middle tab, and choose to superimpose the second data set on the first (rather than interleave). And assign it a different color.

Mathematical details

Binomial distribution

The equation for the probability of exactly X successes in N trials, when each trial has probability P of success is:

```
R=INT(X+0.5)
ExactProb=(P^R)*(1-P)^(N-R) ;exact probability of successes in n trials
NRrearrangments=exp(gammaln(N+1) - gammaln(R+1) - gammaln(N - R +1))
;gamma(J)=(J-1)factorial, or (J-1)! but factorial is not a function within
;NRrearrangments = N!/(R! (N-R)!)
H=ExactProb * NRrearrangments
<A>Y=H
<B>Y=IF(X>cutoff, H, 0)
```

Poisson distribution

The [equation for the Poisson distribution](#) is:

$$Y = \exp(-1 * \text{Lamda}) * \text{Lamda}^X / \text{gamma}(X+1)$$

Note the [definition of the gamma function](#):

```
gamma(i) = factorial(i-1)
gamma(x+1) = factorial(x) = X!
```

2.10 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](#)⁴⁴⁷
- [Using nonlinear regression with an empirical model](#)⁴⁴⁸

2.10.1 Spline and Lowess curves

Curve fitting without a model

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

In some circumstances, you just want to draw a smooth curve to make a graph look attractive, or to use as a standard curve. You don't care about models, and aren't looking for best-fit values that you can interpret.

Prism provides two approaches for fitting a curve without selecting a model. From a table or graph of XY data, click Analyze, and then choose "Fit spline/LOWESS" from the list of XY analyses.

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. Prism always generates a lowess curve with at least four times more line segments than the number of data points, and you cannot reduce the number of segments below this value.

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 if you wish to calculate area under that curve

References

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

2.10.2 Using nonlinear regression with an empirical model

If your goal is just to plot a smooth curve, without worrying about a model, you have several choices.

[Splines](#)^[447] created by Prism go through every point, so may wiggle too much.

[Lowess](#)^[447] 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](#)^[381], 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.

Index

- 4 -

4PL 256, 262

- 5 -

5PL 267

- A -

Abolute IC50 248

Adjusted R2 216

Advice: How to understand a model 12

AIC 48, 51

AIC. Interpreting. 215

Akaike information criterion 48, 51

Allosteric EC50 shift equation 281

Allosteric modulator defined 311

Allosteric modulator of radioligand binding, equation 322

Ambiguous fits defined 197

Ambiguous nonlinear regression 228

Analysis checklist: Deming regression 103

Analysis checklist: Linear regression 91

Analysis checklist: Nonlinear regression, "Ambiguous" 228

Analysis checklist: Nonlinear regression, "Bad initial values" 226

Analysis checklist: Nonlinear regression, "Don't fit" 232

Analysis checklist: Nonlinear regression, "Hit constraint" 231

Analysis checklist: Nonlinear regression, "Impossible weights" 233

Analysis checklist: Nonlinear regression, "Interrupted" 226

Analysis checklist: Nonlinear regression, "Not converged" 227

Analysis checklist: Nonlinear regression, "Perfect fit" 233

Analysis checklist: Nonlinear regression, "Too few points" 232

Analysis checklist: Nonlinear regression, comparing models 221

Analysis checklist: Nonlinear regression, general 218

Analysis checklist: Nonlinear regression, interpolating 118, 223

Analysis of Covariance 89

ANCOVA 89

Angled brackets in writing equations 423

Anscombe's quartet 18

Association binding experiments 325

Association kinetics (one ligand concentration) equation 328

Association kinetics (two ligand concentrations) equation 329

Association then dissociation, equation 331

Asymmetrical (five parameter) dose-response equation 267

Asymmetrical confidence intervals, nonlinear regression 438

- B -

Backfit 104

Bell-shaped dose-response equation 271

Biphasic dose-response equation 269

Boltzmann sigmoid curve 400

Broken line model 374

- C -

Centered polynomial equations 381

Cheng and Prusoff 312, 315

Cheng-Prusoff 164

Cloning an equation 409

Colorimetric chemical assays 9

Column constants 424

Compare tab, nonlinear regression 157

Comparing slopes and intercepts: 89

Competitive binding, defined 311

Competitive binding, one site equation 312, 314

Competitive binding, one site, homologous 321

Competitive binding, one site, with ligand depletion 318

Competitive binding, two sites, equation 315, 317

Competitive bindings, equation for kinetics 333

Competitive inhibition of enzyme 345

Complexity, of equations, defined 414

Confidence bands 165

Confidence bands, computation of 67

Confidence bands, linear regression 92
 Confidence interval of parameters of nonlinear regression 183
 Confidence intervals 165
 Constrain tab, nonlinear regression 158
 Constraints in nonlinear regression, default 437
 Convergence criteria 165
 Covariance matrix 197
 Covariance, analysis of 89
 Cox regression 21
 Cubic spline 447
 Cumulative Gaussian axes, fitting straight lines on 379
 Cumulative Gaussian distribution, equation 389

- D -

D'Agostino-Pearson normality test of residuals. 185
 Damped sine wave 394
 Data set constant 158, 424
 Deleting equations 410
 Deming regression 100
 Dependency 197
 Depletion of radioligands 289
 Diagnostics tab, nonlinear regression 165
 Differential equations 414, 415
 Dissociation constant, ambiguity 285
 Dissociation kinetics equation 326
 Don't fit. Nonlinear regression 232
 Dose-response curves, defined 235
 Dose-response, Five parameters 267
 Dose-response, three parameters 255, 256
 Dose-shift equation 279
 Double exponential decay 360

- E -

EC50 defined 236
 EC90 283
 ECanything 283
 Enzyme inhibition, general 337
 Enzyme kinetics, assumptions 336
 Enzyme progress curve 335
 Equation, cloning 409
 Equilibrium binding 9
 Exponential association equation, one phase 364
 Exponential association equation, two phases 367
 Exponential decay 9

Exponential decay equation, one phase 357
 Exponential decay equation, three phases 362
 Exponential decay equation, two phases 360
 Exponential decay, derived 356
 Exponential defined 355
 Exponential growth equation 368
 Extra sum-of-squares F test 48
 Extra sum-of-squares test 213

- F -

F test to compare two models 50
 F test, extra sum-of-squares 48
 F test, to compare nonlinear fits, interpreting 213
 False minimum 206
 Family of curves, graphing 442
 Fit tab, nonlinear regression 155
 Fitting straight lines to semi-log graphs: 376
 Fitting two line segments 374
 Five parameter dose-response curve 267
 Force line through origin 372
 Fractional occupancy 285
 Functions, graphing 442

- G -

Gaddum/Schild EC50 shift equation 277
 Gaussian distribution, equation 386
 General weighting 31
 Generic curves 447
 GI50 248, 251
 Global nonlinear regression 41
 Graphing tips: Linear regression 92
 GraphPad radioactivity web calculator 290

- H -

Hill slope 240
 Hill slope, in saturation binding 300
 Hit constraint. Nonlinear regression. 231
 Homologous binding defined 311
 Hougaard's measure of skewness 204
 How the F test works: 50
 How to: Linear regression interpolation 76
 How to: Linear regression lines 76
 How to: Nonlinear regression 126

- I -

IC50 261, 262, 264, 265
IC50 defined 236
IC50, absolute 248
IC50, relative 248, 251
IC90 283
ICanything 283
IF() statement in segmental regression 421
Implicit equations 414, 416
Impossible weights. Nonlinear regression. 233
Information theory 51
Inhibition of enzymes by substrate 352
Initial rate, in enzyme kinetics 335
Initial values tab 162
Initial values, rules for 433
Intercept of linear regression 84
Interpolating from a standard curve 104
Interrupted nonlinear regression 226

- K -

Kcat 340
Kd, meaning of 285
Kinetics of competitive binding, equation 333
Kinetics of radioligand binding 325
Kolmogorov-Smirnov test of residuals 185

- L -

Lack of fit, testing for 194
Law of mass action 9, 285
Ligand depletion 289
Linear regression through origin 372
Linear regression, comparing slopes and intercepts 89
Linear regression, goal 14
Linear regression, how it works 15
Linear regression, runs test 90
Linear regression, when to use nonlinear instead 82
Linear vs. nonlinear regression 17
Lines, fitting with nonlinear regression 370
Lineweaver-Burk plot 338
Lineweaver-Burk plots, avoid 19
Local minimum 206
Log Gaussian distribution, equation 388

log((inhibitor) vs. normalized response equation 264
log(agonist) vs. normalized response equation 259
log(agonist) vs. normalized response, equation 258
log(agonist) vs. response -- Variable slope equation 256
log(agonist) vs. response equation 255
log(antagonist) vs. normalized response-- Variable slope equation 265
log(inhibitor) vs. response -- Variable slope equation 262
log(inhibitor) vs. response, equation 261
Logistic curve, four parameter 256, 262
Logistic equation, five parameter 267
Logistic regression 21
Lorentzian distribution, basis for robust regression 59
Lorentzian equation 391
Lowess curve 447

- M -

Marquardt method 62
Mass action, law of 285
Mathematical model 9
Michaelis-Menten model 338
Mixed-model inhibition of enzyme 350
Model, defined 9
Models have the same DF 213
Morrison equation of enzyme inhibition 353
Multiple regression 21

- N -

Nested models 213
Noncompetitive inhibition of enzyme 347
Nonlinear regression, how it works 62
Nonlinear regression, unequal weighting 30
Nonlinear regression, when to use for fitting a line 82
Nonlinear vs. linear regression 17
Nonspecific binding 288
Normality test, residuals of nonlinear regression 185
Normality tests 165
Normalized data and weighted nonlinear regression 35
Normalized dose-response models 243
Normalizing data, pros and cons 243

Normalizing. Does it make sense for dose-response data? 241

Not converged. Nonlinear regression. 227

- O -

Occupancy of receptor 285

Off rate of radioligand binding 325

On rates of radioligand binding 325

One phase exponential association equation 364

One phase exponential decay equation 357

One site -- Fit total and nonspecific binding, equation 293

One site -- Specific binding equation 297, 300

One site -- Total binding equation 292

One site -- Total, accounting for ligand depletion equation 295

One site competition binding with ligand depletion 318

One site competition equation 312, 314

One site competition homologous competition 321

One site with allosteric modulator equation 309

Operational model. Partial agonists experiments 275

Operational model. Receptor depletion experiments 272

Optical density 9

Origin, forcing a line to go through 372

Orthogonal regression 100

Outlier removal, nonlinear regression 53, 57

Outlier removal, when to avoid 54

Outliers table 207

Outliers, graphing 176

Outliers, what to do when you find one 207

Output tab, nonlinear regression 164

- P -

Parallel regression lines, testing for. 89

Perfect fit. Nonlinear regression. 233

Plateau followed by one phase decay, equation 359

Plotting a function 442

Point-to-point 'curve' 447

Poisson weighting 31

Polynomial equations 384

Polynomial regression 21, 381

Power series 404

Prediction bands 165

Prediction bands, computation of 67

Prediction bands, linear regression 92

Probability axes, fitting straight lines on 379

Proportional hazards regression 21

- Q -

Q, ROUT coefficient 53, 57

Q, ROUT coefficient, entering 160

Quartet, Anscome 18

- R -

R squared, linear regression 86

R2 of weighted fits 65

R2, adjusted 216

R2, nonlinear regression 187

Radioactivity calculations 290

Radioligand binding, ligand depletion 289

Radioligand binding, nonspecific binding 288

Range tab, nonlinear regression 164

Regression through origin 372

Regression, Deming 100

Regression, Orthogonal 100

Regression, use of the term 15

Relative IC50 248, 251

Relative weighting 31

Replicates test 194

Replicates, fitting with nonlinear regression 68

Residual plot, nonlinear regression 179

Residuals from linear regression 92

Residuals, testing normality 165

Robust nonlinear regression 59

Robust nonlinear regression, choosing 155

ROUT coefficient, Q 53, 57

ROUT coefficient, Q, entering 160

ROUT method of identifying outliers 53, 57

Rules for initial values in nonlinear regression 433

Runs test following linear regression 90

Runs test, nonlinear regression 193

- S -

Sampling function (sinc wave) 395

Saturation binding, defined 291

Scatchard plots, avoid 19

SD of residuals, from nonlinear regression 187

SE of best-fit parameters of nonlinear regression
183

Segmental linear regression 374

Segmental nonlinear regression 421

Semilog graphs, straight lines on 376

Shapiro-Wilk normality test of residuals 185

Sharing parameters 158

Sharing parameters in nonlinear regression 41

Sigmoidal substrate velocity plots 342

Sinc wave 395

Sine wave 393

Sine wave, damped 394

Skewness of nonlinear regression parameter 204

Slope factor 240

Slope of linear regression 84

Slope, testing if is really zero 88

Sorting equations 410

Specific binding, defined 291

SSreg, from nonlinear regression 187

Standard curve, interpolating 104

Standard errors 165

Steady state, in enzyme kinetics 335

Substrate inhibition of enzymes 352

Summary table, nonlinear regression 164

Sum-of-squares, why minimize 61

sy.x, from nonlinear regression 187

Symmetrical confidence intervals, nonlinear regression
438

- T -

Theoretical curves, graphing 442

Three parameter dose-response curve 255, 256

Three phase exponential decay equation 362

Tight inhibition of enzymes 353

Too complex, defined 414

Too few points. Nonlinear regression 232

Transforms of best-fit values of parameters of nonlinear
regression 438

Two phase exponential association equation 367

Two phase exponential decay equation 360

Two site competition equation 315, 317

Two sites -- Fit total and nonspecific binding, equation
307

Two sites -- Specific binding only, equation 304

- U -

Uncompetitive inhibition of enzyme 349

Unequal weighting 160

User-defined equations 411

- V -

Variable slope vs. fixed slope, dose-response 241

- W -

Weighted nonlinear regression, normalized data 35

Weighted residuals 179

Weighting in nonlinear regression, implementation
65

Weighting in nonlinear regression, math 31

Weighting in nonlinear regression, outliers 65

Weighting in nonlinear regression, R² 65

Weighting in robust regression 65

Weighting, general 31

Weighting, nonlinear regression 30

Weighting, Poisson 31

Weighting, relative 31

Weights tab, nonlinear regression 160