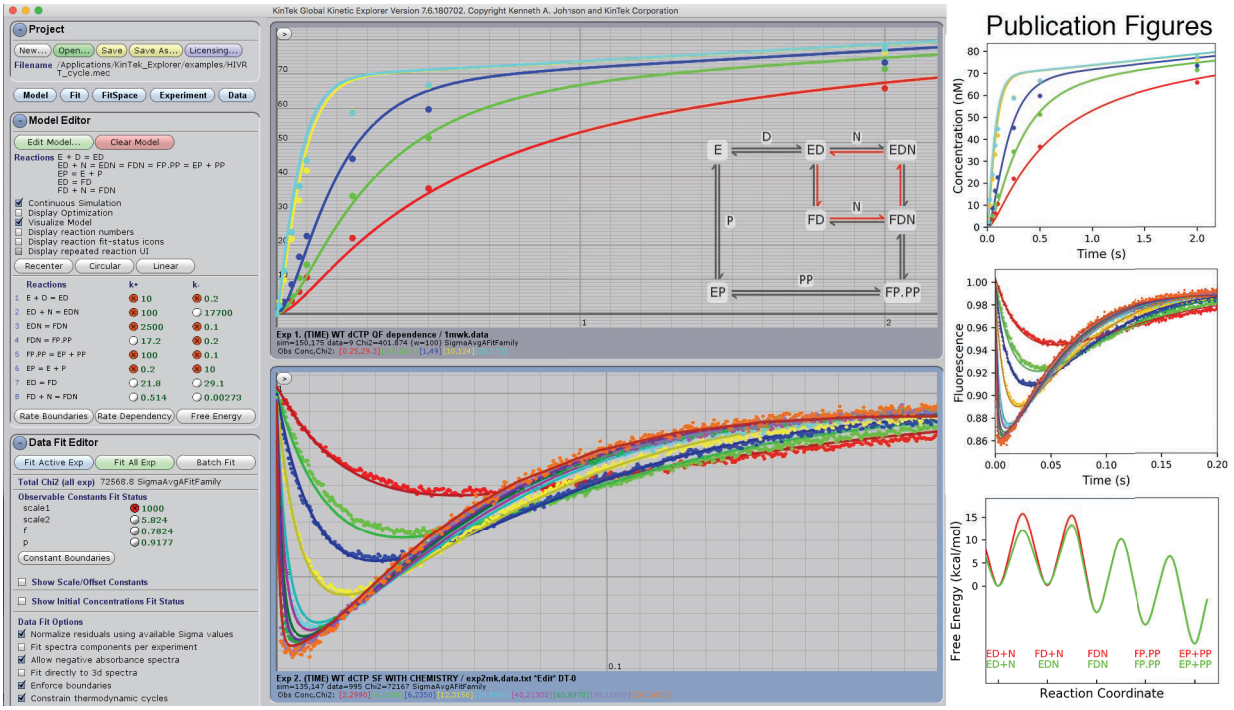
KinTek *Global Kinetic* Explorer™

Version 8.0

with FitSpace Explorer™

and SpectraFit™



This figure illustrates the global fitting of two experiments along with the output of publication quality figures and the free energy profile. The data can be found in the example file HIVRT_cycle.mec from work by Kellinger and Johnson (1). In these experiments, the role of induced-fit in enzyme specificity was established to be a function of the fast conformational change from the open to the closed state and slow dissociation of substrate from the closed state, as can be seen in pathway below. This work resolved a 60-year-old controversy regarding the role of induced-fit in enzyme specificity. Note that for illustration, we also show the alternative minor pathway where the conformational change precedes substrate binding. Data fitting can allow both pathways and we compute the *flux integral* for each.

$$\underset{open}{ED_n} + N \underset{19600\ s^{-1}}{\overset{100\ \mu M^{-1}s^{-1}}{\rightleftharpoons}} \underset{open}{ED_n.N} \underset{0.1\ s^{-1}}{\overset{2500\ s^{-1}}{\rightleftharpoons}} \underset{closed}{FD_n.N} \underset{0.2\ s^{-1}}{\overset{17\ s^{-1}}{\rightleftharpoons}} \underset{closed}{FD_{n+1}.PP} \underset{0.1\ s^{-1}}{\overset{100\ s^{-1}}{\rightleftharpoons}} \underset{open}{ED_{n+1}} + PP$$

Published by KinTek Corporation, Austin, Texas, USA

System Requirements:

We recommend Mac computers for the best performance and the least hassle. Although the code for both Mac and Windows computation is identical, performance is significantly slower when running under Windows.

Mac OSX 10.6 or later, We no longer support the older Mac Power PC computers (pre-2006). Mac OSX 10.12 (Sierra) requires *KinTek Explorer* version 6.0 or later.

Windows 32 bit: We no longer support Windows 32-bit operating systems. If you need an older version for Windows 32 bit, email kajohnson@kintekcorp.com. We can provide an older version from April, 2016 upon request.

Windows 7, Windows 8 or Windows 10: 64-bit operating systems are supported but require at least 1 GB of RAM memory and a graphics card capable of running *open GL*. In general almost any graphics card works. The problems we have encountered are with inexpensive PC systems that lack a graphics card so that the system emulates *open GL* and the operation becomes very sluggish. If you have slow graphics performance, upgrading your graphics card is likely to solve the problem. *Starting with Version 6.0, we no longer support Win32-bit operating systems. Please try the software before ordering the license. In the absence of a license, the software can be run in a "student" mode where all functions are allowed except the importing of new data and exporting of results of the simulation.*

A note on Windows 7: If you are experiencing problems with *KinTek Explorer* (sluggish performance or strange graphic effects), try updating the driver for your graphics card. You can find that latest driver for your graphics card online. If that fails you may need to purchase a new graphics card which supports *open GL*.

On a small fraction Windows 7 laptop computers, we have noted that after setting up a new simulation traces are not displayed until data are included. You can use generate synthetic data to overcome this limitation. However, this problem does not appear with any other versions of Windows, so we have not invested resources in finding the cause and fixing of the problem for a small fraction of computers. *In particular, we have not seen any issues with* ***Windows 10*** *so an upgrade may be in order if your laptop reveals strange graphics problems.*

However, *recent Dell laptop computers* (2017-2018) with Windows 10 have shown problems in running *KinTek Explorer* and we have tracked the problem to an outdated graphics driver provided by Dell. To correct the problems, download an updated driver directly from the vendor (not Dell).

For example, for an Intel graphics card, get the most recent driver here: https://downloadcenter.intel.com/ After you download and unzip the file, go through: Settings => System => Display => Display Adapter Properties => Properties => Driver => Update Driver => Browse My Computer for Driver software. Tell the computer that you have a disk and point to the Graphics folder within the driver package you just unzipped. It should then install.

Table of Contents

Chapter 1. Software Installation and New Features

KinTek Explorer is ideal for students who are first learning kinetics but is also a cutting edge tool for the most challenging data fitting problems. *KinTek Explorer* affords dynamic visualization and global fitting of multiple data sets to a single unifying model. Using a fast and efficient algorithm, *KinTek Explorer* simulates complex reaction schemes by numerical integration, and is built upon a simple and intuitive user interface. A reaction scheme is entered using standard reaction sequence notation. The starting concentrations of reactants and the observable output expression then define each experiment. Using this information, estimates of rate constants and output scaling factors then form the basis for the simulation. Each rate constant, starting concentration, and scaling factor can then be adjusted while the display is dynamically updated to show the effect of each parameter on the observable outputs. This powerful interface provides direct feedback to the user to develop understanding of complex relationships between reaction mechanism, rate constants and observable reactions. Moreover, the visual feedback guides the search for initial parameters required to fit the data by nonlinear regression. Multiple data sets can be fit simultaneously to a single unifying model!

Starting with Version 7.0 there is only one program that can be operated in either an *Unlicensed* state (equivalent to the student version) or in a *Licensed* state (equivalent to the previous Professional version). The *Unlicensed* (*Student*) version is ideal for teaching and learning kinetics since it offers the full power of the *Professional* version, but is limited in that data cannot be imported and the results of simulations cannot be exported. The *Licensed* (*Professional*) version allows your own data to be imported and fitted and the results exported. By purchasing a license, you are supporting our ongoing efforts to provide the very best data fitting software for use in teaching and research.

Please read this instruction manual to learn the full power of the software. Although we have developed an intuitive user interface, there are several unusual features of the software that are not likely to be discovered by trial and error. Moreover, the instruction manual goes beyond the simple how-to approach by giving the rationale behind the use of the software. We also provide a treatise on error analysis in Chapter 9. Video tutorials are available on the website: http://www.kintekcorp.com.

Global fitting of a family of data sets to a single model sets a new standard for extracting mechanistic information from kinetic data, but it must be approached with a good deal of understanding and critical thinking. On the one hand, fitting to a family of curves simultaneously is more robust because it reduces the number of independent parameters required to fit the data and maintains the relationships between rates and amplitudes of reactions that are often lost in conventional data fitting. On the other hand, with the power of computer simulation, it is far too easy to include extra steps in a model that are not defined by the data. *KinTek Explorer* is intended to help you recognized when a model is overly complex both by statistical analysis of fitted parameters, but also by visual exploration of the multi-parameter landscape. *FitSpace Explorer* provides reliable visual feedback and robust estimates of errors on fitted parameters while often revealing complex relationships between fitted parameters. The *FitSpace* confidence contour analysis and the dynamic simulation are features unique to *KinTek Explorer* that enable the exploration of parameter space to reveal which parameters are constrained by the data.

We also include *SpectraFit*, which allows you to analyze time-resolved spectra and fit the data directly to your model based upon numerical integration of the rate equations without the simplifying assumptions that plague other methods of data fitting. *SpectraFit* is described in Chapter 10 and is illustrated with several example files.

Each year I teach an intensive four-day *Kinetics Workshop* in which I illustrate the concepts underlying kinetics and show how to effectively use *KinTek Explorer* to design experiments and interpret results quantitatively. Check our website, http://www.kintekcorp.com, for announcements of the next workshop.

Kenneth A. Johnson, Austin, Texas

1.1 New Features Summarized by Version Number

Version 8.0

Compatibility with Mac OS 10.14 (Mojave). Upgrade to the new Mac operating system has caused significant problems for many programs. *KinTek Explorer* is now fully compatible with Mac OS 10.14.

SVD Analysis and fitting of multiple experiments. We have made major improvements in the fitting of multiple experiments when using time-resolved spectra (or titrations). We can now fit multiple time-resolved spectra collected at different concentrations of reactants. In addition we can include spectra for known species. A video tutorial will be released soon showing how to take advantage of these advanced features.

Enhanced Analytic Fit Capabilities. We have added a feature to show the equation being fit and now allow starting estimates to be scrolled while watching the changes in the shape of the curves. This will be an excellent tool for teaching, but also for finding better starting estimates for data that are difficult to fit using equations. As part of this we have redefined the exponential functions so the amplitude terms are more intuitive. *See Section 5.13.a*

Improved Detection of Closed Thermodynamic Loops. We have improved the methods used to detect and enforce closed thermodynamic loops, especially when there are multiple reactants where conventional methods fail to detect a closed loop.

Added Transmission Coefficient for Free Energy Profiles. We have added a transmission coefficient in computing the free energy profile. See *Section 4.5*. Using a transmission coefficient less than 1 may be more realistic and it also better illustrates the differences between energy barriers because each one is smaller.

Enhanced Output. Numerous changes were made to improve the output of publication-quality figures, especially the *Free Energy Profiles* and *Confidence Contours*. In addition, we offer control over whether to display standard deviation bars for each data point.

Improve Batch Mode. We now offer a more interactive batch mode so you can inspect each file and choose to accept, refine, or discard each fit. See *Section 7.4*.

Enhanced Performance. Several changes were made to enhance the speed of execution and eliminate errors.

Version 7.6

Multiple Concentration Series. It is now possible to vary the concentrations of multiple species within a single experiment. Previously a concentration series was restricted to a single species in a given experiment. Note that in entering multiple concentrations of more than one species, the number of concentrations must match for all species, except for those with only one concentration. When fitting to analytic function, plots of rate or amplitude versus concentration refer to the species in excess. See *Section 5.3*.

Improved x-axis auto-scaling. We automatically set the range of the x-axis so that it begins at zero if the data are close to zero, but otherwise begins with the lowest x-value of the data. In either case, the upper limit for the x-axis display is just beyond the last data point.

Cleaner Data Repository layout. The *Data Repository* layout has been refined to make it easier to keep track of files and display only what is needed. This is especially helpful for *SVD Analysis* of time-resolved or concentration-dependent spectra.

Add a mix-step after a titration. You can now perform an additional mix-step after a titration. This allows more accurate model of experiments where you may incubate two reacts at various concentrations of one (titrant), then perform another addition to initiate a reaction. We then plot the amount of product formed as a function of concentration in the original titration.

Labels for Time or Conc or pH. We added labels to the legend to indicate whether the x-axis represents time, concentration, or pH. \

Export overlaid figures. If you check *Overlay all plots* under the *Experiment Editor*, you can export a publication quality figure the includes all overlaid plots.

Version 7.5

Flux Integrals. We now allow the numerical computation of the integral of the amount of species formed by a given pathway. The term "flux" has frequently been applied to estimates of the fraction of reactions proceeding by a given pathway. We now offer a method to directly compute the amount of reaction proceeding through a given step using reaction integrals.

Embedded Python commands. We now offer the option of downloading a version of KinTek Explorer with commands for exporting professional quality figures embedded in the software so there is no need to install Python separately. This applies only to the licensed version of the software.

Graphical output improvements. The routines for outputting publication-quality figures have been improved especially with respect to enabling the optional inclusion of error bars on output figures and including additional options for configuring the output.

Free energy profile output now more clearly identifies alternative pathways.

Improved User Interface. Based on feedback at the recent *Kinetics Workshop*, we have added information in the legend of each Experiment figure to better define what is being displayed and the linked data files. Similarly, the data files in the *Data Repository* are now flagged to indicate in which experiment they are used.

Better detection of closed loops. We have completely re-written the routines we used to detect and enforce closed thermodynamic loops to work more accurately with complex mechanisms. In addition, the user can now choose which loops to display and how to configure them.

Numbered reactions. We have now added numbers to the listing of reactions and rate constants. This better defines the relationships between rate constants and their confidence contours. Throughout the software, we have avoided references to numbered rate constants, but it is unavoidable in the display of the confidence contours and in specifying the flux integral through individual steps.

Highlighting rates in the display model. Arrows on the display model now turn white when the corresponding rate constant is scrolled in the *Model Editor*. Similarly, in you scroll a rate constant by clicking and dragging on an arrow in the display model, the rate constant is highlighted in the rate constant list in the *Model Editor*.

Continual refinement of the software to improve reliability.

Version 7.0

Softkey licensing. The major advance with Version 7.0 is the ability to license the software using a conventional software licensing key rather than rely upon an USB dongle. Many users have complained about losing their USB dongle, so this solves that problem. One license can be activated on two computers and is then tied to that computer. All future purchases will be based upon the software licensing key.

SpectraFit included. With the switch to softkey licensing, we are also including *SpectraFit* with all software licenses. A separate license is no longer needed to use *SpectraFit*. We hope this will encourage more to take advantage of this powerful method of data analysis.

Error bars on Rate vs Conc plots. We have now added standard error bars on plots of Rate or Amplitude

versus concentration after saving to the data repository and fitting to conventional functions. The error bars will appear on the output publication quality figures.

Improved speed in modeling SPR data. In modeling Surface Plasmon Resonance signals, we have increased the speed of simulations when all of the experiments within a given file represent SRP data. This allows binding steps to be modeled as pseudo-first order reactions at the constant concentration of the analyte in solution.

Importing ITC and Single Molecule histogram data. Both ITC and single molecule data are actually first derivatives of the data for the titration or time dependence, respectively. Therefore, when importing ITC (Isothermal titration calorimetry) data or single molecule histogram data, you can integrate the data so that they are displayed properly for fitting in *KinTek Explorer*.

Version 6.3

In addition to small bug fixes, we made the following enhancements:

Increased speed for Fitspace. We have further optimized the full use of multiple computer cores to improve the calculation of Fitspace confidence contours by doing more calculations simultaneously. Depending on the number of cores in your computer and the number of variables in your calculation this may improve speeds.

Extended aFit functionality. After fitting concentration series data to an analytical function, the rate or amplitude versus concentration can be saved to the *Data Repository* where it can be analyzed by fitting to various functions. For example, after fitting a concentration series of data to a single exponential function, the rate versus concentration can then be fit to a hyperbola after saving the plot to the data repository. This can help in developing a model by revealing patterns in the data.

Improved SVD fitting of time-dependent spectra. We made significant improvements in the ability to simultaneously fit multiple experiments involving time-dependent spectra (or concentration-dependent spectra). In addition, we have added the ability to include known spectra of individual species. We also added an option to fit spectra directly, rather than just the SVD amplitudes. These complex functions will be the subject of a video tutorial. We have also simplified the display of SVD analysis in the *Data Repository*.

Version 6.2

This version removes a number of small bugs. Most importantly, for Mac users of the Professional version of KinTek Explorer, the output of 3D plots was broken by Mac OSX 10.12. The new version requires that you download and install Anaconda Python for the output of publication quality figures. See the installation instructions for more details, or just search for Anaconda Python and install version 2.7.

Version 6.1

This version fixes a number of small bugs in Version 6.0 and eliminates the complications in downloading software for Mac computers—KinTek is now an Apple-approved developer!

Version 6.0

Windows 10 compatibility. Version 6.0 has been adapted to be compatible with Windows 10. Our tests indicate that the odd problems seen with Windows 7 on a minority of computers have been resolved.

Mac OSX 10.12 compatibility. Version 6.0 solves the problems seen on some Mac computers.

Direct output of publication quality EPS and PNG files. In exporting the simulation fitting, we now generate a publication quality EPS vector graphics file. You can now go straight from fitting to publication without the tedious and time consuming process of inputting data and formatting a figure for publication using an external graphics program. If necessary, the EPS file can be directly edited or combined into a composite figure using Adobe Illustrator. In addition to the simulation based fitting results, you can output the results from fitting to analytic functions, and the results of SVD analysis in fitting time-resolves spectra, including 3D plots of the spectra. This feature is available only in the Professional version of the software. Student assignments can use screen capture. *Note if you are using Windows, you will need to install Python to take advantage of this new feature. See installation notes*.

Fitting to define starting concentrations. We have enabled a new function to allow the starting concentrations of reactants to vary in the process of fitting. You can limit the allowed range of variation to a fixed percentage. This is useful to account for slight errors in pipetting to make up solutions or to solve for unknown enzyme concentrations.

Labeled/Unlabeled reactants. We have added functions to allow easy entry of parallel reactions for labeled and unlabeled reactants. See ribozyme.mec for an example involving the use of fluorescently labeled and unlabeled oligonucleotide binding to a ribozyme. Use the ^ **symbol** *after* the species name in the *Model Editor* to designate labeled and unlabeled variants of the same species. The program then creates two parallel sets of reactions with identical rate constants. See *Section 4.1c*.

Lettered rather than numbered linked rate constants. We expanded the number of linked rate constants from 9 to 26 by switching from number 1-9 to letters a-z. See *Section 4.2c. Note: In parts of this manual, the numbered groups are still shown in illustrations.*

Rate or Amplitude versus Concentration Plots. In fitting data to analytic functions for a concentration series, it is useful to plot rate and amplitude versus concentration to infer the underlying mechanism. However, at the extremes of high or low substrate concentration, fitting often returns values with large errors and this throws off the plots. Now with Version 6.0 you can eliminate points from the figure to reveal the concentration dependence from the reliable data. Just *Shift-click* on a data point on the plot and it will be removed and the plot rescaled. For a good example of this, see PNPase-Pf-DADMe-ImmH.mec and use *aFit* with a burst equation. At low and high inhibitor concentration, it is not possible to obtain reliable fits to the burst equation due to the errors inherent in fitting data to multiple exponentials. In conventional fitting these data must be excluded–that is why we fit globally using simulation where all of the data can be included!

SPR Data Fitting. We have made an initial effort to fit data obtained by SPR (surface plasmon resonance) by allowing "mix-steps" to specify a new fixed concentration that is not consumed by the reaction since it replenished by the continuous flow. With each new phase of SPR, the concentration of the reactant (analyte) is specified by the concentration in the solution. Use the syntax: **[%n]** where n is the new concentration (brackets dictate that the concentration remains constant). Without the brackets, **@n** specifies a new concentration, forgetting the concentrations carried over from previous steps. *There are still serious limitations in fitting SPR data rigorously because of complications due to mass transfer and surface density of receptors!* See *Section 5.8.*

SVD analysis of equilibrium titrations. Spectra collected as a function of a titration, including pH dependence, can now be fit using SVD analysis to resolve component spectra. This is illustrated by the pH dependence of absorption and fluorescence spectra. As part of this, we enabled the use of *log scales* in titrations.

Repetitive reactions. It is now easier to enter multiple reactions in sequence with linked rate constants. For example, you can easily program 100 sequential reactions for DNA polymerization or actin nucleation and polymerization. *See Section 4.1d.*

Version 5.2

Log scale concentration series in titrations. We have enabled the input of an exponentially increasing concentration series during a titration. This is especially useful when performing pH titrations where the concentration of hydrogen ion spans orders of magnitude.

Fixed concentrations during a titration. Especially useful for pH titrations, we now enable the user to specify that a given concentration is held fixed during the reaction. Use the brackets to specify that the concentration of the titrant does not change: [@x-y, n]

The syntax for a titration has been changed to specify the number of points to compute in the titration, not the interval between steps. For example, enter @0-1,100 rather than @0-1,0.01 (old syntax).

Plots of Rate or Amplitude versus concentration. The dialog now has checkboxes for the rates or amplitudes so that you can selectively turn on/off plotting for rates (all concentrations). You can also now shift-click on individual points to remove them from the plot. To reinstate them, just toggle the rate checkbox off/on to reset all concentrations for that rate.

Endpoint function. Click on *"Endpoints"* under the *Experiment Editor* and see the concentrations of all species at the end of the reaction. This is useful in modeling complex reactions to see the equilibrium endpoint. The values are updated when parameters are changed such as time of reaction or starting concentrations.

Ability to constrain the product of equilibrium constants at a defined value. In the example of EPSP.mec we know that the product of $K_1K_2K_3K_4K_5K_6 = 180$, the net equilibrium constant for the chemical reaction. You can now specify a constraint to define the net product of individual equilibrium constants.

Version 5.1

Windows 64-bit: The major change with *5.5* is the addition of a Windows 64-bit version, which is about twice as fast as the 32-bit version so that it is almost as fast a Mac. We will be phasing out the 32-bit version over the next year or two. Until then, look for both versions on the website.

Tabs: You can now tab between entries for rate constants or concentrations. This makes it much easier to enter a set of known rate constants in creating a model.

Denaturant Concentration Dependence: In studies on protein folding, the rate constants governing folding and unfolding follow an exponential dependence on the concentration of the denaturant (urea or guanidine hydrochloride). In Version 5.1 we have implemented the functions to globally fit the concentration dependence of folding and unfolding following the input similar to the programming of voltage- and temperature- dependence in the *Model Editor – Rate Dependency*. This is not to be confused with the normal reactant concentration dependence entered in the *Experiment Editor*.

Version 5.0

Version 5.0 is based upon extensive work to completely re-write the data fitting routines to enable boundaries on rate constants and to recognized and enforce thermodynamic cycles (maintaining a net equilibrium constant of one for the product of reactions around a closed loop). In addition, we have greatly improved the *Model Display* and included a graph of the free energy profile.

Upper and Lower Limits: The user can now enter upper and lower limits on rate constants and output scaling factors when fitting data. This can be used to prevent unrealistic values when attempting to fit data

to a model where some rate constants are not well constrained. *See Section 4.2d.*

Thermodynamic Loops: The software will now recognize a closed loop (not an enzyme cycle) where the net free energy change around the loop should be zero and the net product of all equilibrium constants should be unity. The program will now enforce conservation of energy in going around the cycle. Cycles are shown with colored arrows in the model display. See *Section 7.2c.*

New Model Display: The reaction sequence can now be viewed in a meaningful display and the user can arrange the elements of the model in a way that helps with their intuitive understanding of the reaction. To facilitate this, the *ligand* for any biomolecular reaction needs to be specified by entering it second in the reaction sequence. For example in the reaction of A + B = AB the order specifies that B is the ligand and we then model the conversion of A to AB in a reaction whose rate and free energy depend on the concentration of B. See *Section 4.1c.*

Free Energy Profile: We have implemented a display of the free energy profile for the reaction, or any portion of the reaction. Note that physiological concentrations of small ligands (substrates, products, inhibitors, etc) need to be entered to provide a reference state. For any bimolecular reaction, one of the partners needs to be specified as the *ligand* by entering it second in the pathway. See *Section 4.3d.*

Plot of Rate or Amplitude vs Concentration. We now include the option to plot Amplitude versus concentration after a fit to an analytical function, and we include standard error estimates in the plots. See *Section 5.13b*.

Version 4.0

Student Version Open/Save: The student version of the software is now capable of saving files and can open any mechanism file. The only restrictions are that you cannot import data or export the results of fitting or simulations. In addition, to aid in the utility of the software in teaching, synthetic data created in the student version can be exported.

Voltage-dependent rate constants: The software can now fit a family of curves involving voltage-dependent rate constants as in ion channels. The channel is equilibrated at a holding voltage and then the current is recorded as a function of time after a jump in voltage. A family of curves can be fit simultaneously to determine the voltage dependent terms, including the *Partial Gating Charge*.

Temperature-dependent rate constants: The software simulates an experiment in which the system is first equilibrated at T1 and then the relaxation to equilibrium is monitored after a rapid jump in temperature. A family of curves collected at different temperatures can be fit simultaneously to derive the activation energy for each rate constant in the reaction.

Compact 3D confidence plots: As an aid to creating figures for publication, we have implemented a option for a compact display of 3D confidence plots.

Numbers in output expressions: You can now enter numeric values directly in output observable expressions, for example, ES + ES + 2*ES2. The parser will recognize any purely numerical entity as a fixed numeric value in the output expression. See *Section 5.2b.*

Version 3.0

Multithreading: Starting with Version 3.0, a major re-programming effort has enabled *use of multiple computer cores*, which increases speed approximately 6-fold on a Mac 8-core computer. Performance improvements are somewhat less under Windows due to inefficiencies beyond our control. You will see that multiple *FitSpace* calculations are now done simultaneously using different computer cores to greatly accelerate these time consuming calculations.

Mac OSX and Windows Native File Directories: Saving and opening files are now performed using the file directory forms native to Windows and Mac OSX.

One-dimensional FitSpace: As a simplified version of FitSpace, we now offer a 1 dimensional search. Each parameter is varied and the best fit is obtained by allowing all other constants to vary. The change in chi^2 can be used to obtain reliable error estimates for each parameter.

Manual Entry of FitSpace Boundaries: The boundaries over which the program will search for changes in chi2 as a function of each parameter can be entered manually. This is especially useful in preparing a publication quality figure.

Graph Zoom: Left-click and drag on the graph to define a box to zoom in. Use right-click-drag to pan the view window area. Use Ctrl-W to return to the full graph display.

Simultaneous fitting of multiple time resolved spectra. We have improved the simultaneous fitting of multiple time-resolved spectra by solving equations to get a single set of extinction coefficients that best accounts for all spectra. This can be very useful since spectra collected different concentrations of reactants can contribute differently to the accuracy of determining spectra and rate constants governing individual species.

Global Fit Weight: We added a new feature to alter the weight given to different experiments during data fitting. Under the *Experiment Editor*, you can change the *Global Fit Weight* for a given experiment to any positive number. The weight given each experiment will be the ratio of the value for that experiment relative to the value for other experiments. For example if the *Global Fit Weight* = 10 for one experiment and 1 for all others, the chi^2 values will be multiplied by 10 in computing and attempting to minimize the total chi^2 in global fitting. This feature can be useful in seeking a global fit that accounts for all experiments; for example, experiments with fewer data points can be weighted to increase their influence in finding a good global fit. NOTE: The *Global Fit Weight* values should be reset to 1 when evaluating the errors on parameters derived from a final global data fit.

Display of Residuals: For each experiment, the residuals ($y_{observed}$-$y_{calculated}$) can be displayed as an aid to evaluating goodness of fit.

Version 2.6

Modeling Pulse-Chase Experiments: Pulse-chase experiments can be modeled using the **> *symbol***. See *Section 5.10*.

Modeling Equilibrium Titrations: Equilibrium titration experiments can now be modeled using the **@ symbol**. We take a unique approach to modeling equilibrium titrations in that the rates of reaction are used to compute the endpoint of the reaction according to the time allowed for equilibration. This reveals some interesting effects for slow reactions that are not at equilibrium. Moreover, complex reactions can now be modeled precisely without the inaccurate simplifying assumptions required to solve equations describing multiple equilibria. See *Section 5.11*.

Correcting fluorescence for absorption of incident light. Reductions in fluorescence intensity due to absorption of exciting light by a product or reactant can be included in the model as described in *Section 5.2.d*.

Version 2.5

Fitting Time-Resolved Spectra using SVD: Starting with Version 2.5, we include an optional add-on for fitting time-resolved spectra. As described in Chapter 10 on *SpectraFit,* time-resolved spectra are de-convoluted by singular value decomposition (SVD) and the resulting amplitude vectors are then fit to the model programmed by the user to derive the spectra and time-dependence of individual species. Unlike other implementations of SVD, the user is not restricted to preprogrammed models with simplifying assumptions; rather, users fit data directly to the model based upon numerical integration of the rate equations with no simplifying assumptions. Moreover, multiple experiments with various signals and protocols can be fit simultaneously, with comprehensive error analysis .

As shown on the order form, there is an additional charge to cover the cost of this extensive programming effort. If you already own a license, you can add the *SpectraFit* feature by paying an additional licensing fee. However, if you do not need the ability to fit time-resolved spectra, your existing license will still allow you to benefit from our continued advancement in other aspects of the software with no additional costs. Moreover, you can open example files showing the fitting of time-resolved spectra; you just will not be able to import time-resolved spectra and run SVD. The student version of the software also allows four example files to be opened which illustrate the use of SVD to fit time-resolved spectra.

BatchFit. You can now program the software to automatically fit a large number of data files to the same model and report the results in a summary table.

The ***% symbol*** can be used in a double mixing experiment to bring the concentration of a reactant to zero in the second mixing step. This is useful to mimic a rapid gel filtration experiment.

1.2 Installing the program

1. Download the latest version of the software (see list below) from the KinTek website:
https://www.kintekexplorer.com

Software is available for both Mac and Windows platforms:

For PC: KinTek_Explorer_win64.zip or KinTek_Explorer_Py_win64.zip
For Mac (post 2006): KinTek_Explorer_MacOSX.dmg or KinTek_Explorer_Py_MacOSX.dmg

The version with "*_Py_*" in the name have Python graphics commands embedded in the software to you can output publication quality figures.

We no longer offer separate Student and Professional versions. Rather the software can either be run as a licensed version (to enable the attributes of the Professional version; namely, data input and output of results). In the absence of a valid license, the software runs as the Student version were input of data and output of results are not allowed.

2. Uncompress the file and place the KinTek_Explorer_Pro or the KinTek Explorer_Student folder anywhere on your hard drive. No further installation is required, but *you must run the program from the KinTek_Explorer directory*.

On Mac OSX, after double clicking on the .dmg file, drag the KinTek_Explorer folder to the Applications directory, or any other directory of your choice.

On Windows 7, you must NOT place the *KinTek Explorer* directory in the *Program Files* directory. Windows 7 does not allow the program to write to the *Program Files* directory and will cause the program to crash.

You will find several subdirectories:

documents contains this instruction manual, plus a tutorial and a sample exam.

examples contains about 110 example files.

rawData contains sample data files.

models and **output** are empty directories for saving new models and outputting results, respectively.

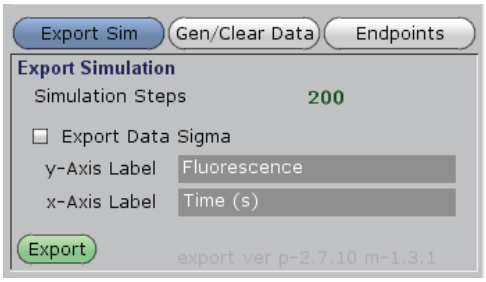
core contains program specific fonts and images that do not need to be accessed by the user.

Follow instructions on the website for purchasing a license and for downloading and installing the license keycode.

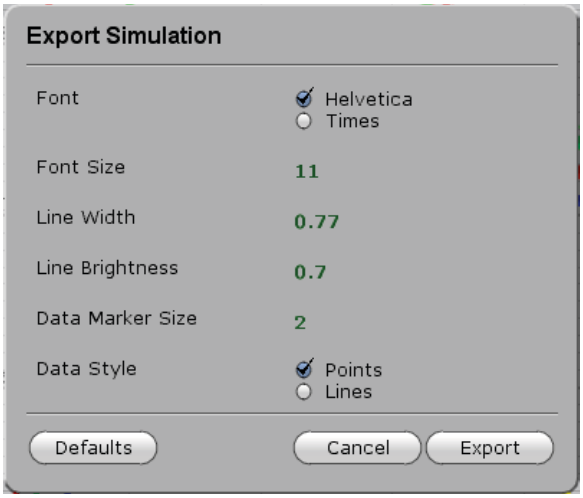
a. Installing Python program for publication quality graphics output.

In order to output of publication quality figures, you can either download the software version with "*_Py_*" in the name, which have Python graphics commands embedded in the software. Alternatively, you can install *Anaconda Python version 2.7.* You can either search for it, or go to the following URL: https://www.continuum.io/downloads. Then follow instructions for installation. DO NOT install Python 3.5, which contains changes incompatible with version 2.7. Because of these incompatibilities, the industry standard is version 2.7. The key test to make sure it is installed is when you click on "Export Sim" in the Licensed version of KinTek Explorer, then click "Export" after entering the axes labels, you should see the menu shown in the figures below. If you do install Python, you will only get the text files necessary to create your own graphics using a different graphics program.

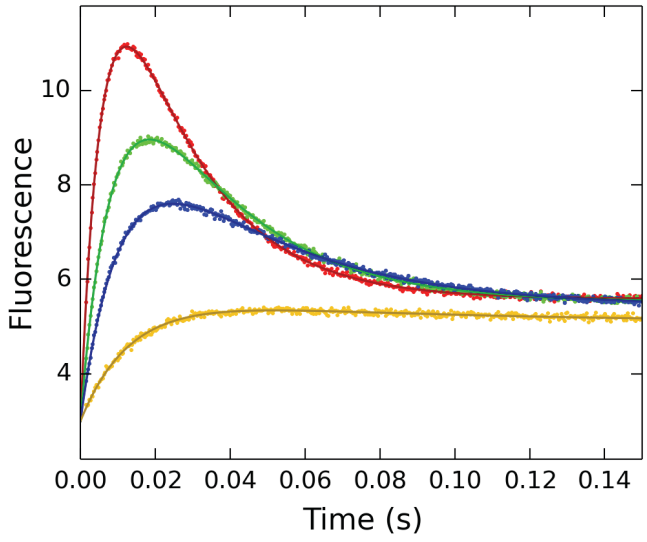
Under the Experiment Editor, click on "Export Sim", enter the axes labels, then click on "Export"



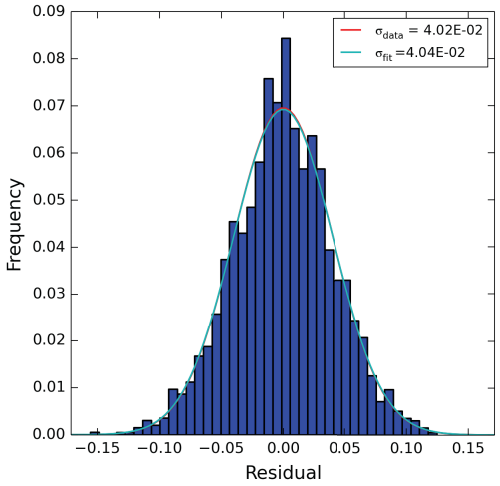
If Python is installed, you will get the menu at the right. Click *Export* and select a directory for saving the image files.



If Python is installed properly, you will get outputs as illustrated below as examples from Tryp_synthase.mec, showing the publication quality figures obtained using default parameters.



This shows the graphic output from the Tryp_synthase.mec example file.



This shows a histogram of the residuals with two lines, red and light blue, showing the distribution expected from the sigma values from the data and from the fitted curves, respectively.

Note the program outputs both .png and .eps files. The .png files are useful for embedding into manuscripts and slides, but the .eps files are vector graphics than can be enlarged to any size and can be edited in Adobe Illustrator, or other graphics programs to change colors or fonts. We offer limited selections of fonts (Helvetic or Times Roman) to conform to journal requirements.

b. Licensing keycode for Licensed version

To run the *Professional* version, a valid license is required. After purchase of the software, you will receive a licensing key tied to your email address. The software can then be installed on two computers.

c. Free Student version

In the absence of a valid license key, the software runs in "student mode." The *Student* version is limited in that you cannot import your own data, or export your results. Other than that, the program retains the full functionality of the *Licensed* version. For student assignments, there are sufficient example files to be used in teaching, and reports can be based upon a screen capture of the graphical display.

1.3 Running the program

Double click on the *KinTek_Explorer.exe* (or *KinTek_Explorer.app*) file to run the program. The program will open with a simple mechanism. You can see other examples by clicking *Open* and selecting from the list. If the example folder is not visible, search for the directory *KinTek_Explorer_Pro/examples*. To open a file, click on it in the directory listing, then click *OK*. You can navigate the directory listing by scrolling up/down using the center mouse wheel or by holding down the right-click mouse button to drag the listing up and down.

To immediately see the power of this program, click on any rate constant, concentration, or scaling factor with the mouse, and while holding the mouse button down, drag the value up or down and see the effect on the shape of the curves.

a. Dynamic simulation

Dynamic Simulation provides a powerful tool that can help you decide which rate constants are the most important and to find parameter sets that can serve as starting points for fitting by nonlinear regression. Careful analysis will soon reveal which constants are constrained by the data and which are not, both before and after fitting. Dynamic simulation can readily reveal when your model is overly complex in that it includes rate constant or output factors that are not constrained by the data.

You can control this function using the *Continuous Simulation* check box; if the box is checked, the simulation is updated as you drag any rate or concentration up/down with the mouse; alternatively, if the *Continuous Simulation* is not checked, the simulation is performed as soon as you let go of the mouse button. The latter function is useful when you are initially setting up a simulation or if the model is very complex or if your computer is slow.

b. Nonstandard features

There are three non-standard features in the current version of the program.

(1) The left hand panel (*Control Panel*) has a fixed width and expands vertically as more information is added. It can be scrolled up and down by right-clicking anywhere within the panel and dragging the mouse up and down; alternatively the control panel can be scrolled by rolling the middle mouse wheel up and down.

(2) Rate constants, concentrations and output factors can be scrolled up or down by clicking on an entry and dragging the mouse up or down; alternatively, if you click and release on the text, you then can enter a value into the text box, as described below (*Left click to change parameters*).

c. Mouse operations

The program takes full advantage of a three-button mouse. You can purchase a 3-button mouse for a Mac computer (including the Magic Mouse) or use Ctrl-click to get a right click function. On a Mac laptop, drag two fingers on the touch pad to get the scroll function.

Left click to change parameters: There are two operations. If you click and release on a parameter, a

cursor will appear in a text box and you can then edit the number or enter a new number. Entry is terminated by the Enter key on your keyboard or by clicking anywhere outside the text box. If you click and hold the mouse button down, you can drag the value up and down with the mouse. Note that rates and starting concentrations can be changed either on the Display Model or in the control panel. Hit the *Esc* (escape) button to cancel a scroll and return to the original value.

Left click to arrange the display model: Left click on a species in the display model and drag with the mouse to a new location.

Right click to move the model: Right click anywhere on the *Display Model* and drag the mouse to move the model to a new location.

Right click on the control panel: The control panel is one continuous list of items that expands as the model increases in complexity or to display fitting options and results. Right click anywhere on the control panel and drag the panel up or down with the mouse.

Middle mouse wheel: Roll the middle mouse wheel to scroll the control panel up/down or to scan up/down in the file open and file save menus. In the right hand (graph) panel, you can use the middle mouse wheel to expand and contract the xy axis scale.

Middle button: Click and drag or use the rotary mouse button to alter the scale on the graph. You can also use the *Autoscale Plot* button in the control panel to turn on and off the auto-scaling function. The middle button also allows you to view the ~ 3D plots of confidence contours from various angles—simply click on a contour and drag the mouse.

d. Controlling limits of data fitting

These three controls allow you to set and clear boundaries to limit the range of data that are fit. After the *shift-click*, a gray band shows the area of data to be excluded from fitting. This is normally intended for use in fitting to an analytical function to derive estimates of the standard deviation of your data as described in *Section 6.4*. However, it may also be useful in exploring the parameter space by forcing the fit to account for a subset of the data.

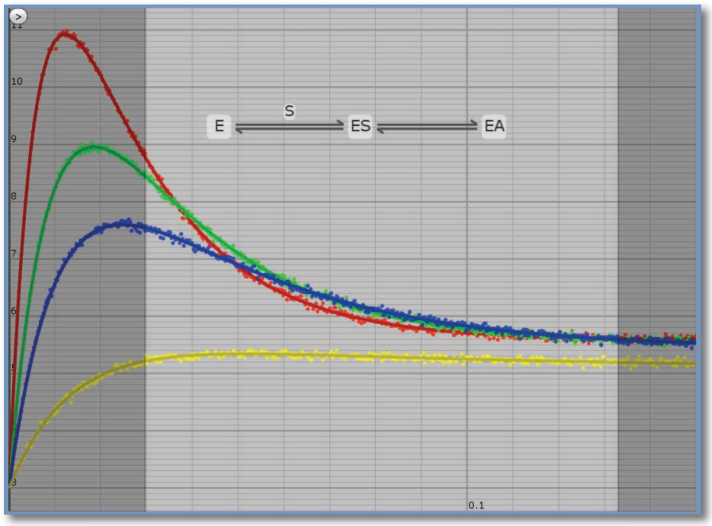
Shift-Left-click: Click on the figure to set a lower limit of data to exclude from fitting.

Shift-Right-click: Click on the figure to set an upper limit of data to exclude from fitting.

Shift-CTRL-click: Clear lower and upper limits.

Limits of data fitting. The darker gray zones show data excluded from fitting, selected by holding the shift key down while clicking on the graph with either the left or right mouse button.

Use *Shift-CTRL-click* to clear the limits and restore data fitting to the full range displayed.



e. Other functions

There are some useful functions accessed by using the *Control* key (Ctrl). Note the even on the MAC, you must use the *Control* key, not the *Command* key.

Click-drag: Zoom in on a section of a graph highlighted by the click-drag action.

Ctrl-W Return to full graph size after zooming in.

Ctrl-D Hide the Control Panel so the graph can be shown full screen; this is useful when using a lower resolution video projector.

Ctrl-B Toggle on/off the display of chi^2 values under each graph.

ESC Hit the escape button to exit from a scrolling operation or text entry and reset the value of the parameter to its starting value.

Ctrl-Q Clear all data from the data repository

Ctrl-R Restore the last saved version of the current file from the disk. Use this function with caution since you will loose all of your current work.

Ctrl-T Toggle on/off the display of simulation steps.

Ctrl-shift-B Toggle the display of bold fonts for scrollable parameters

Ctrl-shift-F Toggle font color for scrollable parameters

You will notice a small icon in the upper left corner of each figure.

 Expand the selected figure to occupy the full graphic space

 Shrink the selected figure back to its original size

f. File extensions used by *KinTek Explorer*

.mec The main mechanism file containing the pathway, rate constants, definitions of experiments and any imported data

.fsp A file containing the results of the *FitSpace* confidence contour analysis; see *Section 8.3.c*

.sim.txt This pair of text files is created when you click on *Export Sim* under the *Experiment*
.dat.txt *Editor*. The *.sim.txt* file contains the data defining the simulation while *.dat.tx*t contains the data. These files can be used to create publication quality graphs. We provide a *MatLab* script that will read these files and create a graph in *eps* format. See *Section 5.13.a*

.fit This is a text file that is created when you click on *Export* after fitting data under the *Data Fit Editor*. The file contains a summary of the fitted parameters, standard error estimates, and covariance matrix derived during nonlinear regression analysis.

.png These are publication graphic output files. The .png files are 300 dpi.

.eps The .eps files are vector graphics that can be edited in Adobe Illustrator and scaled to any desired size.

g. Reserved words and symbols with special meaning

Certain words and symbols have a reserved meaning and cannot be used to define a species.

Logarithmic functions: ***pow, exp, log, ln***; see *Section 5.2c.*

Pointers to concentration, voltage, temperature or pressure: ***CONC, VOLT, TEMP, PRES, TOTAL***.

pH in a mechanism entry specifies the use of pH units rather than [H+].

PATH is used to specify a reaction sequence for the free energy profile.

@ is used to setup an equilibrium titration; see *Section 5.11*.

[ ] are used to specify that the concentration of a reactant does not change; see *Section 5.3*.

> is used to specify a pulse-chase reaction; see *Section 5.10*.

%0 is used to bring the concentration of a species to zero after a mixing step, as in a gel filtration to remove excess ligands; see *Section 5.8*. Starting with Version 6.0, this has been extended allow any concentration, n, with the syntax %n.

[%n] holds the new concentration fixed as in modeling SPR data.

^ is used to specify parallel reactions involving labeled and unlabeled forms of a given reactant; see *Section 4.1c*.

There is one special case: **{ } and (i)** can be used in entering repetitive reactions in the Model Editor. They cannot be used in any other context. *See Section 4.1d.*

Note: In Entering models, in addition to all alphanumeric characters, *only* the following special characters can be used as part of the species names: **# ~ . $:**

1.4 Standard nomenclature

In reference to rate constants in this manual, we will use the standard numbering scheme as illustrated below. Rate constants are numbered sequentially with a positive number for the forward rate constant and a negative number for the reverse rate constant, in the direction written when the model was defined. Equilibrium constants are numbered consecutively as well; for example, $K_1 = k_1/k_{-1} = 1/K_{d,S}$ in the following model.

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} ES \underset{k_{-2}}{\overset{k_2}{\rightleftarrows}} EI \underset{k_{-3}}{\overset{k_3}{\rightleftarrows}} EP \underset{k_{-4}}{\overset{k_4}{\rightleftarrows}} E + P$$

In using the program, no reference to numbered rate constants is needed because the user interface relates rate constants directly to individual steps, except in the *FitSpace* confidence contours for reference to individual rate constants. In addition, in discussing the variables in this manual, we will need to refer to numbered constants.

1.5 Citing KinTek Explorer

Two papers describing the use of the *KinTek Explorer* and *FitSpace Explorer* are published in *Analytical Biochemistry*. Please reference these papers in citing the use of *KinTek Explorer*.

Johnson, K. A., Simpson, Z. B., and Blom, T. (2009) Global Kinetic Explorer: A new computer program for dynamic simulation and fitting of kinetic data. *Analytical Biochemistry* 387, 20-29. http://dx.doi.org/10.1016/j.ab.2008.12.024

Johnson, K. A., Simpson, Z. B., and Blom, T. (2009) FitSpace Explorer: An algorithm to evaluate multi-dimensional parameter space in fitting kinetic data. *Analytical Biochemistry* 387,30-41. http://dx.doi.org/10.1016/j.ab.2008.12.025

A paper describing the use of the program to fit full progress curve kinetic traces and slow-onset inhibition data has been published in *Methods in Enzymology*.

Johnson, K. A. (2009) Fitting Enzyme Kinetic Data with KinTek Explorer *in Methods in Enzymology* 467, 601-626.

Recent examples using *KinTek Explorer* to fit quench-flow and stopped-flow fluorescence data globally:

Kellinger, M. W. and Johnson, K. A. (2010) Nucleotide-dependent conformational change governs specificity of HIV reverse transcriptase. *PNAS* 107, 7734-7739. http://www.pnas.org/content/107/17/7734.full.pdf

Robertson, B. A. et al., (2009) Pre-steady-state kinetic analysis of cis-3-chloroacrylic acid dehalogenase: analysis and implications. *Biochemistry* 48 (49), 11737.

Kellinger, M. W., and Johnson, K. A. (2011) Role of induced-fit in limiting discrimination against AZT by HIV reverse transcriptase. *Biochemistry* 50, 5008-5015. http://www.ncbi.nlm.nih.gov/pubmed/21548586.

Li, A., Ziehr, J.L., and Johnson, K.A. A new general method for simultaneous fitting of temperature and concentration dependence of reaction rates yields kinetic and thermodynamic parameters for HIV reverse transcriptase specificity. J. Biol. Chem. 2017; 292:6695-6702

A recent *Editor's Pick* in the *J. Biological Chemistry* illustrated the power of fitting multiple experiments simultaneously, including fluorescence stopped-flow, equilibrium fluorescence titrations and Isothermal Titration Calorimetry data.

Qian, Y. and Johnson, K. A. The human mitochondrial single strand DNA binding protein displays distinct kinetics and thermodynamics of DNA binding and exchange. J. Biol. Chem. (2017) 292(31) 13068–13084

In these two papers, time resolved spectra were also included in the global fitting:

Schroeder, G. K., Johnson, W. H., Huddleston, J. P., Serrano, H., Johnson, K. A. and Whitman, C. P. (2012) Reaction of cis-3-Chloroacrylic Acid Dehalogenase with an Allene Substrate, 2,3-Butadienoate: Hydration Via an Enamine. *J. Am. Chem. Soc.*, 134, 293-304

Huddleston, J. P., Schroeder, G. K., Johnson, K. A. and Whitman, C. P. (2012) A Pre-Steady-State Kinetic Analysis of the αY60W mutant of *trans*-3-Chloroacrylic Acid Dehalogenase: Implications for the Mechanism of the Wild Type Enzyme. *Biochemistry* 51, 9420-9435

Chapter 2. Overview

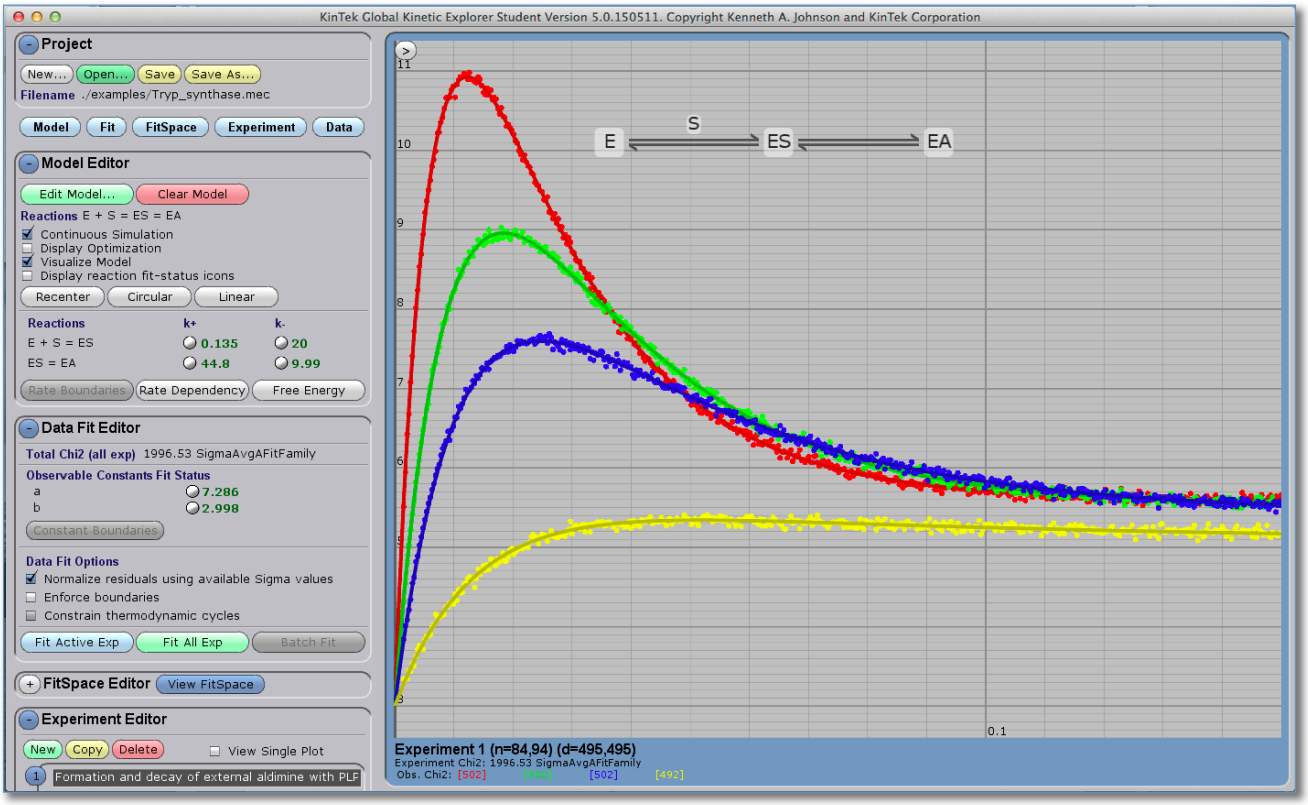


Figure 2.1 *Example: Tryp_synthase.mec*. This figure shows the division of the window into the Control Panel (left) with its six subpanels and the graphic panel (right). The Control Panel can be scrolled up and down to access individual functions and each subpanel can be expanded or contracted by clicking on the + or – next to its name (i.e., *Model Editor*). The *Display Model* shows the current mechanism, rate constants and starting concentrations. Beneath the graph, the numbers display the high and low values for the number of simulated points over a concentration series (n), the number of data points (d) and the chi^2 and individual chi^2 values for each trace.

Control Panel Subdivisions

Project: Open, save and create new projects

Model Editor: Enter model and estimates for rate constants

Data Fit Editor: Fit data and enter values for scaling factors

FitSpace Editor: Examine the space over which parameters can vary

Experiment Editor: Set up each experiment (define starting concentrations and outputs)

Data Repository: Import data, view and fit to analytic functions and assign data to experiments

Note that the subpanels are grouped in the *Control Panel* in an order that optimizes their use. For example, data are fit only after defining the model, the experimental conditions and importing data, but the *Data Fit Editor* is placed just beneath the *Model Editor* so that controls over rate constants will be conveniently located when fitting data. The *Data Repository* is at the end after the *Experiment Editor*, but data can be imported into the *Data Repository* before or after developing a model.

Pressing these buttons caused the control panel to jump to the corresponding submenu.

2.1 Program flow chart

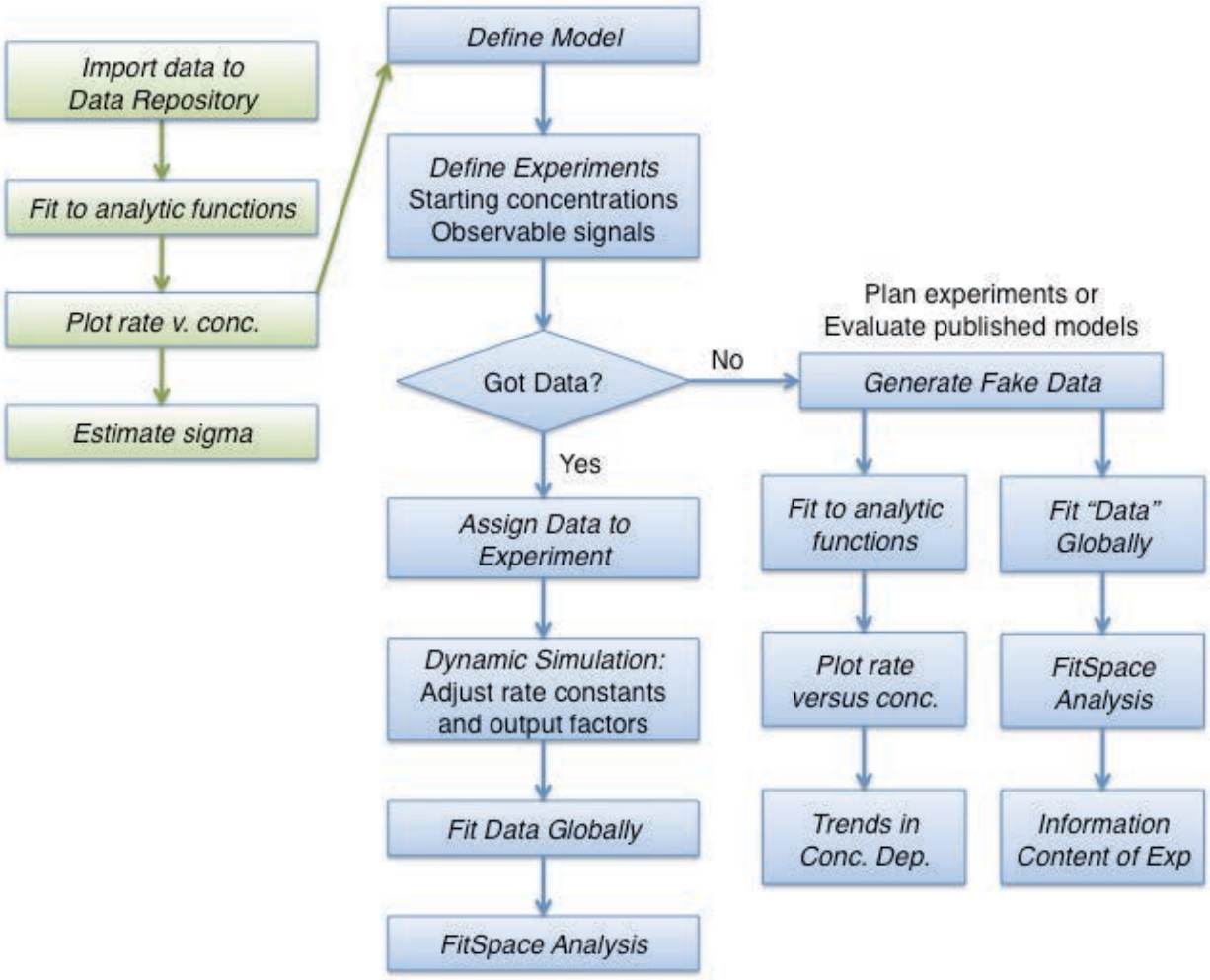


Figure 2.2 *Program Flow Chart* This diagram serves as an introduction to the flow of the program and the order in which various actions must or can be performed. The importing and fitting of data using the *Data Repository*, new in version 2.2, is shown in the leftmost column (*green*), which enables importing of data independent of defining the model. The sections in *blue* define the normal path of performing simulation by first defining a model and experiments (initial concentrations and observable signals) and then entering rate constants and scaling factors to generate an output. Data can then be assigned to a given experiment and then fitted directly to the model to derive kinetic parameters.

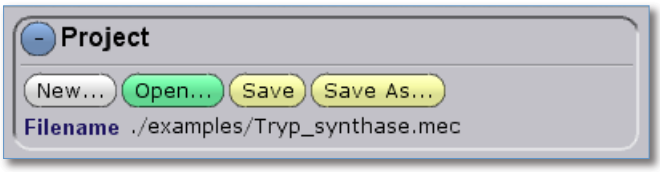
See *Section 7.3* for more details on the steps in fitting data. After a model and experiment are defined, data can be assigned directly to the experiment. Alternatively, if previously imported into the *Data Repository*, data can then be assigned to a given experiment for fitting. Data that have been imported into the *Data Repository* can be fit to analytic functions and plots of observed rate versus concentration can be generated. This is useful to see the patterns in the data that reveal the underlying model. For data to be fitted directly to the model based upon numerical integration, the data must be assigned to an experiment, under the *Experiment Editor*.

The use of the program to generate synthetic data can be quite valuable for learning kinetics, planning experiments, or evaluating published models. You will be shocked at how frequently published models do not reproduce the primary data!

Chapter 3. Project Menu

Use the Project panel to:

- Open and Save Projects
- Create a new Project



New... Create a new project by clearing all of the current model and settings. The data in the *Data Repository* remains in the mechanism file.

Open... Click to open an existing mechanism file (extension *.mec*). Use a single click to navigate to the directory containing mechanism files and the select the desired file. You can also use the middle mouse wheel (or right-click and drag) to scroll up and down the listings. Then click OK to open the file. If the *examples* folder is not visible, find the directory *KinTek_Explorer_Pro/examples*. Navigate the menu by scrolling up and down with the center mouse wheel or with a right-click-drag. The default mechanism file extension is .mec. The figure below shows the Mac OSX menu, allowing use of the efficient Mac OSX directory search engine. A native file dialog is also implemented for Windows.

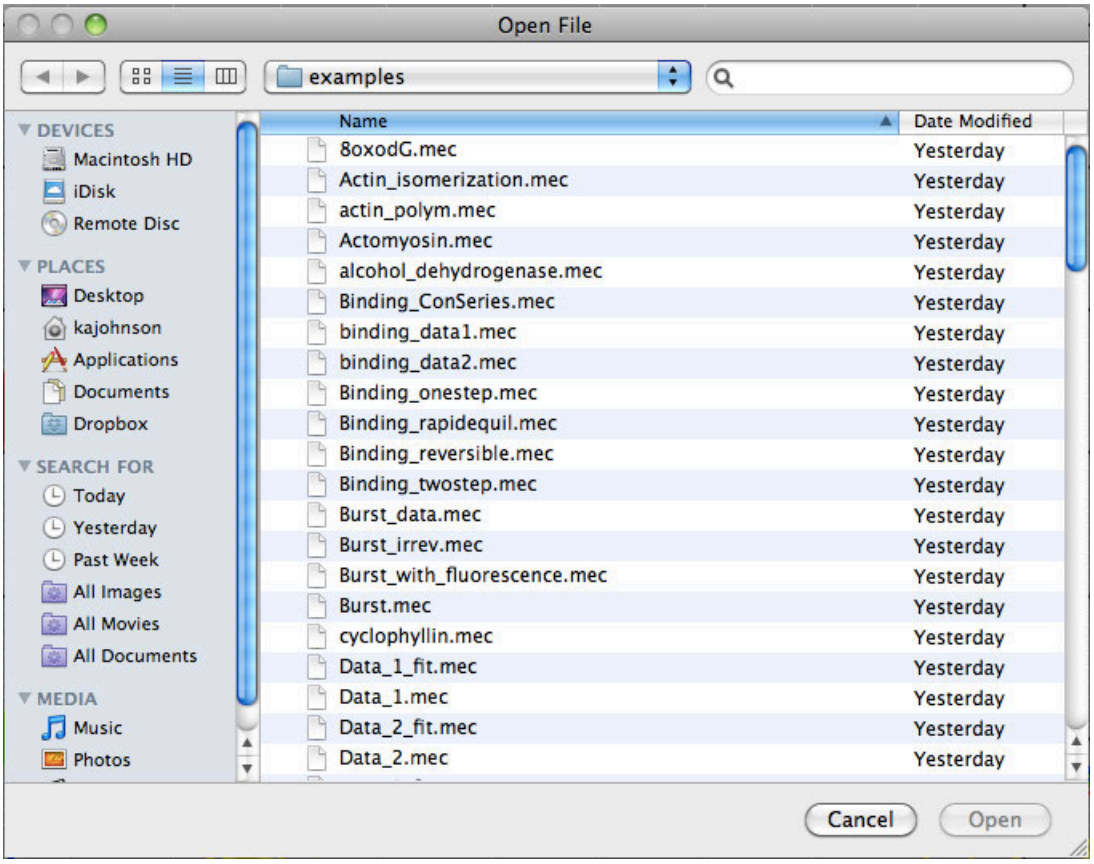


Figure 3.1 *Open File menu*. Native file open/save dialogs are provided for both Mac OSX and Windows.

Save Saves the current mechanism file (*.mec).

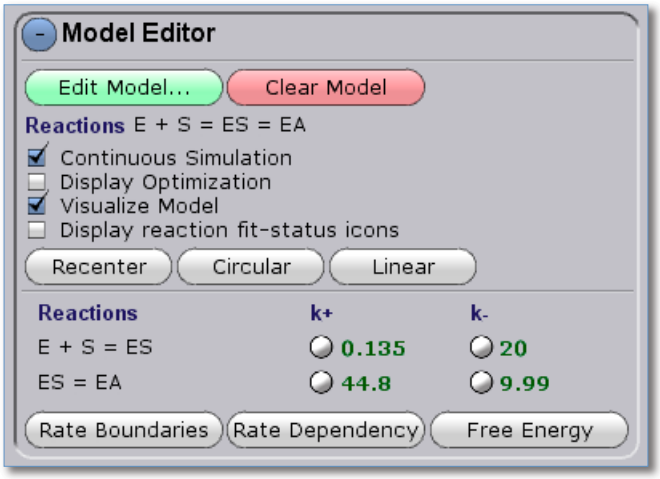
NOTE: Data in the Data Repository or assigned to an experiment will be saved with the mechanism file.

Save As... Saves the current mechanism file under a new name and/or in a different directory.

Chapter 4. Model Editor

Under Model Editor you will:

- Edit or define a model
- Control the *Display Model*
- Enter and scroll rate constants
- Constrain rate constants
- Select display optimization
- Setup voltage- or temperature dependence of rate constants
- Enter Upper and Lower rate boundaries
- Display the free energy profile



4.1 Entering the model

The Model Editor contains functions to edit existing models, create new models, and to enter and control rate constants.

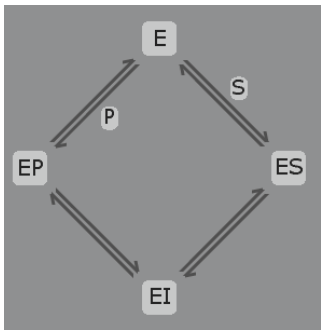
Click to clear the existing model. *Caution:* this deletes all experiments and kinetic parameters, but retains the data in the Data Repository.

Click to edit an existing model or create a new model if none exists. This command opens a text editor into which you can enter the reaction sequence with an = sign to designate a reaction. After editing the model, click *Create* to save changes, or *Cancel* to abandon changes.

For example the reaction:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} ES \underset{k_{-2}}{\overset{k_2}{\rightleftarrows}} EI \underset{k_{-3}}{\overset{k_3}{\rightleftarrows}} EP \underset{k_{-4}}{\overset{k_4}{\rightleftarrows}} E + P$$

is entered as: E + S = ES = EI = EP = E + P



NOTE: It is important that you enter the major species (usually the enzyme) first each reaction. This is necessary for the short-hand notation as illustrated on the right with the minor species displayed on the arrow connecting major species. This is also used in constructing the free energy profile.

As you type, the program will continuously parse the model and warn you of syntax errors. The reaction can continue on one line as long as mass balance is maintained. More complex pathways such as EPSP synthase require multiple lines to maintain mass balance:

E + A = EA
EA + B = EAB = EI = EPQ = EQ + P
EQ = E + Q

Note the error in the following, unless you intended to indicate that B was produced when A reacted with E: E + A = EA + B = EAB (this implies that E + A → EA + B).

a. Species names

The names given for reactants are *case sensitive* and may contain any unique set of alphanumeric characters except the reserved words.

Reserved words: The following words cannot be used since they are reserved for using exponential functions in the output expressions (see *Section 5.2.c*): *pow, exp, log, ln*

In addition, the following special characters can be used, *but no others*: **# ~ . $:**

These characters can be used to help distinguish complex terms such as E.D.dNTP or E.NADPH.DHF, or E.Tyr~AMP.PP. However, you should keep the names as short as possible while using terms that you recognize easily. The EPSP synthase pathway could have been written:

E + S3P = E.S3P
E.S3P + PEP = E.S3P.PEP = EI = E.EPSP.P = E.EPSP + P
E.EPSP = E + EPSP

Note that because the species names are case sensitive, ES and Es would be considered as different species.

Reactions involving the simultaneous collision of three species are not allowed. For example it may be common to write the following enzyme-catalyzed reaction when the order of product release is not known:

E + S = ES = EPQ = E + P + Q

However, that implies the reverse reaction involves the simultaneous collision of P and Q with the enzyme, which is a highly unlikely event. We enforce reasonable reactions at all steps and preclude the entry of such a model because we encourage the fitting of reactions in the forward and reverse directions simultaneously. This reaction sequence can be entered instead as:

E + S = ES = EPQ = EP + Q
EP = E + P

If the order of product release is not known, simply make the rate constant for release of P faster than that for release of Q, so that kinetically both steps occur at a single rate (until or unless you have data to the contrary). You can specify a random order of product release by adding two additional lines, but note that you will have then created a thermodynamic cycle with two energetically equivalent routes from EPQ to E:

E + S = ES = EPQ = EP + Q
EP = E + P
EPQ = EQ + P
EQ = E + Q

b. Species order for bimolecular reactions

Starting with Version 5.0 the order in which you enter two reactants matters because it affects the *Model Display* and the *Free Energy Profile*. We have adopted the following syntax: The second species in the sequence is considered to be the small molecule ligand (or more generally, the species that is in excess). This enables us to make the simplification, for example, that the reaction of E to ES above occurs with a pseudo-first-order rate constant given by the product k_1[S]. We then simplify the Model Display by placing the concentration of S above the arrow as:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \quad \text{becomes:} \quad E \underset{k_{-1}}{\overset{k_1[S]}{\rightleftharpoons}} ES$$

alternatively with enzyme in excess, the rate of disappearance of S is governed by k_1[E]:

$$S + E \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \quad \text{becomes:} \quad S \underset{k_{-1}}{\overset{k_1[E]}{\rightleftharpoons}} ES$$

c. Labeled/Unlabeled Reactants

Starting with Version 6.0, we have made it easier to enter parallel reactions involving labeled and unlabeled variants of the same reactant. For example, in the file *ribozyme.mec (Experiment 2)*, we model the dissociation of a fluorescently labeled oligonucleotide from a ribozyme driven by competition with an unlabeled oligonucleotide. In the original file, we entered two pathways, one for labeled species and one for the unlabeled competitor and then linked the corresponding rate constants to be identical for labeled and unlabeled species as shown below.

Model: ribozyme.mec (old style)
E + S = ES = FS
E + C = EC = FC

Reactions	k+	k-
E + S = ES	1 3.86	3 0.46
ES = FS	2 2.47	4 0.0211
E + C = EC	1 3.86	3 0.46
EC = FC	2 2.47	4 0.0211

In the new system, you can use the special symbol ^ to specify species that exist in both labeled and unlabeled forms, with rate constants automatically linked.

Note: The ^ symbol must be the last character in the string, immediately after the species name.

Model: ribozyme^.mec
E + S^ = ES^ = FS^

Reactions	k+	k-
E + S^ = ES^	3.89	0.496
ES^ = FS^	2.5	0.02
E + S = ES	3.89	0.496
ES = FS	2.5	0.02

Note that in this case, the corresponding rate constants are automatically linked, but you do not need to enter the linkage specifically.

This model entry is also useful for pulse-chase experiments, such as dynein_burst.mec.

Model: dynein_burst^.mec
E + A^ = EA^ = EP^ = E + P^

Reactions	k+	k-
E + A^ = EA^	4	0
EA^ = EP^	64	18
EP^ = E + P^	8.34	0
E + A = EA	4	0
EA = EP	64	18
EP = E + P	8.34	0

d. Entering Repetitive Reaction Sequences

We have developed a special syntax to allow easy entry of repetitive reactions. For example for processive DNA polymerization, at each step of the reaction the DNA can either dissociate from the enzyme or the enzyme-DNA complex can bind nucleotide and react to extend the DNA by one residue. In general we would write the reaction sequence as:

$$E + D_i \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ED_i$$

$$ED_i + N \underset{0}{\overset{k_2}{\rightleftharpoons}} ED_iN \underset{0}{\overset{k_3}{\rightleftharpoons}} ED_{i+1} + PP$$

$$E + D_{101} \underset{k_{-end}}{\overset{k_{end}}{\rightleftharpoons}} ED_{101}$$

where k1, k2 and k3 represent the rate constants for the first, second and third steps in the sequence. These rate constants and the reaction sequence are then repeated for the number of time specified, 100 in this example. We have developed a special syntax to specify a repetitive reaction as illustrated in this example:

{E + D(i) = ED(i)
ED(i) + N = ED(i)N = ED(i+1) + PP}(i=1,100)
E + D101 = ED101

This produces a display of the rate constants as shown below. Note that all reactions in the repetitive sequence are linked. That all of the reactions involving DNA binding are in linkage group ***a***, all for DNA dissociation are in linkage group ***b***, etc. Any changes to linkage groups made in the shorthand notation apply to all member of the groups originally defined when the repetitive reaction sequence was entered, that is the terms within the curly brackets { }. If you wish to break up these linkage groups, you can do in from the complete reaction sequence view:

Shorthand notation:

	Reactions	k+	k-
1	E + D1 = ED1	a 12	b 0.1
2	ED1 + N = ED1N	c 10	d 0
3	ED1N = ED2 + PP	e 30	f 0
301	E + D101 = ED101	12	0.1

This view is shorthand notation for the actual reaction sequence shown at the right, which is truncated after the first four reactions in this display. The shorthand notation is much easier to work with. The full reaction sequence is needed only if you wish to change individual rate constants.

Complete reaction sequence:

Display repeated reaction UI

Recenter | Circular | Linear

	Reactions	k+	k-
1	E + D1 = ED1	a 12	b 0.1
2	ED1 + N = ED1N	c 10	d 0
3	ED1N = ED2 + PP	e 30	f 0
4	E + D2 = ED2	a 12	b 0.1
5	ED2 + N = ED2N	c 10	d 0
6	ED2N = ED3 + PP	e 30	f 0
7	E + D3 = ED3	a 12	b 0.1
8	ED3 + N = ED3N	c 10	d 0
9	ED3N = ED4 + PP	e 30	f 0
10	E + D4 = ED4	a 12	b 0.1

Note: Simulating 3 reactions x 100 repeats takes time, so you must be patient in waiting for the results to appear on the screen. It is recommended that you turn off the continuous simulation under the model editor: ☐ Continuous Simulation. In this mode, the simulation is performed only after you complete an entry rather than continuously as you scroll a rate constant with the mouse.

As noted below, if you make changes to the linkage groups while in the shorthand notation, they apply to all of the members of that group. In this example, $k_{-2} = 0$ and $k_{-3} = 0$ are locked and that propagates to all of the reactions involving nucleotide release and the reverse of chemistry.

Shorthand notation:

Display repeated reaction UI

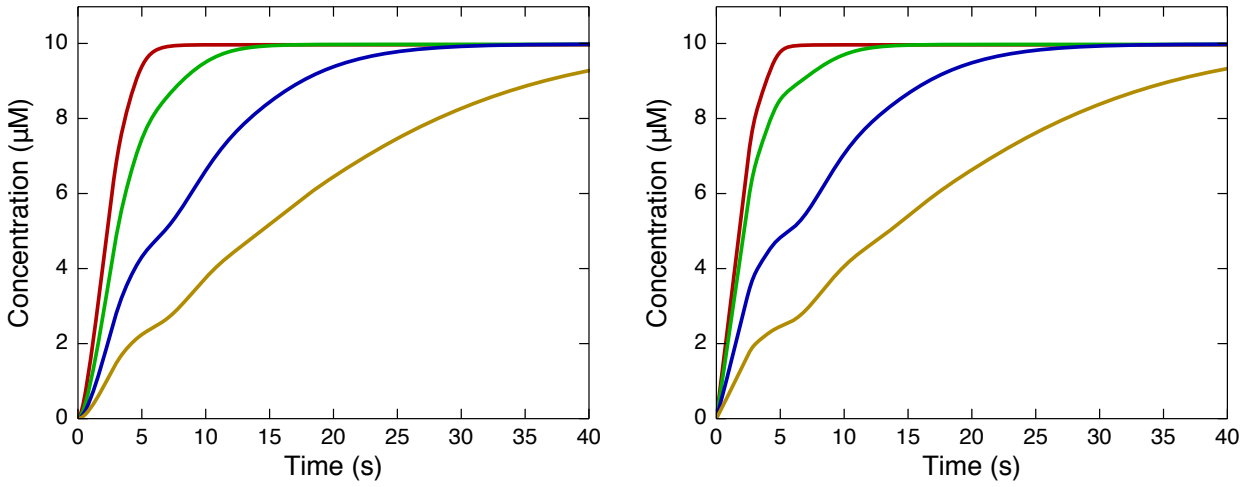
Reactions	k+	k-
E + D1 = ED1	a 12	b 0.1
ED1 + N = ED1N	c 8	0
ED1N = ED2 + PP	e 30	0
E + D101 = ED101	10	1

Note: While in the shorthand notation mode, changes to the rate constants and linkage groups are applied to all of the reactions in the sequence.

Complete reaction sequence:

Display repeated reaction UI

Reactions	k+	k-
E + D1 = ED1	a 12	b 0.1
ED1 + N = ED1N	c 8	0
ED1N = ED2 + PP	e 30	0
E + D2 = ED2	a 12	b 0.1
ED2 + N = ED2N	c 8	0
ED2N = ED3 + PP	e 30	0
E + D3 = ED3	a 12	b 0.1
ED3 + N = ED3N	c 8	0
ED3N = ED4 + PP	e 30	0
E + D4 = ED4	a 12	b 0.1
ED4 + N = ED4N	c 8	0
ED4N = ED5 + PP	e 30	0



Example simulations. *Left:* Example with variable [E], starting the reaction by adding enzyme. Note the lag at the beginning of the reaction due to E binding. *Right:* Example with preincubation to form the ED1 complex, then adding nucleotide to start the reaction. Simulations were performed with [D1] = 0.1 μM, [N] = 100 μM, and [E] = 0.2, 0.1, 0.05 and 0.025 μM and the rate constants listed above. The Y-axis denotes concentration of PP formed. Note the inflection in the data, which results when the polymerase reaches the end of the template strand and must dissociate to rebind additional templates.

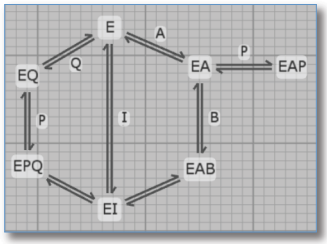
Note: Because steady state data of this type do not provide information to resolve steps after nucleotide binding and before product release, we model the reaction as occurring in two irreversible steps so that $k_2 = k_{cat}/K_m$ and $k_3 = k_{cat}$. It need not be true that either of these reactions is irreversible for this approximation to be valid. See the section on modeling steady state kinetics for more explanation.

e. Display Model control

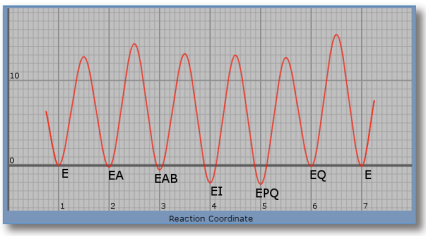
Recenter Circular Linear Click to re-center the *Display Model*, arrange the species in a circular, or linear arrangement, respectively. With complex models, the species will need to be rearranged manually. *Left-click and drag to arrange species. Right-click and drag to move entire model.* The model entry (order of reactants as described above) affects the *Display Model* and the *Free Energy* Profile as illustrated in the example below, from *EPSP.mec.* The second species in a bimolecular reaction is considered to be the ligand (or the species in excess) and is then placed on the arrow connecting major species as shown here.

E + A = EA
EA + B = EAB = EI = EPQ = EQ + P
EQ = E + Q
EA + P = EAP
E + I = EI

Model as entered. Note that species A, B, I, P and Q are listed second in each reaction to designate them as ligands.



Display Model. Note that species A, B, I, P and Q are designated as ligands.



Free Energy Profile (simplified by removing E + I = EI and EA + P = EAP using the *Set Reactions* command.

Other options under the Model Editor are:

☑ Continuous Simulation: When "Continuous Simulation" is checked, the output of the simulation is continuously updated as you scroll a rate constant, concentration or output factor. You may want to turn this function off when defining a new experiment or fitting data, but generally it can be left on.

☑ Display Optimization: This checkbox is intended to overcome display problems on some Windows machines. Unless you hare experiencing problem, there is no need to use this function.

☑ Visualize Model: This checkbox controls whether the reaction model is displayed.

☑ Display reaction fit-status icons: This checkbox controls whether the small dot reaction status indicators are shown on the Display Model (see *Visualize Model*). This is useful to see which reaction rates are fixed and which are linked.

☑ Display repeated reaction UI: This checkbox is only used when the shortcut for entering repeated reactions is used. See *Section 4.1d.*

f. Editing an existing model

Be careful to save your current mechanism file before making changes to the model if you have already programmed experiments with defined outputs. When you alter an existing model, the program has to adapt many of the tables for entry of rate constants and starting concentrations to the new model. If rate constants have already been entered, the program will attempt to match up previous rates with the new model, but you must carefully review the tables.

If you delete a species from the reaction mechanism that is already used as part of a defined observable expression, the new output expression will contain the deleted species as an output factor. For example, suppose you start with the tryptophan synthase model: E + S = ES = EA with an output express of a*ES+b. The program recognizes **a** and **b** as output factors because they are not on the list of species. However, if you change the model to E + S = EA, the program will now interpret your output express by assuming ES is an output factor. *After editing an existing model, always check the list of output factors.*

If there are *syntax errors* in the output expression, it will be flagged as an error with yellow text

under observables in the *Experiment Editor*. The output must be edited to continue. See additional information below on the definition of output functions. If you intend to delete a species that is part of a defined observable output, it is safer to first remove that species from the output expression before deleting the species from the model.

4.2 Entering rate constants

Enter rate constants by putting the cursor over the red/green arrows connecting species, or on the table in the control panel under the Model Editor. Click and drag to scroll values for rate constants up/down, or click and release to enter a new rate.

a. Units

The program does not keep track of units, so you must be consistent. All rate constants need to be in the same units of time as the data you may be fitting, i.e., s^{-1} or m^{-1}. Second-order rate constants must be in the same units of concentration as the concentrations of species used in the mechanism.

We recommend use of time in seconds and concentrations in micromolar, so all second order rate constants are in units of $\mu M^{-1}s^{-1}$. Most second order rate constants are between 0.01 and 100 $\mu M^{-1}s^{-1}$, although an upper limit for diffusion may be as high as 1000 $\mu M^{-1}s^{-1}$. First order rate constants relevant to enzyme catalysis are typically in the range of 0.001 to 10,000 s^{-1}, but can exceed this range.

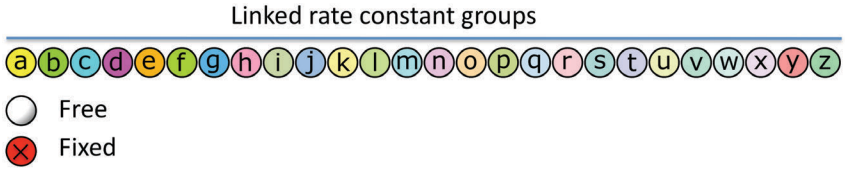
Use of rate constants larger than 10,000 s^{-1} often means that the model is overly complex and the fast step contributes nothing to the observable kinetics, unless you have data collected on the microsecond time scale. In some cases it may be that you only can model a step as a rapid equilibrium or only place a lower limit on a given rate constant. In other circumstances the model may need to be simplified. See (2,3) for more information.

b. Round-off errors

It is best to keep time, rate constants and concentrations between the values of $1x\ 10^{6}$ and $1x10^{-6}$ to avoid round-off errors that can affect fitting when using very large or very small numbers. We recommend using units of concentration that allow numbers close to unity. For example, it is usually inconvenient and inaccurate to use M concentration units. Rather, for most work, units of µM are preferred.

c. Linked or fixed rate constants

Often in fitting, you want to maintain the ratio of two or more rate constants at a fixed value and then change the rates in unison while maintaining that constant ratio. This is particularly important to maintain the equilibrium constant while searching for an optimal or minimal rate. The tool is useful before, during and after fitting. Control is achieved using the following set of symbols adjacent to the rates on the table under *Model Editor*. ***Note:*** Starting with Version 6, we changed from numbered to lettered groups to increase the number of options; in parts of this manual, examples still include numbered groups.



Click on the control to sequentially step through the 28 options. Right-click clears the selection, returning to the *Free state*, meaning that the rate constant vary independently when fitting data. Rates can be grouped into one of 26 sets by selecting one of the lettered symbols listed above. At the time the rate is added to a group, all rates in that group become linked by a constant ratio defined by their values when they are added to the group. For example, for EPSP synthase there are three grouped rates:

$k_3/k_{-3} = 1630/136 = 12$

$k_4/k_{-4} = 351/263 = 1.33$

$k_5/k_{-5} = 83.7/83.7 = 1\ \mu M$

Note that k_5 is first order, but k_{-5} is a second order rate constant. The units on the K_d for Q derive from knowledge kept by the user, not the program.

Reactions	k+	k-
E + A = EA	⊗ 650	⊗ 4500
EA + B = EAB	⊗ 15	○ 284
EAB = EI	ⓐ 1630	ⓐ 136
EI = EPQ	ⓑ 351	ⓑ 263
EPQ = EQ + P	○ 88.6	⊗ 0.06
EQ = E + Q	ⓒ 83.7	ⓒ 83.7
EA + P = EAP	○ 0.113	⊗ 100
E + I = EI	○ 45.8	⊗ 0.01

These constant ratios provide important constraints in fitting the data by including information to define the equilibrium constants for these 3 steps of the reaction.

NOTE: The constant ratio applies both when scrolling the rate constants and in fitting the data.

In order to hold a given rate constant fixed during fitting, select the ⊗ symbol next to the rate constant. The rate constant can still be scrolled or changed by text editing, but it will be held fixed at the value when fitting by nonlinear regression. This is useful when a rate constant is either known based upon data that is not included or is not known or determined by the data. For example, the second order rate constant for diffusion-limited substrate binding can be fixed at a reasonable upper limit (100 to 1000 $\mu M^{-1}s^{-1}$) in order to fit equilibrium constants as described below. In other cases, reverse rates of reaction are often set to zero because there is no information to define the reverse rate. In either case, it is important to explore the extent to which your assumed rate constants affect the fitting of other parameters and to clearly state your approximations when publishing your results.

d. Setting upper and lower limits

When initially attempting to fit data where certain rate constant are not well constrained by the data it is often seen that rate constants go to extreme values that are unrealistic. Under these circumstances it is desirable to set upper and lower limits on each of the rate constants to prevent boundaries on the possible extreme values. For example, second order rate constants always have an upper limit defined by the rate of diffusion-limited collision of approximately $10^9\ M^{-1}s^{-1}$ ($1000\ \mu M^{-1}s^{-1}$). These rate constant boundaries are enforced during data fitting if selected.

☑ Enforce boundaries: Under Data Fit Options this checkbox controls whether the upper and lower boundaries are enforced during data fitting. *Boundaries are not enforced unless this box is checked*.

Note: Rate constant boundaries are turned off during *FitSpace* calculations to avoid misleading chi2 boundaries.

Rate Boundaries Click on *Rate Boundaries* to set upper and lower limits for each rate constant.

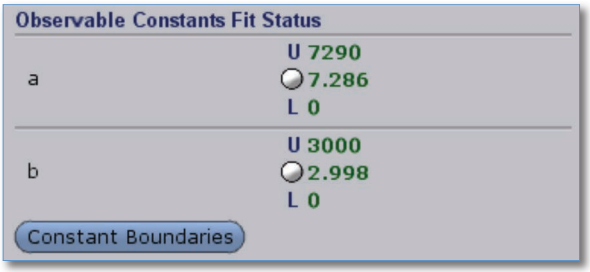
This will open the rate constant entry to include Upper (U) and Lower (L) limits for each rate constant. The default upper and lower limits are set to 1000 x the rate constant and zero, respectively.

Reactions	k+	k-
E + S = ES	U 135 ○ 0.135 L 0	U 20000 ○ 20 L 0
ES = EA	U 44800 ○ 44.8 L 0	U 9990 ○ 9.99 L 0

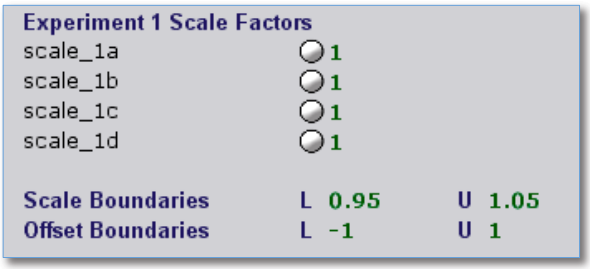
Rate Boundaries | Rate Dependency | Free Energy

Rate Boundaries Once the desired values are set, click *Rate Boundaries* again to accept the changes.

Constant Boundaries A similar set of upper and lower limit boundaries can be set for the output signal scaling constants under the *Data Fit Editor,* accessible via the *Constant Boundaries* button.



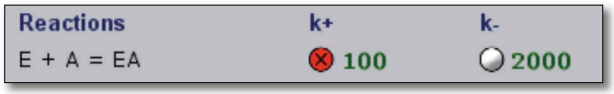
If you select the option to normalized traces within a given experiment by either a *Concentration Series Scaling Factor* or *Offset* under the *Experiment Editor* (*Section 5.7*), you can also place upper and lower limits of the magnitude of the offset or multiplier. These limits should be relatively small because the normalization is intended to correct for small fluctuations in signal intensity from one trace to the next. **Note:** One set of limits applies to all scaling factors.



e. Equilibrium constants

We do not support the explicit fitting of an equilibrium constant because such fitting requires that the reactions be assumed to be infinitely fast, and that is never the case. Often a step in a sequence is thought to be fast enough relative to other rates that it is essentially at equilibrium and there is no information to define the rates. Under these circumstances, simply make the rates faster than neighbors in the reaction sequence. Use of the linked rate constants and dynamic simulation is useful to explore how fast the rates need to be in fitting of the parameters to the data and the chosen model. There are two options when fitting equilibrium constants to a dataset, as illustrated below by fitting a binding constant.

Case 1. You do not know the equilibrium constant, but believe it is rapid. In this example, set the binding rate at the diffusion limit (nominally 100 to 1000 $\mu M^{-1}s^{-1}$), and let the dissociation rate float during fitting to determine the $K_d = k_{-1}/k_1$. This is accomplished as shown below, using the fixed rate function ⊗ to freeze the binding rate, k_1, in this model where K_d = 20 µM. This analysis will return the error estimates for k_{-1}. The percentage error on the K_d is equal to the percentage error on k_{-1}.



Case 2. You think you know the K_d, but want to fit to see how fast the binding and dissociation rates must be to account for the data. In this example, group the binding and dissociation rates so that they are constrained to give the known K_d, and then fit the data allowing both rates to float in unison (in this case k_{-1}/k_1 = 20 µM). In the example shown below, k_1 and k_{-1} belong to group 1 to maintain the constant ratio. FitSpace explorer can then be used to determine the lower limit on the values of k_1 and k_{-1}.

f. Modeling steady state kinetic data

In fitting steady state kinetic data there is not enough information to define all of the rate constants in a minimal pathway. One can only extract values for k_{cat} and K_m from the data. Therefore, one either needs to fix a subset of rate constants at values that represent reasonable estimates so that only two rate constants are varied in fitting the data, or allow all rate constants to float in fitting the data. By either method, one can calculate correct values for k_{cat} and K_m but only the former provides estimates of errors on the parameters. In either case, *KinTek Explorer* can be used to fit steady state kinetic data consisting of a series of initial velocity measurements (signal versus time for a series of concentrations) or full progress curve kinetics, in the presence or absence of added products or inhibitors. The analysis is much more robust and straightforward than conventional methods based upon estimation of initial velocities (involving fitting the data twice through the plot/replot methods). We begin with a minimal model:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftarrows}} ES \underset{k_{-2}}{\overset{k_2}{\rightleftarrows}} EP \underset{k_{-3}}{\overset{k_3}{\rightleftarrows}} E + P$$

As one approach, one can model steady state kinetic data by solving for all six rate constants. The individual rate constants will not be known with certainty and a large number of sets of six rate constants can be found to fit the data. Nonetheless, one can compute k_{cat} and K_m from any one of the parameter sets. For the minimal model shown above:

$$k_{cat} = \frac{k_2 k_3}{k_2 + k_{-2} + k_3}$$

$$K_m = \frac{k_2 k_3 + k_{-1}(k_{-2} + k_3)}{k_1(k_2 + k_{-2} + k_3)}$$

$$k_{cat} / K_m = k_1 \frac{k_2 k_3}{k_2 k_3 + k_{-1}(k_{-2} + k_3)}$$

Note that any parameter set can be used to compute k_{cat} and K_m, but one cannot derive error estimates on k_{cat} and K_m because of the large errors on the individual rate constants, which are not well determined by the data. The model contains more unknowns than can be defined by the data. Thus, the model must be simplified to contain only two unknowns in order to obtained reliable error estimates for k_{cat} and K_m.

One simple approach is to fit the data to a rapid equilibrium model with substrate binding rates fixed at the diffusion limit and then compute k_{cat} and K_m (2,3). The fitting does NOT imply one knows that chemistry is rate-limiting in the steady state or that substrate binding is in rapid equilibrium; rather, this simplification only allows computation of k_{cat} and K_m by fitting based upon a simplified model. Because of the inherent limitations in steady state data, it does not matter whether the explicit values of the rate constants are accurate. The simplifications are only used to compute k_{cat} and K_m.

$$E + S \underset{k_{-1}}{\overset{(1000\ \mu M^{-1}s^{-1})}{\rightleftarrows}} ES \underset{0}{\overset{k_2}{\rightleftarrows}} EP \underset{0}{\overset{k_3 >> k_2}{\rightleftarrows}} E + P$$

$$k_{cat} = k_2; \quad K_m = k_{-1} / 1000$$

A better approach is often to model the data to derive values for k_{cat}/K_m and k_{cat}, with the realization the K_m is simply a ratio of k_{cat} divided by k_{cat}/K_m. This method provides the most accurate estimate for k_{cat}/K_m, the specificity constant, which is the most important steady state kinetic parameter. This is especially useful when the data do not define k_{cat} very well; for example, when concentrations of substrate sufficient to saturate the enzyme cannot be achieved.

$$E + S \underset{0}{\overset{k_1}{\rightleftarrows}} ES \underset{0}{\overset{k_2}{\rightleftarrows}} EP \underset{0}{\overset{k_3 >> k_2}{\rightleftarrows}} E + P$$

$$k_{cat} / K_m = k_1; \quad k_{cat} = k_2; \quad K_m = k_2 / k_1$$

Finally, if the reaction is readily reversible and product inhibition significantly to the kinetics, such as when fitting full time course kinetic data, one must include a more comprehensive model. Depending on the data and the underlying kinetic parameters, it is sometimes possible to derive values for k_{cat} and K_m in both the forward and reverse directions. See *Progress_curves.mec* for an example. Contrary to claims in the literature (4,5), one cannot fit full progress curve kinetic data to derive rates of formation and decay of enzyme-bound intermediates (2,3).

$$E + S \underset{k_{-1}}{\overset{(1000\ \mu M^{-1}s^{-1})}{\rightleftarrows}} ES \underset{k_{-2}}{\overset{k_2}{\rightleftarrows}} EP \underset{(1000\ \mu M^{-1}s^{-1})}{\overset{k_3}{\rightleftarrows}} E + P$$

Forward reaction: $k_{cat} = \dfrac{k_2 k_3}{k_2 + k_{-2} + k_3}; \quad K_m = \dfrac{k_{-1}}{1000}$

Reverse reaction: $k_{cat} = \dfrac{k_{-1} k_{-2}}{k_{-1} + k_{-2} + k_{-2}}; \quad K_m = \dfrac{k_3}{1000}$

4.3 Voltage- and Temperature-Dependent Rate Constants

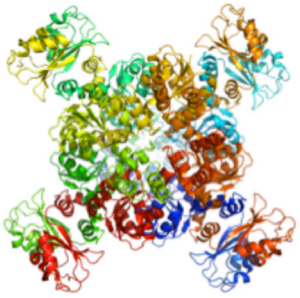
New with Version 4.0, we have included the ability to fit data using voltage- or temperture-dependent rate constants. For example, if you have a serious of experiments covering a range of temperatures, you can fit the family of curves to a single set of rate constants at a reference temperature and a corresponding set of activation energy terms to cover the range of temperatures. This fitting affords the minimum number of unknown parameters defined as by the data. In addition, temperature-jump experiments can be interpreted without the typical simplifications normally employed.

Voltage-dependent rate constants can be used to fit data collected on voltage-gated ion channels, for example. Setting up your model to include voltage- or temperature dependent rate constants is done through the *Model Editor*.

To enable voltage or temperature-dependent rate constant, click on "*Rate Dependencies,"* then select the Temperature or Voltage checkbox as described below. Future releases will include the option for Pressure-jump experiments as well.

a. Voltage-dependent rate constants

The example file "voltage_dependent_rates.mec" illustrates the methods used for data fitting based upon the model published by Smith-Maxwell et al, (6) describing a voltage-gated potassium ion channel. The figure at the right shows the tetrameric structure of a voltage-gated ion channel. Each of the four subunits must undergo a voltage-dependent change in structure before the channel opens leading to a measurable change in current. The system is first equilibrated at a resting voltage, then a rapid jump in voltage leads to an increase in current as the channels open. The lag in the current recordings suggests a multistep sequence leading channel opening according to the model described below with four voltage-dependent rates.


1lnq.pdb

The model uses the simplifying assumption that each monomer has an intrinsic rate of undergoing a structural transition that depends on the voltage. That is, if the rate of the reaction of C1 to C0 is α, then the reaction of C4 to C3 is 4α, the rate of C3 to C2 is 3α, and the rate of C2 to C1 is 2α. In the model, α and β are the intrinsic rate constants for each monomer. There is an additional voltage-independent step.

$$C4 \underset{\beta}{\overset{4\alpha}{\rightleftharpoons}} C3 \underset{2\beta}{\overset{3\alpha}{\rightleftharpoons}} C2 \underset{3\beta}{\overset{2\alpha}{\rightleftharpoons}} C1 \underset{4\beta}{\overset{\alpha}{\rightleftharpoons}} C0 \underset{\delta}{\overset{\gamma}{\rightleftharpoons}} O$$

Rate constants vary exponentially with voltage, where *z* is the *Partial Gating Charge* for the structural transition in each subunit.

$k_{+i} = k_{+i,0} e^{zFV/RT}$ for forward (activation) rate constants and

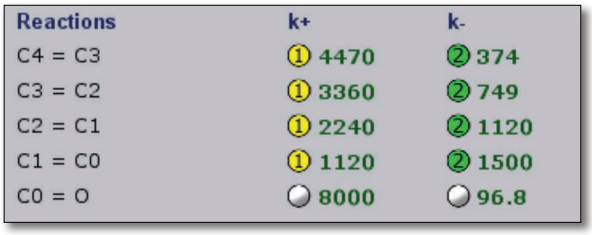
$k_{-i} = k_{-i,0} e^{-zFV/RT}$ for reverse (inactivation) rate constants.

Note: We use a Faraday constant = 0.096485 kJ/mv/gram according to the convention in the field to enter voltage in millivolts and refer to energy in kJ.

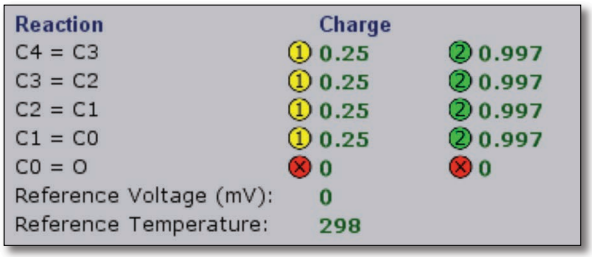
Stepwise Instructions:

Here I give the instructions for setting up the simulation of the model contained in the mechanism file, "voltage_dependent_rates.mec to illustrate the methods.

1. Enter the model as: C4 = C3 = C2 = C1 = C0 = O with the rate constants constrained by the 4, 3, 2, 1 ratios as shown on the right. Values of 4, 3, 2, and 1 are entered for k_{+1}, k_{+2}, k_{+3} and k_{+4}, respectively, then by clicking on the circle they are linked into group 1. The reverse rate constants are linked into group 2.



2. The values are then set to the estimates of k_{+4} = 1020 s^{-1} and k_{-1} = 373 s^{-1}. In the modeling, these are the rate constants at zero voltage.



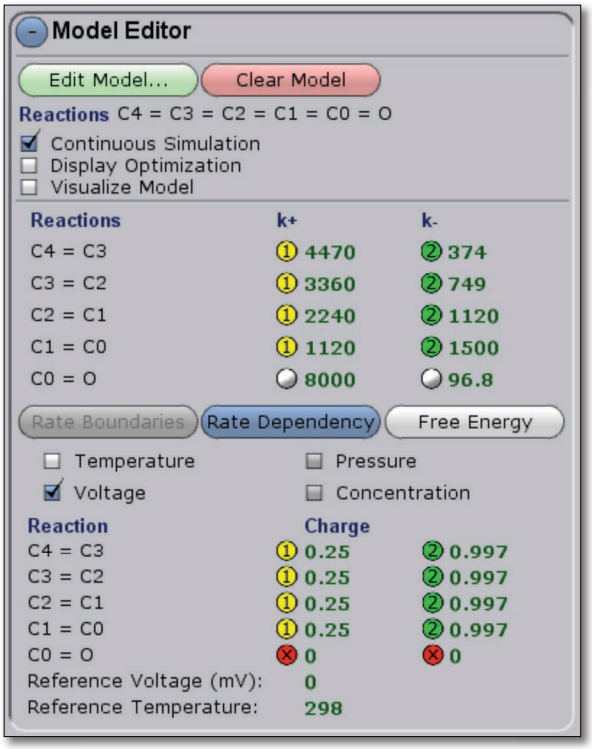
3. Next, click on "*Rate Dependencies*" if it is not already open, the check the rate constants that are voltage dependent. Note that in this example, because the rate constants k_{+1}, k_{+2}, k_{+3} and k_{+4} are linked, we only need to check one of the rate constants; we choose k_{+4} since it is the intrinsic monomer rate constant. Note also that the group designator for all of the linked constants are changed from green to white and can no longer be scrolled or changed since they now all depend on the controlling "amplitude" term, which is the rate constant at the reference voltage. If you want to change the relationships, you will need to first uncheck to box dictating a voltage-dependent rate.

Also, set the "Charge/Ea" terms to estimates of -1.0 and 0.25 for k_{-1} and k_{+4}, respectively, and set:

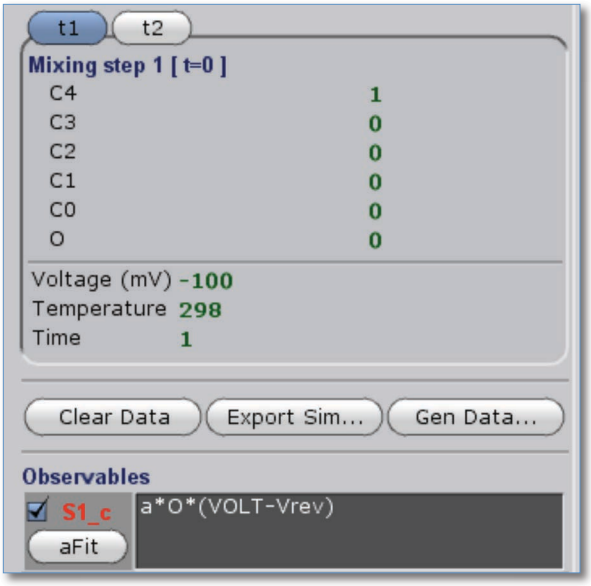
Reference Voltage = 0

Reference Temperature = 298

Note: Voltages must be entered in millivolts



4. Under the *Experiment Editor*, Set up the t1 reaction phase to allow the system to equilibrate at the resting voltage. Set any one of the species concentrations to 1 (here C4 = 1). Note that by default the first two species will be assigned a concentration of 1, so take care to set C3 = 0. Use of a concentration = 1 is just one easy method to normalize the simulation to give you the fraction of species in each state.



Set the resting voltage, time for equilibration and then define the output *Observables*. The measured current depends on the conductance, which is proportional to the fraction of channels that are open and the voltage. So the output observable should be:

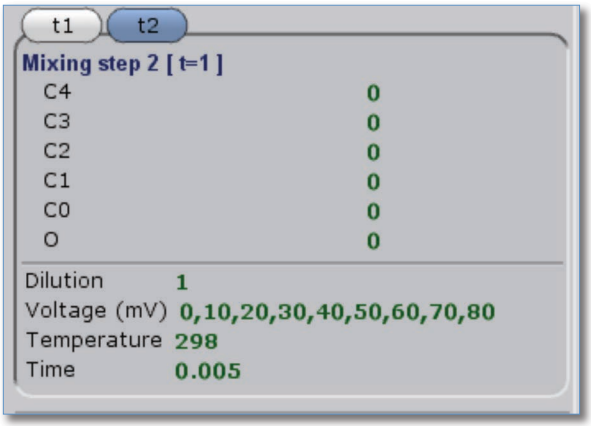
O*MaxConductance*ScalingFactor*(V_{drive}-V_{rev}).

In this example, I have defined the output as the concentration of open channels, O, times the voltage (*using the reserved word "VOLT"*) minus the reverse voltage, here listed as *Vrev*, and ***a*** is a scaling factor used in fitting data.

Note that because we set the starting of concentration of C4 = 1, then the output represents the fraction of channels that are open. ***Note:*** This may not be valid in a model in which the concentrations of other reactants are important. One could also have defined the output to be: O/(C4+C3+C2+C1+C0+O) to explicitly calculate the fraction of channels that are open, or O/Ctotal, where Ctotal is a parameter that must be set to the total concentration of channels. In addition, a further simplification is afforded by including the term 1/Ctotal in the scaling factor, ***a***. For example, if you include a reactant at 1 μM and you believe the concentration of channels is insignificant, set C4 = 0.001. The scaling factor ***a***, will then include normalization of the signal relative to the concentration of ion channels.

Observable*:* O*MaxCondScale*(VOLT-Vrev) = a*O*(VOLT-Vrev).
where VOLT is a reserved work pointing to the voltage for each trace, MaxCondScale, ***a***, is a fitted term combining the scale converting fraction to current and the maximum conductance and *Vrev* is a fitted parameter defining the reversal potential for the ion transporting through the channel.

5. Set up the t2 reaction phase, which represents the voltage jump, by clicking on "New Mix" and then entering the voltages and the times. Note that the concentrations of species listed represents the concentrations added during this "mixing" phase. The concentrations of species from t1 are carried forward with the dilution factor (1 in this case).



Note that after selecting "Rate Dependencies" above, a new line appears giving you the opportunity to enter a series of voltages to describe the family of curves. Voltages must be entered in millivolts because of our choice of constants in calculating the exponential term of the voltage-dependent rate constants.

6. You can now fit the data globally by clicking on "Fit Active Exp." or "Fit All Exp" if you have more than experiment. Note you have fit the entire family of curves with only 7 independent parameters, each of which is nicely constrained by the data. You can also now run FitSpace to get the confidence contours for each parameter. One could also fit to derive the V_{rev}, the reversal potential for K^+ ions.

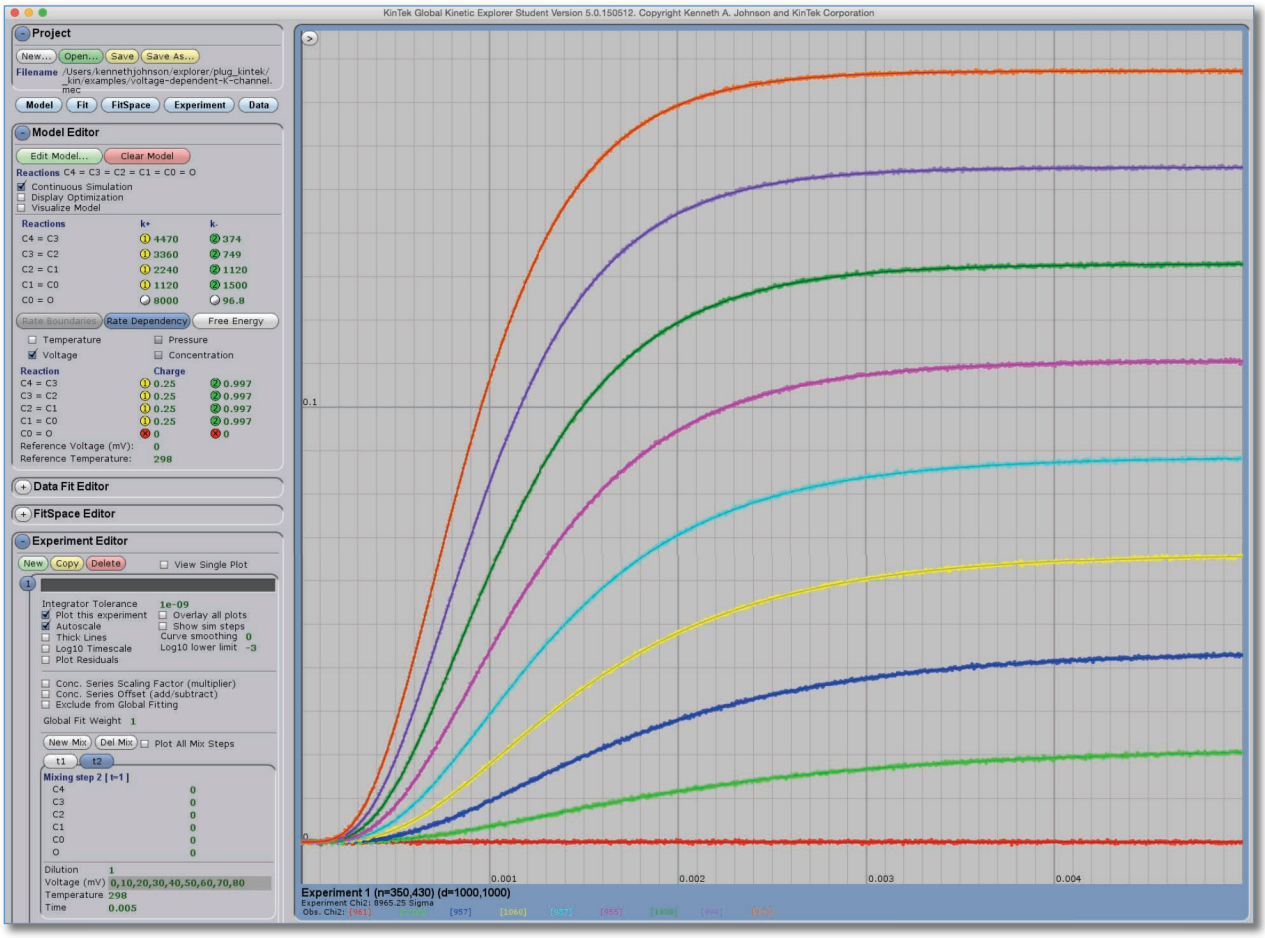


Figure 4.1. *Global fitting of voltage-dependent rate constants.* This figure shows synthetic data illustrating the fitting of voltage-dependent rate constants governing K+ channel opening obtained by recording the ion current after a voltage-jump, according to the model presented by Smith-Maxwell et al (6).

b. pH Dependence

You can model the pH dependence of a reaction by including protons in your reaction scheme, such as

E + S = ES = EP = E + P
E + H = EH; ES + H = EHS; EP + H = EHP
EH + S = EHS = EHP = EH + P

The trick is to then use brackets in specifying the H^+ concentration in the experiment editor to indicate that the H+ concentrations are fixed (by buffers) and not consumed during the reaction. For example, enter [1e-7], or better yet use [0.1] to enter the concentration in μM units to avoid round-off errors in calculations involving small numbers. See *Section 5.3*.

A new syntax is available starting with Version 6.0 where you can simply substitute pH for H in your mechanism. The program will then expect entries of concentration in pH units and instead of individual rate constants for binding and dissociation of protons, you simply enter the pKa. See the example of pH titrations in *Section 10.6.*

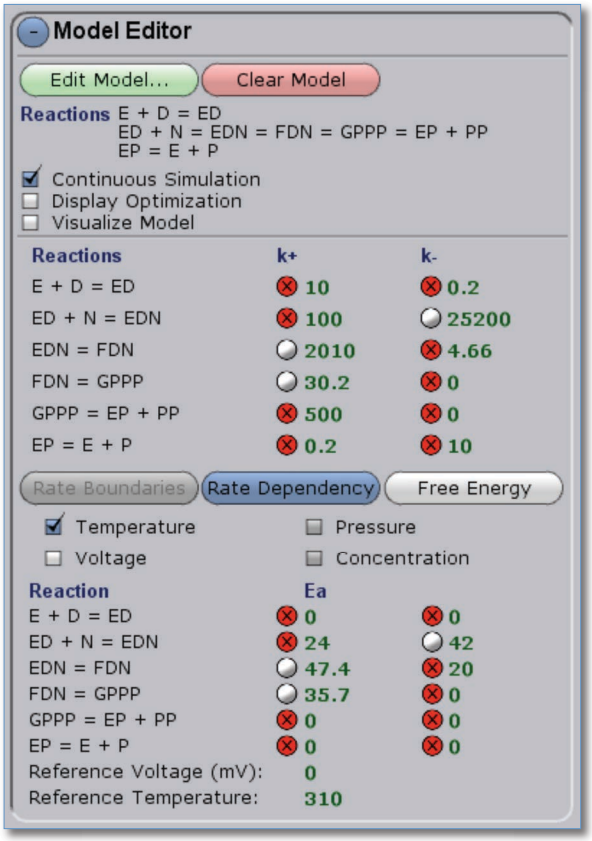
c. Temperature-dependent rate constants.

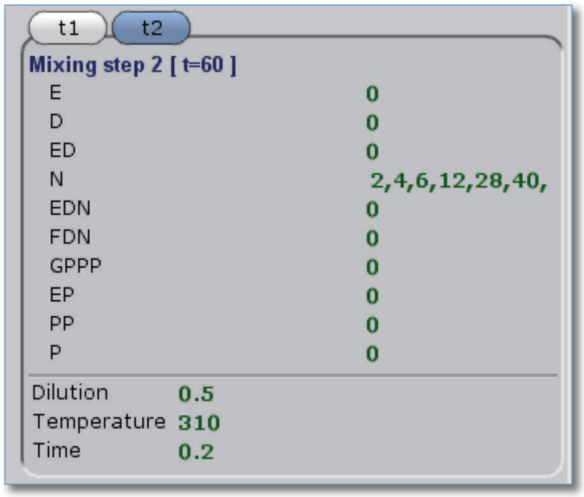
Temperature-dependent rate constants can be modeled in a fashion similar to the voltage-dependent rate constant. After defining the model, click on *Rate Dependencies* and select *Temperature*.

Note: It is important to specify the *Reference Temperature* on this panel. The *Reference Temperature* specifies the temperature at which the rate constants are defined. Experiments at other temperatures are related to the reference temperature by the activation energies (*Ea*) for each step (in kcal/mol by default).

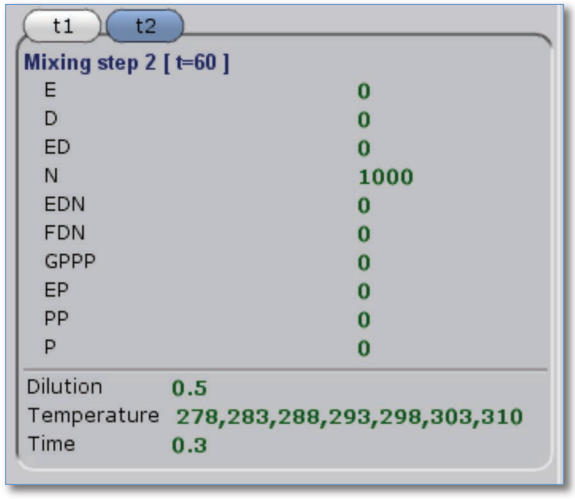
Specify the temperature for each experiment under the *Experiment Editor* (see *Chapter 5*). A given experiment can be performed at one fixed temperature or at a series of temperature analogous to the concentration series as described in *Section 5.3*.

The two figures below illustrate an experiment performed at a series of concentrations of N at one temperature (left) and another performed at a single concentration but over a series of temperatures (right).





Setting up an experiment a single temperature, over a series of concentrations of one species (N).



Setting up an experiment over a series of temperatures at a single concentration of each species.

The procedures for defining experiments are described in the next chapter. If you need to set up temperature dependent experiments, you may wish to come back to this section after you understand how to set up individual experiments.

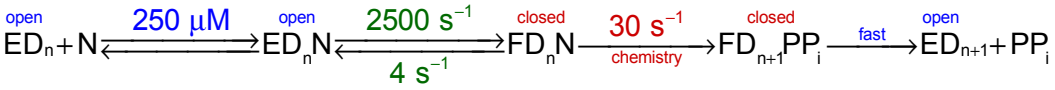
The temperature dependence of each rate constant is modeled according to the simple Arrhenius equation and a reference temperature. The data are fit to give the rate constants at the reference temperature and the activation energy (E_a) terms to account for the observed rates at different temperatures according to the following:

$$k_T = A \cdot \exp(-E_a / RT)$$

$$k_{ref} = A \cdot \exp(-E_a / RT_{ref})$$

$$k_T = k_{ref} \cdot \exp(\frac{-E_a}{R} \cdot (1/T - 1/T_{ref}))$$

The figure below shows a set of four experiments illustrating the use of temperature dependence. Experiments 1-3 show concentration series performed at three different temperatures. Experiment 4 show a series of experiments performed at a high concentration of substrate but at a series of temperatures. These synthetic data would be sufficient to define the K_d for the initial weak nucleotide binding, the maximum rate of the conformational change, the rate of chemistry and the activation energies for the conformational change and chemistry steps according to the following model. Measuring the nucleotide dissociation rate (k_{-2}) requires an additional experiment.



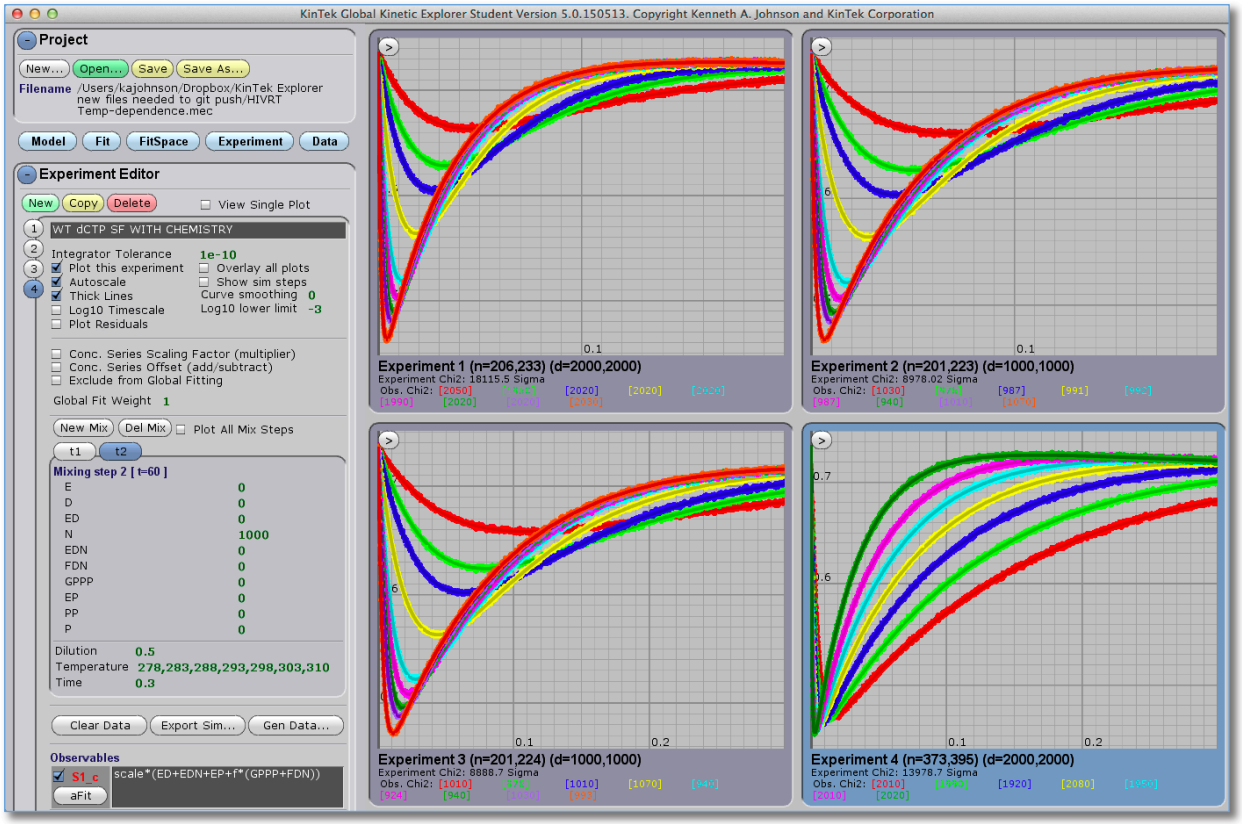


Figure 4.2. *Temperature dependence of DNA polymerase single turnover kinetics*. These experiments are modeled after work performed in HIV reverse transcriptase where fluorescence labeling of the enzyme provided a signal for the conformational changes (1). The fluorescence decreases when the enzyme closes after nucleotide binding, and then increases after chemistry and product release as the enzyme opens again in this single turnover experiment. Experiment 1, 310º C modeled with 0.2 µM Enzyme-DNA mixed with 2, 4, 6, 12, 28, 40, 60, 80, and 200 µM nucleotide (concentrations after mixing). In Experiment 2, the signal at 1 mM nucleotide was modeled for 278, 283, 288, 293, 298, 303, and 310º C. This file is in the examples folder: *HIVRT Temp-dependence.mec*.

4.4 Denaturant Concentration Dependent Rates

In studies on protein folding, the concentration of the denaturant (e.g., urea or guanidine hydrochloride) gives an exponential dependence on the rate constants governing folding and unfolding.

$$k_C = k_0 \cdot \exp(m \cdot C / RT)$$

where k_c is the rate constant at a given concentration, C and k_0 is the rate constant at zero concentration. Data are collected to examine the rate of unfolding after mixing with various concentrations of denaturant and folding after diluting from a high concentration of denaturant to a lower concentration. The data are then fit to derive the rate constants obtained by extrapolating to zero concentration. KinTek Explorer allows global fitting of the complete data set to derive a set of rate constants at zero concentration and m values to define the concentration dependence.

Figure 4.3 shows an example of such global data fitting of folding and unfolding kinetics.

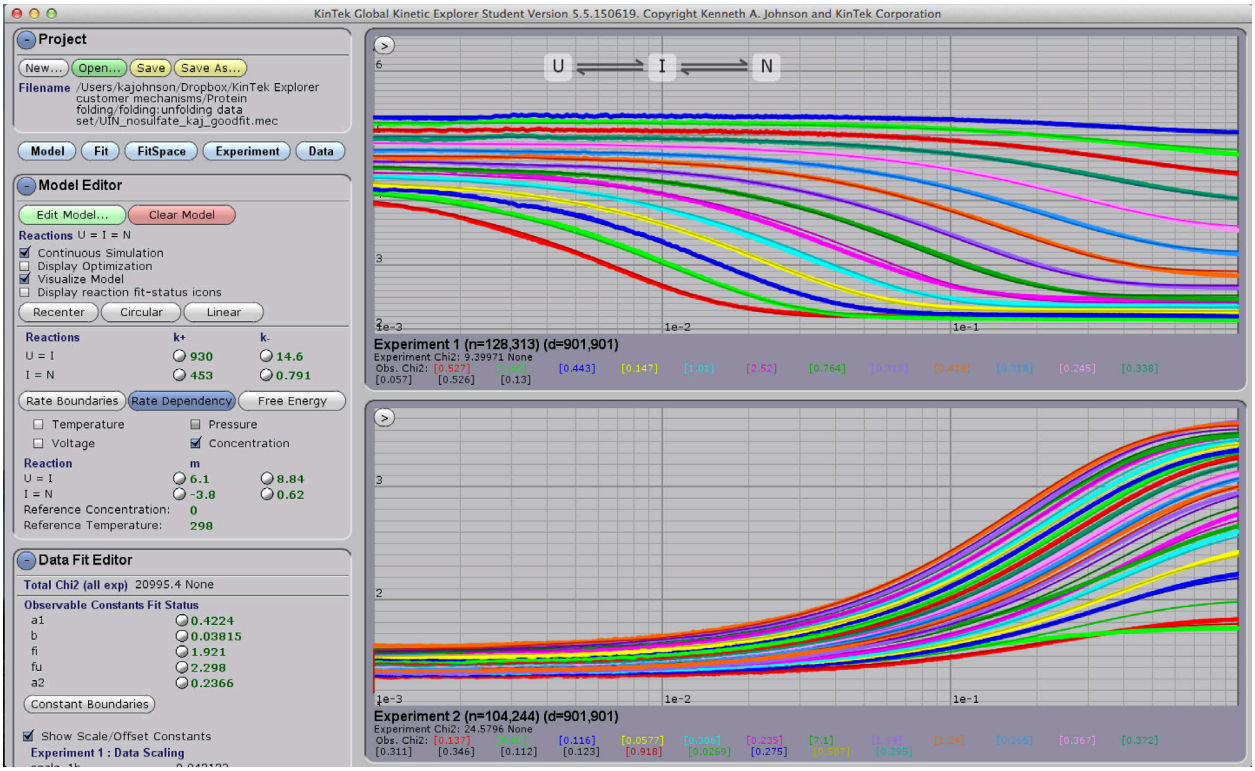


Figure 4.3 *Kinetics of protein folding and unfolding*. Experiment 1: unfolding at various concentrations of urea; Experiment 2: refolding from 8M urea into various concentrations of urea. Data were kindly provided by James Bardwell and Frederick Stull (7). Note the data are shown on a logarithmic time scale.

4.5 Free Energy Profile

The free energy profile is displayed for the reaction mechanism (or portion of the reaction mechanism that you select), the rate constants and a set of reference concentrations of ligands. Here we make a necessary distinction between the species we want to display on the free energy profile and the ligands that binding to each species. For example in the reaction pathway E + A = EA = EP = E + P, both S and P are ligands and the free energy profile will show the reaction progress from E – EA – EP – E as shown below. In addition, we enter an estimated physiological concentration of the ligands A = 20 and P = 0.1 (in this case) to provide a physiologically-relevant reference state rather than the 1 M standard state.

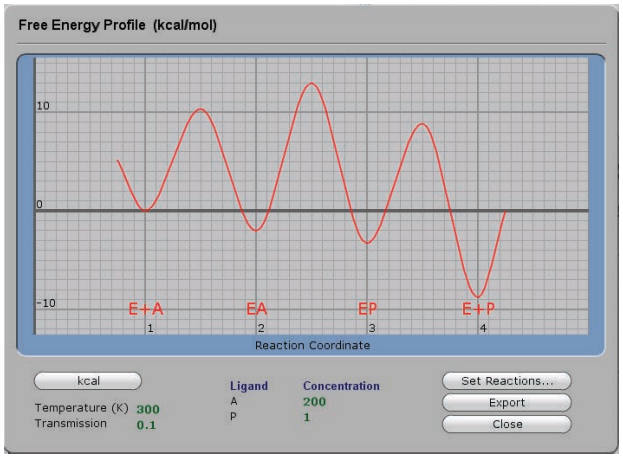


Figure 4.4A Free energy profile with substrate in excess over enzyme, specified by:

E + A = EA = EP = E + P

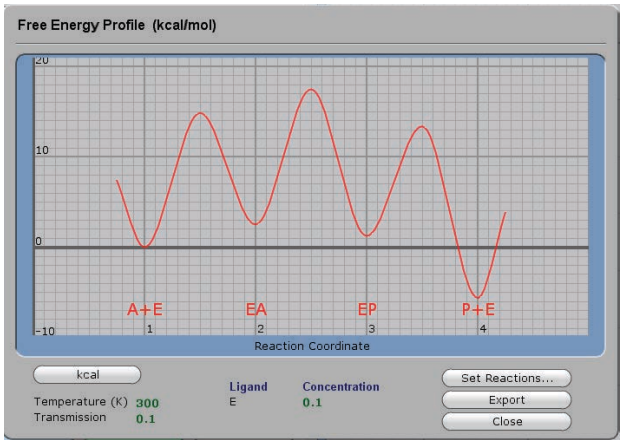


Figure 4.4B Free energy profile with enzyme in excess over substrate, specified by:

A + E = EA = EP = P + E.

The identities of the ligands are specified with the reaction mechanism is entered, by entering the ligand second in the sequence. For example, the sequence E + A = EA = EP = E + P specifies A and P as the ligands. In contrast, the sequence A + E = EA = EP = P + E would specify the E as the ligand in order to observe the conversion of A – EA – EP – P. The important distinction here is related to considering each second order reaction as governed by a pseudo-first order rate constant. The species in excess governs the rate of decay of the minor component:

$$E \xrightarrow{k_1[S]} ES \quad versus \quad S \xrightarrow{k_1[E]} ES$$

Calculation of the free energy profile: The free energy profile is calculated from the following simple transition state theory at 25° C to get the height of each barrier, $\Delta G_i^{\dagger}$. The plot is assembled by adding and subtracting each activation energy, starting with a reference state at zero. The y-axis of the plot is in kcal/mol, which is standard in biochemistry. Alternatively, you can display the graph in kJ/mol.

Starting with Version 8.0, we also allow you to vary the transmission coefficient, A.

$$rate = A \cdot \frac{k_B T}{h} \exp(-\Delta G^{\ddagger} / RT)$$

$$0 < A \leq 1$$

$$\Delta G^{\ddagger} = -RT \cdot \ln\left(\frac{rate}{A \cdot k_B T / h}\right)$$

$$k_B = 1.380x10^{-23}\ J/K$$

$$h = 6.626x10^{-34}\ J \cdot s$$

$$at\ 310K,\ k_B T / h = 6.46x10^{12}\ /s$$

$$k_B = R / N_A$$

$$R = 8.314\ J \cdot K^{-1} \cdot mol^{-1}$$

$$= 1.987 kcal \cdot K^{-1} \cdot mol^{-1}$$

$$N_A = 6.022x10^{23}$$

Note: You can specify the temperature of the reaction, but *changing the temperature does not give you an indication of the change in the profile versus temperature* because such dependence has not been

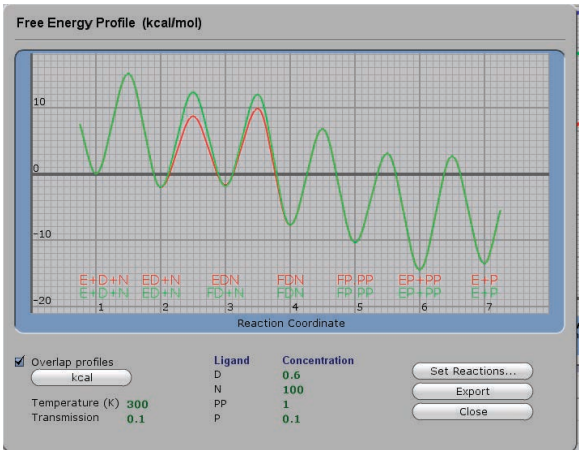
programmed. Changing the temperature setting only affects the y-axis scale. In a future software release we will include activation enthalpies to enable a realistic changes in the free energy profile with temperature derived from analysis of the temperature dependence of the reaction (See *Section 4.3c* above).

Set Reactions... For more complex pathways, the reaction sequence can be edited to eliminate selected reactions to simplify the display and/or create multiple paths as illustrated by these example files: HIVRT_cycle.mec, DHFR.mec, EPSP.mec, Actomyosin.mec, and kinesin_mADP_wt.mec.

Example of branched pathway: HIVRT_cycle.mec. The figures below illustrate various options for showing the free energy profile for this branched pathway. Note the use of the keyword, PATH: to designate each pathway. Changes to the free energy pathway do not affect anything else in the program. Note also the concentrations only affect the free energy profile and should be selected to reflect some desired reference state, such as the physiological concentrations. *When a rate constant is set to zero in the pathway, you will receive a warning since a free energy profile cannot be computed with a value of zero for any of the rate constants*. It is recommended that you then enter a nominal small reasonable value for the rate constant in order to construct the free energy profile. Be sure to describe this in any publications.

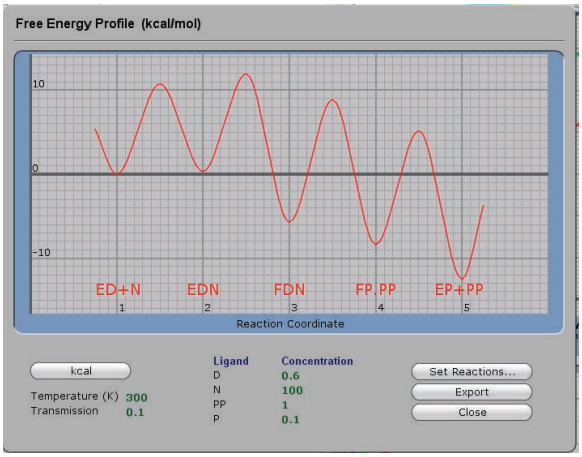
This figure shows the default with all reactions.

PATH:
E + D = ED
ED + N = EDN = FDN = FP.PP = EP + PP
EP = E + P
ED = FD
FD + N = FDN



This figure shows the simple path involving only nucleotide binding, conformational change, and incorporation.

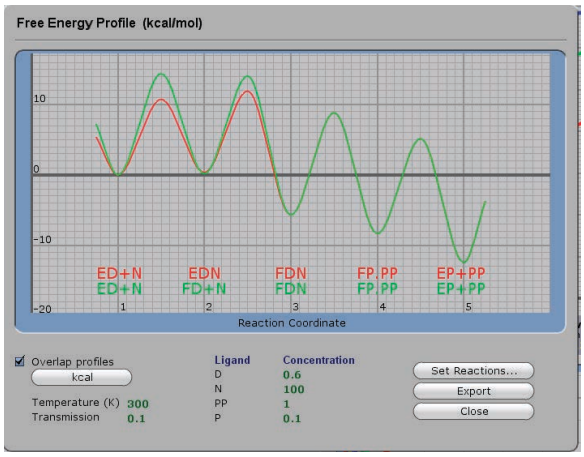
PATH:
ED + N = EDN = FDN = FP.PP = EP + PP



This shows the option of having two paths overlapping to illustrate the branching. Note that reactions are repeated to generate the second path. This figure shows the higher energy barriers for pathway involving a conformational change before nucleotide binding.

PATH:
ED + N = EDN = FDN = FP.PP = EP + PP

PATH:
ED = FD
FD + N = FDN = FP.PP = EP + PP



Chapter 5. Experiment Editor

Under Experiment Editor you will:

- Define new experiments
- Enter starting concentrations
- Enter reaction times
- Set the integrator error tolerance
- Define multiple mixing events
- Define output observables
- Control graphical display
- Generate synthetic data
- Export simulation results
- Fit to analytical functions (aFit)
- Control Scaling or Offset of data within a concentration series
- Specify single molecule data
- Assign a global fit weight
- Specify temperature or voltage
- Display the endpoint concentrations

The *Experiment Editor* contains all of the controls that are specific to a given experiment. Several experiments can be defined within a given mechanism file and globally fit to a single model. Each Experiment is defined by the starting concentrations of reactants and the experimental output signal (concentration, absorbance, fluorescence, etc).

Observable outputs are defined as sums of individual species, weighted by output factors, such as extinction coefficients. In the example at the right, a rapid chemical quench-flow experiment yielded total concentrations of species Q, I and B for the three signals.

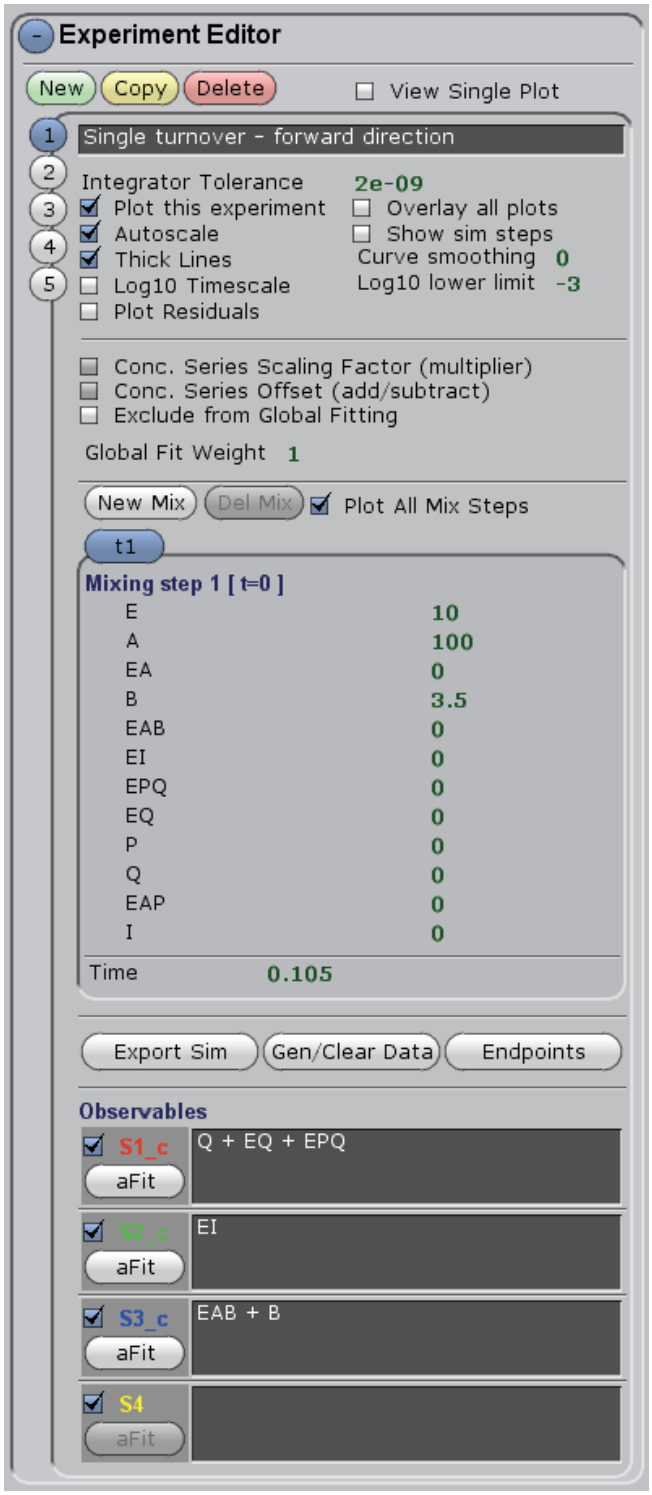


Figure 5.1 *Experiment Editor Control Panel*. Each experiment consists of a set of starting concentrations of reactants, the time of the reaction and the definitions of output expressions that relate the concentrations of species to observable signals. In the example shown above from EPSP.mec, there are four experiments, designated by the numbers 1-4 on the left hand edge. Clicking on each number allows that experiment to be selected so that parameters (concentrations, time, etc) that govern that experiment can be edited.

You must first import data into the *Data Repository* and then *assign* that data to a given experiment output. This allows you to import data without having to first develop a model, and to delete experiments without losing the data. The data stay in the *Data Repository* while you develop or change a model and experiments, and the data become part of the mechanism file when it is saved. See *Chapter 6 Data Repository* for a description of acceptable file formats and for details on how to import data and assign a given data set to a defined output.

5.1 Experiment setup

Each data set needs to be correlated with an experiment defined under the *Experiment Editor*. Specifying the starting concentrations of reactants and formulating a mathematical expression for the observable signal in terms of species concentrations and output scaling factors such as extinction coefficients, define an experiment. In addition, multiple output signals can be defined for one experiment such as in the EPSP.mec case illustrated in Figure 5.1.

New Click *New* to define a new experiment. Entries for the concentrations of species and definitions for output functions will be blank in the new experiment, but any previously defined experiments will be retained.

Copy Duplicates an existing experiment. This is useful if several experiments are similar; after copying an experiment, you can then edit the parameters that differ without having to define all parameters of the new experiment.

Delete Deletes the currently selected experiment.

Gen/Clear Data opens a dialog to clear the data from the current experiment or to generate synthetic data:

Clear Data Clears the data from the current experiment. This does NOT delete the data from the mechanism file; the data are retained in the Data Repository and can be reassigned to a new experiment.

Generate This function generates *synthetic* data to be used in evaluating fitting according to various models. See *Section 5.12 Generating synthetic data.*

Export Sim... This function will write two text files, one containing your data and another containing the results of the simulation in tab-delimited columns beginning with time, y1, y2, etc. These text files can then be read into a graphics program to prepare publication-quality figures. See *Section 5.14 Outputting simulations* and *Chapter 11.*

Endpoints This generates a table to the concentrations of reactants at the endpoint of the current simulation. Values are updated when parameters such as time or starting concentration are changed. With a concentration series, only the first concentration is shown, but upon export, a table with all of the starting concentration is saved.

aFit This allows you to fit data or the simulation output to an analytical function. This *conventional data fitting* to an equation is not to be confused with fitting data directly to the model by simulation, which is the real purpose of the *KinTek Explorer* software. See *Section 5.13 Analytic function fitting*. Fitting to an analytic function can be done either under the *Experiment Editor* or from the *Data Repository*.

New Mix Opens a window to define the new reactant additions for multiple mixing experiments. Previous reactants are diluted by the value defined by the *dilution* term. See Section 5.8 *Multiple-mixing*

experiments.

5.2 Observable output expressions

Outputs are defined to mimic the conditions under which data were collected to get an observable signal. For example, in rapid quench experiments, the observable signal may be the sum of all species with a radiolabel, whether bound to the enzyme or free in solution. In absorbance measurements, the signal may be the sum of all absorbing species weighted by their extinction coefficients. Similarly, in fluorescence experiments, the signal may be a weighted sum of various species. Remember that simulations are performed in units of absolute concentrations of species, while signals are a function of those concentrations, defined by the output factors or extinction coefficients.

a. Examples of output expressions

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} E{\bullet}S \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} E{\bullet}P \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} E + P$$

Fluorescence:	Signal = a*(E + b*ES + c*EP)/(E_0)
Burst of product formation:	Signal = EP + P
Absorbance of S and P:	Signal = a*(S + ES) + b*(P + EP)

In the example in the *Experiment Editor* panel shown above (Figure 5.1), there are three output expressions from this rapid quench experiment:

S1: Q + EQ + EPQ represents the sum of all species containing the reactant Q

S2: EI represents the enzyme-bound intermediate

S3: EAB + B represents the sum of species containing the reactant B

The table below shows an example from burst.mec, with three outputs. Output S1 shows the transient rise and fall of the species EA. Output S2 is a typical burst involving the sum of product bound to the enzyme and free in solution. Output S3 is a hypothetical fluorescence change with different extinction coefficients for each species.

	Signal	Notation
☑	**S1**	[EA]
☑	**S2**	[EP] + [P]
☑	**S3**	a*[E] + b*[EA] + c*[EP]

To define an output, enter the mathematical expression into the box to the right of the signal identifier (S1, S2, etc). New outputs can be entered into the blank space at the end of the list. Within a given output expression, multiple lines of text are allowed to make complex expressions easier to enter and read. Press <Return> to get a new line of text (no continuation symbol is needed). You can also use Ctrl-C and Ctrl-V to copy and past text. An output can be deleted by deleting its definition. *Terminate the entry by clicking outside the box.*

In defining the output, anything that is not recognized as a species name is assumed to be an output factor.

The check box (☑ Sn) allows the display of each output to be toggled on and off.

The color of each output can be altered by right-clicking on the colored signal designator, such as S1. This brings up the menu (shown at the right) for a color to be selected from the color wheel. If there is a concentration series, all of the traces will be made the same color.

All output factors are constrained to be positive numbers. This constraint is necessary to prevent the data fitting routines from wandering into nonsensical space (i.e., negative extinction coefficients). If you need a negative number, subtract a positive number in defining your output expression (see racemase_3step.mec). However, in most instances the use of a negative output factor is probably not valid because nearly all signals are the result of a weighted sum of concentrations of species.

For example, a fluorescence change upon substrate binding can be modeled as:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} FS$$

The general output factor would then be given by the sum of the concentrations of each species weighted by their fluorescence factor (*a, b, c*).

$$a\text{*E} + b\text{*ES} + c\text{*FS}$$

where each scaling factor gives the signal contributed by each species. It is preferably, however, to write the expression as:

$$a\text{*(E} + b\text{*ES} + c\text{*FS)}$$

because in this case, the factor ***a*** scales units of concentration to fluorescence intensity, while the factors ***b*** and ***c*** give the change in fluorescence relative to E. This is important because then it is easier to evaluate whether the changes in fluorescence are reasonable.

If $b < 1$, the signal will decrease with time as the reaction progresses, while the signal will increase if $b > 1$. See Tryp_synthase.mec and Burst_with_fluorescence.mec as examples, and note how the curves change in shape as the relative magnitudes of the output factors change. If there is no change in going from E to ES, then $b = 1$ and the expression can be simplified to:

$$a\text{*(E + ES} + c\text{*FS)}$$

You can use any alphanumeric characters to represent output factors so long as names are distinct from the names of species. Any name that is not recognized as a species gets assigned as an output factor. Look for errors in defining output expressions by scanning the list of output factors. For example, if you mistakenly typed: f1*(E + ES) + f2*EFS + c, there would be four output factors: f1, f2, c and EFS because EFS is not a species, so the program assumes it must be an output factor.

Each output can be toggled on/off with the check box. If there are data associated with a given output, it will be included in global fitting. However, if a given output is not displayed (unchecked box), then the data will not be included in global fitting.

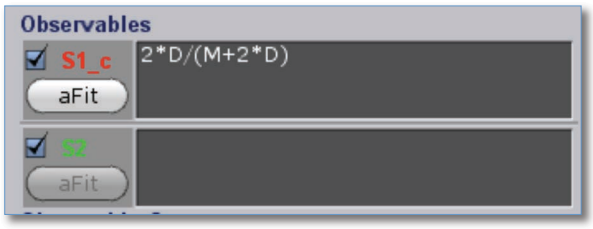
b. Numeric values in output expressions

We have included a provision for incorporating numbers in output expressions directly. This makes for a more simple, clear output expression without the need for a special symbol to specify constant numeric value in the output expression. For example, for a simple monomer-dimer equilibrium, you would express the fraction (θ) of monomer in dimer as given below form the TBP_dimer.mec example.

$$M + M \rightleftharpoons D$$

$$\theta = 2 \cdot D / (M + 2 \cdot D)$$

The output factor "**2"** is recognized as representing a constant numeric value of 2.



If there was a fluorescence change when forming the dimer you could define an output expression as: Signal = f1*2*D/(M+2*D) with output factors "f1" and a numeric value of "2".

c. Global output factors with multiple experiments

When you program the observable outputs for multiple experiments and if you use a given output factor term for more than one experiment, that factor will be identical for each experiment. Alternatively, you must use different terms to describe output factors if you want them to apply to only one experiment. For example, in fitting two fluorescence experiments, you may define the following two output expressions:

Experiment 1: scale1*(E + *a**ES + *b**FS) + bkg1

Experiment 2: scale2*(E + *a**ES + *b**FS) + bkg2

In this case, both experiments will be fit assuming the same percentage change in forming ES and FS, but the different scaling and background factors will allow for variations in setup in each experiment.

d. Exponential functions in output expressions

The following exponential functions can be applied to output observable expressions. These can be used to correct observed fluorescence for the absorption of incident light by a reactant or product as described in section d.

pow(x) = 10^x
exp(x) = e^x
log(x) = $\log_{10}(x)$
ln(x) = $\log_e(x)$, natural logarithm

NOTE: The strings for the exponential functions are *reserved* and cannot be used to define species or output factors.

e. Correcting fluorescence for absorption of incident light

The term, *inner filter effects*, refers to the attenuation of fluorescence due to a species in solution that absorbs light at either excitation or emission wavelengths. For example, there can be changes in fluorescence due to formation of a product that absorbs light at the wavelength required for excitation while monitoring a time-dependence of a reaction. Alternatively, during a titration the fluorescence intensity can be attenuated by the absorption of light by the titrant. The standard protocol in the literature is to correct the data to account for these *inner filter effects*. In contrast, in *KinTek Explorer*, we fit the data while including the *inner filter affects* in our model.

Absorbance of the light is a function of the extinction coefficient (ε), concentration (*c*) and path-length (*l*) according to Beer's Law:

$$A = \varepsilon \cdot c \cdot l = -\log(I / I_0)$$

The intensity of incident light (I_0 = *reference intensity*) decreases as the concentration of the absorbing species increases, and a fluorescence signal will decrease in proportion to the decrease in light intensity. To account for this decrease, the observable fluorescence output should be multiplied by a term to account for the decrease due to absorption of incident light. For example, for an enzyme-catalyzed reaction, the enzyme fluorescence could be a function of the changes occurring during substrate binding and product formation at the active site.

$$E + S \rightleftarrows ES \rightleftarrows EP \rightleftarrows E + P$$

Fluorescence due to enzyme species:

$$F = s \cdot (E + b \cdot ES + c \cdot EP)$$

If the product accumulating in solution absorbs light at the excitation wavelength, the total fluorescence will decrease according to the extinction coefficient of P:

$$F = s \cdot (E + b \cdot ES + c \cdot EP) \cdot I_{ave} / I_0$$

where I_{ave}/I_0 represents the average light intensity in the fluorescence cell relative to the incident light intensity in the absence of the absorbing species. Note that we are not correcting the fluorescence for the effect of the absorption of light, but rather computing the net fluorescence output by including a term to account for the decrease in light intensity. Because the fluorescence signal is the sum of light from the full side of the cell, it is necessary to compute the average intensity. Therefore, if species in solution absorb the incident light, we need to account for the affect of the absorption on the observed fluorescence according to the path-length and the extinction coefficient and concentration of the absorbing species. The intensity of light falls off exponentially as a function of path-length as shown in Figure 5.2, below.

Beer's law:

$$A = -\log(I / I_0) = \varepsilon \cdot c \cdot l$$

$$\frac{I}{I_0} = 10^{-A} = 10^{-\varepsilon \cdot c \cdot l}$$

I = light intensity
I_0 = reference intensity
ε = extinction coefficient
c = concentration
l = path-length

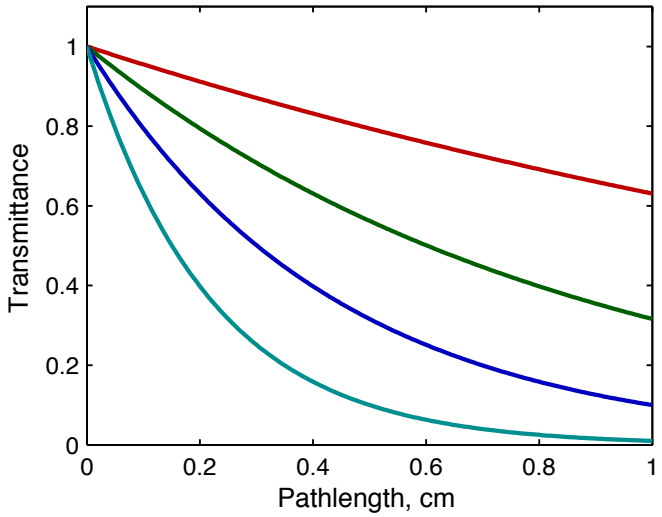


Figure 5.2 *Transmittance versus path-length*. Transmittance is computed as I/I_0 for cases when A = 0.2 (red), 0.5 (green), 1 (blue), and 2.0 (teal).

Linear Approximation: Figure 5.2 shows the transmittance versus path-length for different concentrations leading to net absorbance values of 0.2, 0.5, 1.5 and 2 at *l*=1 cm. At low absorbance values, the average absorbance can be estimated by a linear approximation.

$$\frac{I_{ave}}{I_0} = 10^{-\varepsilon \cdot c \cdot L/2} = 10^{-A/2}$$

where L is the path-length of the cell and A = is the absorbance at that path-length; and so, the average absorbance is A/2. This approximation is probably close enough for absorbance values less than 0.5 where the correction factor calculated from the linear approximation (0.562) is only 5% lower than the value derived from the more accurate calculation (I_{ave}/I_0 = 0.594) as described below. Thus at low absorbance values, one can approximate the correction by the following equation to define the output fluorescence signal.

$$F = s*(E + b*ES + c*EP)*pow(-a*(P))$$
$$= s*(E + b*ES + c*EP)/pow(a*(P))$$

where the half path-length and extinction coefficient have been combined into a single constant,

$a = \varepsilon*L/2.$

Full Correction Term: More precisely, to get the average light intensity across the full path-length of the cell, we integrate over the path-length, L and divide by L.

$$\frac{I_{ave}}{I_0} = \frac{\int_0^L 10^{-\varepsilon \cdot c \cdot l} dl}{L} = \frac{\int_0^L e^{-2.303 \cdot \varepsilon \cdot c \cdot l} dl}{L} = \frac{1 - e^{-2.303 \cdot \varepsilon \cdot c \cdot L}}{2.303 \cdot \varepsilon \cdot c \cdot L} = \frac{1 - 10^{-\varepsilon \cdot c \cdot L}}{2.303 \cdot \varepsilon \cdot c \cdot L}$$

$$\frac{I_{ave}}{I_0} = \frac{1 - 10^{-A}}{2.303 \cdot A}$$

To correct the fluorescence based upon the known absorbance of product, P, in the example given above we would compute the observed fluorescence as:

$$F = s \cdot (E + b \cdot ES + c \cdot EP) \cdot I_{ave} / I_0$$
$$A = \varepsilon \cdot c \cdot L = a \cdot P$$
$$I_{ave} / I_0 = (1 - 10^{-a \cdot P}) / (2.303 \cdot a \cdot P)$$

where the path-length and extinction coefficient have been combined into a single constant, ***a* = *ε**L***. Note that in this equation, the full path-length is used to compute the constant ***a***. Finally, the output factor can be entered using the power function:

$$F = s*(E + b*ES + c*EP)*(1 - pow(-a*P))/(2.303*a*P)$$

Unfortunately this equation gives a divide by zero error when P = 0. One can deal with this by performing the simulation starting with a negligible concentration of product. Alternatively, a small constant term can be added to the equation to prevent a divide by zero error:

$$F = s*(E + b*ES + c*EP)*(1 - pow(-a*(P + d)))/(2.303*a*(P + d))$$

For example, when $a*d$ = 0.0001 and P = 0, then I_{ave}/I_0 = 0.9997, so this would introduce a negligible error in the absence of product and would prevent the divide by zero error. Alternatively, one could define a function to account for the absorption of light by both the substrate and product:

$$F = s*(E + b*ES + c*EP)*(1 - pow(-L*(e1*S + e2*P)))/(2.303*L*(e1*S + e2*P))$$

where e1 = extinction coefficient for the substrate and e2 = extinction coefficient for the product. As before, one could also fit data using a constant which is the product of the extinction coefficient and the path length, such as a1=e1*L and a2=e2*L. Intermediates in the reaction pathway could also be included in the output expression.

Correction of Fluorescence Titrations: During a fluorescence titration (see *Section 5.11*), there can be inner filter effects if the titrant absorbs at either the excitation or emission wavelengths. This is illustrated by the example file, *Titration_innerfilter.mec*. Note that the inner filter effect must be quite large before there is a significant difference between the calculation with the linear approximation and the full derivation.

$$E + L \rightleftharpoons EL$$

Fluorescence due to enzyme species:

$$F = scale \cdot (E + b \cdot EL)$$

Titration of $[L]$:

$$@0.01-100,1$$

Note: the titration begins at a small nonzero value to avoid the divide by zero error.

Fluorescence signal:

$$F = scale*(E + b*EL)*(pow(-a*(L+EL)/2)$$

or

$$F = scale*(E + b*EL)*(1 - pow(-a*(L+EL)))/(\ln 10*a*(L+EL))$$

Note that in this expression, the terms "1", "2" and "ln10" are text terms that will appear as constants that must be defined:

"1" = 1

"2" = 2

"ln10" = 2.303

Alternatively, one could enter "2.303" as the text term.

Example of Output Observable Functions

Model:
E + L = EL

Signal with no absorbance (red)
a*(E +b*EL)

Signal with linear approximation (blue):
a*(E +b*EL)*pow(-e*(L+EL)/2)

Signal with full derivation (green)
a*(E + b*EL)* (1-pow(-e*(L+EL)))/
(ln10*e*(L+EL))

Note: L+EL = L_{total}, ln10 = 2.303

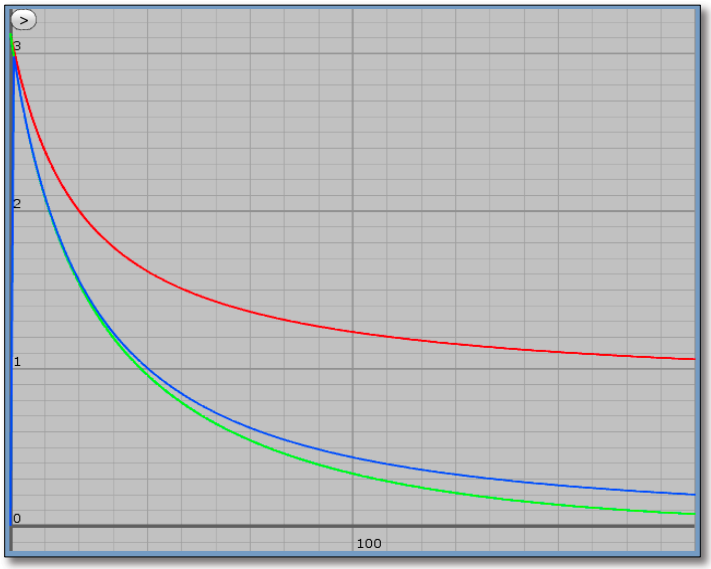


Figure 5.3 *Titration Curves Corrected for Inner Filter Effect*. The figure on the left shows the overlay of three simulations of titrations from the example file, *Titration_innerfilter.mec*. Titrations are shown without absorbance of light by the titrant (red), with the absorbance computed using the linear approximation (blue) and with absorbance computed using the full derivation (green). Always know the extinction coefficients for your titrant at the excitation and emission wavelengths used in your experiment!

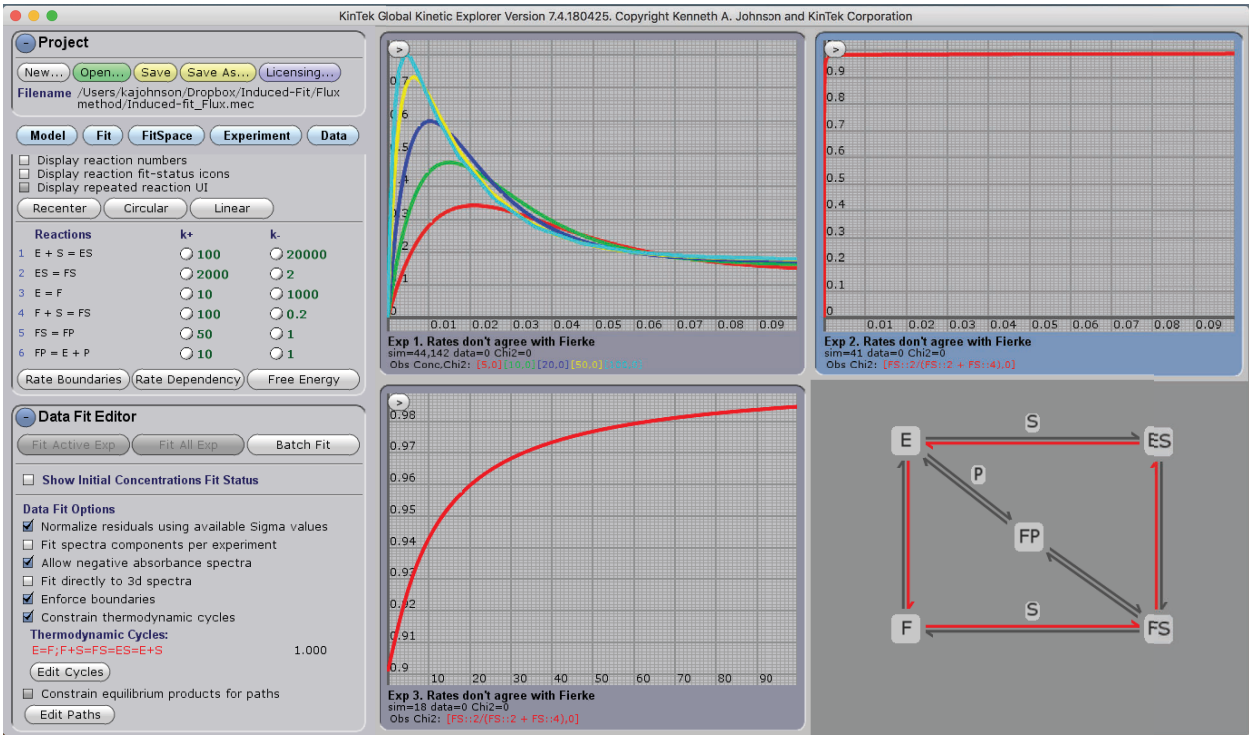
f. Flux integrals

With branched pathways, it is often important to know the fraction of reactants proceeded by each pathway. Although this has previously been approached by analytic methods with simplifying approximations with reactions close to equilibrium, it is better to compute the solution numerically without any approximations. Here we present a method based upon numerical integration of the differentials for each species, the *flux integral*.

Syntax: We allow computation of the *flux integral* for each species using the syntax ***S::n***, where ***S*** is the character string defining a given species and ***n*** is the reaction number. For complex pathways, one only needs to compute the flux integral for the last step in the pathway. This is illustrated with the example below comparing pathways for induced-fit versus conformational selection.

$$\text{Induced-fit: } E + S \underset{}{\overset{1}{\rightleftharpoons}} ES \overset{2}{\rightleftharpoons} FS$$

$$\text{Capture: } E \overset{3}{\rightleftharpoons} F; \quad F + S \overset{4}{\rightleftharpoons} FS$$

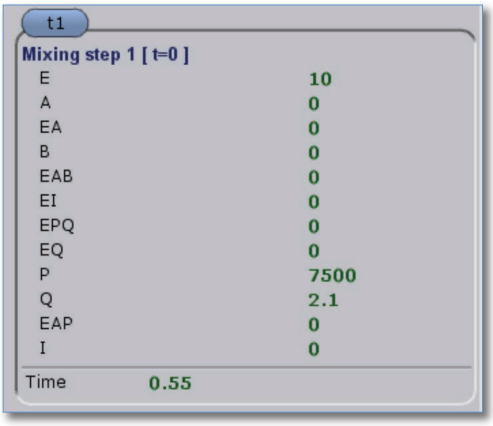


Induced-fit versus capture. In each experiment, the reaction of E to F is allowed to equilibrate, then substrate is added. *Experiment 1*: Simulation of the time dependence of formation of FS after reacting enzyme with 5, 10, 50, 100 μM substrate. *Experiment 2*: Fractional flux versus time to 0.1 s at 100 μM substrate. *Experiment 3*: Fractional flux after 0.1 s versus substrate concentration. The *Fractional flux* via step 2 relative to that via step 4 is defined by the function: *FS::2/(FS::2 +FS::4)*.

This analysis shows that 99% of the flux is via the induced-fit pathway at higher substrate concentrations. This is reduced to 90% at lower substrate concentrations. Note also in this example that the closed loop shown in red is recognized by the software and the rate constants are constrained so that the product of all equilibrium constants around the loop is unity.

5.3 Entering starting concentrations and reaction time

Set up each experiment by specifying the starting concentrations and reaction time, as shown at the right. Enter the concentrations of reactants (final concentrations after mixing to start the reaction).



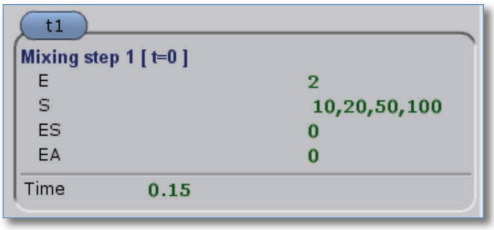
NOTE: The concentrations are those during the reaction, after mixing.

This example, from experiment 1 in the EPSP.mec file, shows the setup for a single turnover experiment in the reverse; note the high concentrations of products.

Concentrations and time values can be scrolled or entered in the text box. Click and drag to scroll. Click and release to open a text box.

The display will automatically rescale to the new time if the *Autoscale Plot* option is checked.

Concentration Series: One of the species can have multiple concentrations separated by commas. This will cause the program to perform simulations at each concentration. In multiple-mixing experiments, a concentration series can be employed only in one of the time periods. When assigning data to an experiment, care must be taken to match the concentrations in the simulation with those in the data file.



NOTE: If you need to vary the concentrations of more than one species, set up separate experiments for each data file.

There are some unique symbols used in entering concentrations:

[x] indicates that the concentration of x is held constant (not consumed, i.e., for pH)

x-y, n indicates an equilibrium titration from x to y with n steps. See *Section 5.11*.

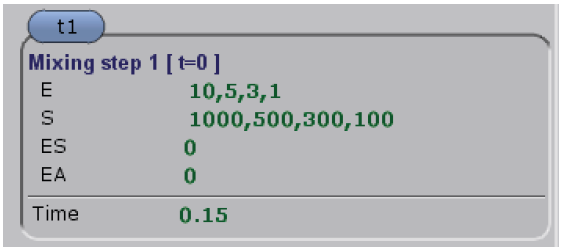
[@x-y, n] indicates an equilibrium titration from x to y with n steps computed

For exponentially increasing step size, use the Log10 Timescale button in the Experiment Editor and make sure the titration does not start at zero, but rather a small number indicating the beginning of your titration. See *Section 5.11*. NOTE: we no longer support the syntax using "exp" to designate a logarithmic titration scale.

Multiple Concentration Series. Starting with Version 7.6, you can enter a series of concentrations for more than one species. Although the entry method remains the same, you can now enter multiple concentrations for multiple species. However, we do not compute traces for a full matrix of concentrations involve all possible combinations of species. Rather, we enter pairs of concentrations as illustrated below. You need to enter the same number of concentrations for each varied species and there is a direct correspondence between concentrations in the two series. Think in terms of a table of concentrations. For example, you may want to enter a series of concentrations varying both the enzyme concentration and the substrate concentration, which we can visualize as shown in this table.

Species	Trace 1	Trace 2	Trace 3	Trace 4
E	10	5	3	1
S	1000	500	300	100
ES				
EA				

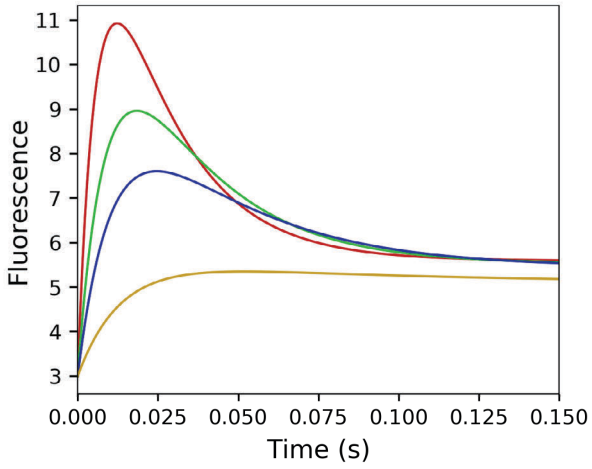
The two series of concentrations are entered into the form in the *Experiment Editor* as shown at the right for this example. Note that you must enter the same number of concentrations for each species and maintain the relationships between entries. That is, the first entry for E corresponds to the first entry for S, and so on as illustrated in the table shown above.

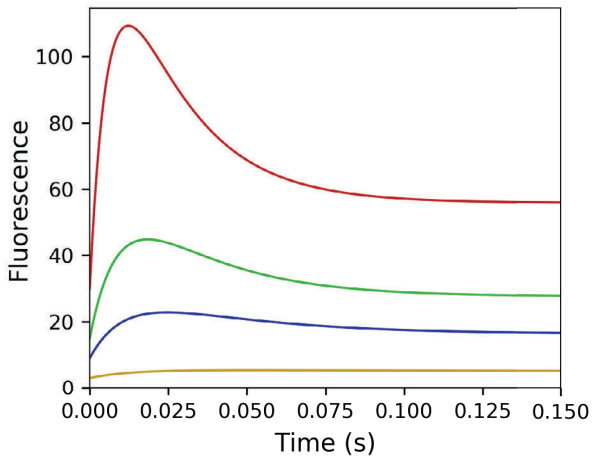


Defining the output observable. Another important consideration when varying both species is in defining an observable output function that normalizes the signal. For example, in this case, the signal is proportional to enzyme concentration, so the traces should be normalized by dividing by the total enzyme concentration (E + ES + EA). With a change in fluorescence in forming ES, the output signal is then defined as:

a*(E + b*ES + EA)/(E+ES+EA)

Without this normalization, the large range of signals makes it hard to see all of the traces on a single plot as illustrated below. If the signal was due to the substrate with enzyme in excess, then the traces should be normalized by dividing by the total substrate concentration.





Normalized traces: a*(E + b*ES + EA)/(E+ES+EA) *Traces not normalized*: a*(E + b*ES + EA)

Fitting to analytic functions: When fitting data to analytic functions, note that the plots of rate or amplitude versus concentration refer to the concentration of the species which is in excess. Remember, the rate of decay to equilibrium (or steady state) is defined by the concentration of the species in excess.

5.4 Multiple experiments

You can click New or Copy to generate additional panels for more experiments. Each experiment is defined by the set of individual starting concentrations and/or output definitions. The rate constants entered on the mechanism will be applied to all experiments. See EPSP_4.mec as an example. Numbered circles on the left of the panel are used to select the currently active experiment. Entries for concentrations and output factors affect only the selected experiment, while changes in rate constants apply to all

experiments. You can enter and fit up to 256 different experiments (sum of all initial concentration sets).

Selecting the active experiment

⇦ Click on a number to select a given experiment. *Note that the currently-active experiment is highlighted by a light blue colored border in the graphics panel.* All entries under the Experiment Editor apply only to the selected experiment. Data fitting can be based upon either the Active Experiment or All Experiments as described in Chapter 7, *Data Fit Editor*.

5.5 Controlling the simulation, display and global data fitting

☑ View Single Plot: When multiple experiments are included, checking *View Single Plot*, enables the display of only one plot at a time. This is particularly useful when working on a laptop computer or when you need to closely inspect one of the experiments. Select the active experiment by clicking on the numbered circle on the left.

Integrator Error Tolerance: (default value 2e-8). This value controls the error tolerance during numerical integration. In general, it is not necessary to change the error tolerance unless you see too few or too many points. Click on *Show Sim Steps* to see the distribution of points. Optimal performance is obtained with 30-300 simulation points. With fewer points, the curves show noticeable straight-line segments. Set the error tolerance to a smaller value to get more closely spaced simulation points to see a smoother curve. The minimum value is 1e-12.

☑ Plot this experiment: This function allows you to turn on/off individual plots when multiple experiments are present. The state of the check box is saved for each experiment; therefore, if an experiment is not displayed you will need to first select that experiment (see Selecting Active Experiment, above) and then change the status of the Plot checkbox.

☑ Autoscale: This function allows you to turn on/off the auto-scaling of both x- and y-axes. By default, the auto-scale is turned on. It is useful to turn this function off if you want to see the trends in the output while changing a rate or concentration that affects the amplitude of the reaction since the continuous auto-scaling tends to obscure the effect visually.

☑ Overlay all plots: This function allows you overlay all visible experiments on a single graph.

☑ Show Sim Steps: Turns on/off the display of the steps of the simulation as square black dots on the simulated curve. Points obtained by spline interpolation are shown in white.

☑ Thick Lines: Toggles between thin and thick lines display of the simulation output. The thinner dark line helps to see the simulation line when there are many data points.

☑ Log10 Timescale: Turns on/off the display of data and simulations on a log time scale.

☑ Log10 Lower Limit (n): Sets the power of ten lower limit of the logarithmic time scale.

Curve smoothing: (range 0-10) This function controls the number of points to add during the spline interpolation between simulation steps to get a more smooth appearance. This parameter does not affect data fitting, only the appearance of the graph on screen. See "Show simulation steps".

☑ Exclude from Global Fits: This allows the selected experiment to be excluded during global fitting. This is useful when fitting multiple experiments in order to explore the effect of a given experimental data set on the global fit by excluding it from the fit.

Global Fit Weight: You can change the *Global Fit Weight* for a given experiment to any positive number. The relative weight given that experiment will be the ratio of the value for that experiment relative to values for the other experiments. For example if the *Global Fit Weight* = 10 for one experiment and 1 for all others, the chi^2 values will be multiplied by 10 in computing and attempting to minimize the total chi^2 in global fitting. This feature can be useful in seeking a global fit that

accounts for all experiments, by increasing the weight for experiments with fewer data points, for example. However, the values should probably be reset to 1 when evaluating the final global data fit.

5.6 Graphics display controls

In addition to the graphics controls described in the previous section, which alter the graphical output of each experiment, we offer features to enhance to expand the display to make it easier to see fine features of the data and the calculated fit.

a. Expand/Contract Graph

When viewing multiple experiments simultaneously, it is useful to be able to expand one experiment to fill the graphics area, especially when working on a laptop computer. You will notice a small icon in the upper left corner of each figure. Click on the icon to expand/contract the graph for a given experiment.

> Expand the selected figure to occupy the full graphic space

< Shrink the selected figure back to its original size

b. Zoom

To zoom in on a area of the graph, *left-click and drag* to create a box outlining the area you wish to expand, the release. The graph is expanded to show the area of interest. Ctrl-W returns to the full wide-screen display mode. *Right-click and drag* to move the graph around within the zoomed window view.

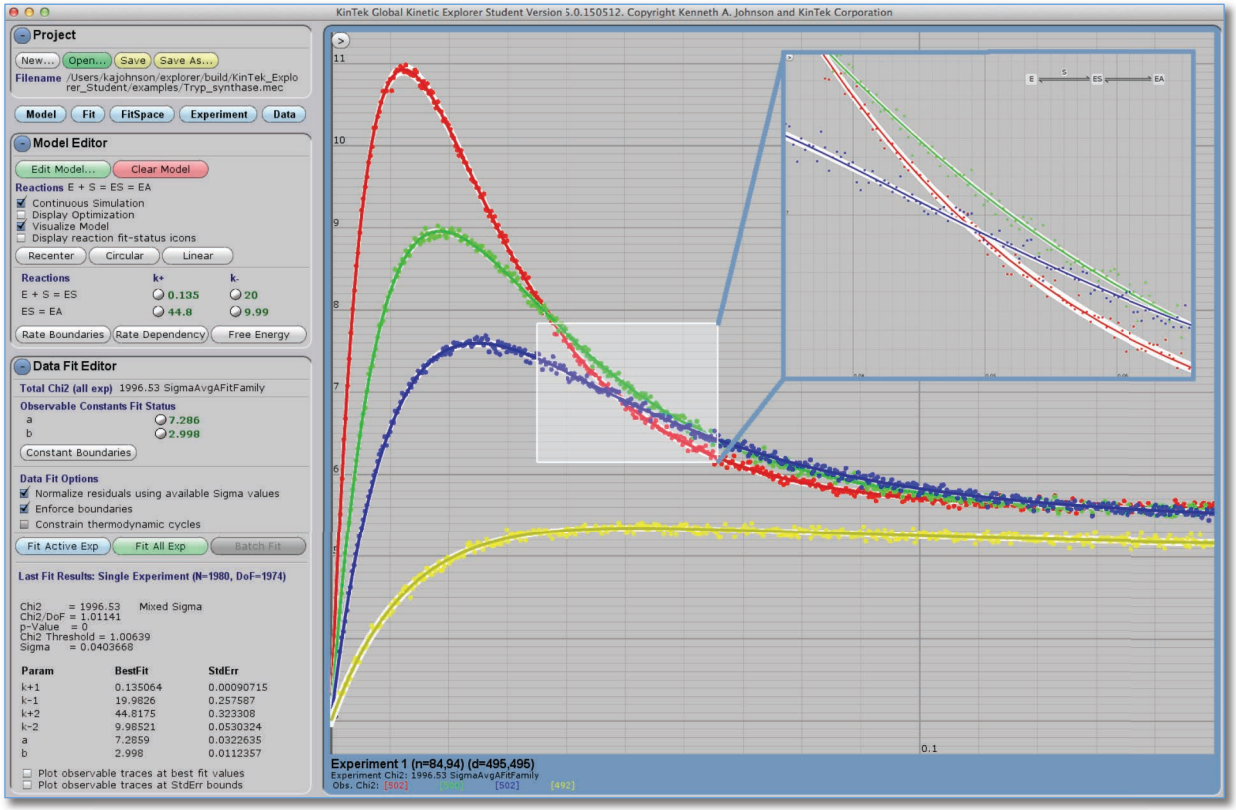
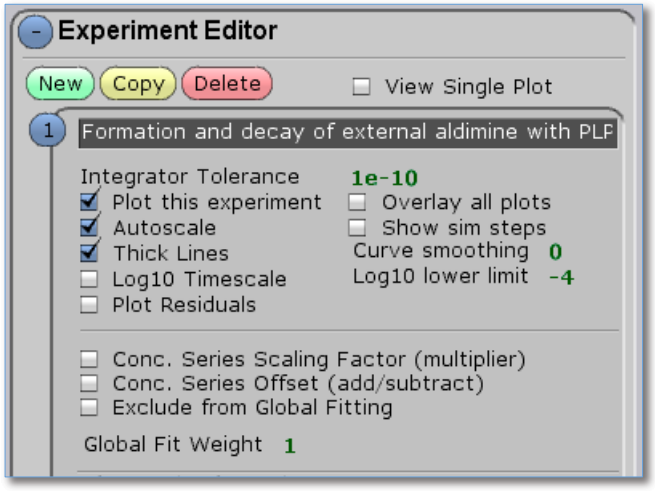


Figure 5.4. *Zooming in.* Left-click-drag then release to zoom in. Ctrl-W to return.

5.7 Scaling or offsetting traces within a concentration series

☑ Conc. Series Scaling Factor (multiplier): This function introduces an additional scaling factor to normalize traces within a concentration series.

☑ Conc. Series Offset (add/subtract): This function introduces an additional offset factor to normalize traces within a concentration series.



a. When and why to normalize data.

Data normalization is useful to correct for fluctuations in lamp intensity in fluorescence traces, or other variations in signal intensity that preclude achieving a global fit to the data. Although this function controls data fitting, it applies only to a single experiment when there is a concentration series. Data are normalized based upon the global fit to the data by applying an additional scaling factor that is applied to each trace to correct for variations in signal intensity. Alternatively, the addition or subtraction of a variable offset term can be used to eliminate fluctuations in reference for Absorbance measurements or in baseline for rapid quench-flow experiments. In either case, proceed with caution and reject any scaling factors or offset factors that are large and alter the inherent relationships within the concentration series.

b. Data normalization example

We illustrate the procedure for normalization by the following example.

1. Open the file, *Tryp_synthase_lamp_error.mec*, which shows an exaggerated example with 5-18% fluctuations in signal intensity as illustrated in Figure 5.5A. Note that a good global fit cannot be obtained because of fluctuations in signal intensity for each trace.

2. Check *Normalize traces in fitting conc. series* under the *Experiment Editor*. A scaling factor is then defined and applied to normalize the traces during fitting. These scaling factors can be seen under *Data Fit Editor*.

Signal a = scale_1a*(original definition)
Signal b = scale_1b*(original definition)
Signal c = scale_1c*(original definition)
Signal d = scale_1d*(original definition)

3. Click on Fit Active Experiment or Fit All Experiments. In the process of data fitting, one scaling factor will be held fixed at unity, while the others are allowed to float during the fitting process in achieving the best fit. The program will select the trace with the lowest chi^2 value to receive the scaling factor set at unity. You will then see the results shown in Figure 5.5B. Note that now the global fit overlays the data much better. As part of the error analysis, we account for the additional variable due to the scaling factors.

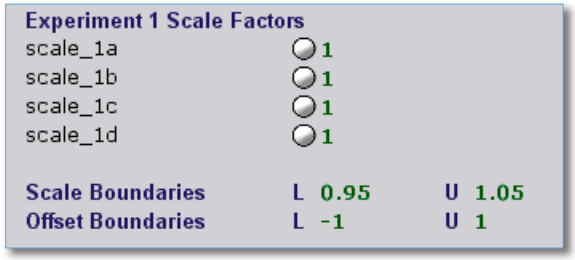
4. Now, you can click Apply To Data. After the normalization factors have been derived, they can be applied to the data as a correction factor by dividing each point in the data file by the normalization factor

for the corresponding concentration (see Figure 5.5C). This is useful because it restores the uniform pattern in amplitudes and endpoints for the traces within a concentration series. The scaling factors will be removed from the definition and applied to the data. If you want to explore the effect of variable scaling factors on the fitted parameter, you can once again check *Conc. Series Scaling Factor (multiplier)* under the *Experiment Editor* and repeat the fitting process. You will now see scaling factors close to unity, but the error analysis will include error terms for scaling factors and reveal the effects on the standard errors for other fitted parameters.

NOTE: data normalization is based upon multiplying or dividing by a constant factor, NOT by subtraction or addition. For this approach to be valid, it is crucial that the data be presented without subtraction of a variable baseline. If different baselines have been subtracted from the individual traces within a concentration series, one must exercise caution to avoid introducing systematic errors when applying a scaling factor.

CAUTION: Proceed with care when using the data normalization function to ensure that you do not remove an important element of your data by normalization. For example, if there is a fast phase at the beginning of the transient leading to loss of signal amplitude, normalization may eliminate the lost amplitude, thereby eliminating that information from your data. Normalization factors should be randomly distributed, not correlated with concentration, and probably less than a few percent (i.e. values ranging from 0.97 to 1.03).

You can the allowable range for scaling factors and offset values by entering Upper (U) and Lower (L) limits for the factors as shown at the right.



☑ Enforce boundaries: Under Data Fit Options this checkbox controls whether the upper and lower boundaries are enforced during data fitting. *Boundaries are not enforced unless this box is checked*.

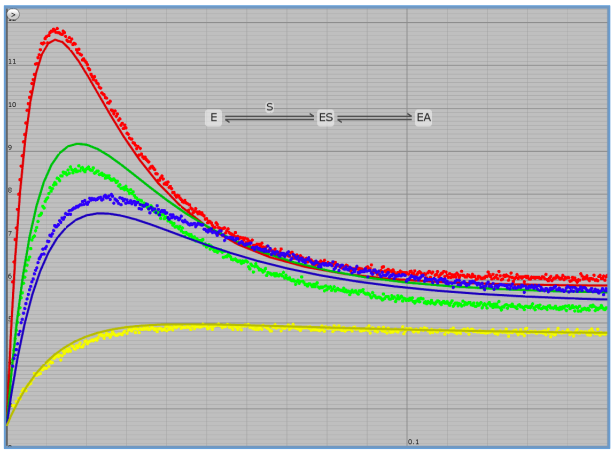
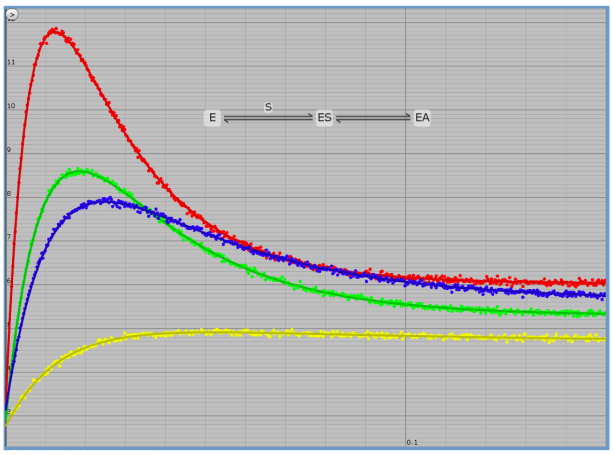


Figure 5.5 *Data Normalization*
A. Data fit without normalization. This figure shows the best fit attainable when there are significant fluctuations in signal intensity for the data collected at the different concentrations.

Figure 5.5B. *Data fit with normalization*. By including a scaling factor to normalize the fluctuations in signal intensity, a much better fit is obtained.

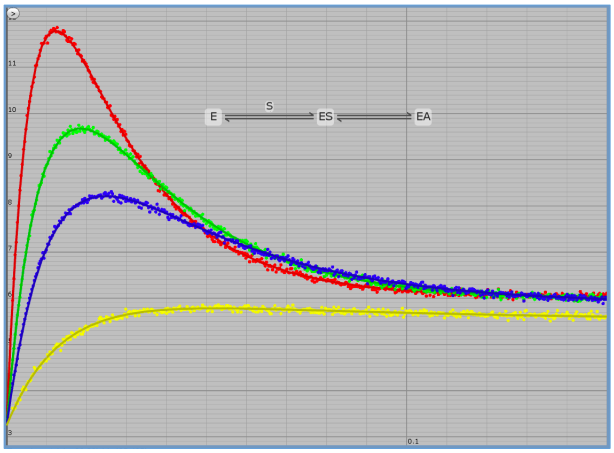


Scaling factors were derived:

Data have been scaled by factors:

scale_1a = 1.174
scale_1b = 1.044
scale_1c = 1.131
scale_1d = 1.0 (fixed)

Fibure 5.5C. *Normalized Data*. After applying the normalization factors to the data, the curves no longer reveal fluctuations in lamp intensity between the traces.



Data have been normalized by factors:

scale_1a: 0.851565
scale_1b: 0.957832
scale_1c: 0.884136

These values are the inverse of the measured scaling factors in Figure B because these represent the values that are *multiplied* by the data during normalization.

c. Scaling factor error analysis

After normalizing the data, the normalization factors are added to the data file and are displayed under the *Experiment Editor*. However, variations in the normalization factors will not be included in the kinetic parameter error analysis. Therefore, to investigate the effect of the normalization factors on the full error analysis, once again check the box ☑ *Conc. Series Scaling Factor (multiplier)*. A new set of normalization factors will appear and after fitting the data again, the values will be near unity. For example, in the Tryp_synthase_lamp_error.mec example above, in the second round of computing normalization factors, the following values were obtained:

Factor	***Value***
scale_1a	1.0002 ± 0.005
scale_1b	1.00015 ± 0.004
scale_1c	1.00009 ± 0.003

These factors will also be allowed to float during the FitSpace calculation in order to evaluate the effects of their variability on other kinetic parameters. As long as *Conc. Series Scaling Factor (multiplier)* remains checked, the scaling factors will remain as a variable during fitting.

If you choose Apply To Data, the new scaling factors, the net scaling factor will be the product of the new and old data scaling factors and the scaling factors will no longer be considered as a variable during data fitting.

Reporting Errors: At the very least, the scaling factors applied to data (along with their standard errors) should be reported in publications. Note that the scaling factors will be applied to the data and included in the text output under the *Export Sim* function. This will enable the use of your favored graphics program in generating publication quality figures.

d. Concentration Series Offset

The concentration series offset functions by adding or subtracting a small term to account for fluctuations in the reference in absorbance measurements or other variable background from one experiment to the next. The variable offset is included as an additional term used in fitting the data to the model. As in the case of the variable offset multiplier, exercise caution when using this feature of the software. Generally, correction factors are acceptable only when they are small relative to the magnitude of the signal.

NOTE: An important difference with the *Offset factors* is that they can be positive or negative. To enter a negative value, click on the value and type in a negative number. You can then scroll the offset term as a negative number.

5.8 Multiple-mixing experiments

You can program up to twelve mixing events. After entering the concentrations and reaction time for the first mixing, click on *New Mix*. You then enter the concentrations of species to be added (their final concentrations), the dilution factor for the previous reactants (a number less than or equal to 1), and the reaction time. Figure 5.6 shows the ribozyme.mec experiment 2 as an example. In this experiment, a fluorescent oligonucleotide is allowed to bind to the ribozyme during the first reaction period, and then the solution is mixed with a competing unlabeled nucleotide in the second reaction period in order to measure the dissociation rate. The mixing involves a 0.5 dilution (1:1 mix).

NOTE: Take care to enter the concentrations of reactants during each reaction phase. For example, in the ribozyme.mec experiment 2 example, E = 0.1 and S = 0.5. In the second mixing, these concentrations get diluted by a factor of 2 (dilutions factor = 0.5) and then C = 5 gets added. Thus, in the final mixture one has E = 0.05, S = 0.25 and C = 5 (units of concentration are μM, but that is only kept in the mind of the user, not entered into the programmed).

New Mix Opens a window to define the new reactant additions. Previous reactants are diluted by the value defined by the *dilution* term.

☑ Plot All Mix Steps: Show all mixing time domains when checked. The alternative is to show only the currently selected time domain. Note also, that assigning data to the experiment is controlled by whether this button is selected. If you have data only for one time domain, show only that time domain before assigning the data and the new data will be brought into that time frame. If the data span the entire experiment time course starting with t1, then check *plot all mix steps* and then assign the data to get data to start at t1 = 0. You can also control the placement of the data within the multiple mixing time frames by using the time offset when importing or editing data under the *Data Repository*. You can also adjust the time offset when importing data to account for a previous mixing time, but this is not the best method.

NOTE: Take care to display only the correct mix step when assigning data to an experiment. The data will be assigned starting at the beginning of the first reaction mix step shown.

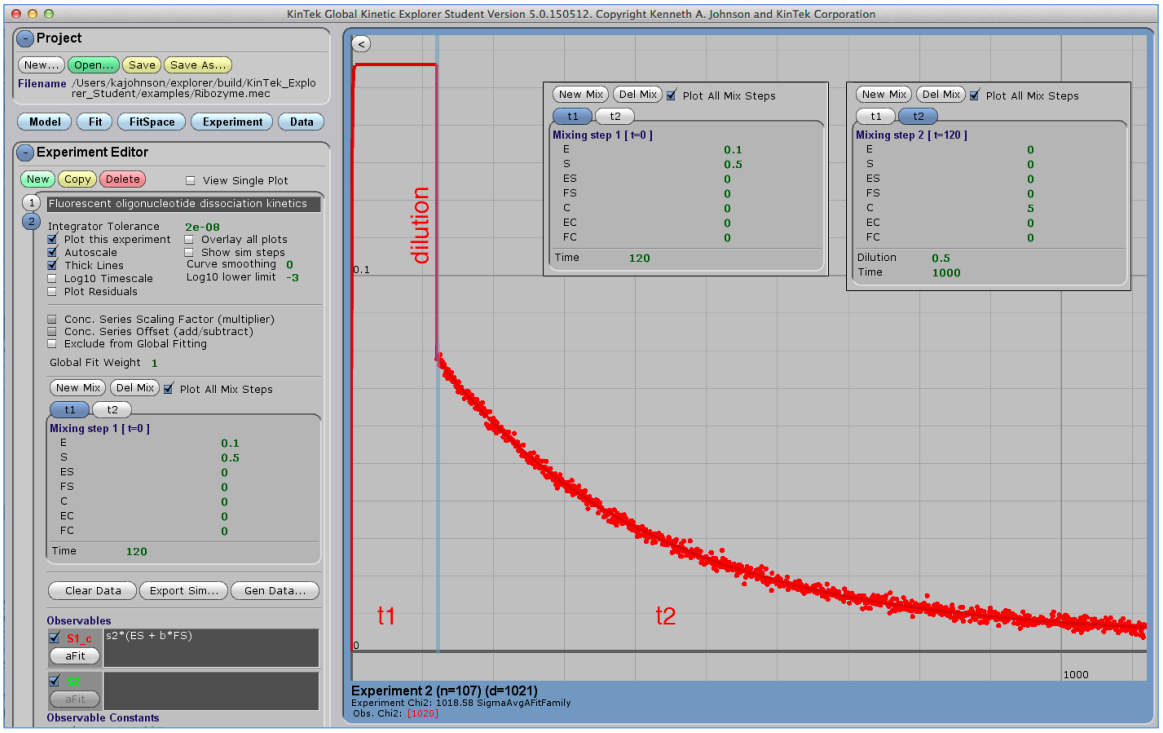


Figure 5.6 *Double-Mix Experiment.* This shows a double mixing experiment from ribozyme.mec. During the first time period (t1) the ribozyme-substrate complex is allowed to equilibrate (0.1 µM E and 0.5 µM S). With a 0.5-fold dilution (1:1 mix) the competing unlabeled substrate is added at a concentration of 5 µM. The change in fluorescence defines the dissociation rate.

The %n Syntax: If you perform a gel filtration step associated with the second mixing step, then you can use %0 to specify that the concentration of a species should be brought to zero at the mixing step. For example in measuring the slow release of an enzyme inhibitor, one can incubate the enzyme and inhibitor in step one, and then perform a gel filtration to remove excess inhibitor and then add substrate. The time dependence of product formation can be fit to establish the inhibitor dissociation rate Entering %0 in place of the concentration of the inhibitor in the second mixing step will force the concentration of free inhibitor to be zero, but will retain the inhibitor bound to the enzyme. This feature is added to allow one to model the experiment precisely without simplifying assumptions or limiting experimental conditions. Alternatively, any value of *n* can be used to specify that the concentration after mixing is a new value, *n*, independent of carryover from prior mixing steps.

Constant Concentrations [n]: If you wish to model a reaction where the concentration of a species does not change during the reaction, such as in a cell where a reactant is maintained at a steady state level, then enclose the concentration on the entry table in square brackets as [25], for example. Dilution during mix steps will reduce the concentration according to the dilution factor. In a new mix step, using the syntax **[%n]** specifies that the concentration after the mix step will be n and will not change during the reaction.

5.9 Modeling and fitting SRP data

Using the fixed concentration syntax, we can model SPR (Surface Plasmon Resonance) data where the flow of new solution across the surface of a cell maintains a fixed concentration. Note however that holding the concentration of a reactant constant presents a challenge with fitting data based upon numerical integration of the rate equations and can cause the simulation and fitting to become very sluggish. We have provided a novel solution to optimize the simulation of experiments with fixed concentrations, but that *can only be applied if all experiments within a mechanism file use fixed*

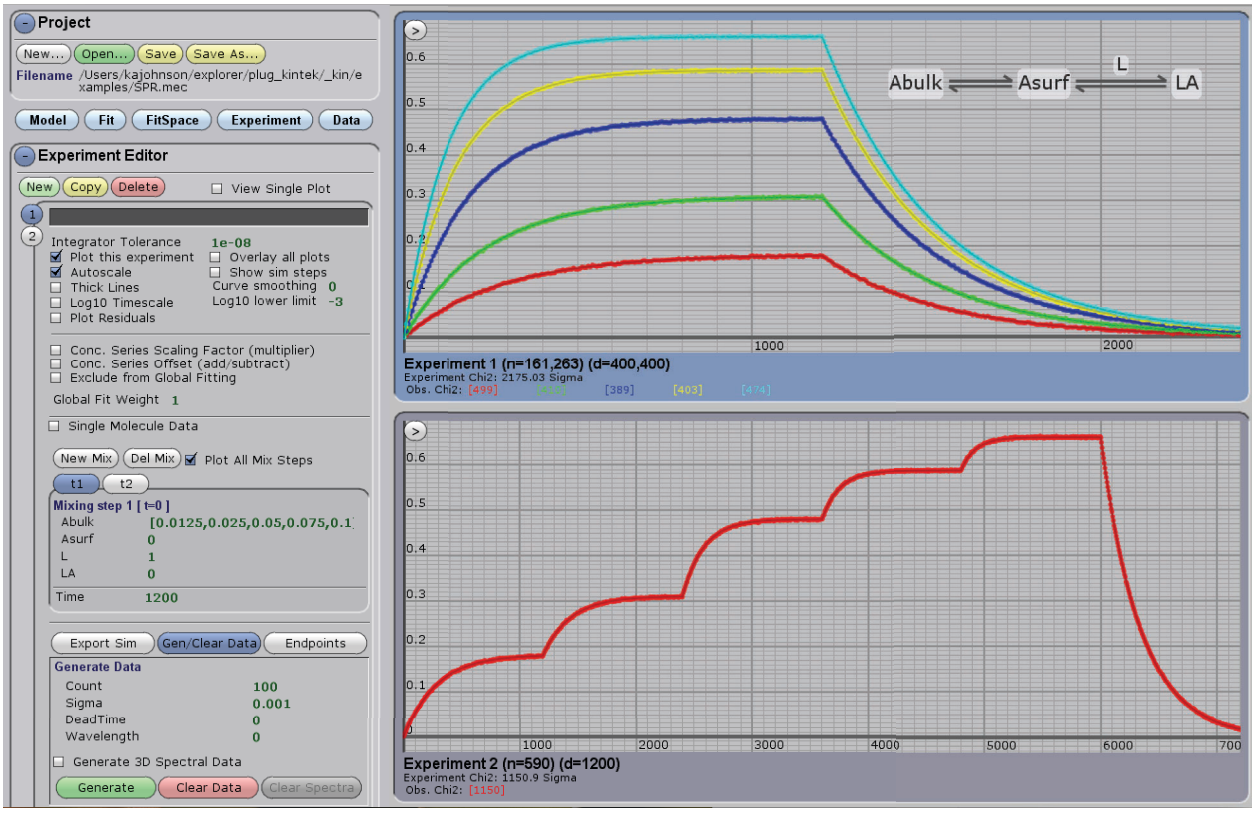
concentrations of the same species. Although you can fit conventional solution kinetics and SPR data simultaneously, the performance will be very slow. In this case, we suggest that you fit the conventional data and the SPR data in separate files initially, the combined then to do a final optimization.

One must exercise a great deal of caution in fitting SPR data because the mass transfer effects can dominate the kinetics. Care must be taken to perform the measurements with multiple loading of the receptor on the surface to control for the effects of mass transfer that arise when the rate of binding caused by a high concentration of receptor exceeds the rate of diffusion into the gel layer covering the surface. In the example below we use a simplified model in which the diffusion of the analyte from the bulk solution (A_{bulk}) to the surface layer (A_{surf}) is considered as a rapid equilibrium. This is then followed by the binding of the analyte at the surface to the ligand, L.

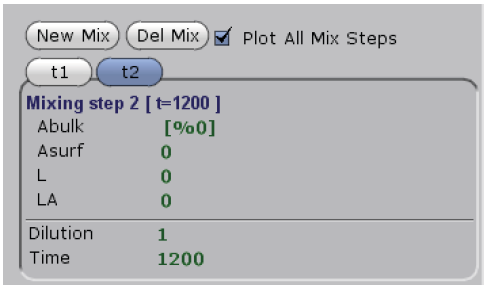
$$A_{bulk} \rightleftharpoons A_{surf}$$

$$A_{surf} + L \rightleftharpoons LA$$

The file SPR.mec shows two typical SPR experiments. In Experiment 1, five separate runs were performed at difference concentrations of the analyte, then washed out in the second mixing step. In Experiment 2, one run was performed with sequential additions of successively higher concentrations of analyte. Note that concentrations of A_{bulk} are listed in square brackets to denote that the concentrations are held constant, [0.0125, 0.025, 0.05, 0.075, 0.1…]. Although A_{bulk} is consumed during the binding reaction, it is replenished by the continuous flow of fresh solution.



The second phase of the reaction to wash out the analyte uses the expression [%0] as shown at the right.



The % specifies a new concentration, neglecting any carryover from the prior mixing steps.

The [0] specifies a fixed concentration of zero.

Using this simple modeling it is each to show how higher concentrations of L greatly perturb the kinetics of binding and dissociation. Try scrolling the concentration of L in the SPR.mec file included in the examples folder. At higher concentrations of L or slower diffusion rates (k_1 and k_{-1}), the curves become much slower. Without performing experiments at different concentration L, one cannot be sure tha the observed kinetics reliably reflect the intrinsic rate constants. This may explain the large systematic error often seen in published SPR data. *Fitting SPR data by simulation using KinTek Explorer does not guarantee that the results and interpretation are rigorously correct! Although that statement is true for all experiments, it is even more important to recognize in the case of SPR data.*

5.10 Pulse-chase experiments

A typical pulse-chase experiment is performed where enzyme is reacted with radio labeled substrate for various time intervals (t1) and then an excess of unlabeled substrate is added and the reaction is allowed to continue for a period of time sufficient to convert all tightly bound substrate to product (t2). The aim of this experiment is to define the kinetics of substrate binding. Of course, during the chase period, substrate can also release from the enzyme leading to smaller amounts of product, and a fraction of the diluted, radio labeled substrate can rebind to the enzyme and react to form product. KinTek Explorer allows you to model the experiment precisely; so all secondary effects are included in fitting the data. There are two options for modeling the data.

Option 1: Model the experiment exactly as it is performed.

E + A = EA = EP = E + P
E + B = EB = EQ = E + Q

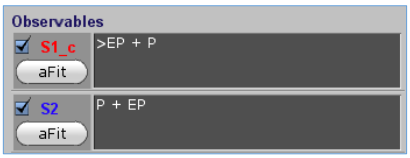
where A is the radio labeled substrate and B is unlabeled substrate. The dynein_burst.mec file provides an example of this approach (see Figure 5.7).

Rate constants are linked so the same rates govern the reaction of labeled and unlabeled enzyme. In this case, additional evidence indicates that k_{-1} and k_{-3} are negligible, so they are set to zero.

Reactions	k+	k-
E + A = EA	1 4.01	0
EA = EP	3 63.7	4 17.7
EP = E + P	5 8.34	0
E + B = EB	1 4.02	0
EB = EQ	3 63.7	4 17.7
EQ = E + Q	5 8.36	0

A double mixing experiment is set up so that the reaction starts in (t1) by the addition of radio labeled substrate (A) and the reaction is followed over 0.1 second. Direct acid quench gives the sum of EP + P formed (red trace). For the pulse chase, we add a high concentration of unlabeled substrate during t2 and set the duration to 1 second, which is sufficient to convert all bound substrate to product. By clicking on "t2" you can verify that the reaction time is sufficient.

Special symbol > We now employ a special symbol in the output definition, which means, "follow the reaction to the end of t2 and plot the result on the t1 time scale." This generates the time course shown in red in Figure 5.7. In contrast, the reaction shown in blue was obtained by stopping the reaction at the indicated times during t1.



The difference between the two curves reflects that fact that the EA complex can partition forward to form product during the t2 reaction phase (the chase). In earlier approximations to fitting this experiment, we were forced to assume that 100% of the EA went on to form product by fitting the pulse-chase experiment as the sum EA + EP + P (8). In the current approach to the problem, the kinetic partitioning of EA in the reverse versus forward directions is accounted for in the simulation.

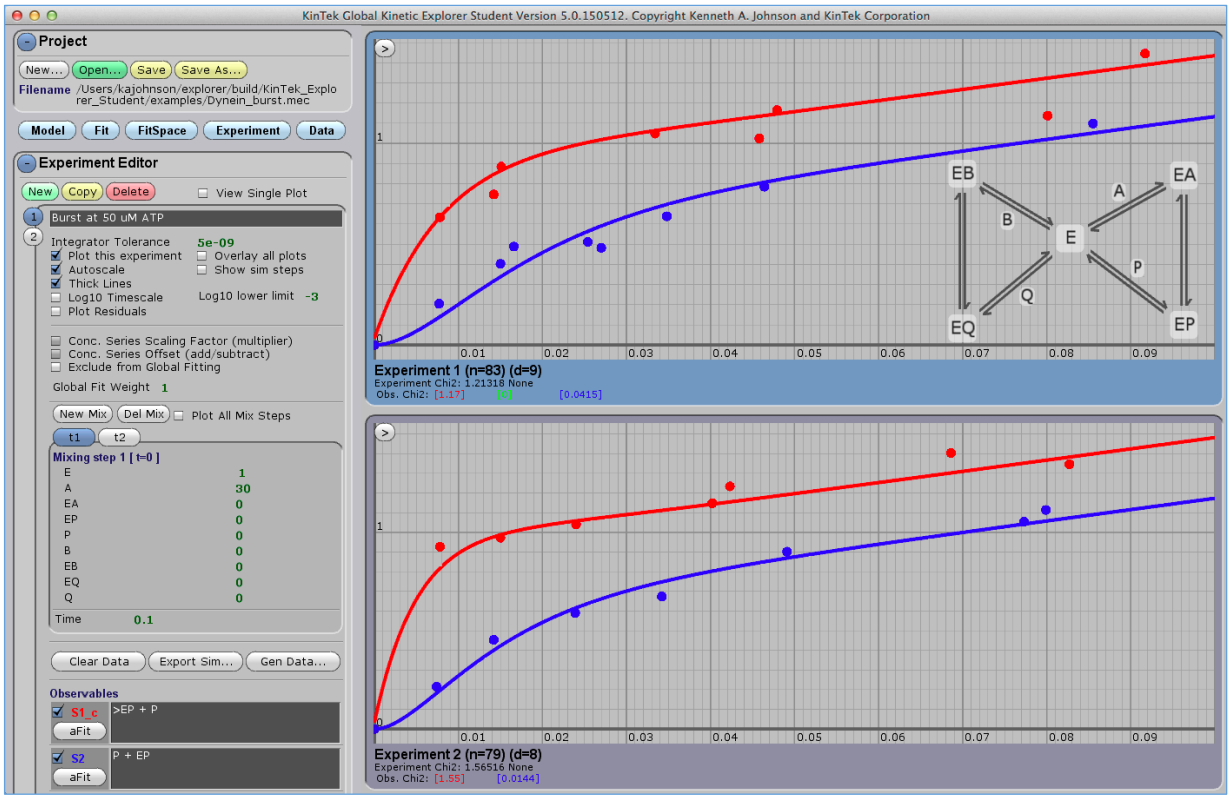


Figure 5.7 *Fitting the dynein ATPase pulse-chase experiments.*

Success of the experiment depends on the prudent choice of the t2 time period. You can inspect the reactions occurring during t2 to evaluate your choice based upon estimates of rate constants used to simulate the reaction. However, it should be noted that because the experiment is modeled exactly, no simplifying assumptions are required to interpret that data. For example, a simplistic interpretation of the experiment may require at least a 100-fold excess of unlabeled substrate; however, the experiment could be performed with only a tenfold excess of unlabeled substrate and rigorously interpreted by fitting with the complete model.

Option 2: Approximate the reaction with a substrate trap.

When reaction mechanisms are complex, it may not be feasible to completely mimic the parallel reactions for labeled and unlabeled substrates. In this case, one can approximate the pulse chase using a trap. This is illustrated with Experiment 2 of pulse-chase.mec.

E + A = EA = EP = E + P

A + trap = Atrap

By making the reaction of A with the trap fast and irreversible, one can approximate the pulse chase reaction by adding the trap during the t2 phase and again using the > symbol to get intermediates on the t1 timescale processed through the chase period.

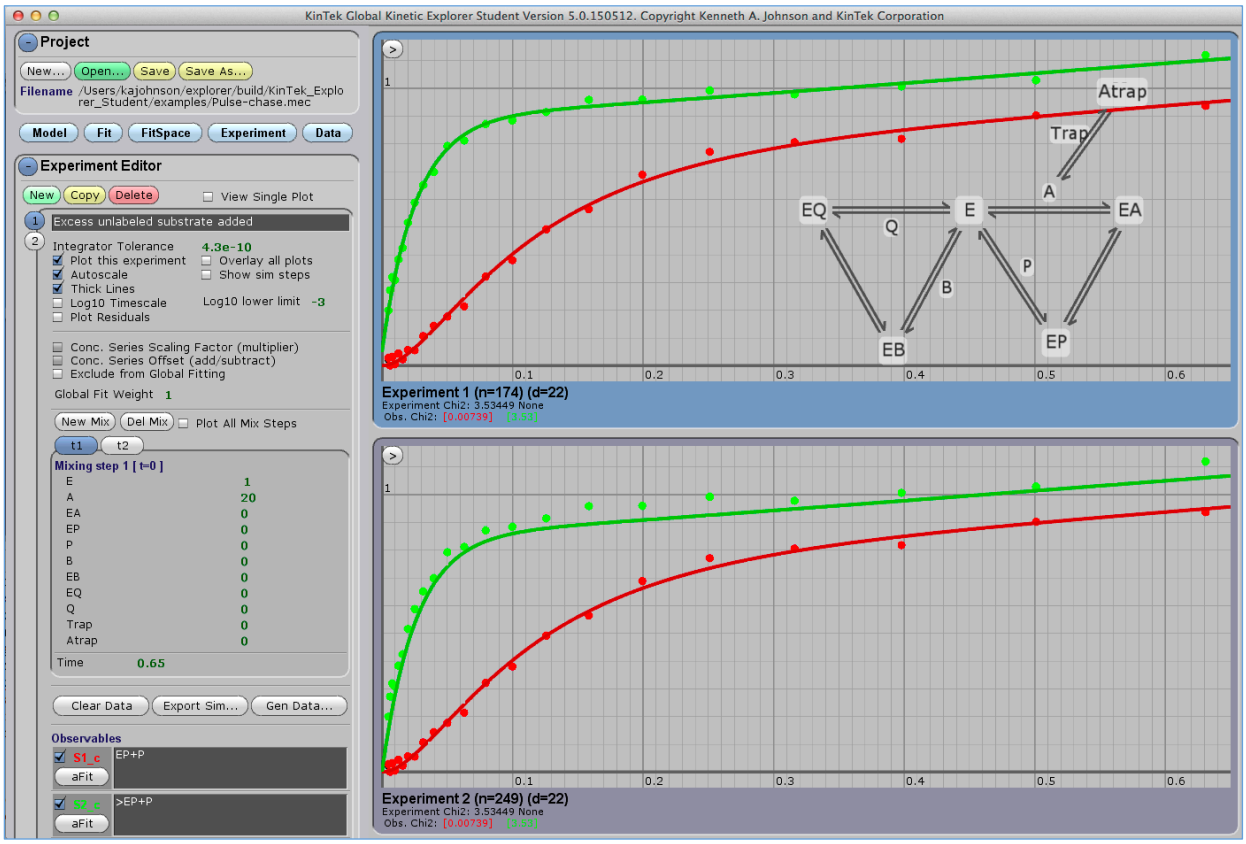


Figure 5.8 *Pulse-chase experiment fit to Substrate competition versus Trap model.*

Figure 5.8 shows a comparison of fitting data to the complete model (Experiment 1) versus the model involving a trap to mimic the experiment (Experiment 2). The only difference is due to the formation of product due to the rebinding of the diluted, radio labeled substrate during the chase period. Close inspection reveals that the simulated curve in Experiment 1 at time zero shows a small amount of product, which is due to the rebinding of the radio labeled substrate during t2. The simulated curve in Experiment 2 is offset from the data by this small amount. One could modify the output definition to be >EP+P+bkg and fit the data with a bkg of approximately 0.05. Alternatively, if one performs a proper control experiment by measuring the amount of product formed from reaction of the radio labeled substrate during the chase period (by mixing enzyme, A and B simultaneously), that amount can be subtracted from the data. One can then accurately fit the data using the simplified Trap model without that additional bkg variable.

Concentration Series

At this time, we do not support a full concentration series during a pulse-chase experiment. As illustrated in the dynein_burst.mec, reactions at different concentrations are accommodated in different experiments.

5.11 Simulating and fitting equilibrium titrations

With KinTek Explorer, we take a unique approach toward fitting equilibrium titration data. Rather than assuming that the reaction is actually at equilibrium, we simulation the endpoint of the reaction based upon the estimated kinetics of the reaction and the time allowed for equilibration. This has revealed interesting results in situations where the reaction was not really at equilibrium. Moreover, based on the power of data fitting based upon solution of complete models, assumptions normally necessary in fitting equilibrium titration need not be applied. Fitting is based upon the complete solution to the comprehensive model. In addition, note that equilibrium and kinetic data can be fit simultaneously, affording a new standard in rigorous data analysis.

There are several examples that illustrate the simulation and fitting of equilibrium titration data: EPSPS_titration.mec, Hemoglobin_pH6.5.mec, Hemoglobin_pH7.5.mec and myoglobin.mec. Figure 5.9 shows the simultaneous fitting of equilibrium and kinetic data for EPSP synthase.

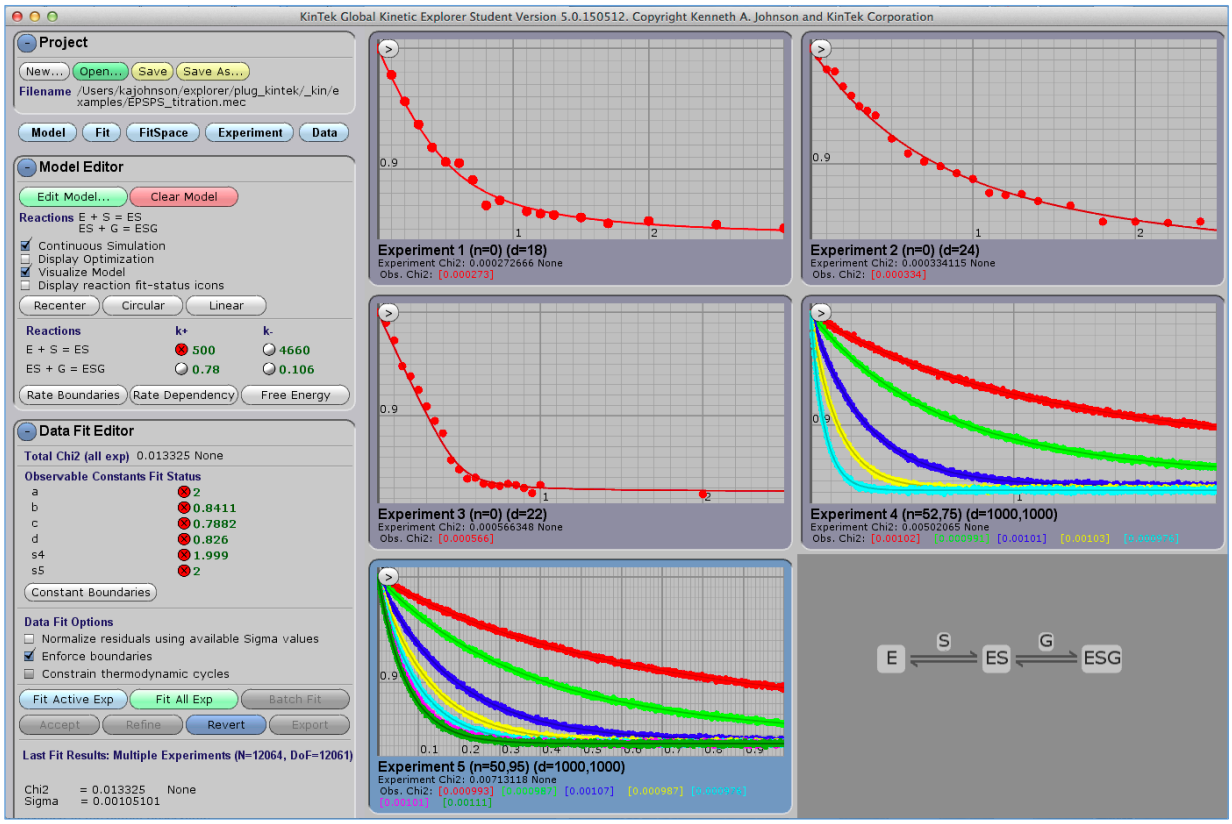


Figure 5.9 *EPSP synthase titration*. Data are shown for the equilibrium titrations and stopped-flow kinetic analysis of EPSP synthase using a change in protein fluorescence after binding S3P and glyphosate. (1) A mixture of 0.5 μM E saturated with 250 μM S3P was titrated with 0-3 μM glyphosate. (2) A mixture of 0.5 μM E with 2.5 μM glyphosate was titrated with 0-2.5 μM S3P. (3) A mixture of 0.5 μM E with 100 μM glyphosate was titrated with 0-2.5 μM S3P. (4) A mixture of 0.5 μM E saturated with 250 μM S3P was reacted in the stopped-flow with 1, 2, 5, 10, and 20 μM glyphosate. (5) A mixture of 0.5 μM E saturated with 20 μM glyphosate was reacted in the stopped-flow with 1, 2, 5, 10, 20, 50, and 100 μM S3P. Equilibrium titration data are from (9).

The data in Figure 5.9 were fit to the model shown at the right to get the values of K_d = 9.3 μM for S3P binding and K_d = 0.14 μM for glyphosate binding to the E.S3P complex. The fluorescence change was modeled with two scaling factors, **a** and **b**, as shown. Because no simplifying assumptions were needed in fitting the titration of E with S3P in the presence of glyphosate, this represents the first completely rigorous fitting of these data. Our published analysis was correct only to a first approximation (9).

$$E + S \underset{4650\ s^{-1}}{\overset{(500\ \mu M^{-1}s^{-1})}{\rightleftharpoons}} ES$$

$$ES + G \underset{0.106\ s^{-1}}{\overset{0.78\ \mu M^{-1}s^{-1}}{\rightleftharpoons}} ESG$$

$$F = a*(E + ES + b*(ESG))$$

Setting up an Equilibrium Titration: An equilibrium titration is specified in the output observable expression by using the following syntax.

@x-y,n
@ x-y specifies a titration with concentrations ranging from x to y
n gives the number of points to compute between the endpoint

For example, in Experiment 1 of *EPSPS_titration.mec*, the titration of E.S3P with glyphosate was specified by the term: **@0-3,100** meaning that the concentration of glyphosate was varied from 0 to 3 μM in increments to give 100 points. The program will compute the "equilibrium" endpoint at concentrations of 0, 0.03, 0.06, 0.18,…3.0 μM.

NOTE: Take care in choosing the value of **n** so that you have enough points to get a smooth curve without jagged straight-line segments, but not so many points that you slow down the simulation. Unless the titration curve has a complex shape with sharp bends, 30-100 points are sufficient to define the curve.

Each concentration requires an individual simulation to derive the endpoint of the reaction. Note that the time of reaction given in the Experiment Editor specifies the time allow for equilibration after the addition of each sample during the titration. The program computes the endpoint derived after the specified reaction time to calculate the "equilibrium" titration.

Optional modifiers:

[ ] Enclosing the term in brackets fixes the concentration of the titrant so that it is not consumed during the reaction. Two examples of when this is useful are with pH titrations, where the proton concentration is held constant by a buffer, and studies with a flow cell where the concentration of a reactant can be maintained by the flow.

For example, **[@0.001-3000,200]** allows 200 data points to be computed from 0.001 to 3000. This is an example for a pH titration, so the concentration of protons is held fixed at each state of the titration. Also you can use the pH designation rather than entering hydrogen ion concentrations.

NOTE: We no longer use the **exp** term added after the n to compute the concentrations on a logarithmic s scale; rather use the Log10 Timescale button in the Experiment Editor and make sure the titration does not start at zero.

☑ Incremental Titration: Click on this check-box when titrations are performed by sequential, incremental additions of the titrant to the same solution. For example, as described above, each reaction is simulated as the addition of 0.1 μM glyphosate to the mixture of species computed at the endpoint of the previous addition. This distinction may be important only if the reactions are slow and may not be at equilibrium during the measurement. In the alternative (unchecked box), each reaction begins by the addition of a given concentration of titrant. Most titrations are done by incremental addition to a single solution.

The figure below illustrates the method of endpoint simulation to mimic a batch titration (A) or an incremental titration (B). In either case, the final endpoint is plotted as a function of concentration (C).

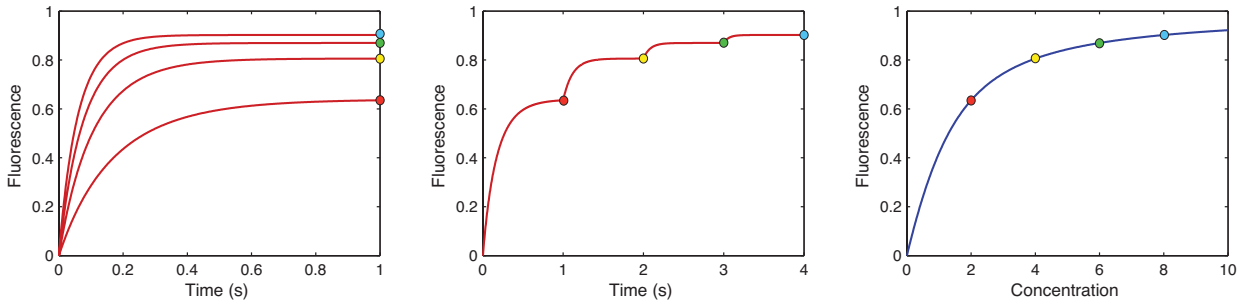


Figure 5.10 *Simulating equilibrium titrations*. A. Batch titration time course, where each reaction begins with a fresh sample of enzyme and substrate. B. Incremental titration time course, were a single enzyme solution is titrated by sequential additions of substrate. C. The concentration dependence of the titration is obtained from the calculated endpoint by either methods (Panels A or B).

5.12 Generating synthetic data

Gen Data... This function generates *synthetic* data to be used in evaluating fitting various models. In generating data, you specify the number of data points (count) and the sigma value governing the Gaussian error added to the data such that:

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2}$$

Note: The sigma value is not a percentage error, but rather has units corresponding to the y-axis scale and should be adjusted accordingly.

You can also specify the dead time of the data collection to generate a data set lacking data shorter than the dead time. For example, a typical stopped-flow instrument may have a dead time of 1-2 ms, so the first 0.001 or 0.002 seconds of the reaction is missed. In a hand-mixing experiment, the dead time can be 5-10 seconds. This function is particularly useful in two regards. First, you can simulate an experiment you plan to conduct based upon available information and then apply our statistical analysis including FitSpace, to evaluate whether the new experiment would address the questions you are asking to distinguish models or better define certain constants. It can also help to define the range of concentrations and time for data collection. Second, this function is particularly useful in evaluating published models. Simply enter the model and rate constants from a publication and then simulate the experiments described by the authors. You will be shocked to learn how often the models do NOT reproduce the original data, illustrating how often conventional data fitting fails. This is generally because the authors neglected to account for the variations in amplitude of the reaction.

Hint: Use this function in conjunction with the *aFit* function to generate synthetic data and then fit it to an analytic function to learn about the concentration dependence of an observed rate derived from a known

model.

5.13 Analytic function fitting (aFit)

This allows you to fit data or the simulation output to an analytical function selected from the list shown below. This *conventional data fitting* to an equation is not to be confused with fitting data directly to the model by simulation, which is the real purpose of the *KinTek Explorer* software. Analytic function fitting is provided here only to afford comparisons between *conventional data fitting* to equations and the modern data fitting directly to the model based upon numerical integration of rate equations.

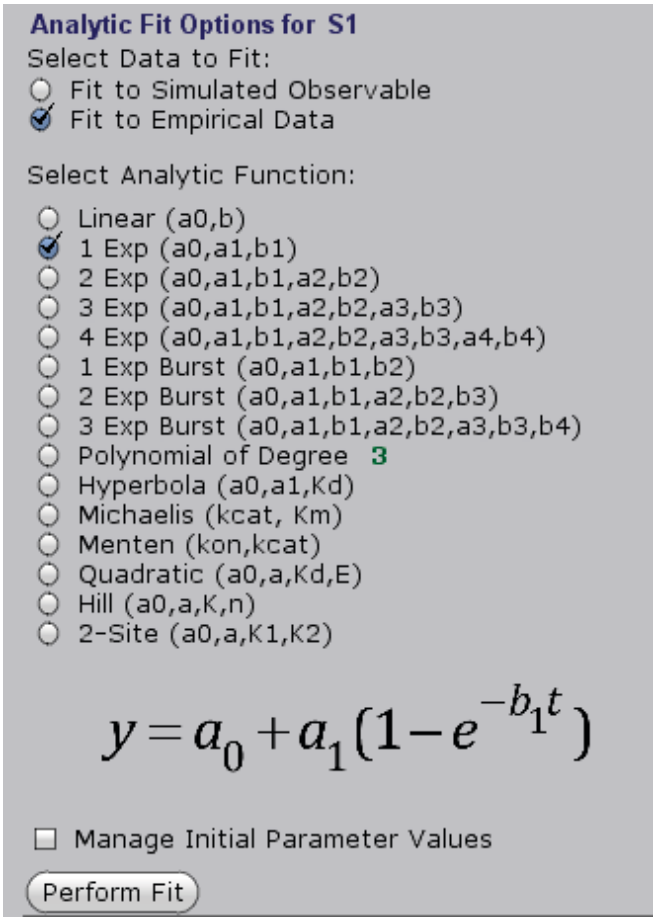
The *aFit* function is available under the *Experiment Editor* and under the *Data Repository*. There are subtle differences in the implementation of the analytic function fitting from the two locations. When calling *aFit* from the *Experiment Editor*, you can fit the data for each observable independently, or you can fit the results of a simulation without data. When calling *aFit* from the *Data Repository*, you fit the entire data set derived from a given data file but you do not have the option to fit the results of a simulation.

After fitting data to analytical function you can then plot the rate versus concentration (see below) to reveal underlying models. Patterns for different models are explained in (10). In addition, these functions are useful in that you can do a simulation of a complex model and then ask what rates you might have observed if you had collected data under a given set of conditions and fit by conventional means using a defined equation. You can then use this information to help to build your understanding of kinetics by fitting synthetic data, or you can predict an appropriate model when fitting real data. Finally, since most published data have been fit by conventional means, it is often instructive to take a published model and use it to generate synthetic data that mimics the original data. You can then fit the synthetic data to analytic functions and check to see whether a published model actually accounts for the original data.

Note that we automatically generate starting estimates for fitting by nonlinear regression and in some instances (particularly with lags), the program may not converge to a good fit because of inaccuracies in the starting estimates. In future releases we may include user input of starting estimates, but for now, please understand that the aFit function is only included to explore possible mechanisms and not a major priority in software development. If you really want more accurate estimates of observable rates from fitting data to multiple exponentials, you should analyze the data using standard software that allows user input of starting estimates to seed the nonlinear regression analysis.

a. Analytic functions

Nearly all useful function for fitting kinetic and equilibrium binding data are included in the *aFit* function. In either the *Experiment Editor* or in the *Data Repository*, you can click on *aFit* to bring up the menu shown below, select the desired function, then click on *Perform Fit*.



a. Select Data to Fit: Select either *Simulated Observable* output or *Empirical Data*.

b. Select *Analytic Function*

c. Click on *Perform Fit*

Figure 5.11 *List of available analytic fit functions.*

When you click on the button next to a function, the graphic below the selections displays the full equation. We use ***b*** rather than ***k*** to denote observed rates obtain in fitting to these equations. This is to emphasize that fitting to equations does NOT yield rate constants, but rather eigenvalues that are complex functions of rate constants in all but the simplest (irreversible reaction) cases.

NOTE: We have redefined exponential functions to make the amplitude terms more intuitively correct. For example for a single exponential function we use:

$$y = a_0 + a_1(1 - e^{-b_1 t})$$

rather than our previous use of:

$$y = a_1 e^{-b_1 t} + c$$

The latter can be confusing since a_1 is negative for a signal that increases over time, and ***c*** is the endpoint. With the new function, a_0 is the starting point of the signal and a_1 is positive for a signal that increases. These changes mainly useful in using our new function to "*Manage Initial Parameter Values*" where you can scroll a given constant and see that effects on the shape of the curve.

b. Table: Analytic functions used in conventional data fitting

Description	*Equation*
Linear	$y = a_0 + b \cdot t$
Single Exponential	$y = a_0 + a_1(1 - e^{-b_1 \cdot t})$
Double Exponential	$y = a_0 + a_1(1 - e^{-b_1 \cdot t}) + a_2(1 - e^{-b_2 \cdot t})$
Triple Exponential	$y = a_0 + a_1(1 - e^{-b_1 \cdot t}) + a_2(1 - e^{-b_2 \cdot t}) + a_3(1 - e^{-b_3 \cdot t})$
Four-Exponential	$y = a_0 + a_1(1 - e^{-b_1 \cdot t}) + a_2(1 - e^{-b_2 \cdot t}) + a_3(1 - e^{-b_3 \cdot t}) + a_4(1 - e^{-b_4 \cdot t})$
Burst with 1 exponential	$y = a_0 + a_1(1 - e^{-b_1 \cdot t}) + b_2 \cdot t$
Burst with 2 exponentials	$y = a_0 + a_1(1 - e^{-b_1 \cdot t}) + a_2(1 - e^{-b_2 \cdot t}) + b_3 \cdot t$
Burst with 3 exponentials	$y = a_0 + a_1(1 - e^{-b_1 \cdot t}) + a_2(1 - e^{-b_2 \cdot t}) + a_3(1 - e^{-b_3 \cdot t}) + b_4 \cdot t$
Polynomial*	$y = a_0 + a_1 \cdot x + a_2 \cdot x^2 + a_3 \cdot x^3 + a_4 \cdot x^4 + \ldots$
Hyperbola (equilibrium binding)	$y = a_0 + a_1 \cdot x / (K_d + x)$
Michaelis-Menten equation standard form	$y = k_{cat} \cdot x / (K_m + x)$
Menten (Michaelis-Menten equation for k_{on})	$y = k_{on} \cdot x / (1 + k_{on} \cdot x / k_{cat})$
Quadratic Equation	$y = a_0 + a \cdot \frac{(K_d + E + x) - \sqrt{(K_d + E + x)^2 + 4 \cdot E \cdot x}}{2 \cdot E}$
Hill Equation	$y = a \cdot x^n / (K_{50}^n + x^n)$
Two-site Equilibrium binding	$y = a_0 + a \cdot \frac{K_1 x + K_1 K_2 x^2}{1 + 2K_1 x + K_1 K_2 x^2}$

*The polynomial fit is useful only for estimating the standard deviation of data or for estimating initial rates. Note that if you fit data to a polynomial and then click on *Rate v Conc*, the program plots only a_1 versus concentration (the initial rate). The time dependence is fit to an nth order polynomial of the form:

$$Y = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \ldots + a_n t^n$$

where n is the degree of the polynomial. Higher values of *n* allow for greater complexity of the function but you must be careful when you have few data points. For example, a 9th degree polynomial will fit exactly to 10 data points with no apparent error.

c. Manage Initial Parameter Values.

We offer a new function to improve the initial parameter estimates for nonlinear regression data fitting or to see the relationships between parameters and the output curve. Click on the check box for "Manage Initial Parameter Values" to open a table of parameters for the selected equation. You can then scroll the values and see changes in the shape of the calculated curve (shown in light green in the figure below). These estimates will be used as starting values for nonlinear regression when you click on *Perform Fit*.

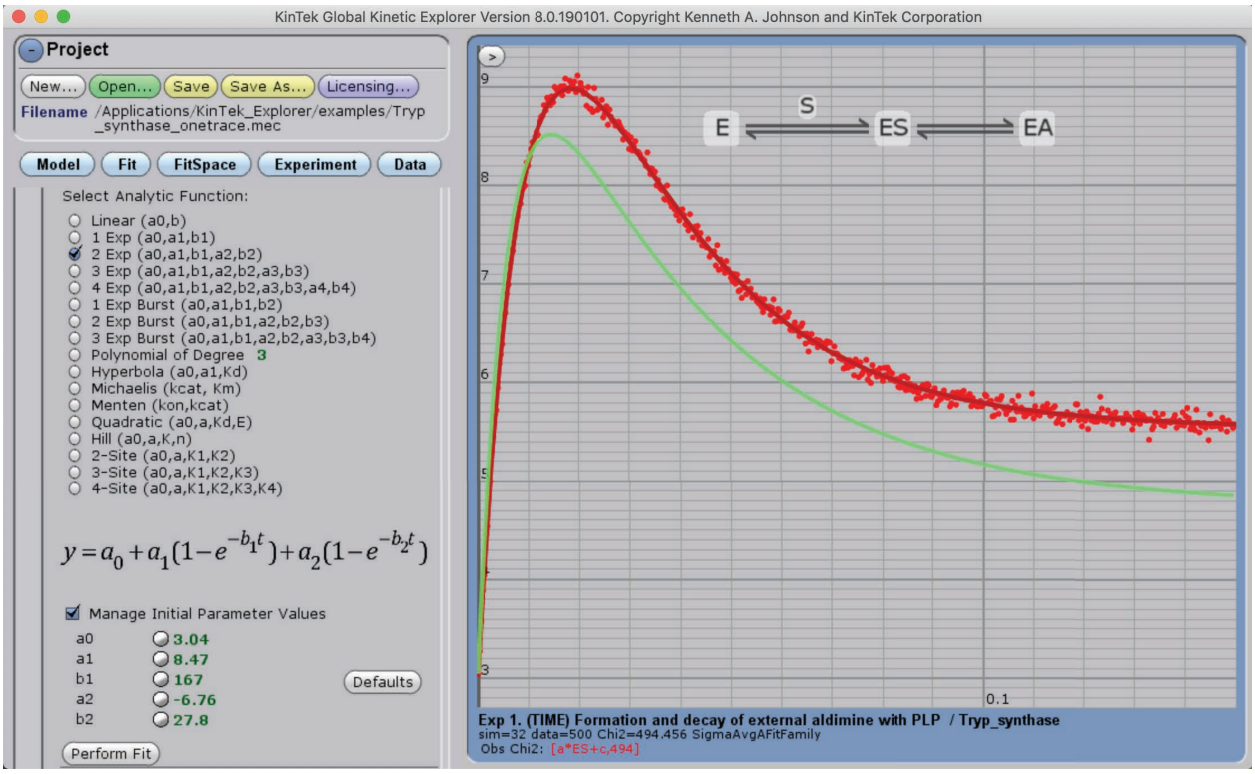
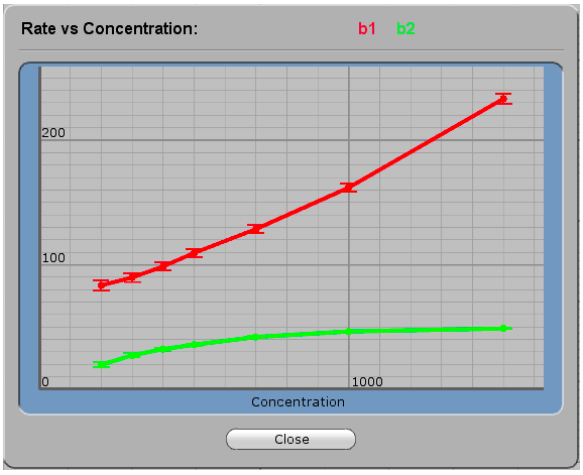


Figure 5.12 *Illustration of managing initial parameter estimate values.* You can scroll any of the values listed under *Manage Initial Parameter Values* so change their values and see how they affect the shape of the curve. This is useful with difficult to fit data and for learning the relationships between parameters and the shape of the curve.

d. Plot rate or amplitude versus concentration

After fitting data spanning a concentration series to an analytical function, you can click on *Rate v Conc* and get the display of the concentration dependence of the observed rates. The example below (Figure 5.13) shows the results after fitting the Tryp_synthase data to a double exponential, revealing the pattern for a two-step reaction. The example demonstrates the kinetic signature of a two-step reaction in which the rates of both steps are comparable. Note: additional concentrations were added to the Tryp_synthase.mec file to show a clearer picture of the concentration dependence of the rate and amplitude.



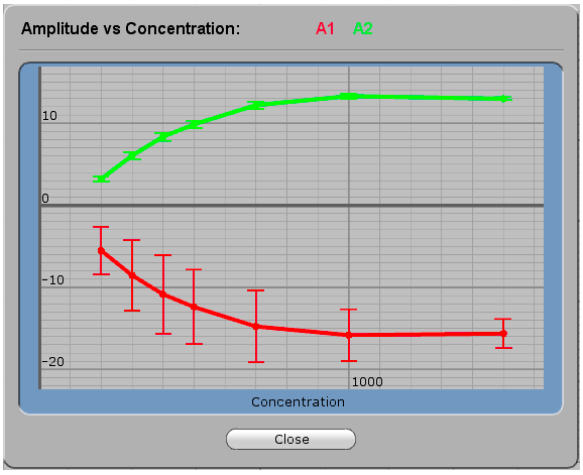


Figure 5.13 A.Rate versus concentration plot. B. Amplitude versus concentration plot.

The slope of the fast phase provides an estimate of the second order rate constant for the binding step (k_1), while the maximum rate of the slow phase gives an estimate of the sum of $k_2 + k_{-2}$ according to the mechanism:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} FS$$

This analysis can guide you in developing a model and getting estimates for kinetic constants. However, note that in fitting more than one exponential there are serious limitations in the ability to extract reliable rate estimates. Analytic fitting should only serve as a guideline to developing the model and deriving initial estimates for rate constants.

NEW starting with Version 5.2. Shift click on a point to eliminate it from the trace. This is useful when the exponential fit function is not valid over the entire concentration range and gives meaningless number that throw off the scale.

New starting with Version 6.0. With the Professional version you can output the results of analytical equation data fitting to generate publication quality figures. This requires installation of Python Anaconda as described in *Section 1.2a*.

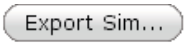
e. Why conventional data fitting often fails

There are numerous reasons why conventional fitting of data fails to reveal the correct mechanism or fails to provide the correct rate constants. (1) Simplifying assumptions in deriving the analytic functions may not be valid (e.g., pseudo-first order conditions with one reactant in 10x excess). (2) Fitting of data to multiple exponentials is unreliable because inherent relationships between rates and amplitudes are lost. (3) Simplifying assumptions in deriving the concentration dependence of the rates may not be valid. (4) The concentration dependence of the amplitude of reactions is often overlooked and may be inconsistent with the model derived from the rate dependence only. It is for these reasons and more that we fit the data globally directly to the model based upon numerical integration of the rate equations rather than fitting to equations to derived Eigenvalues.

5.14 Outputting simulations

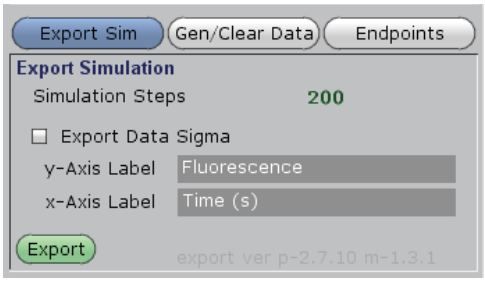
There are three methods to output the results of simulations. First, you can either simply do a screen capture of the simulation results. If you have the software license, you can export the simulations as x,y data that can then be imported into a graphics program of your choice to make publication quality figures. You can also directly generate publication quality figures; this requires the licensed version and installation of a Python graphics package.

a. Exporting simulation results as a publication quality figure



Click *Export Sim.*. under the *Experiment Editor* to output the results of a simulation.

If Python is installed, you will get the menu at the right after clicking on "Export Sim".



Enter the axes labels in the menu shown, then click *Export* and follow the menu instructions for controlling the output. You can select a directory for saving the image files. See Chapter 11 for more information on the output of simulations.

b. Exporting text files for simulation results

The *Export* function outputs publication quality figures in both .eps and .png format (300 dpi). The text files needed to create you own graphics are output in a subdirectory called *Text & cfg files*. See *Chapter 11* for more information.

In addition, when exporting the simulation results, the program will also write two text files, one containing your data and another containing the results of the simulation in tab-delimited columns beginning with time, y1, y2, etc. These text files can then be read into a graphics program of your choosing to prepare publication-quality figures. Two files will be output, one labeled filename.sim.txt containing the results of the simulation and another, filename.dat.txt outputting the data, where "filename" is the name you assign when exporting the simulation. If there are multiple concentrations, the output is an array in which the first column contains the time and subsequent column contain the simulation outputs for the various concentrations.

In the /documents subdirectory, we provide a MATLAB script, *SimPlot.m,* that will read the two text files and generate a publication quality graph. The script generates both a .PNG file and a publication-quality .EPS file. The EPS file can be imported into Adobe Illustrator for further editing, such as altering font size and labeling. Two examples of the output are shown here:

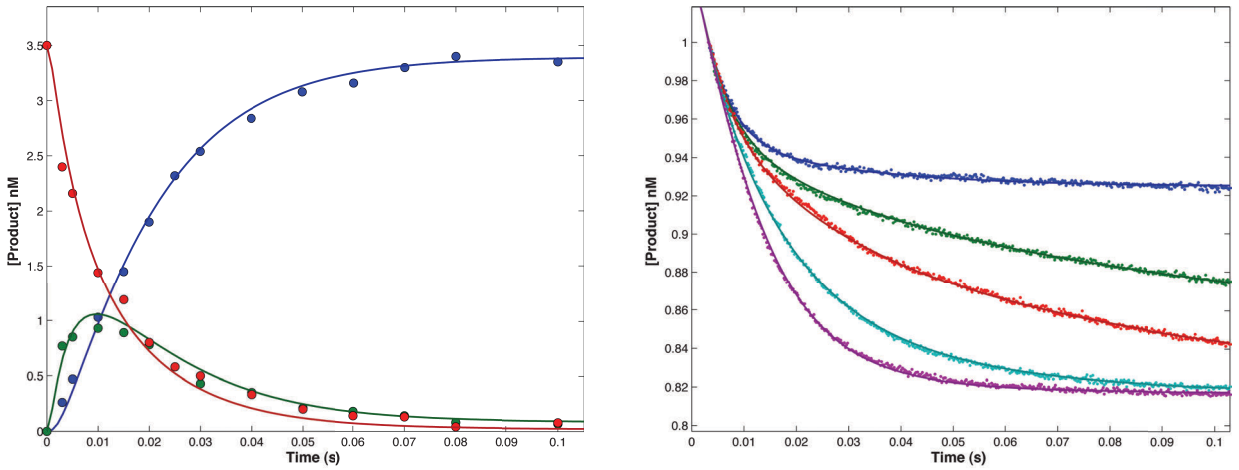


Figure 5.14 *Publication quality figures*. These figures were generated using the MatLab script; SimPlot.m from the text file outputs from the example files EPSP.mec and kinesin_mantADP_wt.mec.

c. Screen capture of simulation

For lab reports and notebooks, it is often useful to record the output of a simulation using a screen capture function. On the Mac, use the included utility "Grab" to capture and save the entire window from the simulation program or a selected portion. Under Windows, there are various utilities. Ctrl-PrintScreen captures the entire screen to memory. The image can then be pasted into a graphics utility program, such as Photoshop or Paint. Another useful alternative is to use one of the free screen capture utilities such as *Gadwin* so that the Print Screen key automatically saves the screen to a file. On Mac OSX, the function *Grab* can be used or the shortcut *Command-Shift-4* to select a portion of the screen to be saved to the desktop.

Chapter 6. Data Repository

Starting with version 2.2, you must import data into the *Data Repository* and then, in a subsequent step, *assign* that data to an observable output for a given experiment. This allows you to enter data without having to first develop a model, and the data stays with the mechanism file (*.mec) in the *Data Repository* while you develop or change a model and experiments. In a second step, you will *assign* the data to a given experiment in the model. See *Chapter 10* for importing time-resolved spectra.

6.1 Data file formats

Data to be imported must be a tab-delimited text file, which can be created using a spreadsheet program or text editor. In Microsoft Excel, save as "Tab Delimited Text (.txt)." There are several types of data files that can be imported as described in tables the below. It is highly recommended that you prepare your data files using a spreadsheet program such as Excel or the free Google documents spreadsheet editor. In either case, save the file as tab-delimited text. Use of a standard text editor can cause problems if you enter extra tabs in an attempt to make the columns line up on your computer display. Also take care that the spreadsheet program has not rounded off the numbers by displaying too few decimal places; this can lead to identical numbers in sequence in the time column which will be flagged as an error and prevent the importing of data.

In addition to the data, there can be a *File Header* with text to describe the experiment; the program will recognize and skip over the header. The line immediately preceding the data should contain *column headers* that describe the content of each column as either *"Time"*, a *species name*, or a *concentration* as illustrated by the following examples.

Table 6.1 *Data File Structures*

Type I: Single Trace

Time	P
$t_{i...}$	$y_{i...}$

Type II: Concentration series, one timescale

Time	5	10	20	50
$t_{i...}$	$y_{i...}$	$y_{i...}$	$y_{i...}$	$y_{i...}$

Type III: Concentration series, multiple timescales

Time	5	Time	10	Time	20
$t_{i...}$	$y_{i...}$	$t_{i...}$	$y_{i...}$	$t_{i...}$	$y_{i...}$

Type IV: Multiple Species, one timescale

Time	S3P	EPSP	Int
$t_{i...}$	$y_{i...}$	$y_{i...}$	$y_{i...}$

Type V: Multiple Species, multiple timescales

Time	A	Time	B	Time	C
$t_{i...}$	$y_{i...}$	$t_{i...}$	$y_{i...}$	$t_{i...}$	$y_{i...}$

If the column headings are not part of the text file, you can add them while importing the data into the program. *You must use the word "Time" (case insensitive) to denote the column containing the time data*. Do not enter units such as Time (s). In parsing the data file, the program looks for spaces between items and will assign (s) to a new column header. Similarly, concentrations should be numbers only, not [S] = 10 or 10 µM, for example. This enables the program to read these headers and use the concentrations subsequently, in plotting rate versus concentration after fitting the data to an analytic function and in assigning data to experiments.

There is no limit on the size of data files. However, if you experience slow performance in fitting a large number of data files with more than 1,000 - 2,000 data points per trace, you can improve performance without sacrificing information content by reducing the number of points by data averaging the data when it is imported (see below).

The menu for importing data is shown below (see ahead, Figure 6.2), and is illustrated using a concentration series (variable substrate) for a simple binding reaction. Note that the first two lines of the text file contain information about the experiment that is recognized by the program and displayed in the *File Header* box. Column headings can be edited (or added) when the data are being imported.

a. Including sigma values in data files

If you have measured the standard deviation for each data point, or know the average standard deviation, the program will take advantage of that information in fitting the data. See Chapter 9 on *Statistical Analysis in Data Fitting* for more explanation for why you should always estimate sigma. To include the sigma values, add a column of data after each column of Y-values with the header having the reserved word "sigma" (case insensitive) as illustrated below. If sigma (σ) values are known, then chi^2 is calculated using the known sigma values (often called a weighted least squares). The data table can be formatted to have a sigma value associated with each data point as shown in Table 6.2 or with a single sigma value for each trace as shown in Table 6.3.

Table 6.2 *Data File Structure with Individual Sigma Values*

Single Trace

Time	P	sigma
$t_{i...}$	$y_{i...}$	$\sigma_{i...}$

Concentration series, one timescale

Time	5	sigma	20	sigma	etc...
$t_{i...}$	$y_{i...}$	$\sigma_{i...}$	$y_{i...}$	$\sigma_{i...}$	

Concentration series, multiple timescales

Time	5	sigma	Time	10	sigma
$t_{i...}$	$y_{i...}$	$\sigma_{i...}$	$t_{i...}$	$y_{i...}$	$\sigma_{i...}$

Multiple Species, one timescale

Time	S3P	sigma	EPSP	sigma	etc...
$t_{i...}$	$y_{i...}$	$\sigma_{i...}$	$y_{i...}$	$\sigma_{i...}$	

Multiple Species, multiple timescales

Time	A	sigma	Time	B	sigma
$t_{i...}$	$y_{i...}$	$\sigma_{i...}$	$t_{i...}$	$y_{i...}$	$\sigma_{i...}$

This table illustrates the data format where each time-point has an associated y-value and a sigma value.

Table 6.3 *Data File Structure with Average Sigma Values*

Type I: Single Trace

Sigma	σ
Time	P
$t_{i...}$	$y_{i...}$

Type II: Concentration series, one timescale

Sigma	σ_1	σ_2	σ_3	σ_4
Time	5	10	20	50
$t_{i...}$	$y_{i...}$	$y_{i...}$	$y_{i...}$	$y_{i...}$

Type III: Concentration series, multiple timescales

Sigma	σ_1		σ_2		σ_3
Time	5	Time	10	Time	20
$t_{i...}$	$y_{i...}$	$t_{i...}$	$y_{i...}$	$t_{i...}$	$y_{i...}$

Type IV: Multiple Species, one timescale

Sigma	σ_1	σ_2	σ_3
Time	S3P	EPSP	Int
$t_{i...}$	$y_{i...}$	$y_{i...}$	$y_{i...}$

Type V: Multiple Species, multiple timescales

Sigma	σ_1		σ_2		σ_3
Time	A	Time	B	Time	C
$t_{i...}$	$y_{i...}$	$t_{i...}$	$y_{i...}$	$t_{i...}$	$y_{i...}$

This table illustrates the data format when each column of y-values (each trace) has a single average sigma value (σ_i). If all data within a file have the same standard deviation, then enter the same sigma value for each column.

Note: If sigma values are not known, we compute the average sigma value after fitting the data to an appropriate function:

$$\sigma_{AVE} = \sqrt{\frac{\sum_{i=0}^{N-1}\left[y_i - y(x_i)\right]^2}{N-1}} \text{ at the best fit valus of } a_0 ... a_{M-1}, \text{ where } y(x_i) \equiv y(x_i \mid a_0 ... a_{M-1})$$

This is done using the aFit (Analytical Fit) function as described *Section 6.4*. The best fit of each to an analytical function sets the standard for an optimal global fit based upon the estimated sigma values. In Chapter 9, we describe the importance of having an estimate of the standard deviation (sigma) during data fitting, and explain our rationale for fitting to an analytical function to estimate sigma values when they are not known.

b. Use “null” or "NaN" for missing data

A missing data point in the y-value listing can be identified by the reserved word "null" or "NaN", which is the Matlab standard for "not a number". The entry is not case sensitive, so "nan" can be used as well. Otherwise, leaving blank results in an entry with a value equal to zero. When parsing the file, the program will know to skip this entry. This is useful for quench-flow data where you may have planned an experiment to collect all data on the same time scale, but something happened (i.e., a tube broke or was dropped) so that a single time point was lost.

The example below illustrates how the absence of a value can be indicated by the term “NaN.” Note that the real data in the EPSP.mec file does not have null values at these positions. It also illustrates

the file heading used to designate multiple species for a given experiment. In this case, you must then assign each column of data to a given species output signal (**S1**, **S2**, **S3**, etc).

Table 6.4 *Example data file showing multiple species and null values.*

File header ⇨	EPSP single turnover experiment performed with 10 uM enzyme, 3.5 uM PEP and 100 uM S3P.			
Column header ⇨	Time	PEP	Int	EPSP
Data ⇨	0.0	3.5	0.0	0.0
	0.003	2.4	0.77	0.26
	0.005	2.16	0.85	0.47
	0.01	1.44	0.93	1.03
	0.015	1.19	0.89	1.45
	0.02	0.8	**NaN**	1.90
	0.025	0.58	0.58	2.32
	0.03	0.5	0.43	2.54
	0.04	0.33	0.35	2.87
	0.05	0.2	0.21	**NaN**
	0.06	0.14	0.18	3.16
	0.07	0.13	0.14	3.3
	0.08	0.04	0.08	3.4
	0.1	0.08	0.07	3.35

c. Importing time-resolved spectra

For importing and fitting time-resolved spectra or spectra as a function of concentration, please see Chapter 10. We allow two data file formats: one with wavelengths listed by row (Table 6.5) and one with wavelengths listed by column (Table 6.6). Note that in this example, the spectrum at zero time was based upon taking an absorption spectrum before mixing. The spectrum of the starting material was then manually added to the table of time-resolved spectra.

Table 6.5. *Sample data file format (tab-delimited text) with wavelengths by row*

Wave/t	0.0000	0.0010	0.0022	0.0037	0.0052	0.0067	0.0082
323.8	0.0732	0.0753	0.0811	0.0860	0.0892	0.0942	0.0910
324.6	0.0748	0.0768	0.0800	0.0848	0.0903	0.0927	0.0944
325.5	0.0748	0.0744	0.0796	0.0864	0.0895	0.0918	0.0910
326.3	0.0763	0.0749	0.0799	0.0864	0.0893	0.0916	0.0908
327.1	0.0762	0.0748	0.0775	0.0837	0.0865	0.0887	0.0901
327.9	0.0798	0.0775	0.0780	0.0840	0.0868	0.0909	0.0901
328.8	0.0815	0.0754	0.0797	0.0836	0.0882	0.0902	0.0914

Table 6.6. *Sample data file format (tab-delimited text) with wavelengths by column*

Time/w	323.8	324.6	325.5	326.3	327.1	327.9	328.8
0	0.0732	0.0748	0.0748	0.0763	0.0762	0.0798	0.0815
0.001	0.0753	0.0768	0.0744	0.0749	0.0748	0.0775	0.0754
0.0022	0.0811	0.0800	0.0796	0.0799	0.0775	0.0780	0.0797
0.0037	0.0860	0.0848	0.0864	0.0864	0.0837	0.0840	0.0836
0.0052	0.0892	0.0903	0.0895	0.0893	0.0865	0.0868	0.0882
0.0067	0.0942	0.0927	0.0918	0.0916	0.0887	0.0909	0.0902
0.0082	0.0910	0.0944	0.0910	0.0908	0.0901	0.0901	0.0914

6.2 Importing data

Import... After clicking on *Import*, a dialog box appears as shown in Figure 6.1. The default data file extension is .txt.

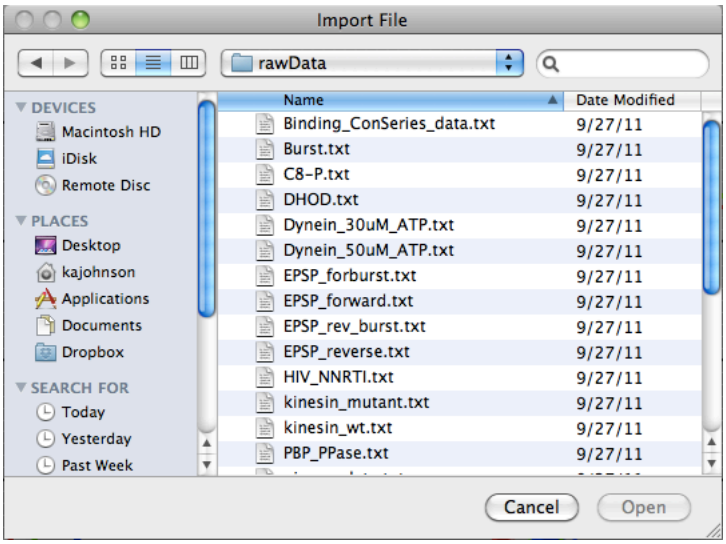
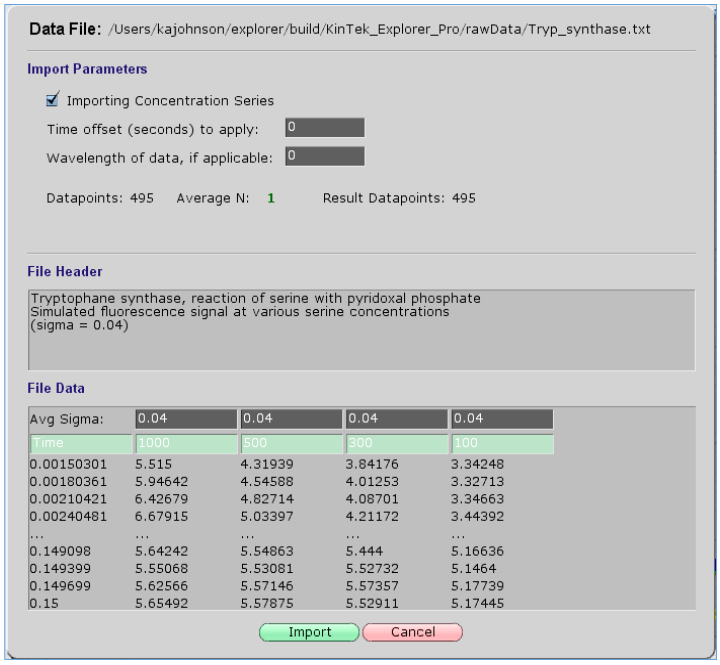


Figure 6.1 *Import File menu*.

a. Data import menu



The figure on the left shows the menu for importing or editing data.

When the data are being imported, you can apply a time offset to correct for the data lost during the dead time or mixing time in your experiment. Also, you can average *N* number of data points to reduce the total number.

The program will recognize text at the beginning of a file as a *File Header*.

The line preceding the number is recognized as the column header. The word "Time" or "time" must be used to designate the column(s) containing the time data. The remaining column headers can contain the concentration or the name of the species (see below). DO NOT include units. Values for the column headers can be part of the file or can be entered in the *Data Import Menu*.

Figure 6.2 Data Import Menu.

Enter *Average N* to establish the number of points that will be averaged. For example if the data file has 2000 points and N=4, then you end up with a total of 500 data points, each of which was derived from the average of 4 points. We do not recommend using this simple averaging if your data is on a

logarithmic time scale.

b. Dead time offset

The *time offset* can be set to correct for the dead time of the instrument, as shown in this example where 0.001 seconds will be added to all time values. This is needed because the simulations always start at time = 0 after mixing and amplitudes of the reaction are an important parameter in fitting data by simulation.

In Figure 6.3 below at the right, the data begin at 1.5 ms, the approximate dead time of the stopped-flow instrument. Note how the fitted curves extrapolate to the same starting point at each concentration (Figure 6.3, right). Without the time offset, each trace appears to begin at a different starting point (Figure 6.3, left). Entering an appropriate time offset is essential for performing global data fitting.

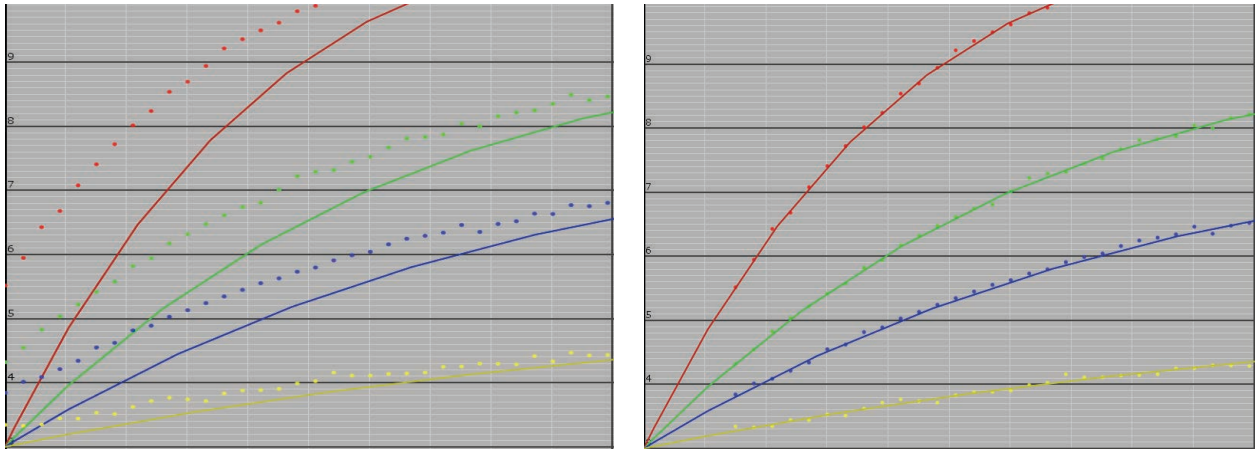


Figure 6.3 *Time offset.* These figures illustrate the difference between data with no time offset (left) and data with the appropriate time offset (right).

c. Displaying, editing, fitting and deleting data

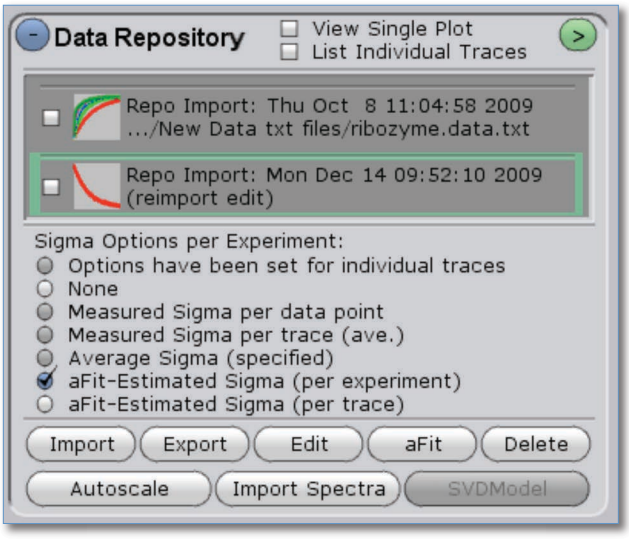
Each data file is listed in the text box of the *Data Repository* along with a thumbnail view of the graph.

Click on the *checkbox* and graph will appear showing the selected data (Figure 6.4) with a green border.

Click on the data set to select a given experiment in order to alter the *Sigma Options* or *Edit* the data, fit to an analytic function (*aFit*) or *Delete* the data.

☑ List Individual Traces applies to the selected experiment.

To hide the Data Repository, click on the minimize button.



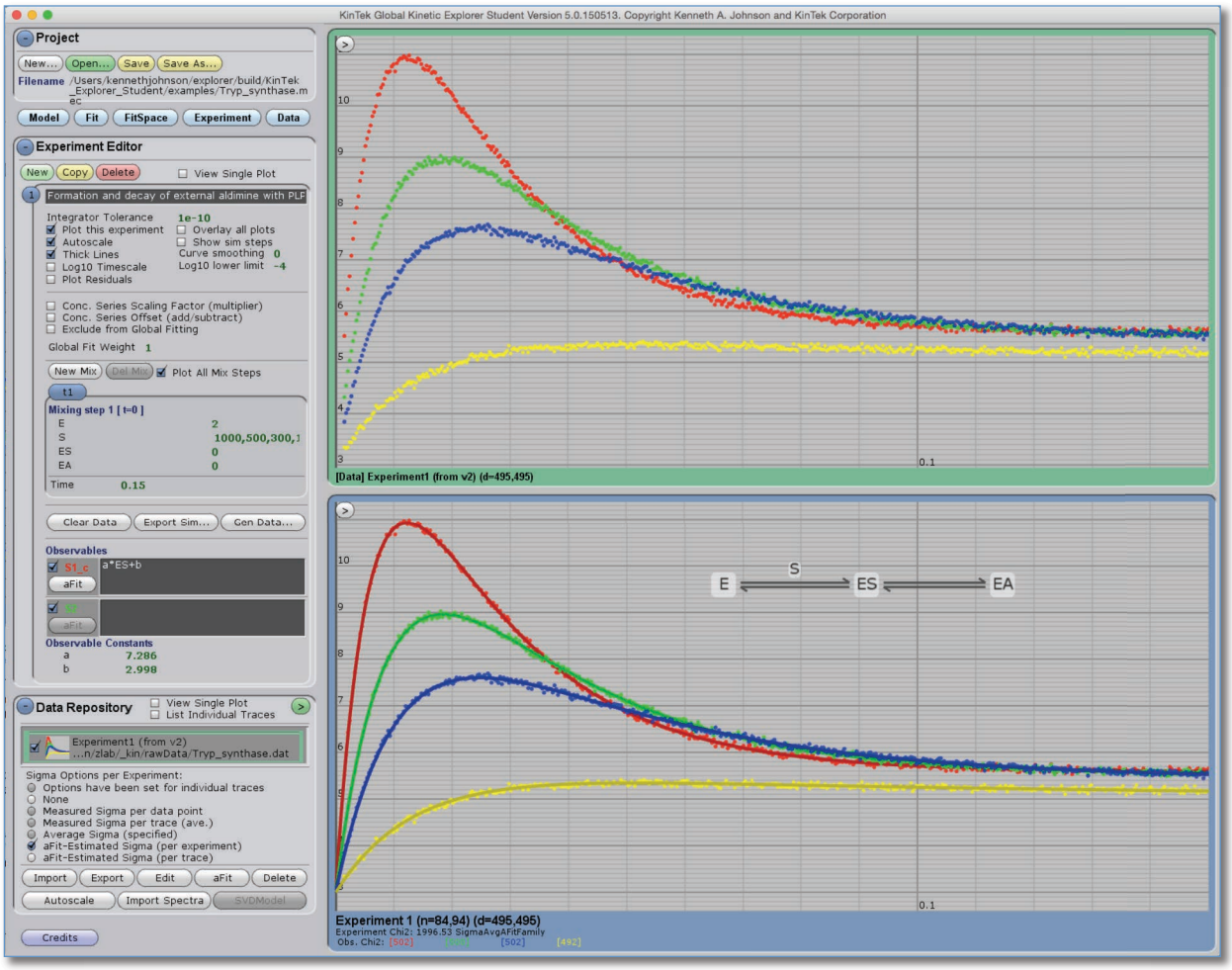


Figure 6.4 *Data Repository Display*. This figure shows the distinction between the display of data in the repository (upper figure outlined by a green boundary) and the simulation output (lower figure outlined by a blue boundary). When the data are *assigned* to a given experimental output, they are mapped onto the selected experiment, as described in *Section 6.5*.

Click on the text listing in the *Data Repository* to select a given dataset. The selected data is outlined in green, here and on the plot (if the checkbox is checked). The selected dataset can then be edited, fit or deleted:

Edit... As if re-importing the data the file can be edited and re-imported to change the dead time offset and increase the number of points averaged to achieve a final data set. Note: you cannot reduce the number of points averaged since the original data is lost in averaging. To reduce the number of points averaged, re-import the original data file.

aFit The data can be fitted to an analytical function to get a plot of observed rate versus concentration for a concentration series, for example. See *"aFit Analytic Function Fitting"* for more details. Click *Back* to close the fitting dialog. This function is also used to estimate sigma values (standard deviation) for data sets that do have a known sigma estimate.

Delete Click to delete data from the .mec file.

d. Sample data file with multiple timescales

Figure 6.5 illustrates how to import data collected on different time scales.

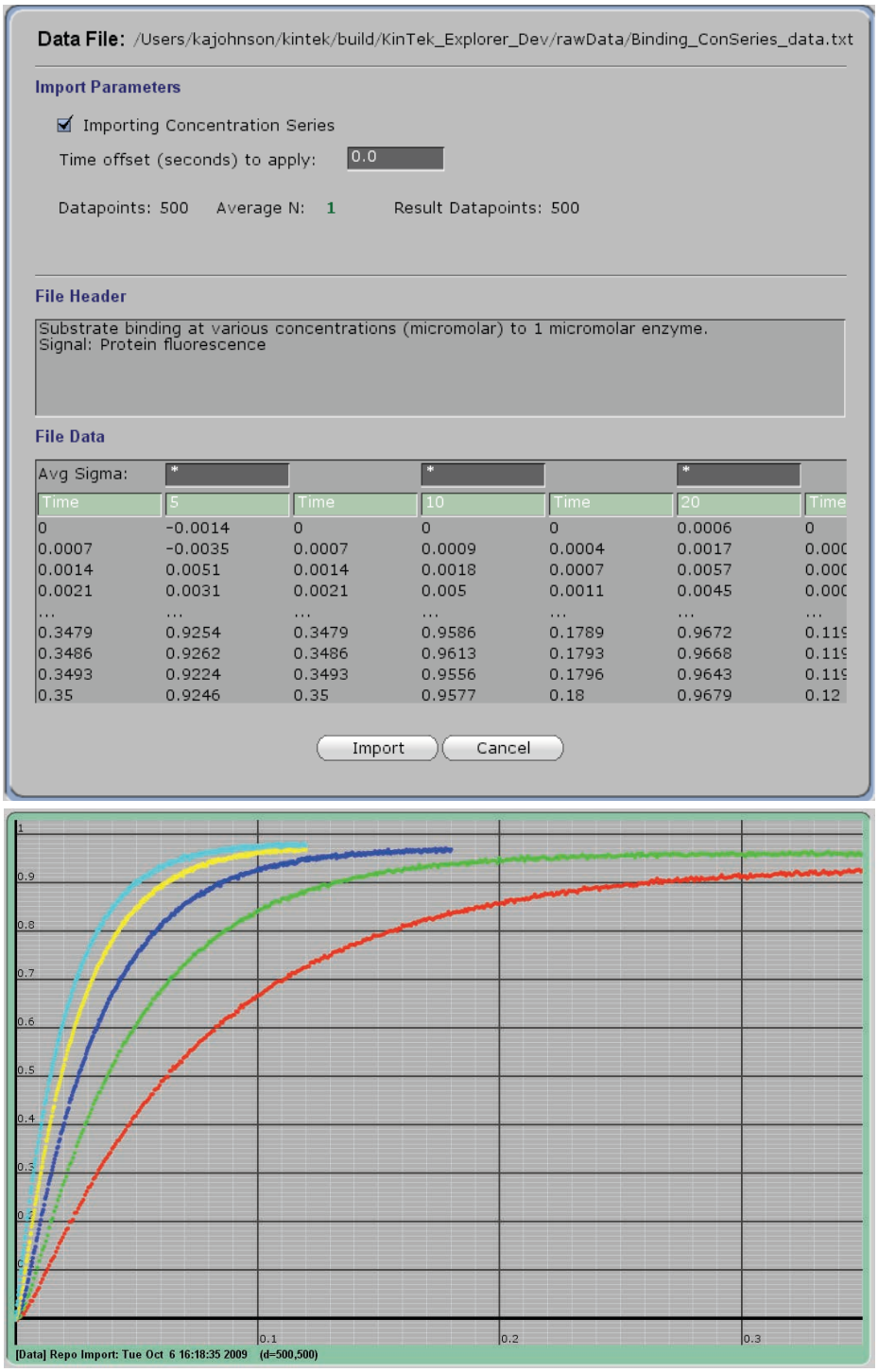


Figure 6.5 *Data file with multiple time scales*. Note the column headers identifying the times and variable concentrations in the table. NOTE: Right-click and drag the data table to see all the columns.

6.3 Fitting data to analytic functions

After data have been imported into the Data Repository, the data can be fit to an analytic function as described above under *"aFit Analytic Fitting"*. Analysis of the concentration dependence of the observed rates can serve as a guide in developing a model and can be used to estimate the standard deviation of the data.

Under *Data Repository*, click on the data text description or the thumbnail data figure to select a given experiment, then click *aFit*, select the function and then click *Perform Fit*. The results of the fitting will be listed and the graph will show the overlay of the best fit on the data.

Click *Back* again to close the fitting dialog.

Click *Rate v Conc* to display a graph of rate versus concentration.

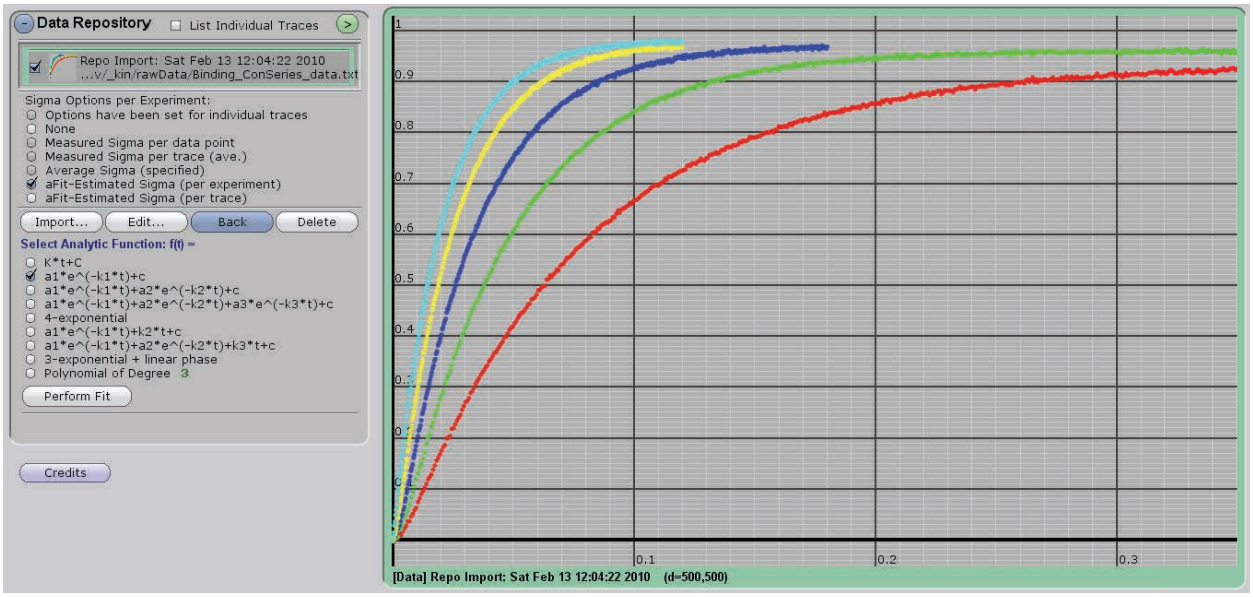


Figure 6.6 Example of data collected on different time scales.

Clicking *Rate v Conc* displays the concentration dependence of the rates derived from the analytic fit, in this case to a double exponential function. *Note* that the fast phase cannot be resolved at the highest concentrations of substrate and the scale of the figure was truncated at 50 µM. The initial slope at lower substrate concentrations defines the rate constant for substrate binding. The maximum rate of the slow phase defines the rate of the isomerization, $k_2 + k_{-2}$. As note here, the results from analytic fitting can be misleading, so just use them as a guide to begin developing a model.

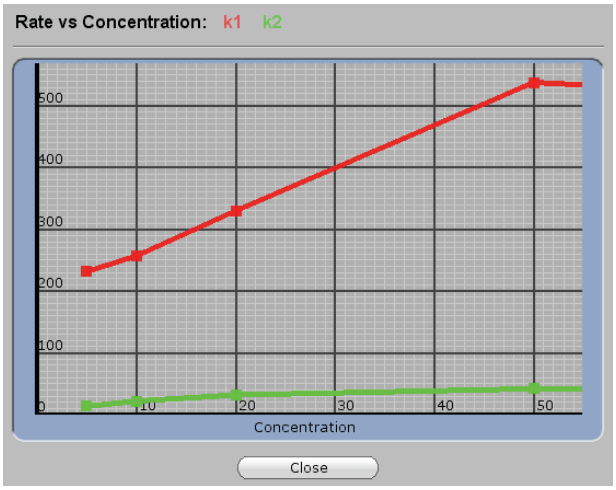


Figure 6.7 *Rate versus concentration plot.*

Model used to generate synthetic data: $E + S \underset{100\ s^{-1}}{\overset{10\ \mu M^{-1}s^{-1}}{\rightleftharpoons}} ES \underset{1\ s^{-1}}{\overset{50\ s^{-1}}{\rightleftharpoons}} FS$

6.4 Estimating sigma values

The standard deviation of a measurement should be estimated when data are collected, but that is not always feasible and you may have data to fit for which the standard deviation is unavailable. Starting with Version 2.2, we require an estimate of the standard deviation for global data fitting and we provide routines for estimating the standard deviation when it is not available.

Our method for estimating sigma (standard deviation) is based upon the assumption that all data within a given kinetic trace have the same standard deviation. We then fit each trace to a chosen function that mimics the data sufficiently to allow an estimate of the average sigma value. This initial fitting is based upon finding a minimum chi^2 value assuming that sigma is the same for each data point.

$$\chi^2 = \sum_{i=1}^{N}\left(\frac{y_i - y(x_i)}{\sigma_{AVE}}\right)^2 = \frac{N}{\sigma_{AVE}}\sum_{i=1}^{N}\left(y_i - y(x_i)\right)^2 \qquad \text{where } y(x_i) \equiv y(x_i \mid a_0 \ldots a_{M-1})$$

The actual function used to fit the data and the errors on individual parameters derived during fitting are *not relevant*. Rather, all we need is a function that runs smoothly through the center of the distribution of scatter in the data. From that fit, we then calculate the average sigma value based upon the residuals using the fitted curve.

$$\sigma_{AVE} = \sqrt{\frac{\sum_{i=0}^{N-1}\left[y_i - y(x_i)\right]^2}{N-1}} \text{ at the best fit valus of } a_0 \ldots a_{M-1}, \text{ where } y(x_i) \equiv y(x_i \mid a_0 \ldots a_{M-1})$$

This calculated average sigma value is then used as a basis to normalize residuals for global fitting.

We illustrate the estimation of sigma using data collected on kinesin as shown in Figure 6.8. These data have some obvious glitches (due to mixing microtubules in a stopped-flow). In panels A-D, individual traces were fit to various functions. The fit to a single exponential (A) was not adequate as seen by the systematic deviation of the fitted curve from the data. A double exponential function was better and may be adequate (B, σ = 0.000826). However, both C and D (fits to 3 exponentials and a 12th order polynomial, respectively) provided better fits for estimating sigma (σ = 0.00067 and 0.00065, respectively), in that the fitted curves run through the center of the distribution of scatter in the data. Fitting data globally provided an average sigma value of 0.00136 (E), which can be improved to 0.0012 by normalizing the traces during fitting (F). Given the apparent glitches in the data, this is deemed to be a good global fit in that it provided an average sigma value that is comparable to that obtained based upon the best fit to individual traces. In using the estimated sigma values during fitting, the important parameter to examine to evaluate goodness of fit is the ratio chi^2/DoF, which for a good fit should approach unity. In the present case, chi^2/DoF = 1.8. More accurately, we compute a p-value to evaluate the probability that the observed chi^2 is attributable to the random errors as reflected in sigma.

a. Procedure to estimate sigma

1. Under *Data Repository* or *Experiment Editor*, click on *aFit*, then select the appropriate function and fit the data.
2. Evaluate the fit by eye. If the fitted lines do not traverse the midpoint of the spread in the distribution of data, choose another function or limit the range of data fitting (*shift-click*).
3. Once a good fit is obtained, click, *Record Sigma*. This value will be retained with that data and the program will use this value to normalize the residuals during global data fitting.

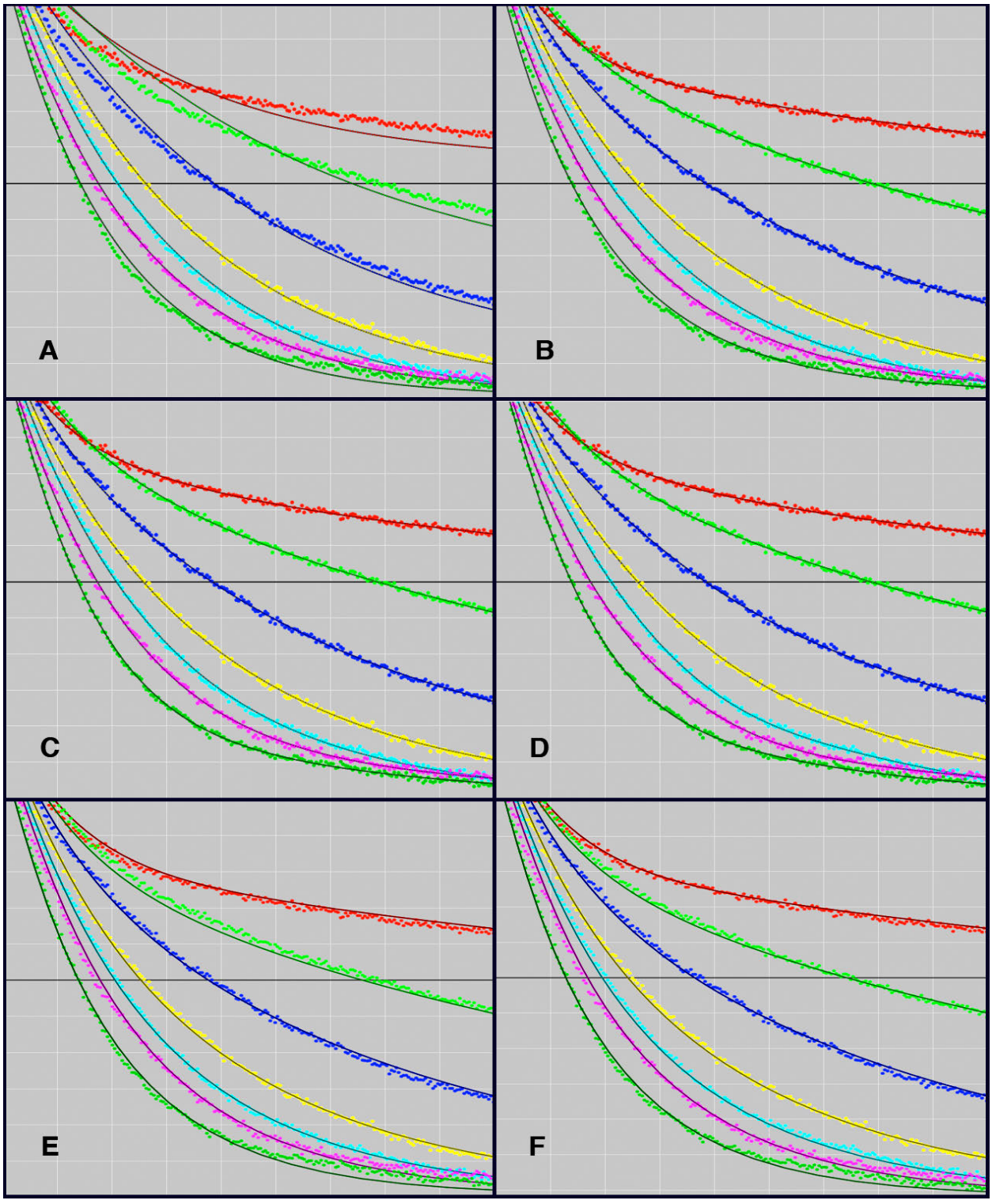


Figure 6.8 *Computing and Using Sigma in Global Fitting*. A. Fit to a single exponential (σ = 0.00216). B. Fit to a double exponential (σ = 0.000826). C. Fit to 3 exponentials (σ = 0.000676). D. Fit to a 12th order polynomial (σ = 0.000653). E. Global fit to model (σ = 0.00136). F. Global fit to the model with normalized traces (σ = 0.0012).

b. Limiting the data fit time range

Complex outputs often cannot be fit to simple mathematical functions over their entire time course and therefore, it may be difficult to use the fitting to an analytical function as a basis for estimating sigma. However, a good fit can often be obtained for a limited sub-set of the data. Setting the upper and lower limits of the range of data to be fitted is accomplished by using the *shift-click* functions. These limits can also be applied to global fitting based upon simulation.

Shift-Left-click: Set lower limit of data to exclude from fitting.

Shift-Right-click: Set upper limit of data to exclude from fitting.

Shift-CTRL-click: Clear lower and upper limits.

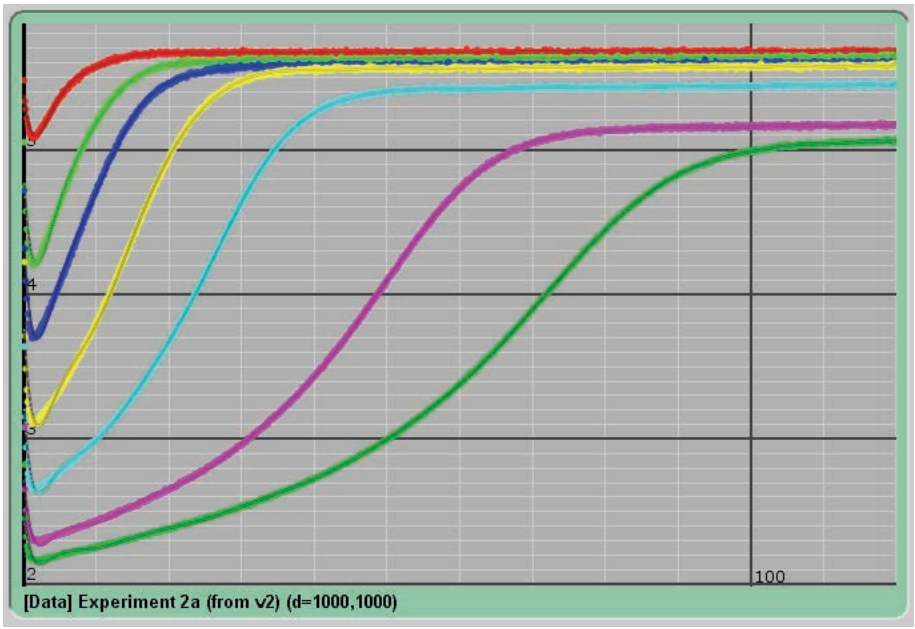


Figure 6.9A *Attempted polynomial fit to full time scale.* Note deviation of the fit from the data at short times.

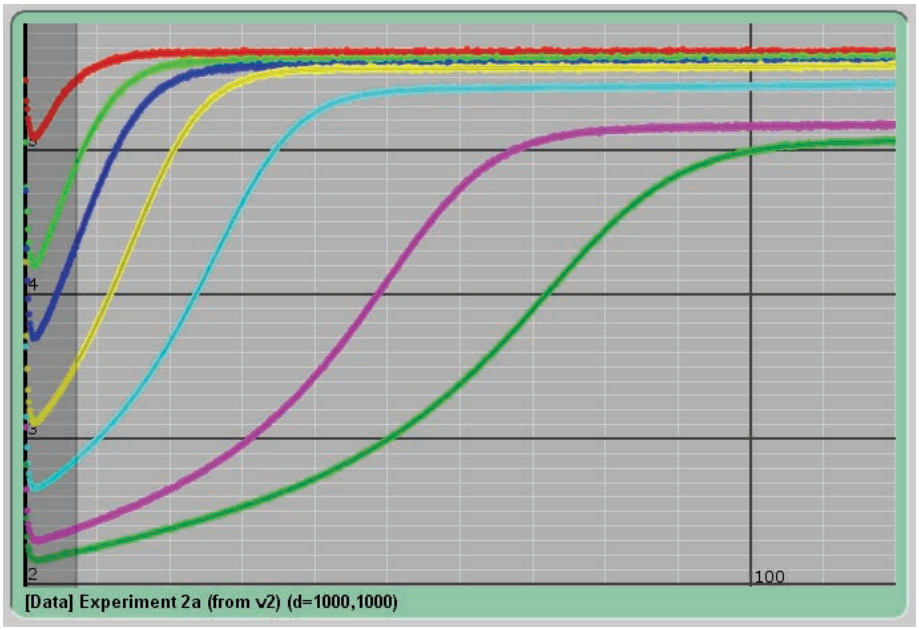


Figure 6.9B *Polynomial fit to truncated time scale.*
A shift-click on the graph is used to designate the region of data to omit during fitting, as denoted by the gray area on the left of the figure. The polynomial fits for the remainder of the curves are adequate for estimating sigma.

6.5 Assigning data to an experimental output

Assigning data to a given experiment is achieved through a simple drag and drop action. If there are multiple traces within a given experiment such as in a concentration series, assigning data to an output requires that the number of concentrations within the series equals the number of data sets within a given data file and that the order of concentrations is the same in both the data file and the experiment setup. If there are multiple species, then the individual traces within single or multiple data files can be assigned to each output individually.

After importing data into the *Data Repository*, simply drag the data from the *Data Repository* to the appropriate output signal as illustrated below. In the example shown in the Figure 6.10, four traces in a concentration series from the Tryp_synthase.mec file are being dropped onto signal S1. You can also select and drag individual traces.

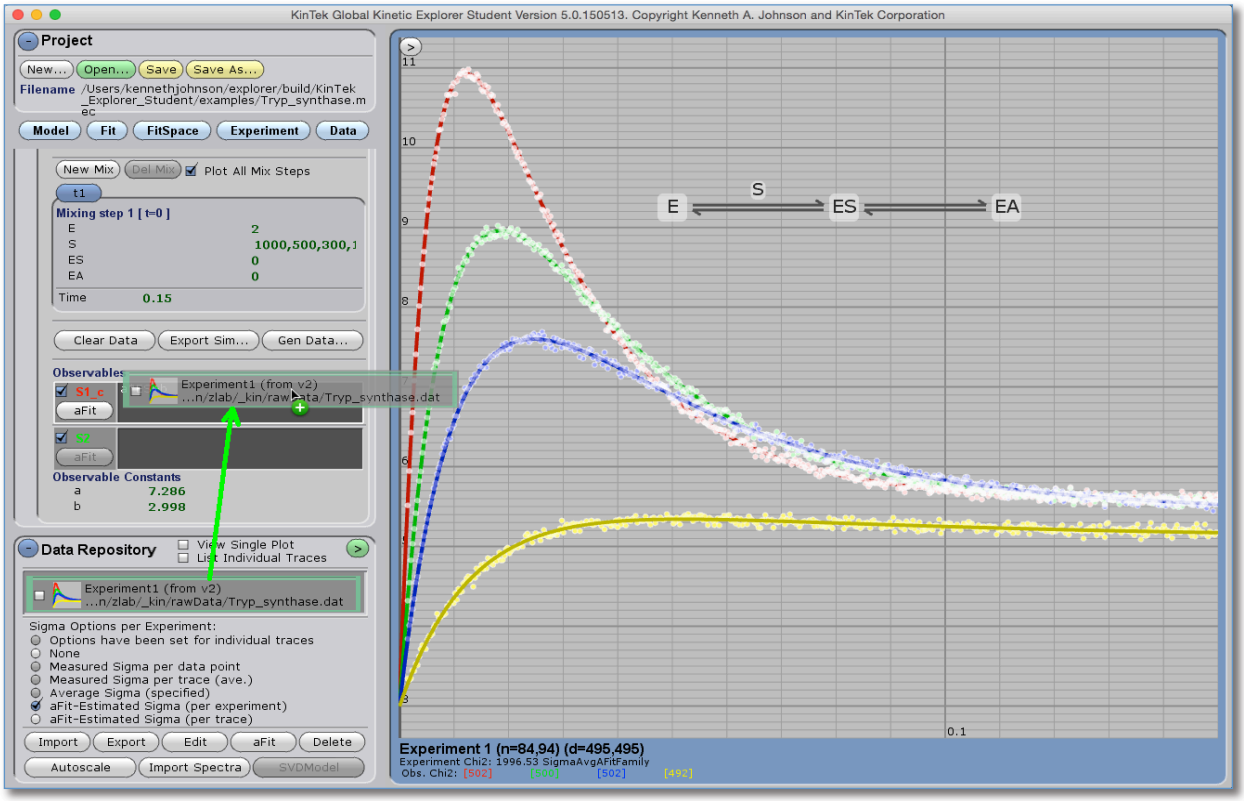


Figure 6.10 *Assign Data.* This figure illustrates the drag & drop method for assigning data an observable output.

Note as soon as you click on the data, points appear in white on the graph, flashing on and off to illustrate the selected data. If the data are not visible, they may be off scale, but can still be assigned to the experiment. If *Autoscale* is checked, then as soon as the data are dropped onto an experimental output, the graph will be scaled to display the full range of the data. You can drag & drop an individual time course or an entire set of data, such as a concentration series illustrated above. Note also that the background of the data set that is being dragged is red until you have it hovering over a valid output definition, when it turns green, indicating that it can be dropped.

Note: When assigning data to an experiment with multiple mixing events, take care to display the correct reaction period (t1 or t2 for example) prior to assigning the data. The data will be assigned to the displayed reaction time period.

If there are multiple output observables, such as in the example in Figure 6.11, each data trace can be dropped onto the appropriate output signal. After dragging and dropping the data onto the appropriate

observable output, close the *Data Repository* menu. See also EPSP.mec for an example of data assigned to different outputs.

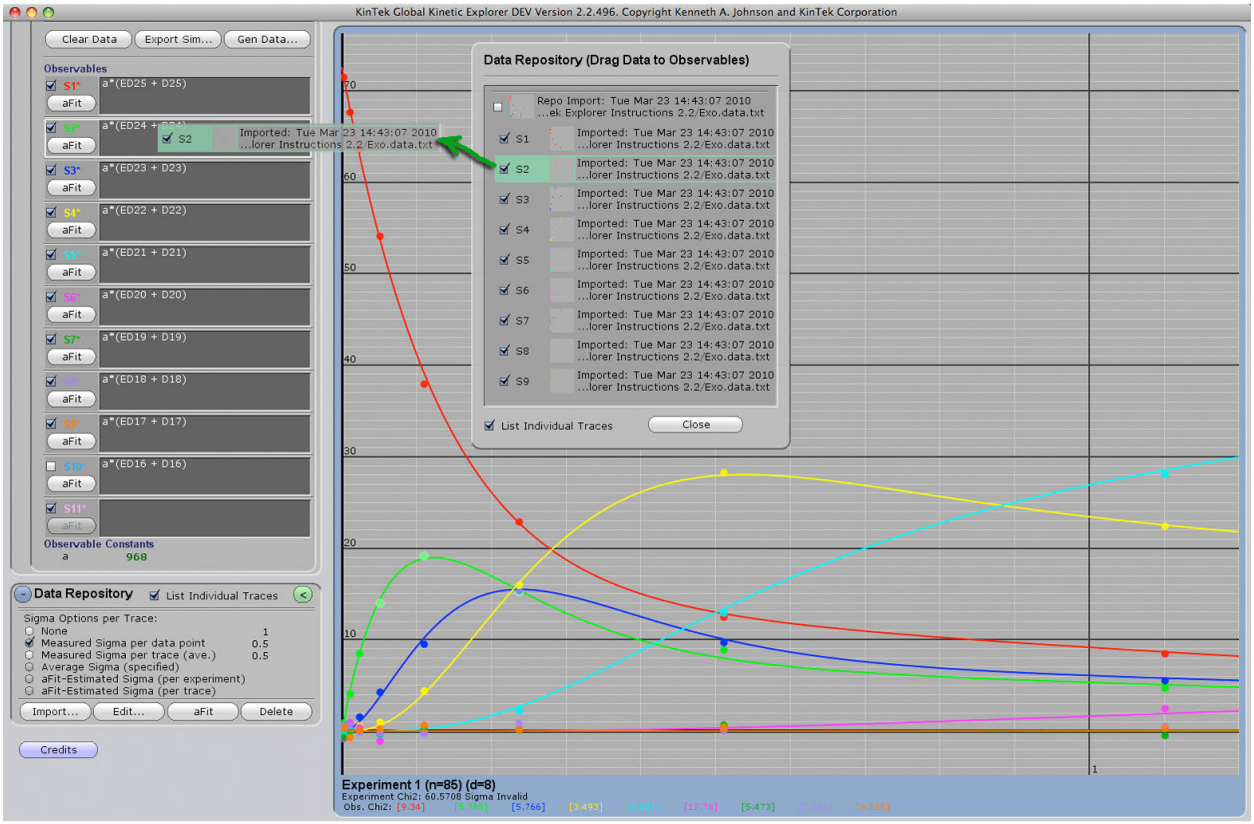


Figure 6.11 *Drag & Drop Individual Traces.* This figure illustrates the drag and drop method when individual traces within an experiment correspond to different outputs. It also shows how the *Data Repository* can be separated from the *Control Panel* by clicking on the > button. This is useful when there are a large number of observable outputs and data files.

Sigma Options: You can control how the available standard deviation data (sigma) are used when fitting each data set using the check boxes shown in Figure 6.12.

- *None:* Sigma values not available or ignored
- *Measured Sigma:* associated with each data point in the imported data.
- *Average Sigma:* entered in the column header when importing data.
- *aFit-Estimated Sigma:* derived by using aFit functions to estimate sigma

Sigma values can be defined for each point, averaged for each trace, or averaged over a set of traces within an experiment (such as a concentration series).

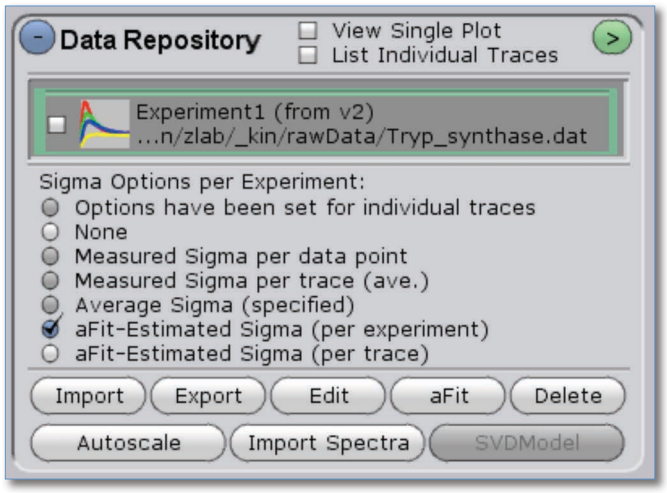
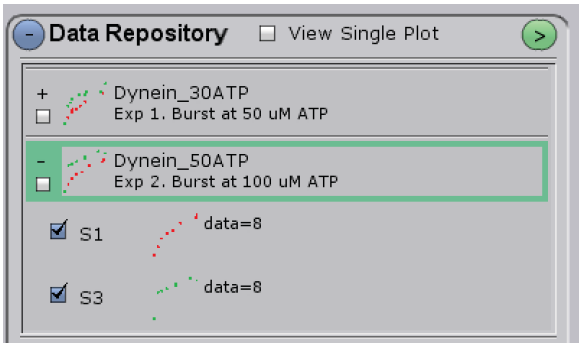


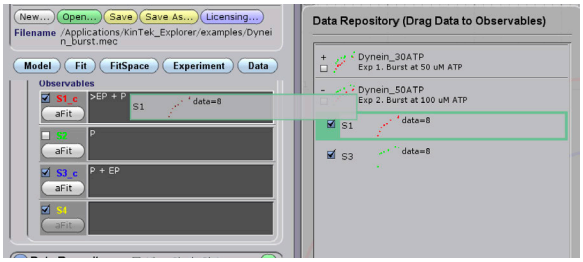
Figure 6.12 *Data Repository Sigma Options.*

Starting with Version 7.6 we have streamlined the listing of data in the Data Repository by providing more controls over what to display.

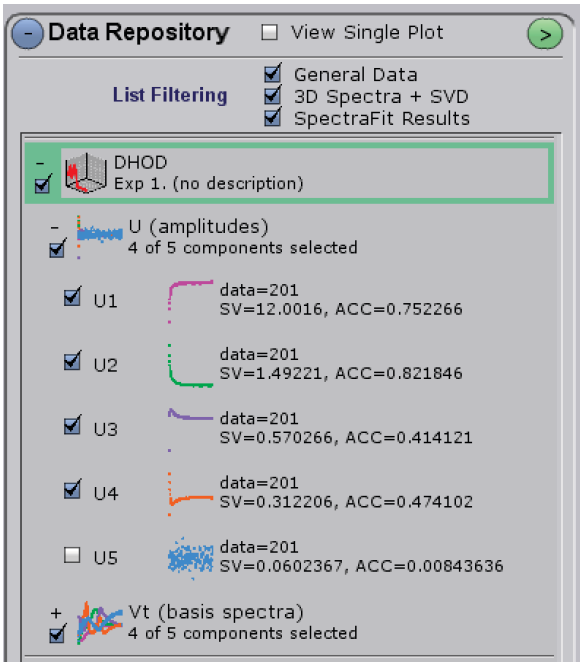


- Click on the checkbox to display the graph.
- Click on the + to show the individual traces.

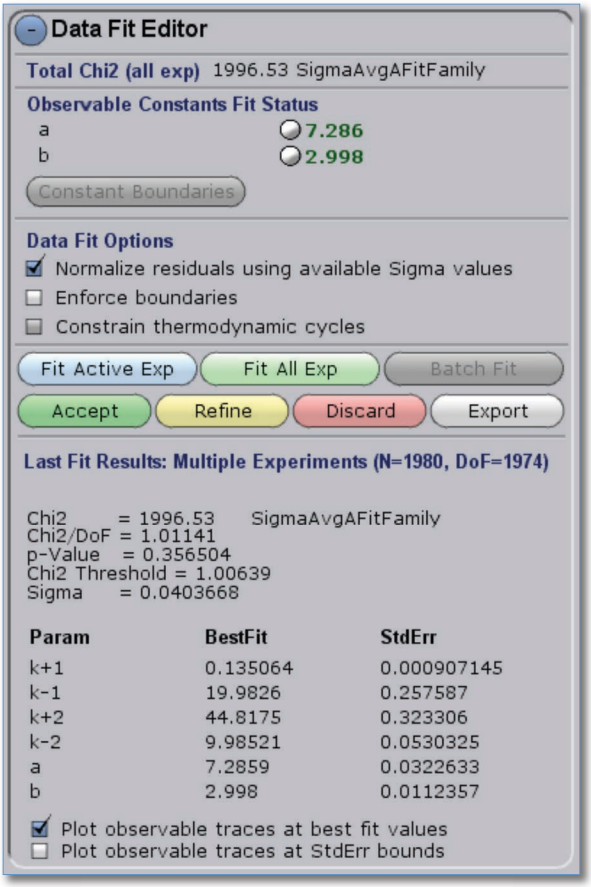
After revealing individual traces, you can then drag a given trace to a given output as illustrated here.



In the case of SVD analysis of time- or concentration-dependent spectra, there are additional options to control the display of the large number of figures and data that are held in the *Data Repository* under *List Filtering*. As described above the +/- toggle controls whether to show individual components under each heading.



Chapter 7. Data Fit Editor



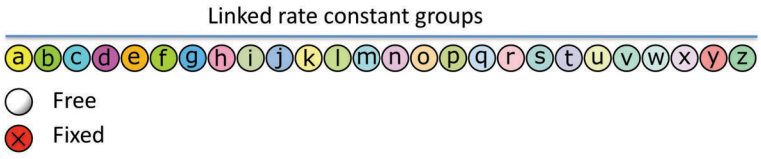
Use the Data Fit Editor to:

- Fit one or all experiments directly to the model based upon simulation
- Constrain output factors
- Toggle on/off the use of sigma during fitting
- Enforce Upper and Lower limits
- Constrain thermodynamic cycles
- Overlay 95% inference band on the fitted curves based upon the standard error
- Examine statistical parameters

The *Data Fit Editor* controls fitting of data directly to a model based upon the numerical integration (simulation) of the model. In addition to the control for whether the observable output constants are held fixed or allowed to vary (or "float") in fitting the data, the fitting routines also obey the constraints placed upon the rate constants under the *Model Editor*.

7.1 Controlling rate constants

The symbols adjacent to the lettered controls whether an observable output factor is held fixed or allowed to vary in fitting the data.



If the symbol is checked (⊗), the output factor will not change during fitting of the parameters to data under the *Data Fit Editor*. Alternatively, un-checking the control allows the output factor to float during the fitting process. In either case, output factors can be changed manually (either by scrolling or inputting new values).

Holding certain constants fixed, or in a known ratio provides an important constraint to achieve a meaningful fit to the data. Alternatively, rate constants can be grouped so they maintain a constant ratio with respect to other members of the group. See *Model Editor* for more details.

7.2 Fitting experiments to the model

KinTek Explorer fits multiple data sets directly to a given model by nonlinear regression analysis, which is based upon an iterative search to find a set of parameters that gives a minimum chi^2 (see Chapter 9).

a. The concept of fitting data to a model

When fitting experimental data based upon a theoretical model, one can describe the process as fitting a model to the data or fitting data to a model. Regardless of the semantics, the process of fitting experimental data is greatly enhanced by eliminating the intermediate step of deriving equations and the complications of then fitting the concentration dependence of observed rates and amplitudes to estimate rate constants. Throughout this text, I will use the shorthand description in terms of *fitting data to an equation* versus *fitting data directly to a model* to make the distinction between the conventional data fitting methods and those based upon numerical integration of rate equations, respectively. Both methods involve fitting data based upon a model, but the fitting of data directly to the model offers numerous advantages do to the more robust requirement that the model account for both the rate and amplitude of observed reactions. However, fitting data directly to a model requires a significant paradigm shift in understanding the information content of data and getting reliable error estimates on kinetic parameters. *KinTek Explorer* provides the tools needed to get the most out of your data and to ensure that models derived in fitting data are fully constrained by the data or to clearly identify those parameters that are not well constrained.

b. Fitting data with starting estimates by dynamic simulation

Nonlinear regression requires a set of starting estimates of parameters from which to begin the iterative search. The starting parameter estimates must yield predicted curves that are close enough to a final solution for the process to converge. There are several ways to get initial estimates. In the traditional method, one inspects the curves to obtain an eyeball estimate of fitted parameters, but that is difficult to do when the parameters are intrinsic rate constants. A second method is based upon conventional data fitting to get estimates of observed rates, which are composites of intrinsic rate constants and then use derived equations to provide estimates of individual rate constants. By far the easiest, and perhaps the least biased method to obtain starting estimates, is based upon using the *dynamic simulation* to dial in a set of parameters that approximately fit the curve. As an example, open the file Tryp_synthase_no_fit.mec. You can easily scroll the output factors **a** and **b** to get a y-axis scale that mimics the data, then scroll the individual rate constants until the family of curves is close to the data. How close to the data do you need to be? Experiment and find out! The answer will depend on the complexity of the model. *Dynamic simulation* is also extremely important in searching for other areas of parameter space to seek other possible combinations of parameters that might fit the data equally well.

To fit the data, click on one of the buttons below:

Fits only the selected experiment (outlined with a bright blue border).

Globally fits all experiments in the current .mec file, with the exception of the experiments that are flagged with ☑ Exclude from Global Fits (see *Experiment Editor*).

Fit all data sets within a given directory. This feature is described in *Section 7.4*.

In either case, the program performs nonlinear regression to find the best fit for all parameters by fitting directly to the model based upon numerical integration of the rate equations and the starting conditions for each experiment. The program will iteratively seek the minimum chi^2 for all data.

☑ Normalize residuals by known sigma values: This function is essential when fitting multiple data sets with greatly varying Y-axis ranges. However, you are advised to use it always because the normalized chi^2 provides a metric to estimate goodness of fit as explained in *Chapter 9 Statistical Analysis in Data Fitting*. See the example file: *Fluor_&_quench_data.mec,* which illustrates need for this normalization. The Y-axis of the quench-flow data spans the range of 0-1000 while the fluorescence data is on a scale of 0-1.5. Without normalizing the residuals by dividing by the estimated sigma value, the larger absolute scale of the quench flow data overwhelms the relatively small signal and correspondingly small residuals, so that the fit is heavily weighted towards the former.

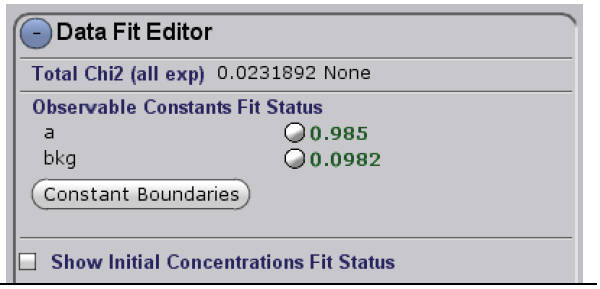
c. Optimizing starting concentrations when fitting.

There are many instances in which including the starting concentrations among the fitted parameters is desirable. For example, there may be minor pipetting errors that prevent achieving an optimal global fit for data collected over a series of concentrations. In this case allowing a modest change in starting concentrations can significantly improve the goodness of fit. In other cases, for example during an active site titration, the concentration of enzyme in an experiment may be an unknown, and fitting to derive the starting enzyme concentration is the goal of the experiment.

With Version 6.0 of KinTek Explorer, individual or groups of starting concentrations can be optimized when fitting data, and the allowed percentage change in concentration can be limited. We illustrate this utility with two examples. We start with a simple example, cyclophyllin.mec. This is a minimal data set that is difficult to fit and posed problems when originally published. The stated enzyme concentration was 300 nM, but inspection of the data suggests that the active enzyme concentration may be closer to 250 nm based on the intersection of two limiting asymptotes.

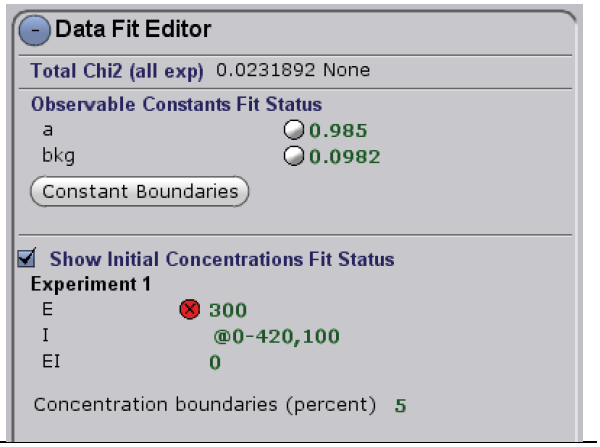
In order to allow the starting enzyme concentration to vary during data fitting, go to the *Data Fit Editor* and check the box:

☑ Show Initial Concentrations Fit Status

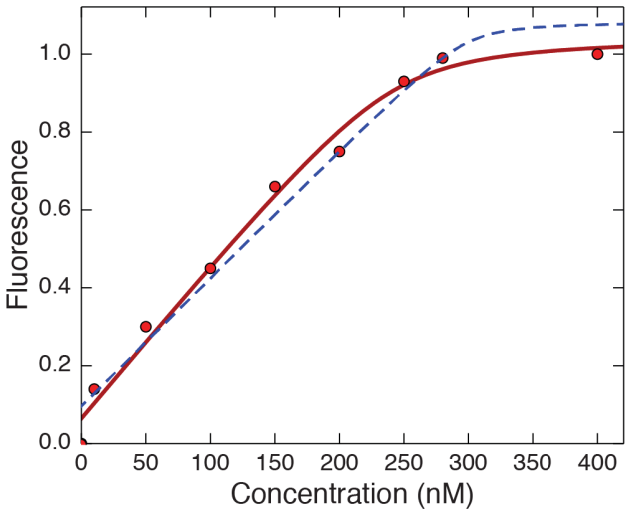


Each experiment is then listed (in this case only one experiment) along with a list of starting concentrations. The ⊗ symbol preceding the concentration enzyme denotes that the value is locked. This is the default status.

Re-fitting the data after unchecking the red x allows fitting the enzyme concentration get a value of 244 ± 33 nM, with a correspondingly better fit as shown below.

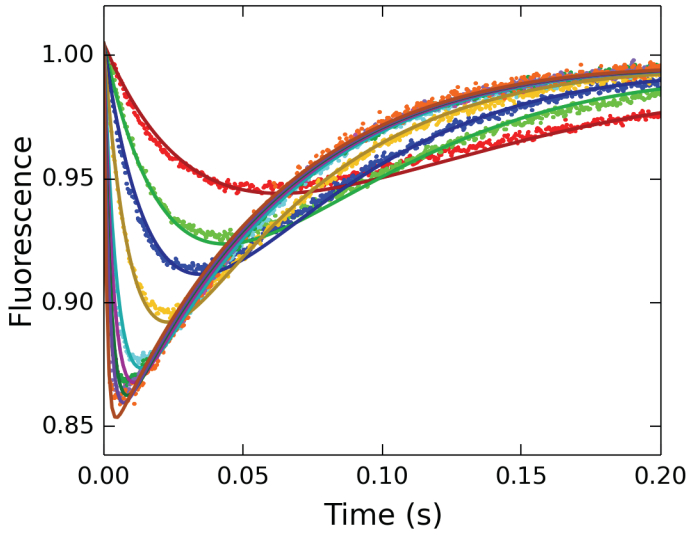


This figure shows a comparison of the two fitted curves. The blue dashed curve is calculated based on the assumed enzyme concentration 300 nM. The red curve was obtained by allowing the enzyme concentration to be a fitted variable returning a value of 244 nM. There are two take home messages from this example: (1) Active site titrations can be used to define the active enzyme concentration. (2) In this particular case, it is clear that more data are needed at higher ligand concentrations to distinguish the two possible fits.



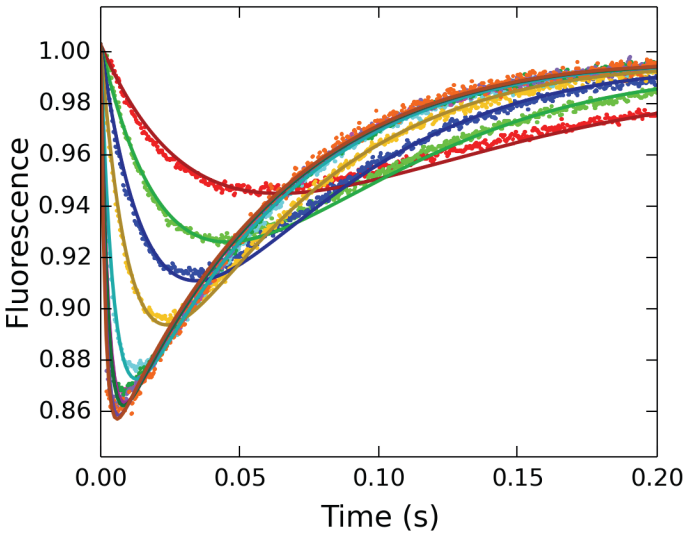
A second example based on HIVRT_fluor-qf.mec is more complex because there are multiple experiments and multiple mixing steps in each experiment, but the principles are the same. In this example, we allow the starting concentrations of substrate to vary by only ± 10% to reflect reasonable limits on pipetting errors.

Experiment 2	
E	0.4
D	0.6
ED	0
N	0
EDN	0
FDN	0
FP.PP	0
EP	0
PP	0
P	0
Mixing step 2	
E	0
D	0
ED	0
N	2,4,6,12,28,40,60,80,200
EDN	0
FDN	0
FP.PP	0
EP	0
PP	0
P	0
Concentration boundaries (percent)	10



The figure at the right shows the fitted curves without allowing the concentrations of N to vary—they are locked by the ⊗ symbol by default. To allow the concentrations of N to vary, uncheck the ⊗ next to the concentrations of N in the series, then set the *Concentration boundaries (percent)* to 10.

Param	BestFit	StdErr
k-2	24365	637.481
k+4	14.8585	0.0722306
scale2	5.44737	0.00237242
f	0.844544	0.000469851
p	1.00785	0.000494367
Exp 2 Mix 2		
N	1.76467	0.0431693
	3.39566	0.0816214
	5.63171	0.136148
	10.4625	0.258196
	27.7894	0.751351
	53.0152	1.68137
	58.7618	0.163231
	88.0806	3.61697
	105.616	5.03057

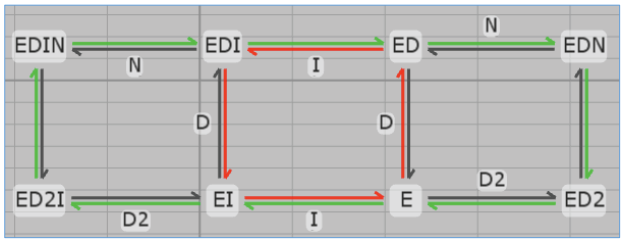


After fitting the data, there is a modest improvement in the goodness of fit. Note also that now the values of the concentrations and standard errors are listed under the fit results.

d. Enforcing thermodynamic cycles

When the program recognizes a closed loop (thermodynamic cycle), it is highlighted by colored arrows in the display model and you are given the option to maintain a net free energy change of zero in progressing around the cycle to return to the starting point (net product of equilibrium constants = 1) when fitting data.

☑ Constrain thermodynamic cycles. This checkbox controls whether thermodynamic cycles are maintained when fitting data.



Thermodynamic Cycles:	
E=ED=EDI=EI=E	1.00
E=EI=ED2I=EDIN=EDI=ED=EDN=ED2=E	1.00

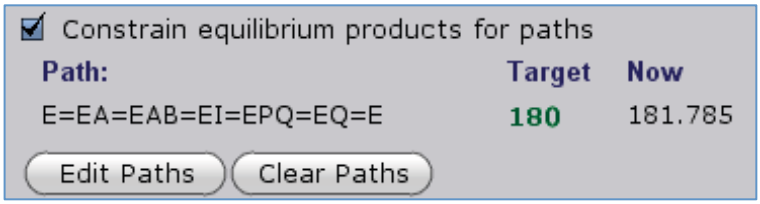
These examples are from the HIV_NNRTI.mec file. There are two cycles, an inner cycle shown in red and an outer cycle shown in green. The net product of the equilibrium constants is given in the *Data Fit Editor* and is updated when you scroll rate constants.

☑ Constrain equilibrium [constant] products for paths.

You can also enforce a user-defined product of a series of equilibrium constants. For example, in EPSP.mec we know that the product of $K_1K_2K_3K_4K_5K_6$ = 180, the net equilibrium constant for the chemical reaction. In order to specify a constraint to define the net product of individual rate constants, you first define the path, then set the product.

1. Click on *Edit Paths*. A text box opens, which you can edit to remove steps in the reaction sequence that are not involved in the equilibrium.
2. Enter the value for the net product of the equilibrium constants for the reaction sequence.

Note that the reaction sequence is abbreviated by eliminating ligands: for example E + A = EA is abbreviated as E = EA.



e. After the fit: accept, refine, discard, revert

After fitting, the graph will display the fit in gray in addition to the colored output for the starting parameters, and several new buttons appear. ***Accept, Refine or Discard***.

Accept The results of the fitting are copied to your working model. After accepting a fit, you still have the opportunity to *refine* the fit, or *revert* to the previous parameters as described below.

Refine When you click *Refine*, the program will accept the results of the last fitting, then run the fitting routine one more time, starting with the new values derived from the previous round of fitting. Each time you click *Refine,* the process will repeat and you will see the best-fit parameters update. Re-starting the fitting process involves a re-initialization of parameters that control the nonlinear regression in ways that are transparent to the user, and allows the fitting to be refined.

Discard Discard the last fit results and return to the previous model parameters.

Revert Even after accepting the results of a fit, you can inspect the parameters and fitted curves and still *revert* to the previous parameters by pressing this button. This acts as an "undo" of the *accept* function. You can use this function to toggle back and forth between two possible fits to the data.

☑ Plot observable traces at best fit values: This turns on and off a thin black line showing the best fit.

f. Exporting fit results

Export Press Export to save a text file containing a summary of the fitted parameters and standard error. Additional statistics are also given, including the covariance matrix.

Text files suitable for generating publication-quality graph of each experiment can be exported using the *Export Sim* function under the *Experiment Editor*.

Starting with Version 6.0, this function also outputs publication quality figures in both .eps and .png format. See Chapter 11 for a detailed description of this function.

g. Plotting confidence intervals

☑ Plot observable traces at StdErr bounds: This turns on and off an overlay of all possible fits to the data obtained by generating curves based upon the extremes of parameter values derived from the standard error. For each upper and lower bound of each variable parameter, the program will find the combination of all parameters yielding the best fit for each individual parameter extreme as shown below in Figure 7.1. The white lines show the range of curves attainable for the limits of error on all fitted parameters. For example, curves are generated for $k_{+1} = 0.60$ and $k_{+1} = 0.62$ based upon the estimate of $k_{+1} = 0.59 \pm 0.01$ $\mu M^{-1}s^{-1}$. For the curve at $k_{+1} = 0.60$, the data are fit again allowing other parameters are allowed to float in finding the best fit for that parameter extreme, then plotted on the graph. The process is repeated for each parameter.

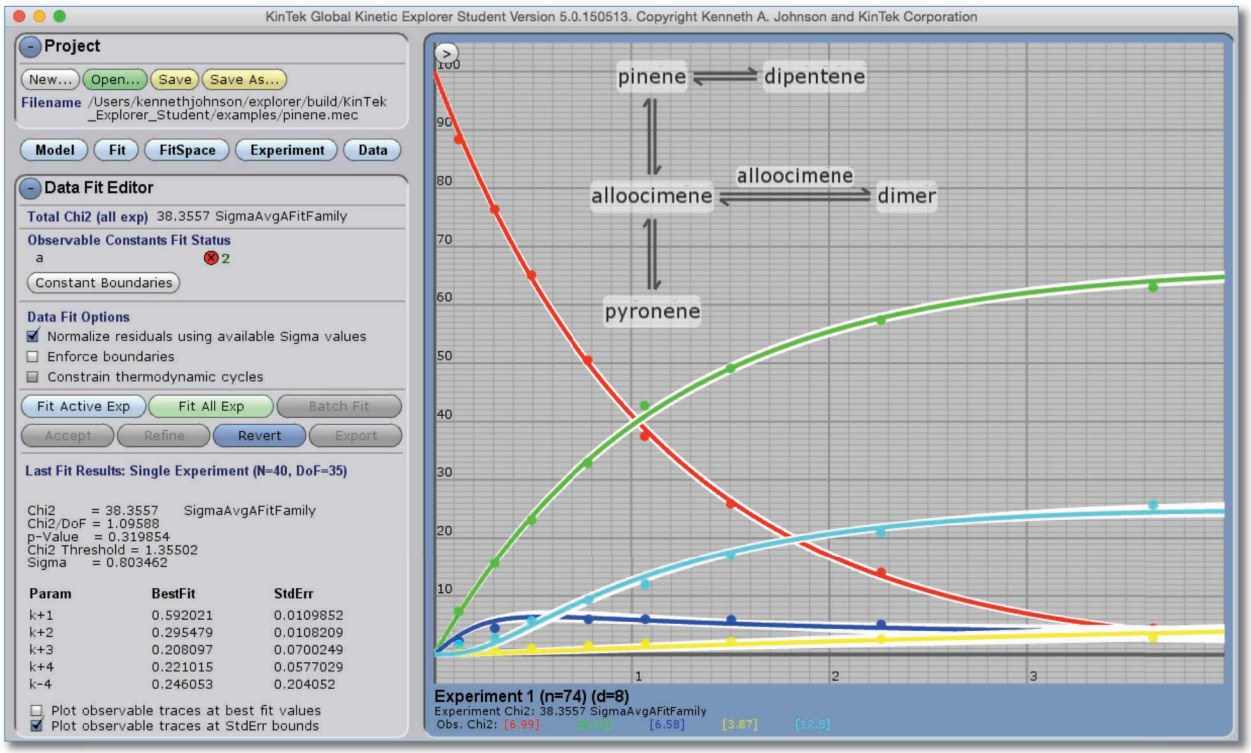


Figure 7.1 *Plotting 95% inference confidence intervals.* This figure shows the overlay of curves computed at the extremes of the parameter confidence limits from the file, pinene.mec.

This analysis is useful in that it shows the variability in the fitted curves according to the standard error estimates. Examine each of the examples to understand the relationships. In some cases, when parameters are not well constrained by the data, you will see overlaid traces that are indistinguishable from the best fit (i.e., racemase_3step.mec). In another case, dynein.mec, the ATP binding rate is constrained only to 8 ± 3.7, the overlay reveals the limitations in the data that prevent tighter limits on k_{+1}.

NOTE: The FitSpace calculation, described below, produces cleaner displays of overlay traces because it overcomes many of the limitations of computing standard error when parameters are not well constrained.

g. Error analysis

Data fitting by nonlinear regression produces a standard error estimate based upon the covariance matrix (11,12). This standard error analysis can be misleading when applied to multiple variable fitting if the parameters are not well constrained; therefore, one should exercise caution in interpreting the standard error analysis. When parameters are not well constrained, there are large errors in computing the covariance matrix. For example, in the racemase.mec file we previously noted in using Version 1 of *KinTek Explorer* that the fitting gave that k_{-2} = (6.45 ± 0.03) x 10^8 s^{-1}. However, with the release of version 2.0 in which we changed to double precision (64-bit), the fitting gave a value of k_{-2} = (6.64 ± 6.3) x 10^8 s^{-1}. In this example, if k_{-2} and k_3 are scrolled downward by 5 orders of magnitude, one then gets equally unreliable estimates showing that k_{-2} is not well determined at all. Thus, the increase in precision of the calculations has improved the ability of the nonlinear regression routines to indicate when parameters are not well constrained. This result explains why other programs, such as Dynafit, suggest such lower error estimates on fitted parameters, such as in the case of alanine racemase (4,5). The extent to which parameters are constrained is most readily revealed using the *Dynamic Simulation* and by the *FitSpace* analysis described below and in (2,3).

Users are encouraged to take advantage of all that the *Dynamic Simulation* has to offer by

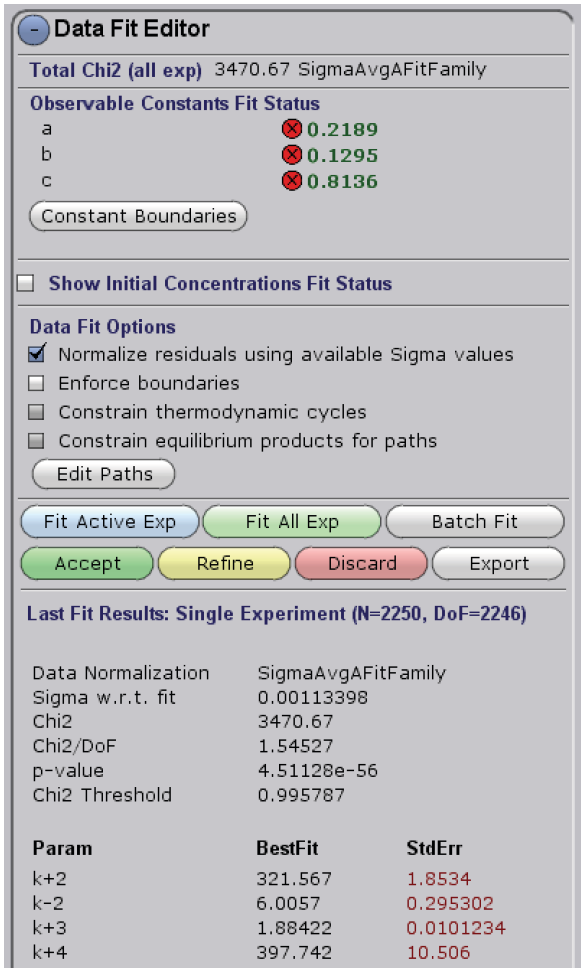
exploring the range over which parameters can vary by scrolling individual constants or pairs of linked constants while monitoring the visual appearance of the curves and the chi^2 calculations. Exploration of the landscape after fitting using the *Dynamic Simulation* will do much to tell you about the range over which a given rate constant can be varied. Also, think carefully about the model and use the ability to link rates in a constant ratio to explore the multi-dimensional fitting landscape *without perturbing the net equilibrium constant*. If k_{-2} and k_3 are linked and varied in synchrony, the full 5 order of magnitude range over which they can be varied is revealed.

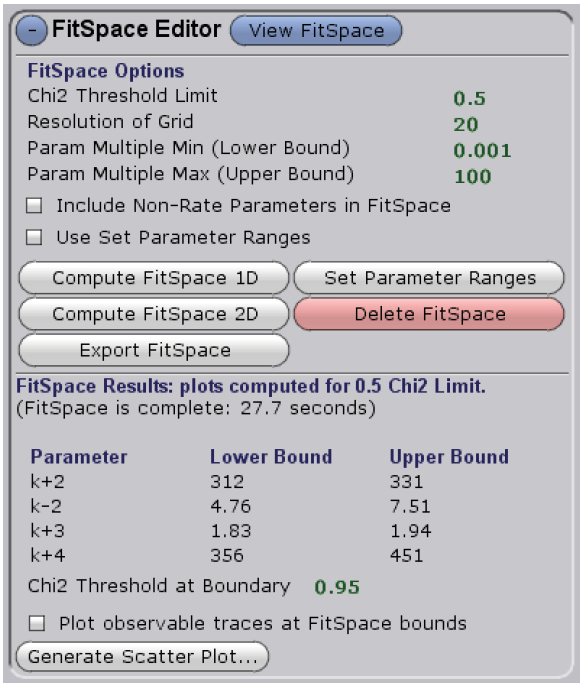
Fitting data to a model is often difficult because one needs to include all steps in a pathway even though some rates may not be known. Therefore it is important to recognize which rates are constrained by the data and which are not. In particular, it is often the case that a rate constant for a given step may only have a lower limit because as long as the rate is greater than that lower limit, changing the value does not affect the curves. This situation is typical when a model is under-determined. In fact, the appearance of a rate constant, which is much faster than neighboring rates in a reaction sequence, should be taken as a red flag indicating the model may be overly complex. Nonetheless, it is often appropriate to include such fast steps for completeness and use the software to just establish a lower limit on the magnitude of the rate constants.

Figure 7.2 shows two panels giving statistical analysis from fitting data. The *Data Fit Editor* panel shows the results from nonlinear regression analysis, while the *FitSpace Editor* (see Chapter 8) shows the results from confidence contour analysis. We chose the example file Kinesin_mantADP_wt.mec because it contains high quality stopped-flow data and the fitting provides rate constants that cannot be obtained by conventional fitting. Moreover, it illustrates the kind of deviations from ideality that occur with real data, in this case from mixing microtubules in the stopped-flow. Therefore, we are faced with the challenge of evaluating goodness-of-fit and estimating standard errors when the data do not conform to the requirement that all data are *independently and identically distributed (the IID assumption)*. The data come close to meeting those criteria, but not quite. Accordingly, the statistical analysis comes close to providing valid estimates of errors on the parameters, but need to be evaluated carefully.

Data Fit Editor Statistics

Chi2 = 3470	This is the total Chi^2, with a note indicating that Chi^2 was normalized based upon an average *Sigma* value estimated using the AFit function (SigmaAvgAFitFamily). See Chapter 9.
Chi2/DoF = 1.54	Chi^2 divided by *Degrees of Freedom*, the number of data points minus the number of fitted parameters. For a good fit to ideal data, this should approach 1.0. Larger values indicate deviations from a good fit. The value of 1.54 is good for real data, but the p-Value is a more precise metric.
p-Value = 4.8 e-56	The low p-Value suggests that Chi^2 cannot be explained simply as due to random errors in the measurement according to the expectations based upon the *Sigma* value. That is, there are some deviations from ideality, which are evident on close inspection of the data.
Chi2 Threshold = 0.996	As described in Chapters 8 and 9, we compute a threshold representing a fractional increase in Chi^2 that gives 95% confidence intervals. The computed threshold is overly optimistic. We suggest a value of 0.95 is more conservative. **NOTE:** Starting with Version 6.0, we now use an inverse function so the threshold compared to previous versions of the software. This provides a better graphical representation of the best fit.
Sigma = 0.001134	The standard deviation of the data.





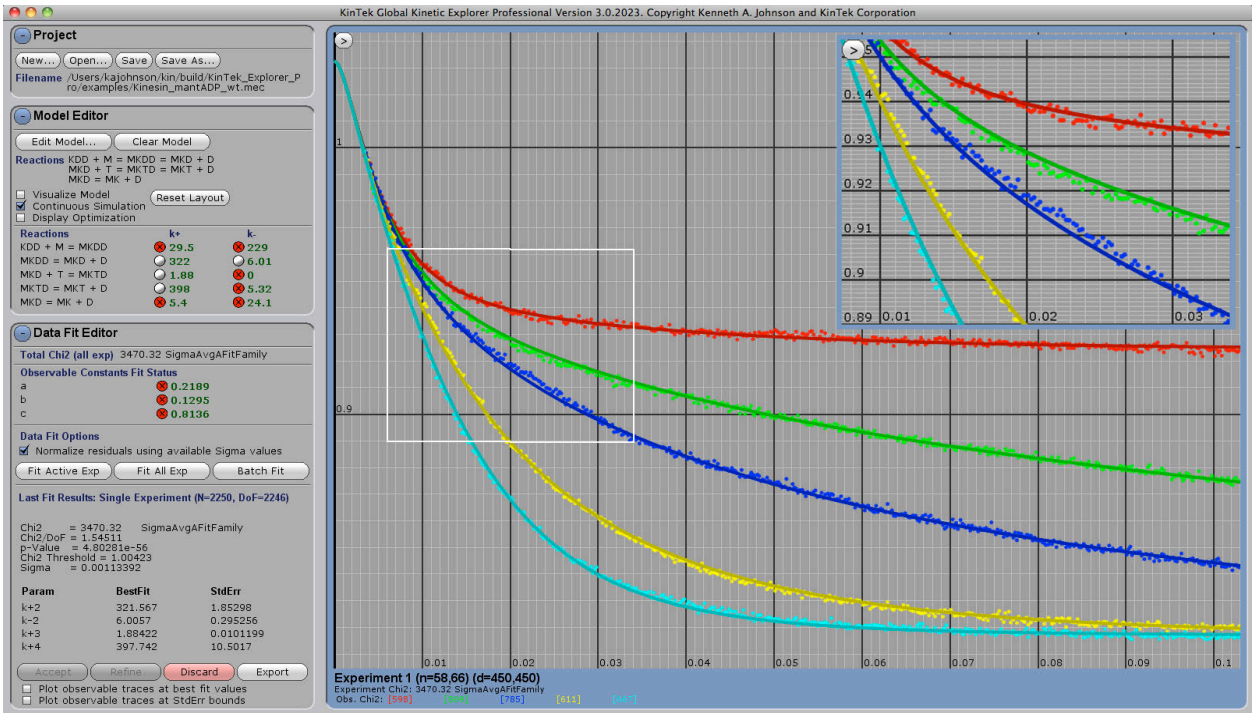


Figure 7.2 *Statistical analysis in fitting Kinesin_mantADP_wt.mec* Here we show the Data Fit Editor and FitSpace Editor panels and the main graphics panel. Note the slight deviations of the data from the fitted curve, most noticeable in the blue curve at around 0.02-0.03 s, shown in the inset.

In *Chapter 8*, we describe computation and use of *FitSpace* to evaluate the extent to which parameters are constrained by the data. The *FitSpace* confidence contour analysis provides a more realistic assessment of errors in the data, and is essential in identifying parameters that are not well constrained by the data. As summarized in the *FitSpace Editor* panel, the analysis provides upper and lower limits for each parameter, allowing for asymmetric error distributions, which are quite common.

Constant	best fit ± std err	lower limit	upper limit
k_2	321 ± 2	312	332
k_{-2}	6.0 ± 0.3	4.61	7.51
k_3	1.88 ± 0.01	1.83	1.94
k_4	400 ± 10	348	458

Note the unrealistically small standard error estimates derived by nonlinear regression. The confidence contour analysis with a chi^2 threshold of 0.95 provides much more realistic limits. This is described in more detail in *Chapters 8* and *9*.

In *Chapter 9*, we provide a brief description of error analysis and parameters that can be used as a guide to evaluating goodness of fit. In particular, if known sigma values are included in computing chi^2 during data fitting, one expects that a good fit will be achieved when chi^2 is equal to the degrees of freedom (DoF), the number of data points minus the number of parameters. That is, *Chi^2/DoF* approaches unity. More precisely, we compute the *p-value,* which estimates that probability that the observed chi^2 is within a range attributable to random errors expected from the measured sigma (see *Section 9.4*). However, this metric will depend upon the inclusion of all possible sources of error, which is often difficult to derive for global data fitting. For example, one needs to include standard deviation of measurements, errors in reactant concentration, and possible lamp fluctuations between experiments. For these reasons, these metrics are not always reliable in evaluating goodness-of-fit for global data fitting. They are included here for your evaluation while we are continually working to find new ways to reliably evaluate errors.

7.3 Summary of steps in fitting data

1. *Create Data Files:* Prepare tab-delimited text files containing data following one of the patterns described under *Data Repository* (Chapter 6). If the concentration of a single species is varied across a family of curves, they can all be in one file. Alternatively, if different data sets involve changing more than one species, they must be placed in individual files.
2. *Import Data:* Starting with version 2.2, data can be imported at any time, before or after a model is developed as described in Chapter 6. After the data are imported, fitting by conventional methods to an analytical function using aFit (Section 6.3) may provide important clues to suggest the model and provide estimates on rate constants. Conventional data fitting is useful in that it illustrates trends in the data that reveal the underlying mechanism. Finally, it is important to use the aFit functions to get an estimate of the *standard deviation* of your data if that information is not already part of the data file.
3. *Enter Model:* Decide on an appropriate mechanism and enter the mechanism using the model editor, as described in Chapter 4.
4. *Define Experiments:* In *KinTek Explorer*, define one "experiment" to correspond to the conditions used to collect data in each individual data file. Define the output functions that correspond to the signal of observables for each experiment.
5. *Assign Data:* Assign the data to each experiment setup. When assigning data to an experiment, you match the data to a given simulation output. Therefore, it is necessary to define the simulation before assigning the data. If you have data covering a concentration series (a series of experiments performed over the same time scale at several concentrations on one species), then the program will assume that the order of the series in the experiment setup corresponds to the order of columns in your data file.
6. *Get Initial Estimates:* Adjust rate constants and output factors in attempting to mimic the data. At this stage of the fitting, you can begin to get an idea of which rate constants may be determined by the data and which might not. Fix any rate constants that are known from other experiments and not established by the current experiment. Look for pairs of rate constants that might be linked.
7. *Fit Data:* Fit the model to the data using the functions under *Data Fit Editor* (Chapter 7).
8. *Evaluate Goodness of Fit:* There are two metrics to evaluate goodness of fit. First, if care was taken to estimate sigma (standard deviation) of the data measurements, then chi^2/DoF approaches unity for a good fit (chi^2 = DoF, the degrees of freedom). More precisely, the p-value should be greater than 0.01 (see *Section 9.4*). Secondly, careful visual examination of the fit relative to the data ("chi by eye") is essential.
9. *Explore the range* over which parameters can vary by scrolling individual rates or pairs of constants while monitoring the output. Try moving parameters into new ranges and fit the data again; this can be done by fixing the new rate or allowing it to float. This may be the most important part of the fitting process where you can use the dynamic simulation to explore the extent to which individual constants are constrained by the data.
10. *Critically evaluate your model* and your fit to the data. Can the data be fit equally well to a model with fewer steps? Edit the model and seek the simplest model to explain your data by including only those steps that are well constrained by your data. If steps must be included for which there is no information to define their rates, provide minimal estimates for the magnitude of the rate constants and clearly identify them as such in all publications. There is nothing wrong with stating that a given rate was set at a given lower limit because the data did not constrain it to a known value. In particular, it is often necessary to assume a diffusion-limited collision rate (typically 100 to 1000 $\mu M^{-1}s^{-1}$), but clearly state the assumption.

CAUTION! The real danger in fitting data globally to a model is in trying to convince yourself and your colleagues that a model is well determined when it is not. Such overstatements of the utility of global fitting actually undermine confidence in the method! You do not get something for nothing simply because you fit your data globally. The *information content* of the data still limits the conclusions that can be made. What you gain is a more rigorous interpretation of the data because more of the information content (particularly amplitude information) can be extracted from the data, while assumptions necessary for mathematical solutions are eliminated.

NONETHELESS, fitting globally does yield information not attainable in fitting data in a piecemeal fashion. The whole really is greater than the sum of the parts. This conclusion is based on the requirement that the rate and amplitude of all reactions must be accounted for by one universal model. Confidence contour analysis demonstrates the *information content* of the data.

Dynamic simulation was implemented in designing *KinTek Explorer* so that you can see the relationships between rate constants, output factors, the model and the data. Take advantage of the powerful method to fully explore the fitting landscape, critically evaluate your model, and carefully state your conclusions. In Chapter 8, we present our methods to explore the fitting landscape systematically through our *FitSpace* function.

7.4 BatchFit

New with Version 2.5, we facilitate the fitting of a large number of identical trials to a single model using the *BatchFit* utility. Moreover, multiple experiments (as defined within the *KinTek Explorer* mechanism file) can be fit for each trial set, described below.

1. Prepare your data files with sequentially numbered file names according to the following format.

xxxx.yy.anyname.txt

xxxx is a four digit *trial number* to define each sequential data set number.

yy is a two-digit *experiment number* starting with 01 to define the *experiment* number according to the numbering within the *KinTek Explorer* mechanism file.

anyname is any text sequence you choose to identify your data sets. Note that the program ignores the *anyname* portion of the filename so it can be different for each data set and so you can use it to help you to identify the differing reaction conditions.

For example, the following shows the names for three *trial* sets, 0001, 0002, and 0003. Each reaction condition was used to perform two experiments, 01 and 02. Each set of the two experiments will be fit to the model after loading experiment 01 and associating it with *Experiment 1* and loading experiment 02 and associating it with *Experiment 2* in your mechanism file.

0001.01.compound_1253.txt
0001.02.compound_1253.txt
0002.01.compound_3245.txt
0002.02.compound_3245.txt
0003.01.compound_675.txt
0003.02.compound_675.txt

Each file should contain data in a format (tab-delimited text) such that importing the data to the specified experiment may proceed automatically–this can be tested by manually importing an appropriate file for the currently selected experiment. You should be able to accept the imported data without editing and see that the data maps correctly to the observables as you would expect when you drag the data onto the appropriate experiment. You can also check the pressing Fit All Exp yields a good fit to the data. That is

your starting estimates contained within the *master.mec* file are close enough to the average that the fitting process will converge.

Note: when batch-fitting data, the .mec file that is loaded when the batch fitting process is initiated specifies the initial parameter values, including concentrations, for *all* subsequent fits.

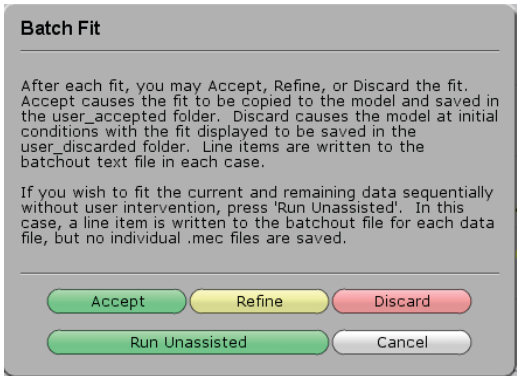
2. Load the *master.mec* mechanism file that is the "base" for all subsequent fits. The mechanism file that you have loaded should contain the number of experiments that you have data for, and want included in the global fit. Thus if you have 3 experiments in your *master.mec* file, you must have <trial#>.03.anyname.txt data files to provide data for the third experiment. You can use any name you choose for the *master.mec* file.

3. Click on 

from the *DataFit Editor*, and then a "Choose File" dialog appears. Choose the data file that is the first trial that you want included in your results (e.g. 0001.01.anyname.txt). You are not restricted to starting with number 0001; you could start with 0125, for example. The program will start with this trial, load the data, fit it and then look for sequential numbering to find the next file within the same directory. The process ends when the last numbered file in the directory is reached.

As the batch fitting process proceeds, fits are left onscreen momentarily to allow a visual overview of your data applied to the loaded mechanism. It is actually entertaining to watch the automatic loading and fitting of data, at least for the first 100 trials or so!

New with Version 8.0, we allow you to run the batch-fit with more interaction and time for inspection of each fitted data set. The menu shown at the right will pop up when you start a batch-fit run so you have time to inspect each trace with the options to *Accept*, *Refine* or *Discard* the fit. In either case, we now save a .mec file for each data set so that you can go back to look at the results later. If the batch fit seems to be running well and/or you have a lot of data sets, you can click *Run Unassisted* to proceed without further input.



A dialog will tell you when the process is complete, how many trials were fit, and the name of the output summary file (the batchout.txt file). If the batch fitting process tries to import data that does not exist, the process will terminate, and the normal dialog will display the number of trials fitted

During the fitting of the data in each trial, a line is added to a text file summarizing the results. The file will list the trial number, chi2 (before the fit), chi2fit (after the fit) and then each rate constant and its standard error. For example:

Sample Batchout.txt file

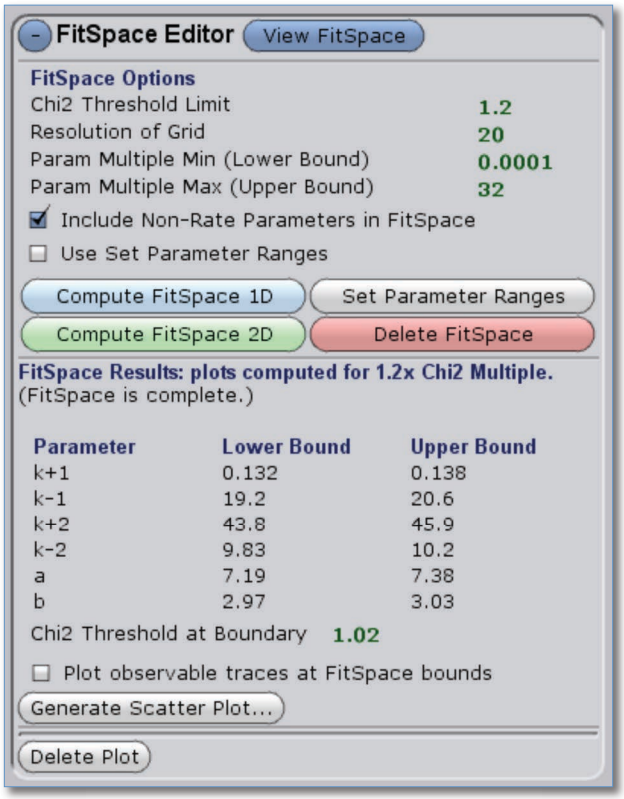
trial	chi2	chi2fit	k+1	err	k-1	err	k+2…
0001	0.532	0.00253	0.674	0.0297	33.465	0.9561	11.422
0002	1.941	0.00562	0.498	0.1617	69.761	4.0567	20.705
0003	0.285	0.00861	0.487	0.0937	34.006	2.1563	13.217
….							
0015		0.00510	0.426	0.0790	36.581	1.9150	15.552

This text file can then be imported into Excel or another program to examine the results. If a particular trial failed to converge to a good fit as indicated by the chi2fit value, or if there are large errors in the rate constants, you can load that data set and look to see why the fit failed. As with any automated process, there is no substitute for loading and fitting several data files individually to find starting estimates of rate constants and output factors that are close to an average value so that the fitting of each data set is more likely to converge without altering the starting estimates.

Chapter 8. FitSpace Editor

Use the FitSpace Editor to:

- Set parameters limiting the *Fitspace* search
- Compute *Fitspace*
- View *Fitspace*
- Set the chi^2 boundary error estimation
- Plot traces to display the error limits
- Get upper and lower boundaries on parameter values



In our data fitting routines, standard error analysis is based upon the covariance matrix resulting from nonlinear regression. The standard errors reported using these methods always understand true experimental errors because the methods are based upon the assumption that all errors are *independently and identically distributed*; often this is not the case because of non-uniform variation from ideal behavior. More importantly, in many cases standard errors are extremely misleading and imply a well constrained fit, when, in fact, there is a large space over which parameters can vary but the various parameters are related by complex, nonlinear functions. Here we make an important distinction. If the nonlinear regression converges to produce a fit to the data, it can be described as a good fit in that the errors have reached a minimum with Chi^2/DoF approaching unity and the fitted curve appears to go through the center of the distribution of data points. However, this does not mean that the parameters are well constrained or that the fit is unique. The computation of *FitSpace* allows you to explore the landscape over which parameters can vary while still achieving a good fit to the data. This is described in more detail in our recent papers (2,3).

8.1 Confidence contours

After a fit to a data set is achieved, you can simply click on either button below to begin the calculation.

This allows computation of chi^2 as each parameter is varied individually. At each value for the individual parameter, all other parameters are allowed to float in finding the best fit by nonlinear regression. The resulting chi^2 value is then displayed as a function of the variable parameter. This information is sufficient to define upper and lower boundaries for each parameter.

Compute FitSpace 2D This function provided a pair-wise 2 dimensional search. This calculation is computationally much more expensive than the 1D analysis, but can reveal complex relationships between parameters.

a. Compute FitSpace 1D

The one dimensional FitSpace calculation explores the range over which each parameter can vary while still allowing an acceptably good fit. While holding a given constant fixed, all other parameter are allowed to vary in seeking the best fit to the data. The process is repeated over a range of parameter values to generate a plot of normalized chi^2 values (chi^2 divided by the minimum chi^2) versus parameter value as shown in Figure 8.1.

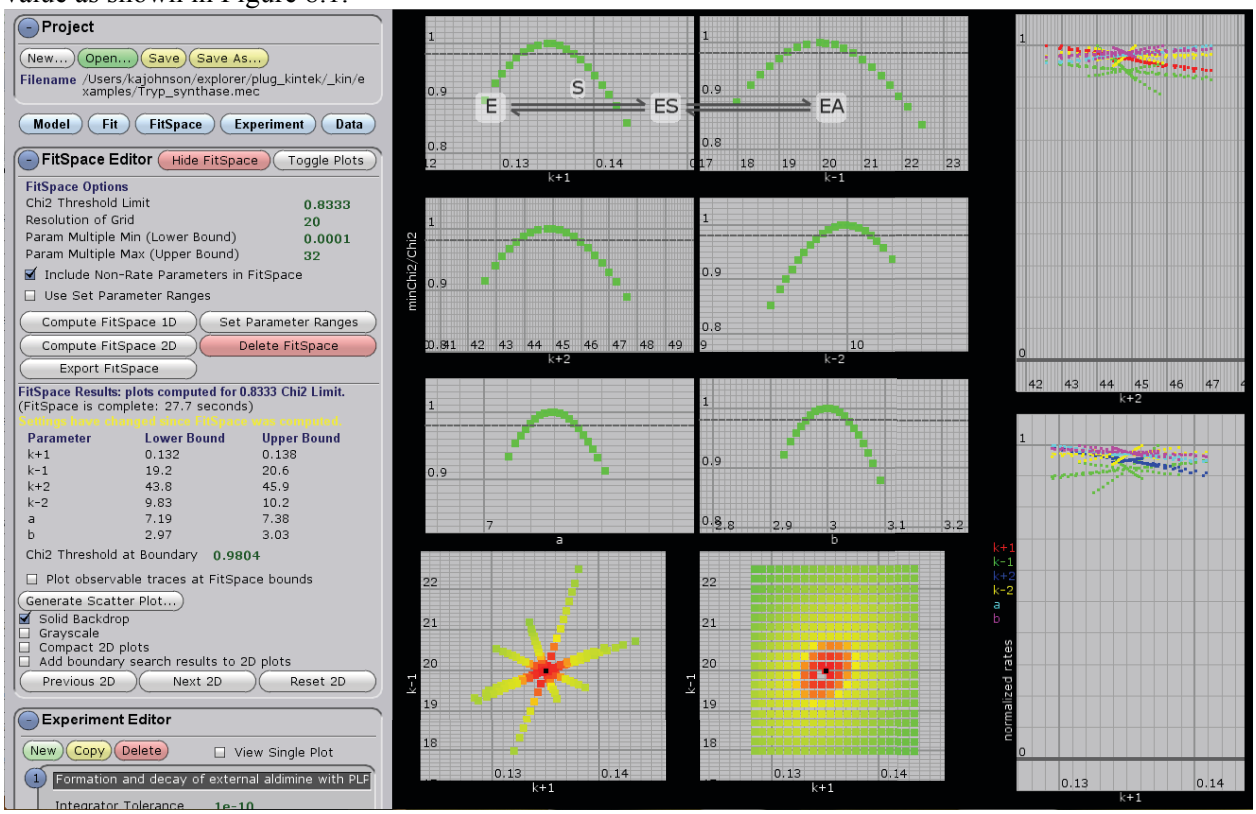


Figure 8.1 *Example of well-constrained parameters in 1D FitSpace.* This sample is take from the Tryp_synthase.mec file, and shows that all four rate constants and two fluorescence output factors are well constrained by the data. The chi2 plots show well defined upper and lower boundaries for each parameter. Setting a threshold in the chi2 plots allows estimation of upper and lower error boundaries for each parameter.

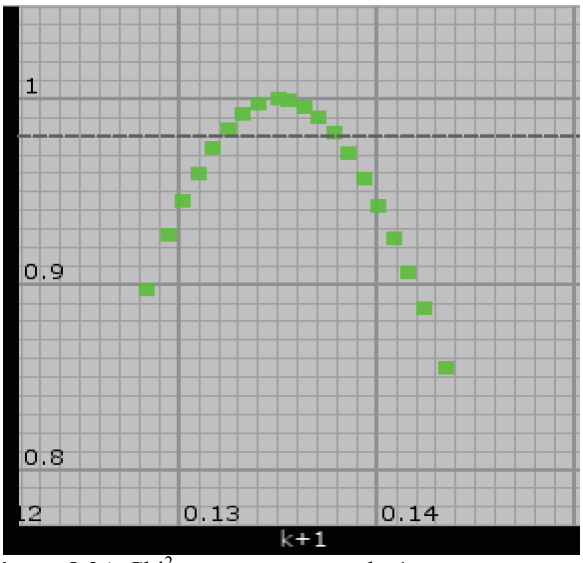


Figure 8.2A Chi^2 versus parameter k+1.

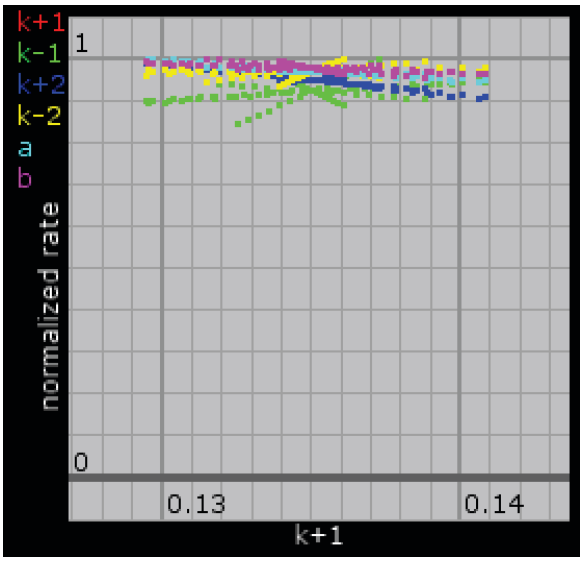


Figure 8.2B Range of normalized parameter values versus parameter k+2

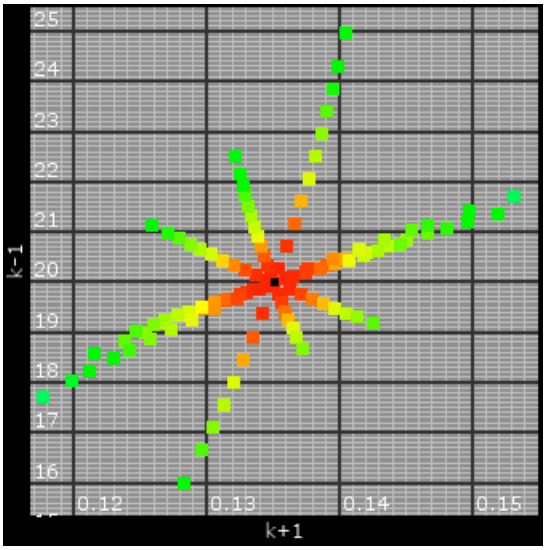


Figure 8.2C Composite of all 1D plots for k-1 versus k+1.

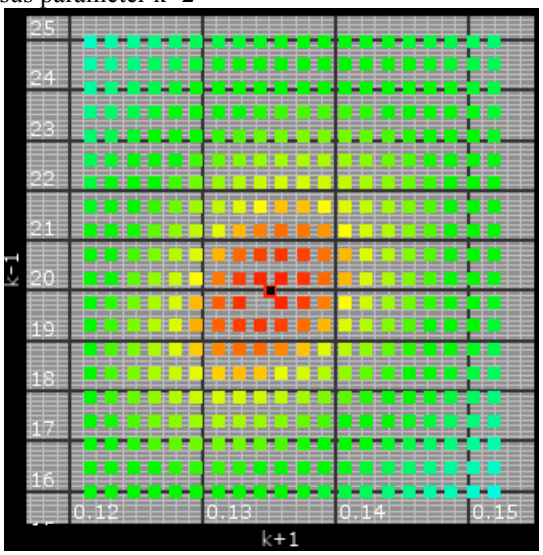


Figure 8.2D 2-Dimensional Confidence Contour.

Figure 8.2A shows the plot of $Chi^2/MinChi^2$ versus the rate constant k_{+1}. Using a threshold of 0.98 provides upper and lower limits of 0.132 and 0.138, respectively, with a best-fit value of 0.135.

Figure 8.2B shows a plot all values (normalized by dividing by the maximum value for each parameter) that produce a chi2 within 1% of the minimum (threshold of 0.99). This plot is useful in that it illustrates relationships between parameters when they are not well constrain. In this case, all parameters are within about 90% of their maximum value indicating that the parameters are well defined.

Figure 8.2C illustrates a composite resulting from the 1D FitSpace calculations using a false color to display the normalized chi2 values and illustrating the relationship between k_{-1} and k_{+1}. Each time a given constant is examine as in Figure 8.2A, values for all other constants are derived as well. These are then all added together to generate Figure 8.2C. One can consider this as a composite of various trajectories across the three dimension confidence contour illustrated in Figure 8.2D.

Figure 8.2D shows the complete surface computed by filling in a grid of calculations systematically examining pairwise combinations of k-1 and k+1. The central red area defines the range of values giving a chi2 within a set threshold (1.1 in this case).

b. Compute FitSpace 2D

In principle, one would like to examine the data fitting landscape in *n+1* dimensions, for *n* independent parameters. Because that is not possible, we examine parameters in all possible pair-wise combinations to examine the extent to which any two parameters can co-vary and produce a good fit, as judge by chi^2. To begin the exploration of *FitSpace*, the program steps through each variable parameter (rate constant or scaling factor) and holds it fixed, while allowing all other variables to float as nonlinear regression seeks the best fit. The process is repeated until the range of each parameter is mapped. We then plot a 3D graph showing the dependence of the reciprocal of the normalized chi^2 value versus each parameter pair (x, y axes) as shown below. That is, at each x,y pair on this grid, chi^2 is calculated for the best fit, attained by allowing all other parameters to be adjusted. We then plot the $Chi^2_{min}/Chi^2_{x,y}$ on the vertical axis so that it falls on a scale of 0 to 1, with 1 being the best overall fit. The color-coding and the scale on the right gives the absolute chi^2 value calculated. This example shows clearly that the rate constants k_{+2} and k_{-2} in the model are well constrained. The process is repeated for all of the pair-wise combinations of variable parameters to generate a graph as shown on the following page, derived from the tryptophan synthase example (13).

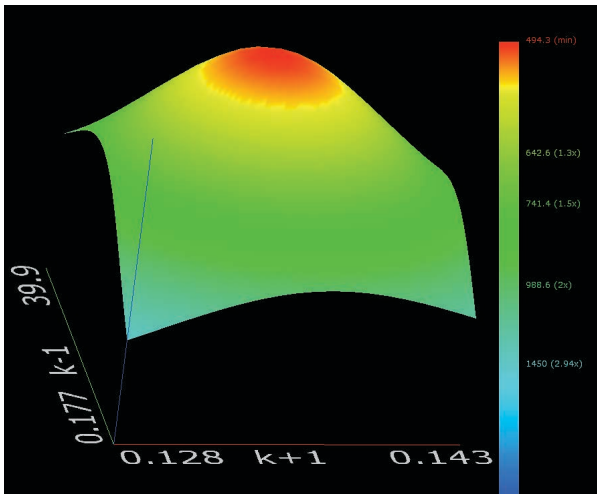


Figure 8.3 *Confidence Contour.* The figure at the right shows the 3-dimensional surface generated in computing the normalized best-fit values of chi^2 as k_{+1} and k_{-1} are systematically varied in the Tryp_synthase.mec example.

Standard errors in parameters can be obtained by determining the valued at which chi^2 (χ^2) crosses a threshold defined by:

$$\frac{\chi^2_{threshold}}{\chi^2_{\min}} = 1 + \frac{p}{n-p} F^{\alpha}_{p,n-p}$$

where $p =$ the number of parameters

$n =$ the number of data points

$F^{\alpha}_{p,n-p} =$ the F distribution for 1-α error limits.

After fitting data, the program displays a recommended chi^2 threshold value computed from the equations given above, calculated as described by Bates & Watts (11).

Starting with Version 6.0 we use an inverse function of the threshold which is better for graphical display because as χ^2 gets larger, the inverse function approaches zero rather than infinity.

$$Chi^2 \text{ ratio threshold} = \frac{\chi^2_{\min}}{\chi^2_{threshold}}$$

Although this theory establishes the principle of using a threshold in chi^2 to estimate confidence intervals, in our opinion, the thresholds set by this calculation greatly overestimate confidence in

parameters when fitting data sets with thousands points because a single trace does not consist of thousands of independent and identically distributed points; rather the data are linked, thereby reducing the effective number of independent data points. See (2) for a discussion of this issue. For now, we generally recommend a 5-10% increase in chi^2 as the basis for setting 95% confidence limits. This is based upon the assessment that there are approximately 10-30 independent line segments within a given trace, not 1000 independent points, for example. Calculations of the F-distribution based upon only 10-30 data points per trace provide chi^2 threshold values that are more useful. Moreover, as described below, the most important conclusions derived from the confidence contour analysis do not depend greatly on the value chosen for the chi^2 threshold. Rather, it is most important to establish order of magnitude estimates for the extent to which each parameter is constrained by the data.

Note: The chi^2 threshold is a scrollable parameter representing the Chi^2 ratio threshold:

Chi2 Threshold at Boundary 0.9.

8.2 FitSpace display

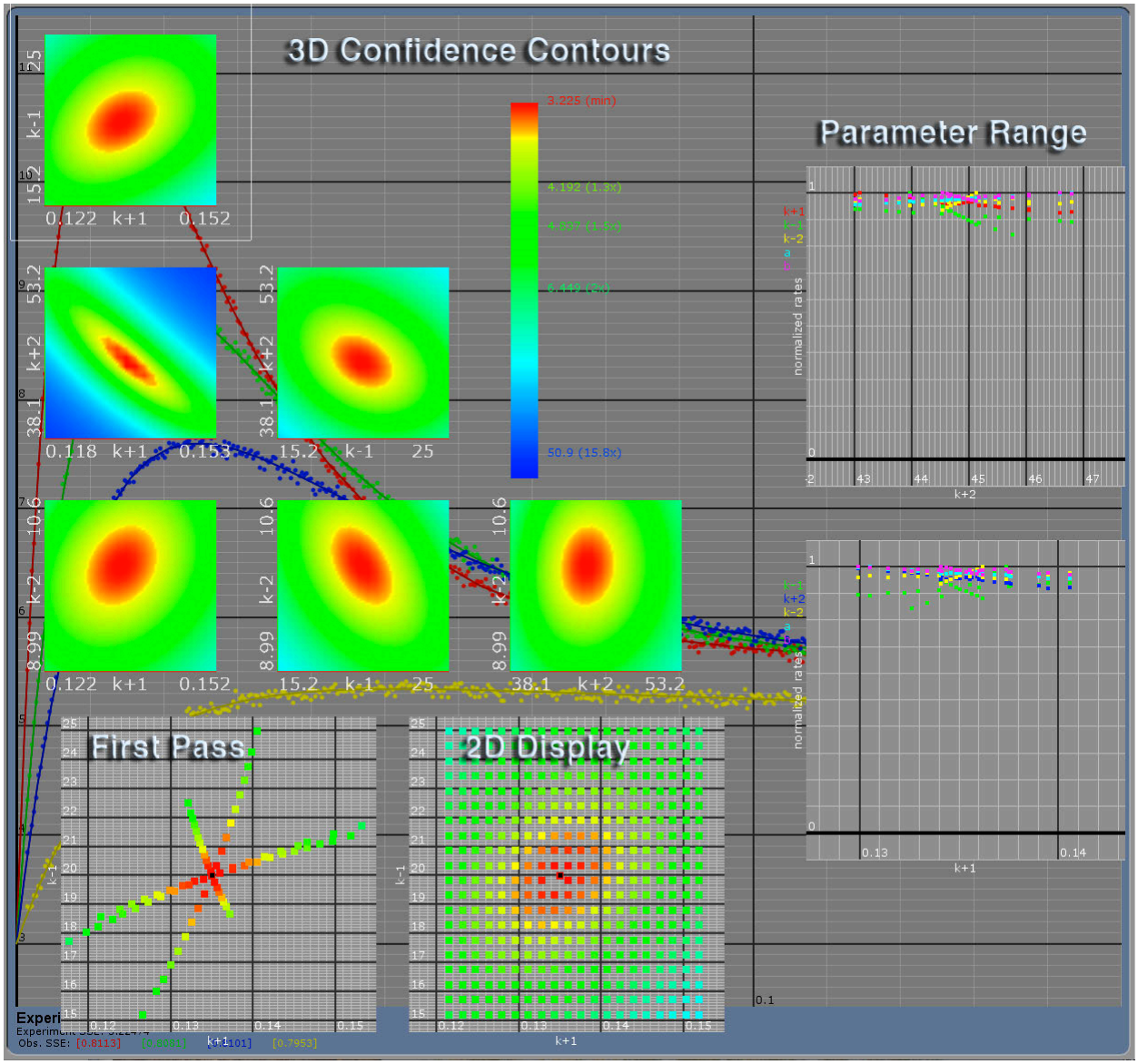


Figure 8.4 *FitSpace Display.* This figure illustrates the various parts of the FitSpace display:

First Pass shows the results the initial excursions to map the boundaries of a good fit.

2D Display shows the grid of x,y pairs obtained from a stepwise grid search.

Parameter Range shows the dependence of all parameters on each forward rate constant that was allowed to vary. On this graph, each point represents a parameter that produced a fit within 10% of the minimal chi^2. The parameters are normalized to their maximum value. Note the parameters are clustered within 0.8 to 1, implying that in this example, the parameters are well constrained. Contrast this display with that shown on the following page.

3D Plots region show the 3D representation of the chi^2 variation for each pair of rate constants. Note that each pair of rate constants is well constrained with a well-defined local minimum. The ridge in the relationship between k_{+2} and k_{+1} indicates that the product $k_{+1}*k_{+2}$ is more constrained than either value individually.

The data contained in the Tryp_synthase.mec file provides well-constrained estimates for all six parameters, consisting of four rate constants and two fluorescence-scaling factors. The ability to constrain six parameters to achieve a good fit for the tryptophan synthase data is dependent upon data collected over a range of concentrations. The concentration dependence of the rates and amplitudes, as well as the starting and ending points, allow definition of the rate constants and scaling factors. As a point of contrast, we show the results of a *FitSpace* calculation where data from only one of the concentrations is included. In this case, the parameters are clearly not well constrained. Note especially that there is no lower limit for k-1, while the other parameters can vary over a wide range of values, as evidenced by the *Parameter Range* plots as well as the shape and range of the 3D plots. These examples illustrate how the *FitSpace* calculation gives immediate visual clues to indicate whether a fit to a data set is well constrained or not.

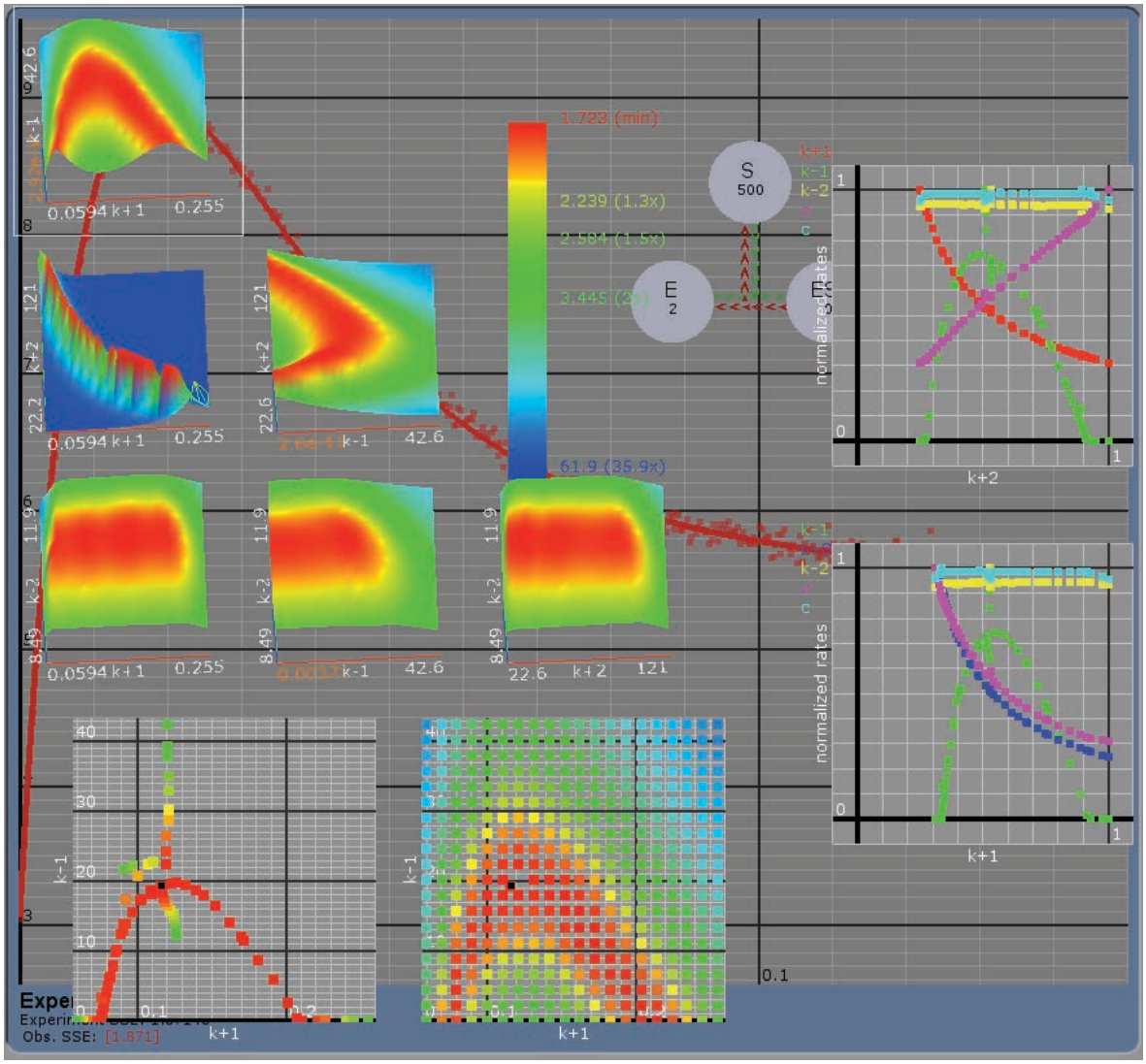


Figure 8.5 *3D-display of confidence contours*. Note that the 3D graph for k_{+2} versus k_{+1} shows an extended ridge defining a function $k_{+1} \cdot k_{+2} = C$, a constant. Because of the grid sampling, this appears as a series of peaks, but it is a continuous function.

Examination of the *FitSpace* contours can provide more reliable error estimates on individual parameters than that obtained by conventional nonlinear regression analysis. A given threshold of the normalized chi^2 values can set a boundary to define the acceptable limits on a given parameter. As a visual cue, the orange/yellow boundary represents a 20% increase in chi^2 and may represent a reasonable boundary for acceptable fits to the data. We use this boundary to provide estimates for upper and lower bounds on parameters, given in the table under *FitSpace* Options.

8.3 Controlling the FitSpace search

After a fit to a data set is achieved, you can simply click on either button below to begin the calculation as described above.

This allows computation of chi^2 as each parameter is varied individually.

This function provided a pair-wise 2 dimensional search to look for relationships between parameters. This calculation is computationally much more expensive than the 1D analysis.

NOTE: The *FitSpace* calculation will include all experiments in the mechanism file. If you want to omit some a given experiment from the *FitSpace* calculation, check the box *Excluded from Global Fitting* under the *Experiment Editor* for that experiment.

CAUTION: It is not advisable to run *FitSpace* if there are parameters that are clearly not well constrained. The purpose of the *FitSpace* calculation is to determine the extent to which parameters are constrained by the data and to look for relationships between parameters. There are several ways to look for under-constrained parameters before running *FitSpace*. In particular, unconstrained parameters are revealed by:

- Nonlinear regression returns a value of "ND" (not determined) for the standard errors
- Fitting returns a value close to zero and scrolling the parameter has no effect on the curves. In this case it is advisable to lock the parameter at zero.

In these cases, examine the model and find ways to simplify it before attempting the *FitSpace* calculation. You can certainly run the *FitSpace* calculation, but it will take a long time in searching for fits with parameters that are not well constrained.

a. Parameters to control the *FitSpace* computation

Chi^2 Multiple Max: range 0.1 to 10, default 2.0. This factor sets the limit on the variation in chi^2 as the program searches the space over which a parameter can be varied. For example, when set to 0.6, the program stops searching when the chi^2 exceeds 60% of the minimum value. A smaller number provides better resolution around the boundary and also limits the search when parameters are not well constrained. A larger number allows a broader search.

Resolution of Grid: range 5-20, default 10. This parameter sets the resolution of the grid search; for example, a value of 10 gives a 10x10 grid. Higher numbers give smoother 3D graphs but require more computation time.

Param Multiple Min (Lower Bound): range 0.0001 to 1, default 0.001. This parameter sets a limit on the fold variation while decreasing a parameter before the program stops searching for a lower boundary.

Param Multiple Max (Upper Bound): range 1-100, default 32. This parameter sets a limit on the fold variation while increasing a parameter before the program stops looking for an upper boundary. For example, if a parameter has no upper boundary in that increases in the value of the parameter do not increase chi^2, then the program will stop when the variable exceeds 100-fold the starting value.

After setting the parameters, click *Compute FitSpace* to perform the

calculation. The calculation can take anywhere from a few minutes to an hour depending upon the complexity of the system and the speed of your computer. Note that you can stop the calculation, save the file, and resume later. Also, there is a great deal of information derived from the initial pass to explore the limits of parameter space and so generation of the full contours may not be needed.

☑ Autosave Progress. If this is checked, the results of the FitSpace calculation will be saved during the calculation.

Delete FitSpace The results of the FitSpace calculation are stored with the *.mec* file. If the model is changed, the results may no longer be relevant and should be deleted.

After the *Fitspace* computation is complete, a table showing upper and lower bounds of each parameter appears as illustrated in Figure 8.6. This is based upon the threshold set in chi^2; a threshold of 0.9 represents a 10% increase in chi^2.

b. Setting the *FitSpace* chi^2 threshold

Boundaries reflect fits within Chi^2 Multiple **0.98** This value defines the threshold in the confidence contours that is used to estimate errors on each parameter. It can be scrolled or edited and the 3D confidence contours, and upper and lower limits on rate constants are updated accordingly. The threshold is illustrated in the 3D confidence contours as the yellow band between the red (best fit) and green areas.

View FitSpace Hide FitSpace

This toggle controls the display of the *FitSpace* graphs. The results of a *FitSpace* calculation are stored with the *.mec file and can be reviewed pressing *View FitSpace*.

Previous 2D Next 2D

This pair of buttons allows the user to scroll through the individual 2D plots.

Toggle Plots Toggles the *FitSpace* display between full display, an enlarged version showing only the 3D plots and a display showing the 1D plots.

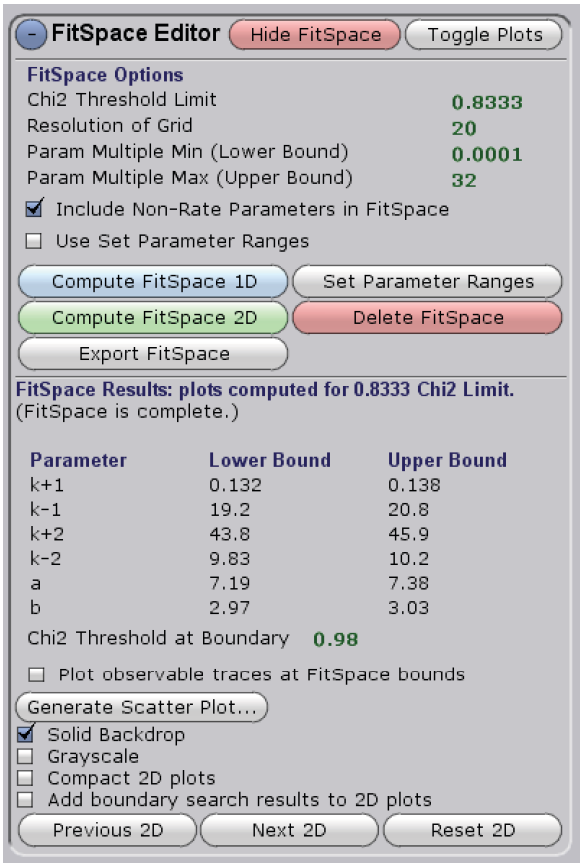


Figure 8.6 *FitSpace Editor* control panel.

☑ Plot observable traces at boundaries: This overlays the resultant confidence intervals on the individual traces—see *Section 8.4.b.*

☑ Solid Backdrop. This checkbox toggles between a black background and a grayed version of the data screen.

☑ Grayscale. This checkbox toggles between colored and grayscale version of the 3D plots according to your personal preference.

☑ Compact 3D plots. This checkbox removes spaces between 3D plots making for a better publication quality figure from a screen capture.

☑ Add boundary search results to 3D plots. This checkbox allows additional points obtained in searching for boundaries of acceptable fits to be added to the 3D contour plots.

Mouse Control: By clicking on one of the 3D plots with the middle mouse button and dragging the mouse, the angle of the 3D view can be changed to see the contours more clearly.

c. Set Individual *FitSpace* Bounds

Pressing this button brings up the table shown at the right, allowing manual entry of upper and lower limits for the FitSpace 2D figures. As an aid to entering values, you can select Use Param Multiple to get the default values found in the initial search, or you can click on Use Chi2 Threshold to pull values from the current upper and lower limits according to the Chi^2 set threshold.

This is particularly useful when you want to create a publication quality figure with a good display of the confidence contours. For this it is useful to set the Chi2 Threshold higher than the value you will use to establish upper and lower boundaries on each parameter in order to have a clear definition of the area of best fit. Then click on Use Chi2 Threshold and run the FitSpace calculation.

☑ Use Individual Parameter Bounds. Check this box to tell the program to use values from the manual entry table of parameter boundaries.

c. Publishing *FitSpace* contours

For publications, we recommend performing a screen capture of the *FitSpace* contour display using a high resolution computer display. The images can then be edited in Adobe Photoshop and Illustrator in order to generate publication quality labels.

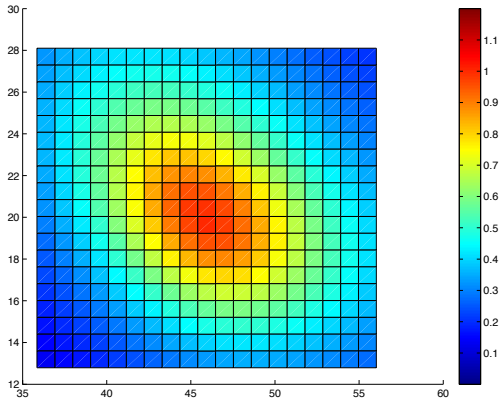
Alternatively, the data defining the confidence contours can be used with an external program to generate a three dimensional plot. When a mechanism file containing a *FitSpace* calculation is saved, the data defining the confidence contours are output to a *.fsp text file. In this file, there is a table of chi^2 values for each pair of rate constants as illustrated in the example shown below, taken from the Tryp_synthase.mec.

$k_2\backslash k_{-1}$	12.8	13.6	14.4	15.2	16.0	16.8	17.6	18.4	19.2	20.0	20.8	21.7	22.5	...	28.1
35.85	0.13	0.15	0.17	0.19	0.22	0.24	0.27	0.29	0.31	0.33	0.35	0.35	0.35	...	0.26
36.92	0.15	0.17	0.19	0.22	0.25	0.28	0.31	0.35	0.37	0.40	0.41	0.42	0.42	...	0.28
37.98	0.17	0.19	0.22	0.25	0.29	0.33	0.37	0.41	0.45	0.48	0.49	0.50	0.50	...	0.31
39.04	0.19	0.22	0.25	0.29	0.34	0.38	0.44	0.49	0.53	0.57	0.59	0.59	0.58	...	0.33
40.10	0.21	0.24	0.28	0.33	0.39	0.45	0.51	0.57	0.63	0.67	0.69	0.69	0.67	...	0.35
41.16	0.23	0.27	0.32	0.38	0.44	0.51	0.59	0.67	0.73	0.78	0.79	0.78	0.75	...	0.36
42.22	0.25	0.30	0.35	0.42	0.50	0.58	0.67	0.76	0.83	0.88	0.89	0.86	0.81	...	0.37
43.28	0.28	0.33	0.39	0.47	0.55	0.65	0.75	0.84	0.92	0.95	0.95	0.91	0.84	...	0.37
44.35	0.30	0.36	0.42	0.51	0.60	0.70	0.81	0.90	0.97	1.00	0.98	0.92	0.84	...	0.36
45.41	0.32	0.38	0.45	0.54	0.64	0.74	0.84	0.93	0.98	0.99	0.96	0.90	0.82	...	0.35
46.47	0.34	0.40	0.48	0.56	0.66	0.76	0.85	0.92	0.95	0.95	0.91	0.84	0.76	...	0.33
47.53	0.35	0.42	0.49	0.57	0.66	0.75	0.82	0.88	0.89	0.88	0.83	0.77	0.70	...	0.31
48.59	0.36	0.42	0.49	0.57	0.65	0.72	0.78	0.81	0.82	0.79	0.75	0.69	0.63	...	0.29
49.65	0.37	0.42	0.49	0.56	0.62	0.68	0.72	0.74	0.73	0.71	0.67	0.62	0.56	...	0.28
50.71	0.37	0.42	0.47	0.53	0.58	0.62	0.65	0.66	0.65	0.63	0.59	0.55	0.50	...	0.26
51.78	0.36	0.41	0.46	0.50	0.54	0.57	0.59	0.59	0.58	0.55	0.52	0.48	0.45	...	0.24
52.84	0.35	0.39	0.43	0.47	0.50	0.52	0.53	0.52	0.51	0.49	0.46	0.43	0.40	...	0.22
53.90	0.34	0.37	0.41	0.43	0.45	0.47	0.47	0.46	0.45	0.43	0.41	0.38	0.36	...	0.21
54.96	0.33	0.35	0.38	0.40	0.41	0.42	0.42	0.41	0.40	0.38	0.36	0.34	0.32	...	0.19
56.02	0.31	0.33	0.35	0.37	0.38	0.38	0.38	0.37	0.36	0.34	0.33	0.31	0.29	...	0.18

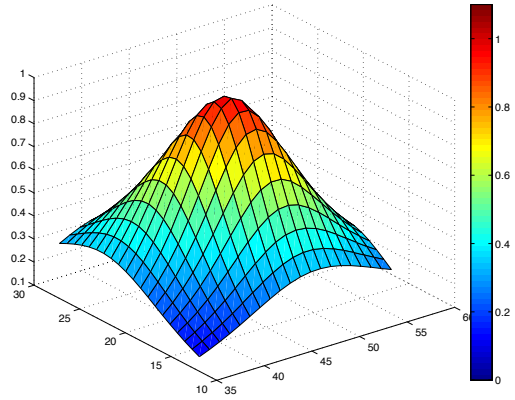
This table shows a truncated version of the 20x20 matrix of chi^2 values where the first row defines the values of the rate constant k_{-1} and the first column defines the values of k_2. Several programs are available for creating a three dimensional graph from these data. The simple Matlab script below can read the text file and generate one of several graphs illustrated below. You must first edit the text file to replace the "k2\k-1" text with a 0 as a numeric placeholder, so that the data can be easily read into Matlab.

d. Sample *Matlab* script

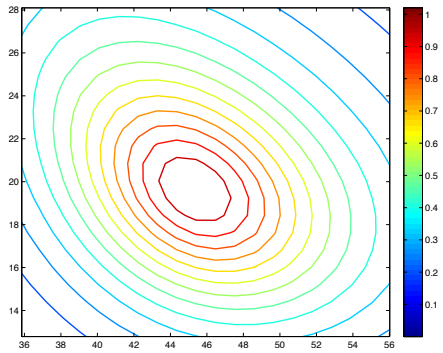
```
close all;
clear all;
input_dir='/Users/kajohnson/Desktop/';
filename='TSk2k-1.txt'  % Input file name
cd(input_dir);
data=importdata([filename]);

X = data(2:end,1);
Y = data(1,2:end);
Z = data(2:end,2:end);

colormap jet; %set the color scale, others are available
figure(1);
surface(X,Y,Z); %create the graph
colorbar %add the color index
caxis([0 1.2]); %modify range of color mapping
print(gcf, '-depsc', strcat(filename,'_surface')) %output EPS file
```



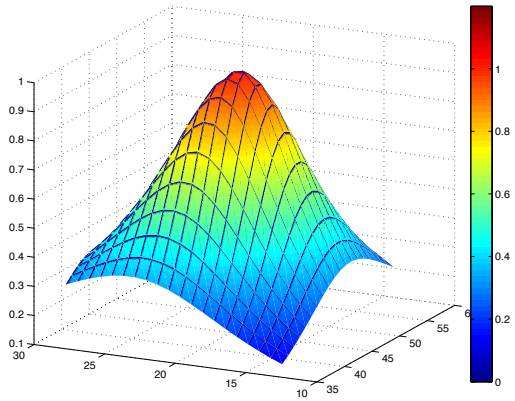
Command: surface(X,Y,Z);



Command: surf(X,Y,Z);



Command: contour(X,Y,Z, 12, 'Linewidth', 1.5);



Command: surf(X,Y,Z); shading interp; view(-64,18);

Figure 8.7 *Examples of 3D rendering from Matlab*. Each figure shows the result obtained with the different commands indicated in the legend for each figure.

8.4 Errors on fitted parameters

The *FitSpace* calculation allows more reasonable estimates of errors on the parameters than are achieved by conventional nonlinear regression. However, it should be noted that this is an area of less than exact science. We all take comfort in reporting the 95% confidence intervals from the nonlinear regression, but the standard errors always underestimate the true error, sometimes to a large extent.

a. FitSpace boundary error limits

Boundaries reflect fits with Chi^2 multiple X: This scrollable parameter sets a threshold on the *FitSpace* confidence contour for computation of the ranges allowed for fitted parameters. Note that as you scroll the value, the color boundary on the 3D plots changes to show the new threshold as the yellow boundary between red and green (or the edge of the white center in a grayscale rendition). After you release the mouse button, the table showing Lower and Upper Bounds will be updated. For most experiments, a threshold of 0.9 gives a reasonable range for errors. With thousands of data points that are independently and identically distributed, a value of 0.9 to 0.99 may be appropriate (2,3). With only 40 data points, a value of 0.75 may be more appropriate. The software computes a recommended threshold, which is based upon the assumption that the data are all independently and identically distributed and that a good fit is achieved so the observed chi^2 can be accounted for by the known sigma values. This often not the case with real data, so we recommend a more conservative (lower) threshold than suggested.

The table below illustrates the problem based on error analysis of parameters derived in fitting the tryptophan synthase sample data. According to the nonlinear regression, each parameter is known within less than 1% error. However, the *FitSpace* shows that several of the parameters can vary over a much larger range. In particular, k_{-1}, should be reported as 20 ± 5, which is a more realistic estimate than 19.9 ± 0.37

Parameter	Best Fit	Std Error	Lower Bound	Upper Bound
k_{+1}	0.135	0.001	0.128	0.144
k_{-1}	19.9	0.37	16	25
k_{+2}	45.1	0.46	42.8	46.5
k_{-2}	10.0	0.072	9.5	10.7
a	7.33	0.046		
b	3.00	0.014		

A more troubling example is shown in the racemase.mec file. Nonlinear regression reported that k_{+3} = (2.55 ± 0.044) x 10^9 s^{-1} and k_{-2} = (6.6 ± 0.17) x 10^8 s^{-1}. However, *FitSpace* exploration and dynamic simulation each reveal that these constants are not well constrained at all. They are bounded on the lower limit only by k_{cat} in each direction, while there is no upper limit for either parameter.

Note added January 23, 2009: Recent improvements in data fitting, principally due to performing all calculations in double precision (64 bit), have overcome some of the problems with error estimates derived by nonlinear regression when systems are under-constrained. In particular, the current version of the software (Version 2.0.1059) returns errors for the racemase_4step.mec of k_{+3} = (2.44 ± 0.63) x 10^9 s^{-1} and k_{-2} = (6.64 ± 6.3) x 10^8 s^{-1}. Although this represents a significant improvement in dealing with situations where error terms are large, nonlinear regression standard errors still fail to reveal the full extent to which parameters are under-constrained.

b. Plotting confidence intervals

We have added a new function to show the overlay of all possible fitted curves within the boundaries defined by the *FitSpace* confidence contour threshold.

☑ Plot observable traces at boundaries: This option appears after completing a FitSpace contour analysis. When this item is checked, all of the reasonable fits to the data will be overlaid with the data and the best fit, as shown below. The curves are computed based on the parameter sets defined by the lower and upper limits of each variable parameter according to the value set for "Boundaries reflect fits within chi^2 multiple *x*", where *x* is the user input value for the threshold. After the threshold is changed, the display is updated if the *Plot observable traces at boundaries* function is checked. It is important to note that these curves are not simply the traces obtained by varying each parameter individually while keeping all other parameters fixed at their best fit values. Rather, for each overplayed trace, one parameter is set to its upper or lower limit, while all other parameters are adjusted to display the best fit attainable within the set chi^2 threshold. The process is repeated for the upper and lower limits of each variable parameter, to overlay the family of acceptable traces (in white) behind the data and the best fit (in color). For comparison, consider the examples such as 8oxodG.mec, where there are fewer points to define the parameters as accurately; although, they are still well constrained. In racemase_3step.mec, the parameters are totally unconstrained (except for lower limits), yet all of the traces superimpose on the data. This function provides one more tool to evaluate the relationships between the data and the allowable range on the fitted parameters.

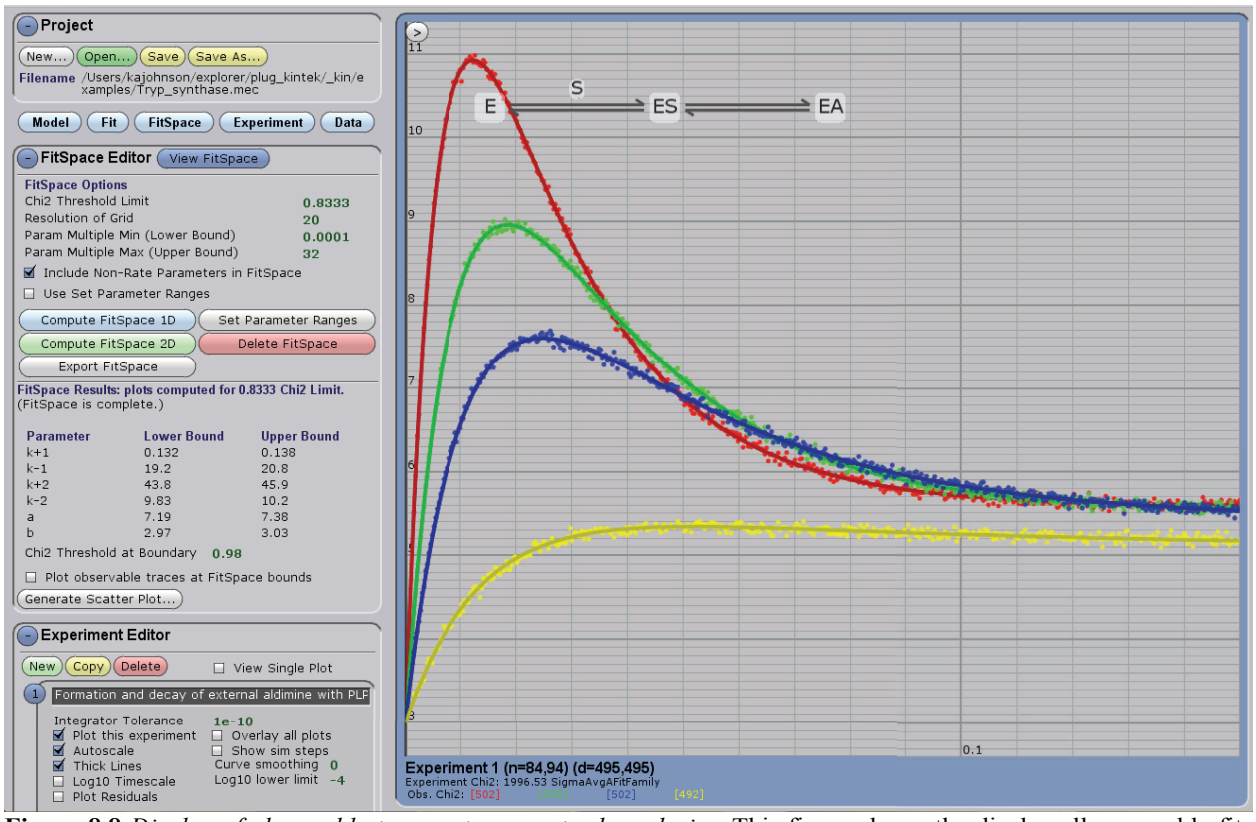


Figure 8.8 *Display of observable traces at parameter boundaries.* This figure shows the display all reasonable fits to the data obtained for Tryp_synthase.mec within a chi^2 threshold of 0.98.

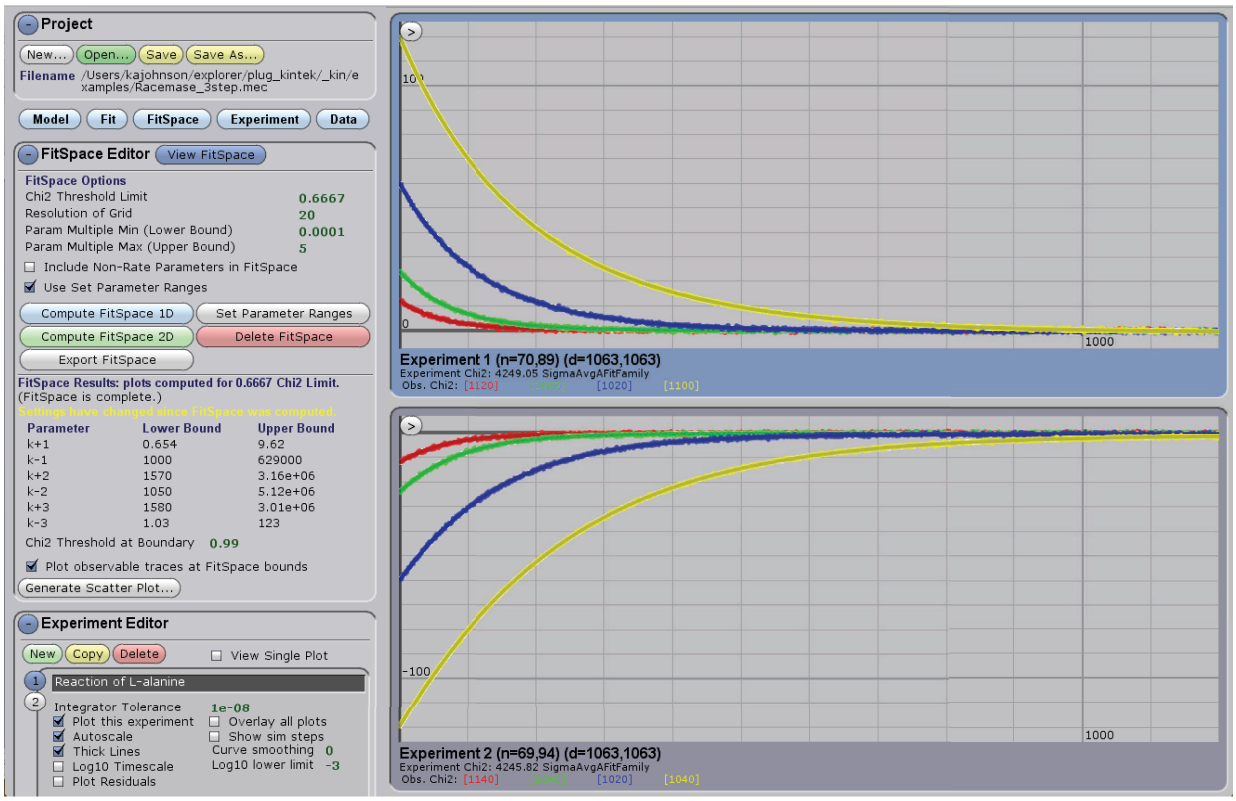


Figure 8.9 *Display of observable traces at parameter boundaries*. This figure shows the display all reasonable fits to the data obtained for *Racemase_3step.mec* within a chi^2 threshold of 0.99. Note the wide range of the parameters used to generate the plots shown here in spite of the high threshold.

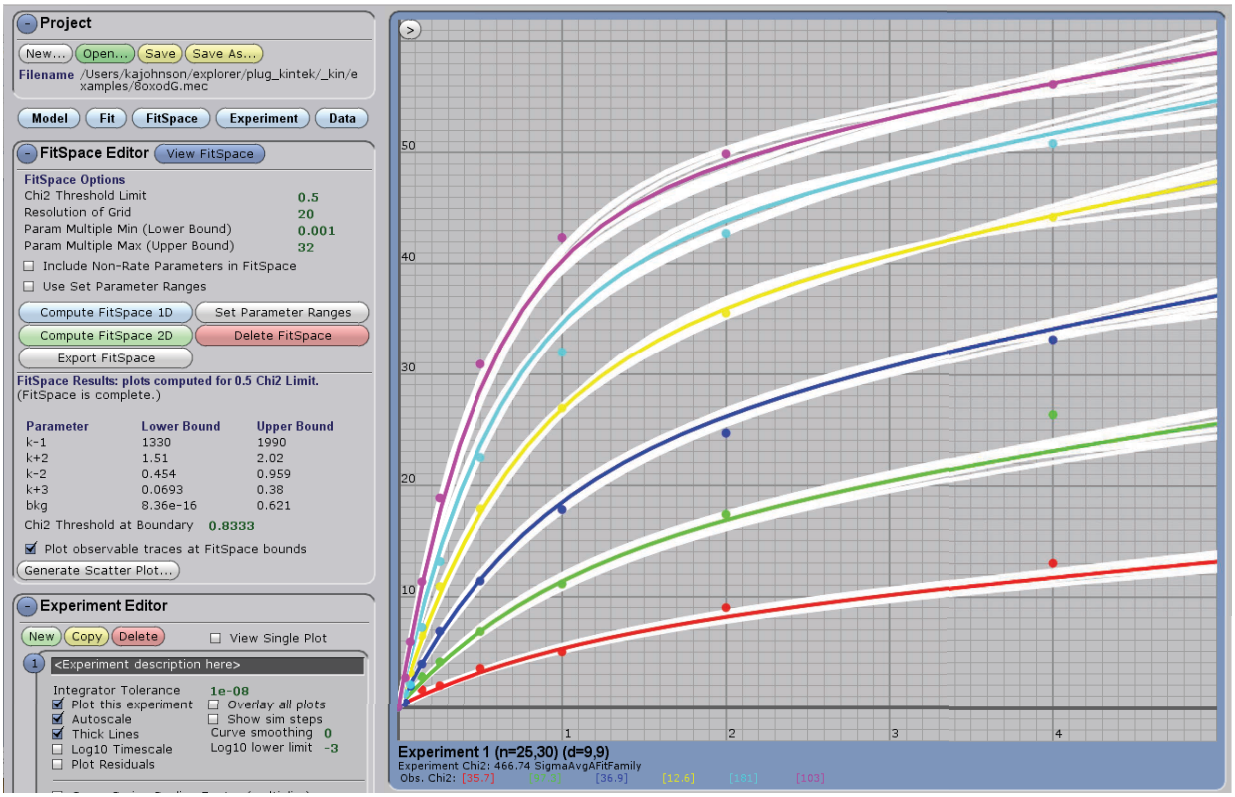


Figure 8.10 *Display of observable traces at parameter boundaries*. This figure shows the display all reasonable fits

to the data obtained for *8oxodG.mec* within a chi^2 threshold of 1/1.2.

Chapter 9. Statistical Analysis in Data Fitting

We present here a brief description of the statistics of sampling and data fitting. This brief tutorial provides a definition of terms such as sigma and chi^2 that are used throughout this manual. See *Numerical Recipes* (14) for more information on data fitting.

There are two important criteria to be used in evaluating whether a fit to data represents the best minimal model. First, we can evaluate goodness of fit: does the fit to the data adequately mimic the observations. Second, we must determine with the kinetic parameters are well constrained by the data. If the model fails to adequately account for the data (fails a goodness-of-fit test) then it may be that another area of parameter space needs to be explored or the model may be too simple. On the other had, if parameters are not well constrained, then the model may be overly complex. The ideal minimal model is achieved when all parameters are well constrained and a good fit is achieved.

9.1 Standard deviation

Data are subjected to random variations resulting from errors in measurement. The noise is an inherent property of the data and is quantified by the standard deviation of the measurement, sigma (σ). When multiple measurements are made under the same conditions, sigma can be quantified as the square root of the sum of the residuals squared divided by the number of measurements minus one as shown in *Equation 9.1*.

$$\text{residual}_j = x_j - \bar{x} \text{ where } \bar{x} \text{ is the mean: } \bar{x} = \frac{\sum_{j=1}^{N} x_j}{N}.$$

(9.1)

$$\text{Standard Deviation} = \sigma = \sqrt{\frac{\sum_{j=1}^{N-1} (x_j - \bar{x})^2}{N-1}}$$

$$\text{Variance} = \sigma^2$$

$$\text{Standard Error} = \sigma / \sqrt{N}$$

The standard deviation does not change when making multiple measurements, but the accuracy of determining the standard deviation and the mean improve as more data are included. The standard error of the mean is equal to sigma divided by the square root of the number of measurements.

The standard deviation defines the distribution of individual measurements and can be used to estimate the probability that an individual measurement will fall within a given range of the real value as illustrated in Figure 9.1. For example, a given measurement has a 95% probability of falling within 2σ of the real value, independent of the number of measurements, if there are no systematic errors. However, the accuracy with which we estimate the "real" value by calculation of the mean depends upon the number of measurements. Here with state that the mean has a 95% probability of being within $2\sigma / \sqrt{N}$ of the real value. As the number of measurements increases, the estimated accuracy improves as the square root of the number of measurements. This accuracy estimate is based upon assuming a normal (Gaussian) distribution.

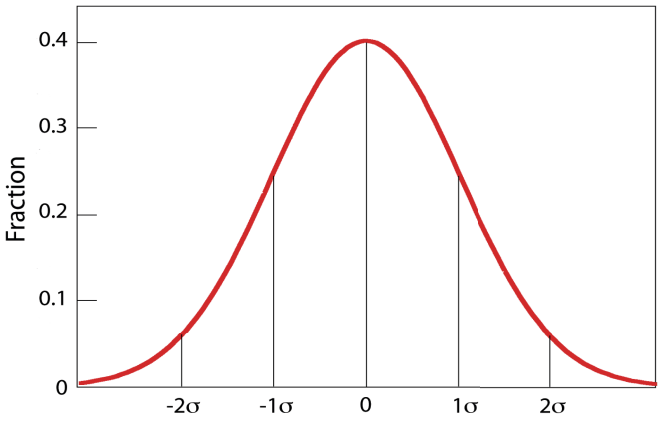


Figure 9.1 *Standard Deviation.* This figure shows the normal distribution of measurements about the "real" value and defines the percentage of points that fall within a given range based upon the standard deviation (σ). As defined by the area under the curve, 68.2% of the measurements fall within the range of ±1σ and 95% fall within the range of ±2σ.

It is generally assumed that the shape of the distribution is known and therefore, the center point is known with greater certainty than if the shape were not known. If all data are *independently and identically distributed* about the mean, the shape of the distribution follows a *Gaussian* distribution, also referred to as a *normal* distribution (Figure 9.1). Data fitting routines and estimations of errors in parameters assume a *normal* distribution, and this is one reason that error estimates are overly optimistic. However, as the number of data points increases, the impact of the assumed normal distribution becomes less important. Figure 9.2 shows the effect of increasing number of data points on the observed distribution of data. In each case, the sigma value used to generate the data was 0.2. The distribution is not changed by sampling more data, but the accuracy in defining the distribution and the midpoint (mean) are improved as the number of data points increases.

The effects of increasing number of data points on the estimated sigma value and the standard error are listed in Table 1. Synthetic data from the figure were analyzed to calculate the, mean, standard deviation and standard error, as summarized in the table below. This illustrates several important points. First, it is clear from the graph that 95% of the points fall within ±2σ (± 0.4). However, the standard error estimates, which are based upon the number of measurements and the standard deviation, indicate a much smaller range for confidence in determining the mean. Even with only 10 data points, the standard error is only 0.1 and with 1000 data points that is reduced to 0.007. With the lower numbers of data points, it should be obvious that the confidence limits on the mean depend upon the assumed shape of the distribution.

Table 1. *Statistical Analysis of Synthetic Data*

N	mean	$\sigma_{calculated}$	Std. Error
10	1.0716	0.331	0.1047
100	1.0102	0.222	0.0222
1000	1.0006	0.213	0.00675
10000	1.0026	0.199	0.002

This simple analysis of the standard error and mean derived by measurements of a single parameter has important implications for estimating several parameters based upon the simultaneous fitting of multiple experiments.

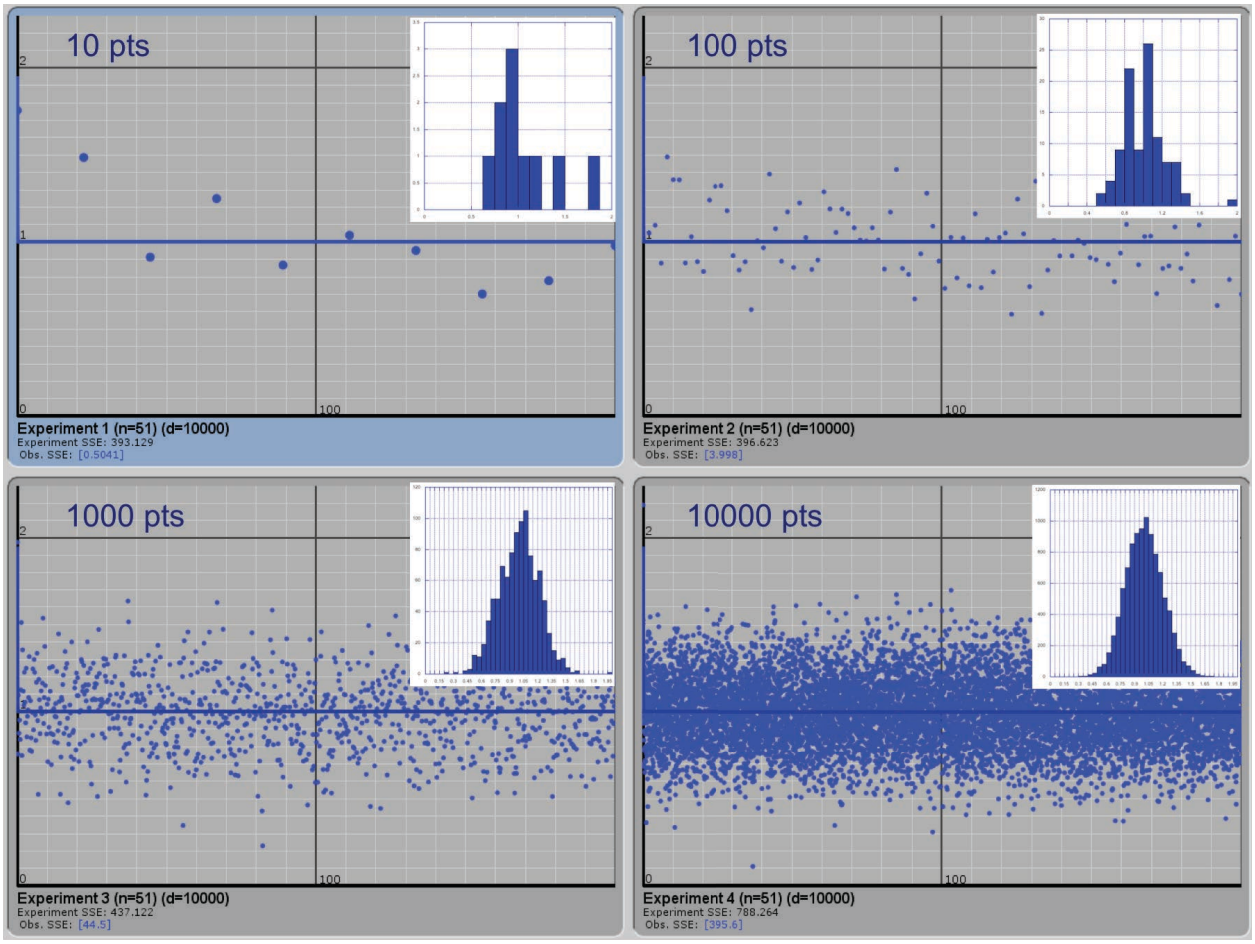


Figure 9.2 *Standard Deviation and the Number of Measurements.* This figure shows the effect of increasing the number of measurements on refining our ability to define of the shape of the distribution. Synthetic data were generated with the addition of random error with a sigma value of 0.2. The distribution about the mean is plotted in the inset of each graph.

In the process of fitting data, there are three important distinctions to make in evaluating the quality of the fit. First, a fit is said to have *converged* during regression analysis when a minimum chi^2 is achieved (defined below); that is, no further improvements in chi^2 can be achieved in successive iterations. However, the *goodness of fit* must be assessed by other means; namely, by evaluation of the chi^2 value with respect to the sigma values for the data, and by visual inspection of the fitted curve to look for systematic deviations of the data from the fitted curve. However, achieving a good fit does not imply that all of the variable parameters are *constrained* by the data. This is especially important when fitting several experiments simultaneously to derive multiple parameters. Thus, the final assessment of the quality of a given data fit must be based upon evaluating the extent to which parameter values are constrained by the data according to the proposed model. The background provided here is necessary to understand the principles governing data fitting and the tools that are used to find a good fit and then to evaluate whether the fit is unique.

9.2 Fitting by chi^2 minimization

Data fitting by regression analysis is based upon seeking a set of parameters that produce a minimum chi^2 (χ^2), defined as the sum of the residuals squared, normalized by dividing by the sigma value for each data point (*Equation 9.2*). That is, ideally, the standard deviation of each datum should be known. Unfortunately, in many cases sigma is not known, and we will deal with that case later.

(9.2) $$\chi^2 = \sum_{i=0}^{N-1}\left(\frac{y_i - y(x_i)}{\sigma_i}\right)^2$$ where $y(x_i) \equiv y(x_i \mid a_0 ... a_{M-1})$

y calculated at x_i with parameters a_0 to a_{M-1}

The calculated y values are a function of the x value and a set of parameters, defined as the unknowns in an equation during regression analysis. For example, below we describe the fit of kinetic data to a double exponential function with five variable parameters, two amplitudes (A_1 and A_2), two rates (λ_1 and λ_2) and an endpoint (C).

(9.3) $$Y = A_1 e^{-k_1 t} + A_2 e^{-k_2 t} + C$$

In fitting data based upon simulation, the unknown parameters are the rate constants and output scaling factors of a model that are used to calculate y values by numerical integration of rate equations. In either case, the process is the same. Regression analysis is an iterative process where estimated parameter values are updated by examination of the dependence of chi^2 on each parameter until a minimum in the chi^2 is achieved. If sigma values are known for each data point, then chi^2 is computed using the residuals normalized by their individual sigma values as described in Equation 9.2. When normalizing the residuals by the known sigma values, a good fit is achieved when the distribution of data about the fitted curved is the reflects the statistical distribution of errors in the data at each point. Visually, this is the case when the fitted curve goes right through the center of the distribution of the data as illustrated in Figure 9.3. A good fit is achieved when the minimum chi^2 is equal to the degrees of freedom (N-M).

(9.4) $$\chi^2 \approx N - M$$

N = no. of data points

M = no. of variable parameters

N - M = "degrees of freedom"

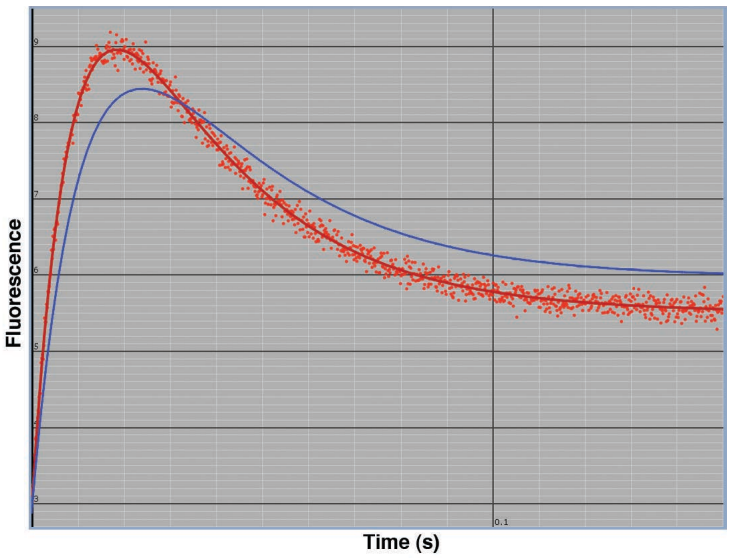


Figure 9.3. *Fitting by chi^2 minimization.* This figure represents the process of nonlinear regression analysis based upon seeking a minimum chi^2. Data consisting of 1000 points were generated with a standard deviation of σ = 0.1. In the process of fitting, chi^2 was decreased from 26800 (blue) to 962 (red). In fitting to a function with 5 variables, the "degrees of freedom" = 995, so this represents a good fit. One can also distinguish a good fit in seeing that the data are distributed uniformly above and below the fitted curve.

9.3 Data fitting without known sigma values

It is often the case that the sigma values are not known for the data. Under these circumstances the data are fit assuming that all measurements have the same standard deviation. One can then calculate an average sigma value for the data relative to the best fit achieved by finding the parameters giving the minimum chi^2 value.

(9.5)

$$\chi^2 = \sum_{i=0}^{N-1}\left(\frac{y_i - y(x_i)}{\sigma}\right)^2 \qquad y_{calc} = y(x_i \mid a_0 \ldots a_{M-1})$$

$$\chi^2 = \frac{N-1}{\sigma^2}\cdot\sum_{i=0}^{N-1}\left[y_i - y(x_i)\right]^2$$

We can then calculate the average sigma value:

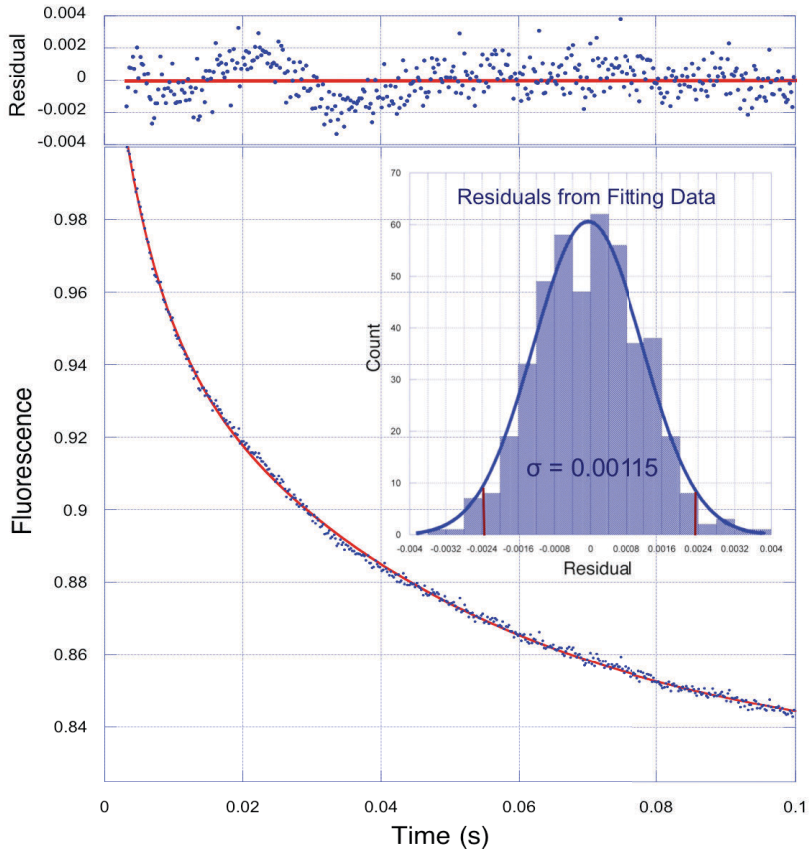
(9.6)

$$\sigma_{AVE} = \sqrt{\frac{\sum_{i=0}^{N-1}\left[y_i - y(x_i)\right]^2}{N-1}} \qquad \text{at the best fit valus of } a_0 \ldots a_{M-1}$$

Here, it should be obvious that the calculated average sigma value based upon the fit should be equal to the sigma value describing the distribution of the data about an ideal center point if a good fit is achieved. This is the equivalent of saying that the properly computed chi^2 (with residuals normalized by dividing by a known sigma value) will equal the degrees of freedom for a good fit.

Figure 9.4 shows stopped-flow fluorescence data with the best fit to a double exponential function. The upper figure shows the residuals and illustrates a uniform distribution of the data about the fitted curve except for a brief oscillation in the data at about 20-30 ms. In fact, this data set was chosen for analysis because of this unusual bump in the data to see how it affected the distribution of residuals. However, as shown in the inset, the distribution of residuals appears normal. The reason for this is apparent upon closer inspection of the residuals. The maximal excursion of the bump in the data is still within the amplitude of the noise in the data and is equally distributed on both sides of the data. Thus, the bump in the data may have slightly increased the net standard deviation, but the effect is small.

In this example, we calculate an average sigma value of 0.001. If the sigma value was known to be 0.001, then we could have computed chi^2 by dividing the residuals by the known sigma values and would have achieved a chi^2 equal to the degrees of freedom, and this would have provided an important criterion to evaluate goodness-of-fit. Thus, it is important to estimate the standard deviation when data are originally collected. For example, one method applicable kinetic data is to simply sample the noise after a reaction has gone to completion and use that calculate an average standard deviation. Alternatively, if multiple samples can be collected for each time point, then a mean and standard deviation for each datum can be used in fitting the time dependence.



$$Y = A_1 e^{-k_1 t} + A_2 e^{-k_2 t} + C$$

$$A_1 = 0.085 \pm 0.001$$
$$k_1 = 171 \pm 5\ s^{-1}$$
$$A_2 = 0.132 \pm 0.0007$$
$$k_2 = 21.6 \pm 0.4\ s^{-1}$$
$$C = 0.829 \pm 0.0006$$

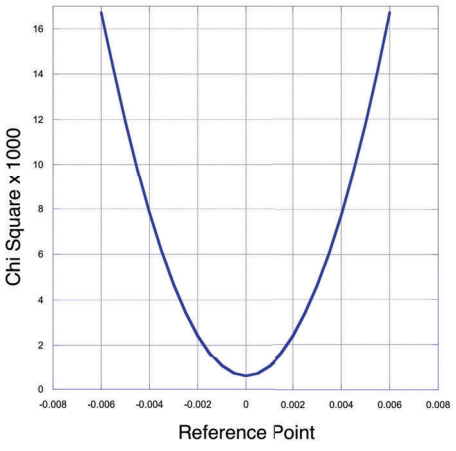
$$\chi^2 = 0.00059$$
$$R = 0.9995$$

$$N = 450$$
$$M = 5$$

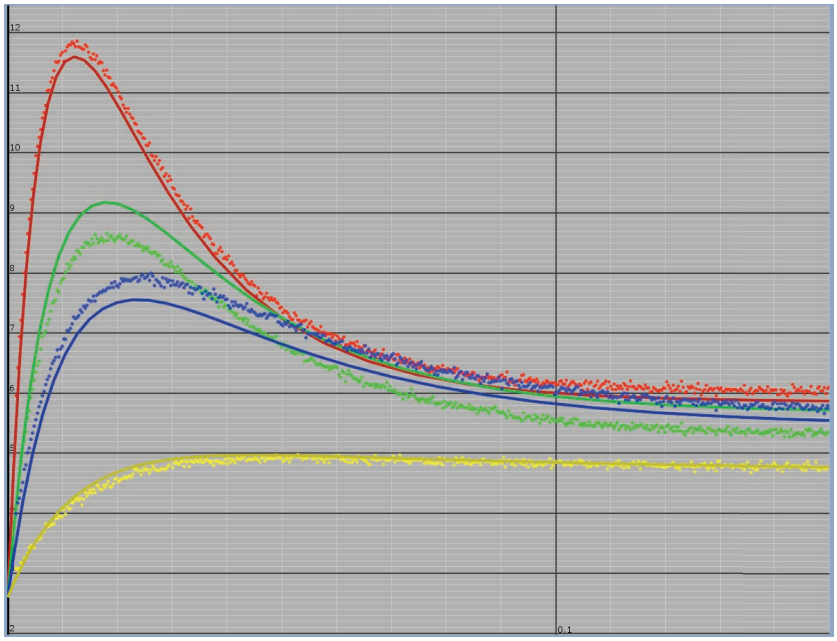
$$\sigma_{AVE} = \sqrt{\chi^2/(N-M)} = \sqrt{\frac{0.00059}{445}}$$
$$= 0.00114$$

Figure 9.4 *Best fit to stopped-flow data.* Stopped flow data are shown with the best fit to a double exponential function (see side panel) based upon nonlinear regression. The upper graph shows linear time-dependence of the residuals, while the inset displays the distribution of residuals. This data set was chosen because there is a bump in the trace, attempting to show the effects of systematic deviations from the fitted curve. *Side note:* These data were taken from the kinesin_mantATP_wt.mec file, which is a function of three kinetically significant steps. This example illustrates how exponential fits to data can be misleading and that rigorous fitting of data requires simulation based on a realistic model.

Figure 9.5 *Confidence contour.* This figure shows a plot of chi^2 as a function of one variable parameter (C) while holding the other 4 parameters fixed.



In the search for the best fit, one seeks by trial and error the point at which chi^2 is a minimum. The change in chi^2, as the value for the constant term (C) in Eq. 3 is varied systematically, is shown in Figure 9.5. The steep dependence of chi^2 leads to very small error estimates on the value for the parameter, C.



$\sigma_{AVE} = 0.281$

$\sigma_{DATA} = 0.04$

DoF = N-M = 1980-6 = 1974

Corrected $\chi^2 = 97{,}500$

χ^2/DoF = 49

Clearly NOT a good fit.

Figure 9.6 *Simultaneous fit of four curves with artificial lamp drift.* This figure shows synthetic data with a drift in the lamp between collecting each of the four traces. This drift precludes getting a good fit to the entire data set. The lack of a good fit is evident based upon several criteria, including the visual evaluation noting the deviations of the data from the fitted curves.

Figure 9.6 shows the fitting of synthetic data with artificial lamp drift between runs. This is NOT a good fit based upon several criteria:

σ_{AVE} >> σ_{DATA}; the sigma values derived in fitting are much greater than the sigma values used to generate this synthetic data (in this case the sigma values are known, but were not used in computing χ^2).

Corrected χ^2 >> N-M; calculation of the chi^2 values based upon normalization of the residuals by dividing by sigma gives a corrected chi^2 which is much greater than the number of data points minus the number of parameters, indicating a poor fit.

$$\chi^2_{FIT} = (N - M) \cdot \sigma^2_{AVE} \qquad \text{So normalized } \chi^2 = \chi^2_{FIT} \,/\, \sigma^2_{DATA}$$

Chi by eye; the fitted lines systematically deviate from the data.

It should be noted that in the absence of a known sigma value for the data, it is only the last criterion, chi by eye that is available for evaluating goodness of fit. Thus, it is important to provide an estimate of the standard deviation of individual measurements during data collection.

9.4 Evaluating Goodness of Fit

The figure below shows the results of a good global fit for two sets of data with widely varying signal intensity and sigma values. Here it is essential that the data fitting be based upon residuals normalized by known sigma values; otherwise, the data with the larger y-axis values would dominate the computation of chi^2 in finding the best fit. The data were fit properly based upon finding the minimum value of chi^2, computed by dividing the residuals by the known sigma values (often called a weighted least squares).

$$\chi^2 = \sum_{i=0}^{N-1}\left(\frac{y_i - y(x_i)}{\sigma_i}\right)^2 \qquad \text{where } y(x_i) \equiv y(x_i \mid a_0 \ldots a_{M-1})$$

y calculated at x_i with parameters a_0 to a_{M-1}

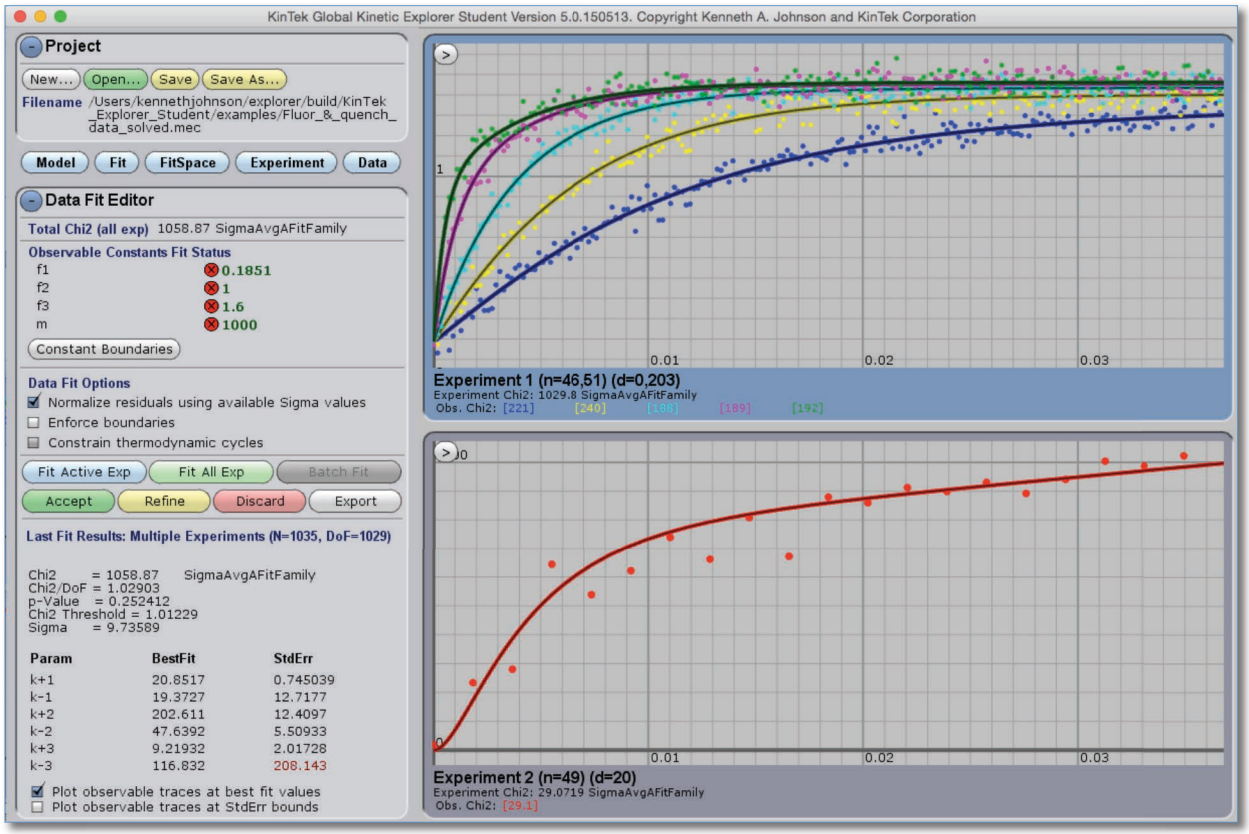


Figure 9.7 *Good Global Fit.* This shows an example of a good global fit obtain for experiments with widely varying signals intensities. Experiment 1 represents fluorescence data consisting of 5 traces of 203 points each with an average sigma value of 0.04. Experiment two represents rapid quench data consisting of 20 points with a sigma value of 60. Taken from example file: Fluor_&_quench_data.mec. Note also that individual chi^2 values (listed under each figure) are approximately equal to the number of data points in each trace.

a. Evaluating chi^2

A good fit was achieved as indicated by a chi^2 value approximately equal to the degrees of freedom. In this case there are a total of 1035 data points and 6 variable parameters leaving 1029 degrees of freedom. The ratio of chi^2/DoF is approximately 1, indicating a good fit.

Figure 9.8 shows the results of global fitting when the sigma values of the individual data sets are not known (or are ignored). Here chi^2 is computed as the sum square residuals:

$$\chi^2 = \sum_{i=0}^{N-1}\left(y_i - y(x_i)\right)^2 \qquad \text{where } y(x_i) \equiv y(x_i \mid a_0 \ldots a_{M-1})$$

y calculated at x_i with parameters a_0 to a_{M-1}

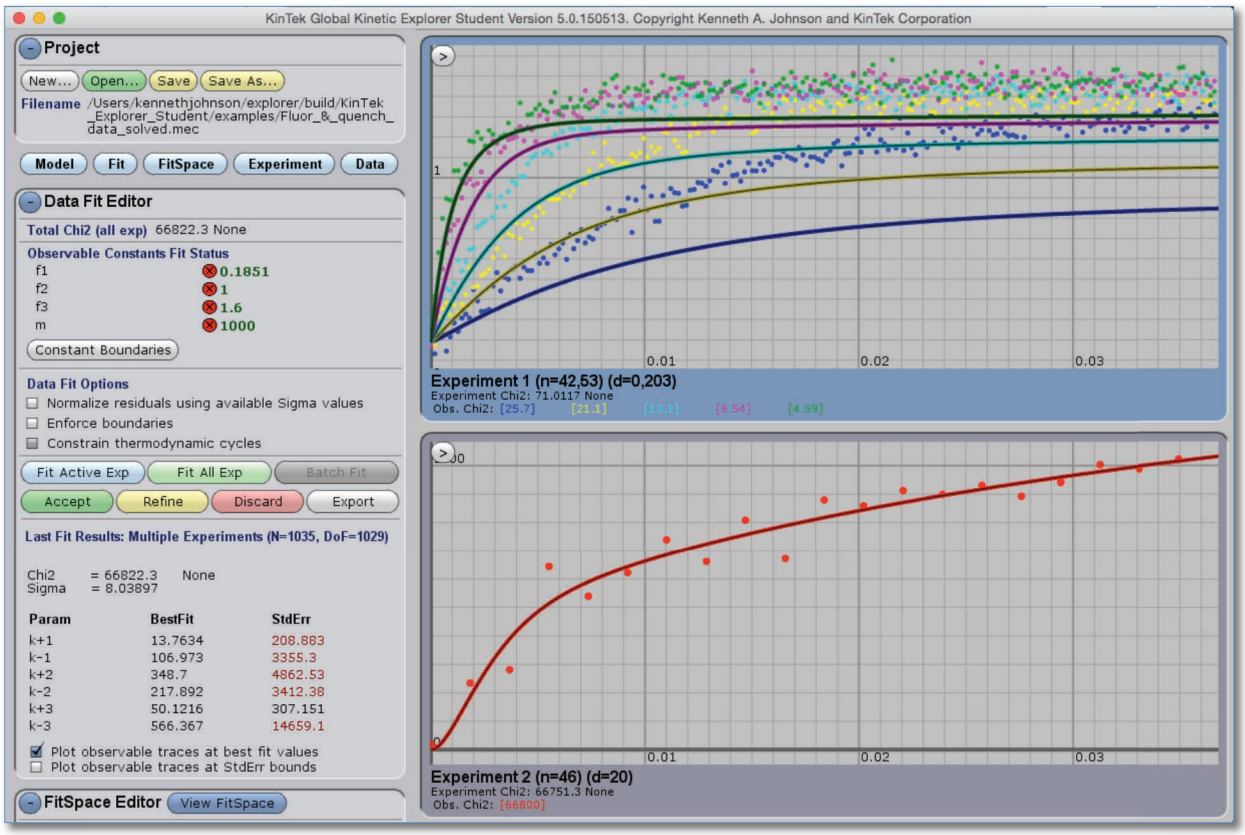


Figure 9.8 *Bad Global Fit.* This shows an example of a *bad* global fit obtain for experiments with widely varying signals intensities as described in Figure 9.7 (Good Global Fit). Note also that, in this case, individual chi^2 values (listed under each figure) are arbitrary and difficult to compare. Experiment 2 dominates the chi^2 minimization attempts so that Experiment 1 is largely ignored. Because the calculated chi^2 value is arbitrary, the calculation of chi^2/DoF is meaningless.

In this example, the quench-flow data dominate the fitting process so that the fit to the stopped flow data is rather poor. Without knowing the sigma values for the data, one is left with a chi^2 value that is arbitrary and only valuable as a relative term. The "best fit" is achieved at a minimum chi^2 value, but there is no independent metric to evaluate goodness of fit. One is left with "chi by eye." Moreover, it is difficult to fit globally sets of data from different experiments with markedly different signal intensities.

Three points should be obvious:

1. You can fit a single data set without knowing the standard deviation of the measurements, but evaluation of goodness of fit is entirely based on chi by eye. Visual evaluation (or plotting of residuals to aid your eye) is the only method to evaluate whether the fit mimics the data. This may be good enough and is an important part of data fitting in any case.

2. If you make the effort to estimate the standard deviation of your measurements, then you get the added criterion for evaluating goodness of fit. The calculated chi^2/DoF should approach unity for a good fit. More precisely, the p-value should also approach 1 for a good fit; values lower than 0.01 may indicate that factors other than normally distributed errors may contribute to the observed chi^2.

3. If you want to globally fit data sets from different experiments, with differing y-axis ranges and differing standard deviations, you must include the sigma values in your fitting so that fitting is based upon properly normalized chi^2 values.

b. Using the p-value

If you have entered estimates of the standard deviation (sigma) of your data, the program will compute chi^2/DoF and the p-value. The p-value statistic provides a more accurate estimation of whether the observed chi^2 value is within acceptable limits for what one might expect from random errors. That is, the p-value represents the probability that the null hypothesis, that the chi^2 value is not explained by normally distributed errors predicted by the observed sigma values, is false. The double negative gets confusing. We would like to say that the p-value represent the probability that the observed chi^2 is within the limits of random error expected from the known sigma values, and that is approximately correct, except for the caveat that statistics can never prove anything, but only serve to negate a null hypothesis as highly unlikely.

Given that the p-value provides a means to evaluate whether the observed chi^2 is within acceptable limits expected from the measured sigma values, we ask how valuable this parameter is in evaluating global fitting. The answer is that it is marginally useful in evaluating the global fits to multiple experiments. The major problem is that there are always terms that affect the errors in global fitting that go beyond the standard deviation of measurements within a given data set. For example, errors in actual concentrations of reagents from one experiment to another, or fluctuations in lamp intensity from one trace to another, can lead to increases in chi^2 that go beyond the expectations from a simple analysis based upon the standard deviation within a single data set. For example, the stopped-flow data from kinesin_mantADP.mec give the results shown below in Figure 9.9.

The best fit gave a value of chi^2/DoF = 1.32, which is not bad for a global fit of four traces. Moreover, the fit mimics the data quite well, as can be seen by visual inspection of the results. However, the p-value is 2.5e-23, indicating that the chi^2 cannot be explained by random fluctuations expected based upon the measured sigma value. The p-value extremely small because there are so many data points and the statistical test is based upon the assumption that more points replicate the measurement. The relatively minor fluctuations in signal, seen in the 15-30 ms range of the green and blue curves, increase chi^2 sufficiently to throw off the p-value statistic. That is, these fluctuations are beyond the expectations based upon sigma. Therefore the p-value shows that not all of the errors can be explained by statistical variation due to sampling error based upon the known sigma. This is apparent in the visual inspection of the data.

In the end, the p-value has limited utility in evaluating global fitting because in global fitting the p-value almost always indicates that there are errors in the data that are beyond the expectations based upon the measured sigma values. The answer to this dilemma is to provide a better estimate of sigma by taking into consideration all of the possible sources of error. Even then, the calculated p-value is only one of several criteria that one should use in deciding whether the fit is a good one. Another option would be to include an arbitrary function to model the lower frequency oscillations, but we have chosen not to implement such refinement because of the errors that could be introduced in masking aspects of the real data. In the end, there is no substitute for visual evaluation of the fitted curves with a keen eye and an understanding of the mechanism and the data.

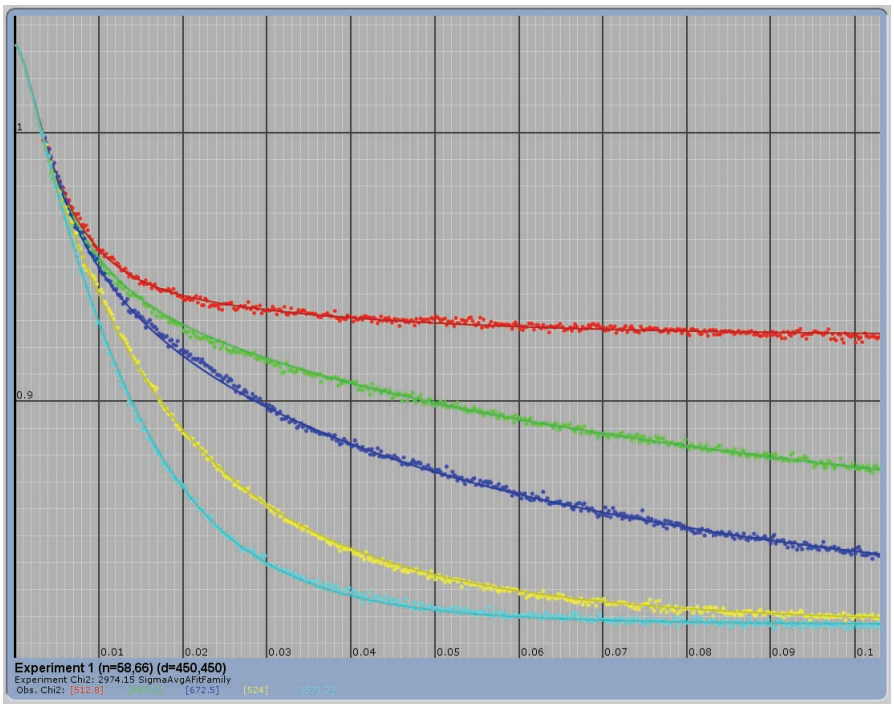


Figure 9.9 Global fitting of kinesin_mantADP_wt.mec.

9.5 Deriving and using estimated sigma values

If sigma values are not available, we provide a means to get estimates of standard deviation based upon fitting the data to an arbitrary analytic function. Our method is based upon the premise that one can find an arbitrary function to fit the individual traces that mimics the data sufficiently well to serve as a basis for computing the standard deviation. Here we rely upon visual evaluation to see that the fitted curve traverses the midpoint of the distribution of measurements so that it can be used to compute the standard deviation, as you would in computing a mean and standard deviation from multiple measurements. This estimated sigma value is then used in global fitting.

The process may sound circular, but it is not in the sense that global fitting of multiple traces to a model places much more stringent demands upon the process of data fitting than that of fitting an individual trace to an arbitrary function. It is easy to find a function to fit a given trace, but much more difficult to find a global fit to account for all data. Fitting each trace to an arbitrary function provides a standard for the best fit that might be attainable; and so, we use this to estimate the standard deviation. During global fitting, we can then properly calculate chi^2 by normalizing the residuals by dividing by the estimated sigma valued. Although this process may not carry with it the presumed legitimacy of basing the fit on an independently measured sigma values, it is better than having no estimate of standard deviation at all.

The process for deriving estimates of the standard deviation is described in *Section 6.4* on *Estimating Sigma Values*.

Chapter 10. SpectraFit SVD analysis

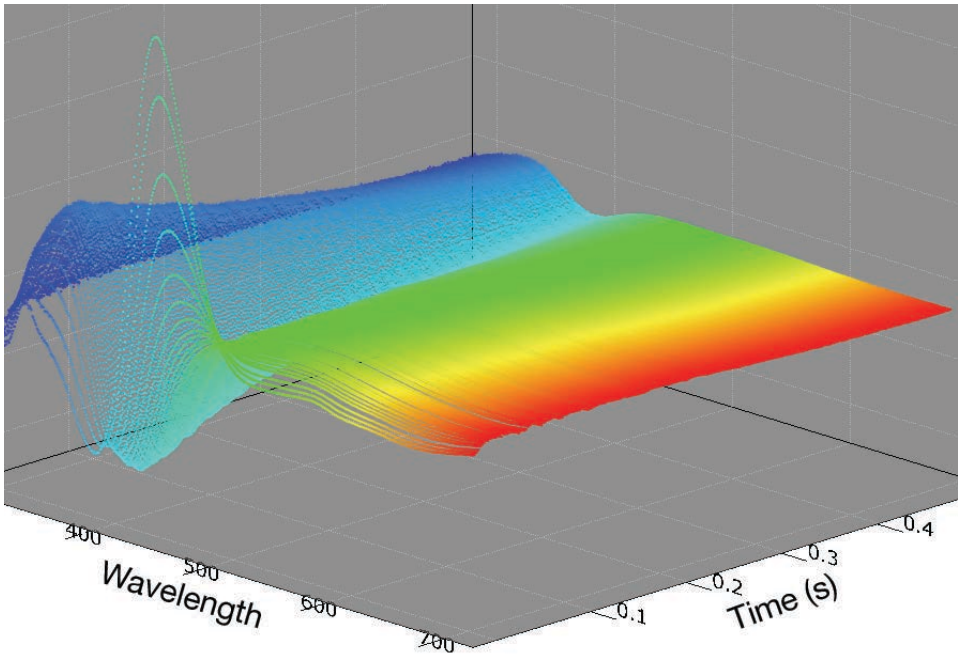


Figure 10.1 *Resolution of time-dependent spectra into components by Singular Value Decomposition*. This figure shows an example of initial time-resolved spectra that are used in this manual to illustrate the method for data fitting using singular value decomposition. These data were kindly provided Elizabeth Karnas, Jonathan Sessler and Shunichi Fukuzumi from (15)

Singular Value Decomposition (SVD) provides a method to de-convolute the highly redundant data contained in time-resolved spectra in order to obtained the time-dependence of individual species and define the spectrum of each unique species (16-18). Unlike the restrictions imposed in other implementations of SVD, *KinTek Explorer* allows the resulting time dependence to be fit to any model entered by the user and the fit is based upon numerical integration of the rate equations with no simplifying assumptions. Moreover, multiple experiments can be fit globally to a single model. These instructions assume a basic working knowledge of *KinTek Explorer* in terms of the methods for entering a model, defining an experiment, and fitting data.

Fitting spectra is accomplished by the executing the steps described below. Data can be imported and examined before or after entering a model. Perhaps the most difficult aspect in fitting the time-dependent spectra is to determine the number of species with distinct spectra. *KinTek Explorer* offers several tools to aid in making a judicious choice. Moreover, the data fitting process can be repeated using different models, until a final fit is obtained that satisfies the data and other available information.

Figure 10.2 shows an overview of the process. When data are imported, SVD is performed automatically. One then enters a model, defines an experiment and identifies the observable species. Fitting the data to the model resolves the individual spectra and time-dependence of each species. Finally, the model is used to reconstitute the original time-resolved spectra.

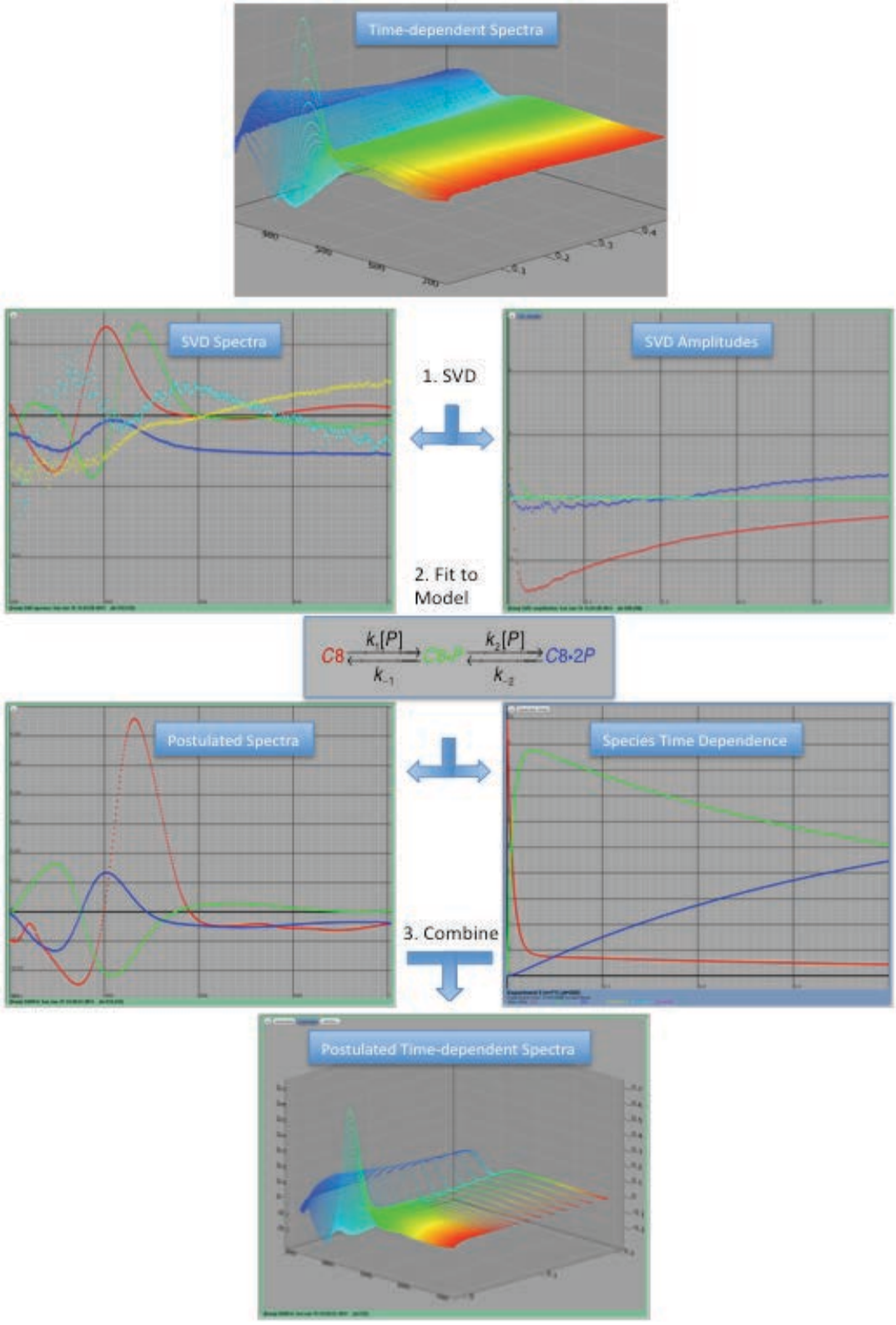


Figure 10.2 *Overview of SVD based fitting of time-resolved spectra*. **1**. SVD analysis produces Eigenvectors for the spectra and amplitudes, representing a mathematical de-convolution without constraints of physical reality; for example note the negative absorbance and negative concentrations! **2**. Fitting the SVD amplitudes to a model allows resolution of real spectra and the time dependence of species. **3**. The original time-resolved spectra are regenerated from the model.

10.1 Import Spectra

First, prepare your data file in *tab-delimited text* format. Data can be edited in MS Excel and then saved as tab-delimited text. As illustrated in Table 1, the first column of data should contain the wavelengths and each subsequent column should contain the absorbance values. The keyword *wave* or *wavelength* (or any word beginning with wave) identifies the data set as time-resolved spectra with the wavelength data in the first column. The remaining column headers, shown here with a pale green background, contain the time values for each column. The units of time do not matter, but must be consistent with the units used for rate constants. See *Section 4.2* for more information on choice of units for time, rate constants and concentrations of reactants. The decimal point must be specified by a "." not a comma. Time and wavelength should each be listed in ascending order, right to left and top to bottom, respectively.

We allow two data file formats: one with wavelengths listed by row (Table 10.1) and one with wavelengths listed by column (Table 10.2). Note that in this example, the spectrum at zero time was based upon taking an absorption spectrum before mixing. The spectrum of the starting material was then manually added to the table of time-resolved spectra.

Table 10.1. *Sample data file format (tab-delimited text) with wavelengths by row*

Wave/t	0.0000	0.0010	0.0022	0.0037	0.0052	0.0067	0.0082
323.8	0.0732	0.0753	0.0811	0.0860	0.0892	0.0942	0.0910
324.6	0.0748	0.0768	0.0800	0.0848	0.0903	0.0927	0.0944
325.5	0.0748	0.0744	0.0796	0.0864	0.0895	0.0918	0.0910
326.3	0.0763	0.0749	0.0799	0.0864	0.0893	0.0916	0.0908
327.1	0.0762	0.0748	0.0775	0.0837	0.0865	0.0887	0.0901
327.9	0.0798	0.0775	0.0780	0.0840	0.0868	0.0909	0.0901
328.8	0.0815	0.0754	0.0797	0.0836	0.0882	0.0902	0.0914

Table 10.2. *Sample data file format (tab-delimited text) with wavelengths by column*

Time/w	323.8	324.6	325.5	326.3	327.1	327.9	328.8
0	0.0732	0.0748	0.0748	0.0763	0.0762	0.0798	0.0815
0.001	0.0753	0.0768	0.0744	0.0749	0.0748	0.0775	0.0754
0.0022	0.0811	0.0800	0.0796	0.0799	0.0775	0.0780	0.0797
0.0037	0.0860	0.0848	0.0864	0.0864	0.0837	0.0840	0.0836
0.0052	0.0892	0.0903	0.0895	0.0893	0.0865	0.0868	0.0882
0.0067	0.0942	0.0927	0.0918	0.0916	0.0887	0.0909	0.0902
0.0082	0.0910	0.0944	0.0910	0.0908	0.0901	0.0901	0.0914

Click the Import Spectra button and select the desired file. Figure 10.3 shows the spectra as a function of time in a 3D representation after import.

You can click on the figure and rotate the display to view the data from various angles. Click on "Ortho" to toggle between the normal display to viewing from the "Time" axis or the "Waveln" (wavelength) axis.

Click the "Animate" button to see the spectra evolve over time. When in the animate mode, press CTRL-P to pause, then use the arrow keys to step through individual traces (*Left Arrow*: previous trace, *Right Arrow*: next trace, *Up Arrow*: first trace, *Down Arrow*: last trace.

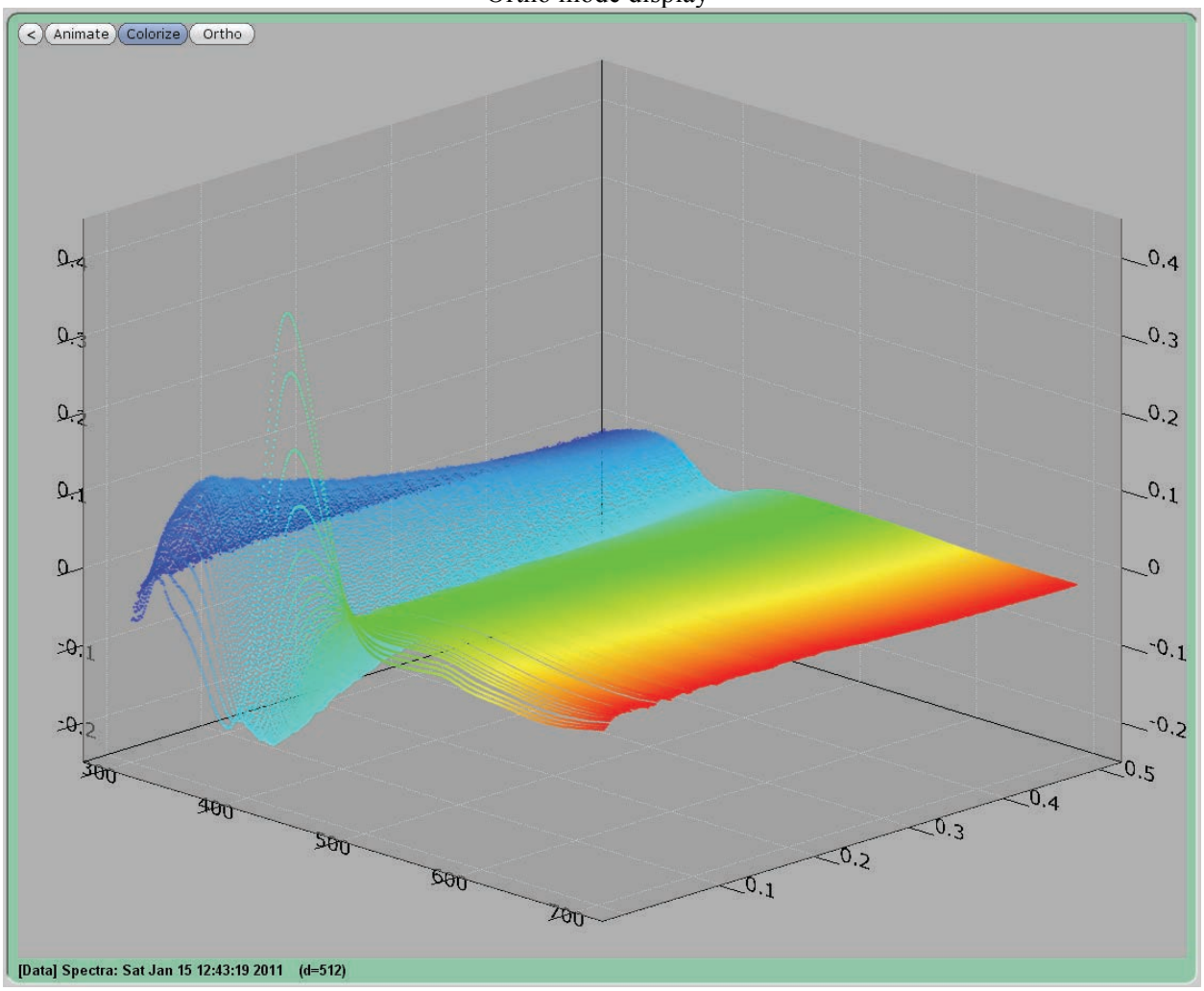
➔ next trace

➔ previous trace

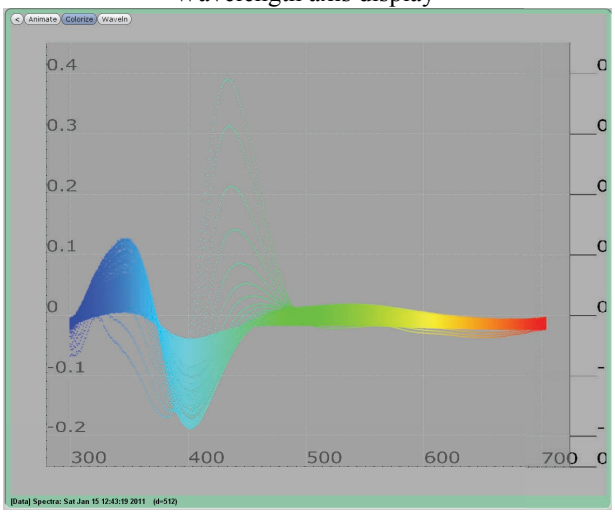
➔ first trace

🡻 last trace

Ortho mode display



Wavelength axis display



Time axis display

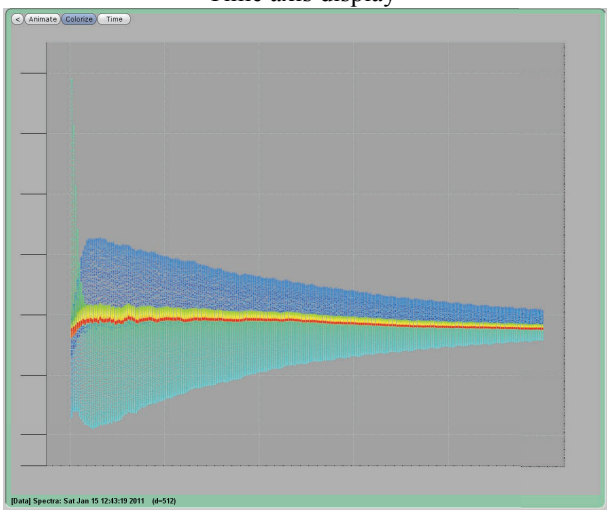


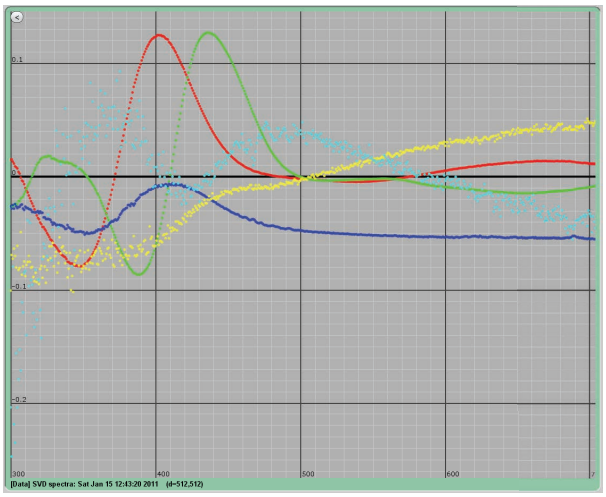
Figure 10.3 *Display of time-resolved spectra*. Click on the figure and rotate to view data from various angles. Toggle between different display angles using the *Ortho* - *Time* - *Waveln* buttons. Click on Animate to see the spectra evolve over time.

Singular value decomposition (SVD) is performed immediately when the data are imported. As shown in Figure 10.4, SVD provides a de-convolution of the data into two component matrices, one defining the spectra and the other defining the amplitudes as a function of time. This is not the final result, but rather an intermediate step on the way to providing a model that conforms to physical reality.

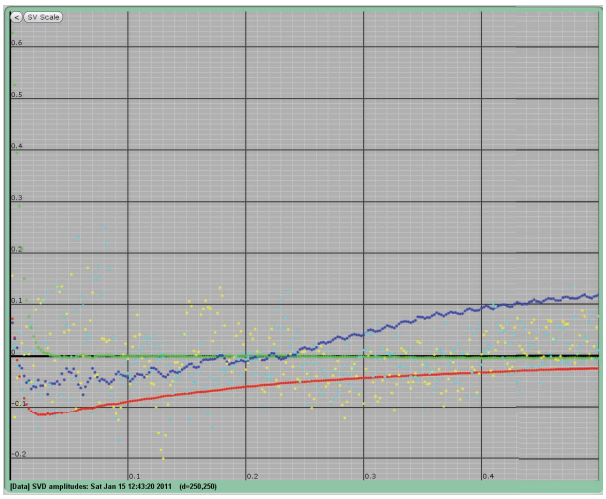
The figure labeled "SVD spectra" shows the time-independent SVD spectra for each species.

The figure labeled "SVD amplitudes" shows the time-dependence of changes in concentrations of species, scaled by some arbitrary factors, as explained below.

Note that in this example, there are 5 Eigenvectors displayed in 5 different colors, two of which represents noise in the data (cyan and yellow traces). The Eigenvectors containing mainly noise can be removed from the display as described below. In addition, as an aid to data fitting, the amplitude vectors can be scaled according to their associated singular values (magnitude of the Eigenvector) as shown in Figure 10.4C.



A. *SVD spectra*

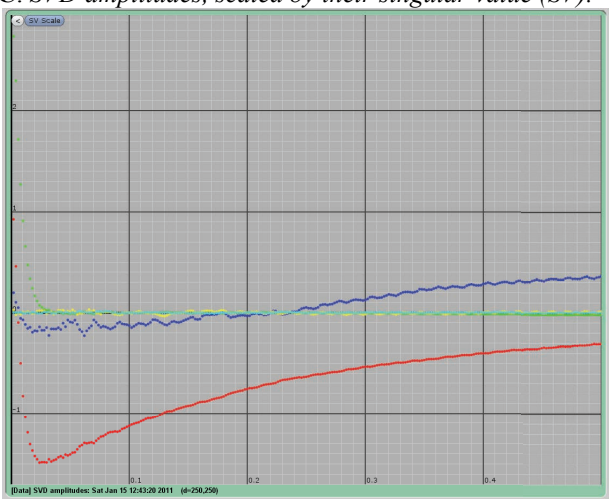


B. *SVD amplitudes*

Figure 10.4 *Results of SVD.* The SVD analysis resolves the spectra (A) and amplitude (B) Eigenvectors. Display of the SVD amplitude vectors can be scaled by their singular value to indicate their relative significance (C).

Note that there are negative absorbance values and negative amplitude terms. When multiplied together, the SVD spectra and SVD amplitudes recapitulate the original data; thus, the SVD provides a solution that is mathematically sound, but one that may not make sense physically! In addition, you will note some Eigenvectors represent noise in the data, but can be removed from further analysis as shown in Figure 10.4C.

C. *SVD amplitudes, scaled by their singular value (SV).*



10.2 Select significant Eigenvectors

In this example, there are 5 Eigenvectors, displayed in the figures and listed in the *Data Repository* in the order of the most to least significant. To complete the fitting process, you can first select which Eigenvectors you think represent distinct species and de-select those you believe to represent noise. You should be left with the number of Eigenvectors matching the number of species with distinct spectral characteristics. By repeating the fitting procedure, you can readily explore different models with different number of species.

The figure at the right shows the process of de-selecting the least significant (noise) Eigenvectors by unchecking the associated box under the *Data Repository*.

1. Click on the *SVD amplitudes* thumbnail (or the text description) to select that dataset. The selected dataset will be highlighted with a green border and the individual Eigenvectors within the set will be displayed. Make sure that "List Individual Traces" is checked at the top of the *Data Repository*.

2. Uncheck the boxes for the Eigenvectors that you believe represent noise in the data as shown at the right. Note that magnitude of each Eigenvector is listed, for example, SV = 12.7 for U1, while SV = 0.17 for U4, indicating a greater significance for the former. In addition the low autocorrelation coefficients (ACC) suggest that there is no signal associated with U4 and U5.

Note: This step is not necessary but is used only in aiding model development. When the data are dragged to the experiment, only the number of Eigenvectors corresponding to the number of species in the model is used.

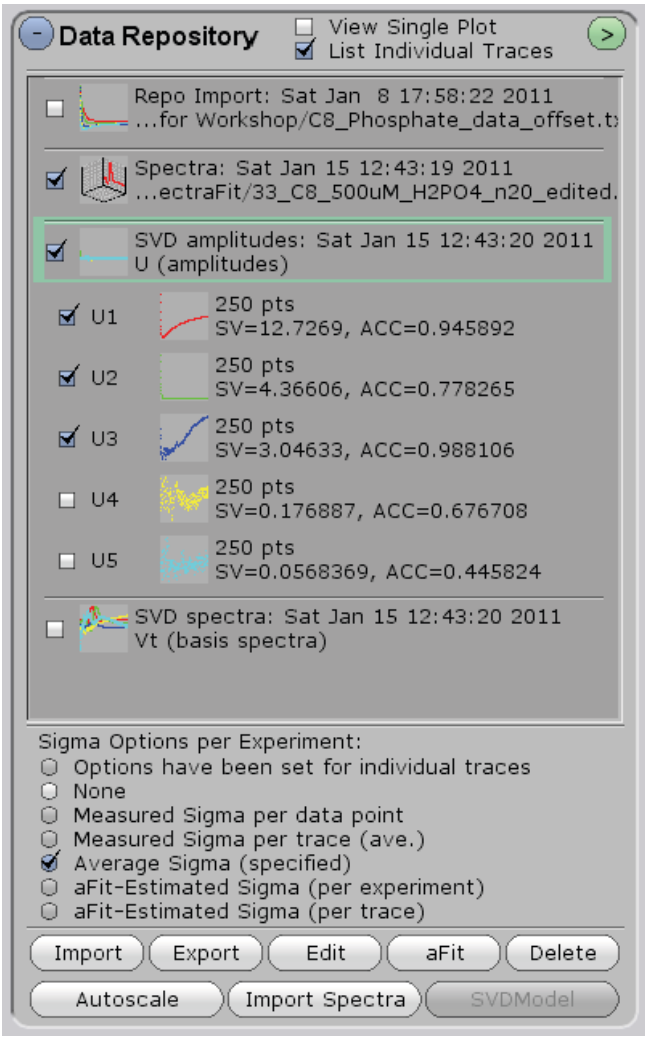


Figure 10.5 Selecting significant Eigenvectors

Note: In navigating the control panel, when the cursor is over the *Data Repository*, you can scroll data within the repository up and down. When you reach the upper or lower limit within the *Data Repository*, continued scrolling shifts the *Control Panel*.

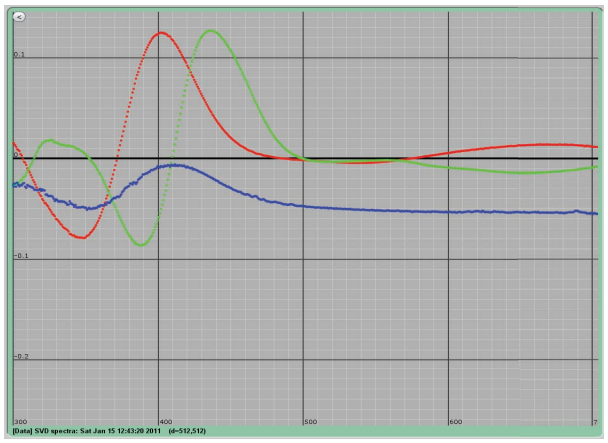


Figure 10.6A *SVD spectra without noise vectors*

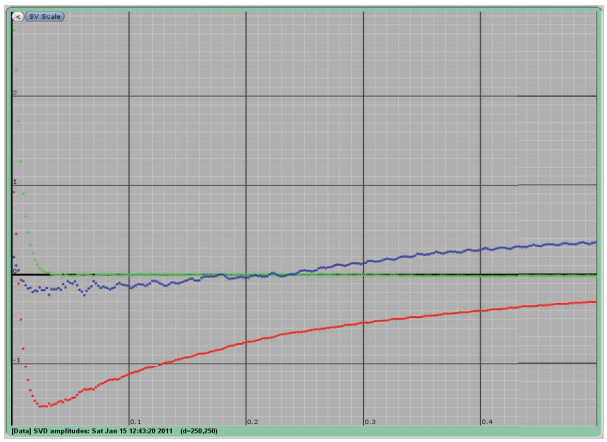


Figure 10.6B *SVD amplitudes without noise vectors*

10.3 Optional SVD model function

We also have implemented an algorithm that can be used in examining data to develop a model for the data fitting(19,20). This can be performed as an intermediate step in the process of fitting data.

SVDModel Under the *Data Repository*, first select the *SVD Spectra* that you want to model, and then click on the *SVD Model* button. Note that the *SVD Model* button is disabled until you select a given *SVD Spectra* data set. The program then searches for a solution to the time dependence of the observables while attempting to eliminate negative absorbance values and negative concentrations. For example, using the DHOD.mec data set, *SVD Model* produces the figures shown below. This is not helpful in the case of C8-P.mec because the data represent difference spectra, which can have negative values.

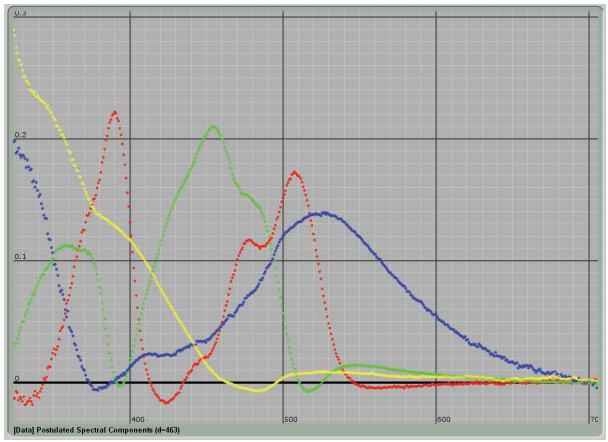


Figure 10.7A SVD-Model Spectra

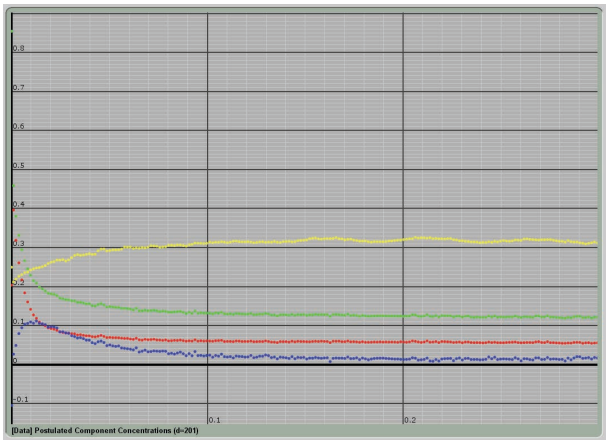


Figure 10.7B SVD-Model Amplitudes

This analysis can be useful in evaluating the data to determine the number of distinct species and getting a better estimation of their time-dependence. However, the result is not used in data fitting, other than to the extent that it can influence the development of a model and early estimates of rate constants. For example, note that the red and blue traces representing the most significant amplitude terms show positive signals decreasing with time, which is more realistic than the negative signals increasing with time as seen in the original SVD amplitudes (see the DHOD.mec example file). Data were kindly provided by Bruce Palfey (21).

10.4 Enter a model

As shown in Figure 10.6, the displays of the SVD Spectra and SVD Amplitudes indicate that there are three significant Eigenvectors, suggesting that there are three species with distinct absorption spectra. We next define a model and set up the experiment.

This experiment was performed with starting concentrations of 10 μM cyclo[8]pyrrole (C8) and 500 μM phosphate. C8 binds two phosphate anions, so we enter the model as:

$$C8 + P = C8P$$

$$C8P + P = C8P2$$

Next we create a new experiment, enter the concentrations of reactants and time of reaction.

Finally, based upon the three distinct Eigenvectors, we assign distinct species as:

S1: C8
S2: C8P
S3: C8P2

Note: By assigning these output signals, we are proposing that each species has a distinct absorption spectrum that we will resolve in subsequent fitting.

Figure 10.8 Experiment Editor setup.

Other functions on the Experiment Editor menu:

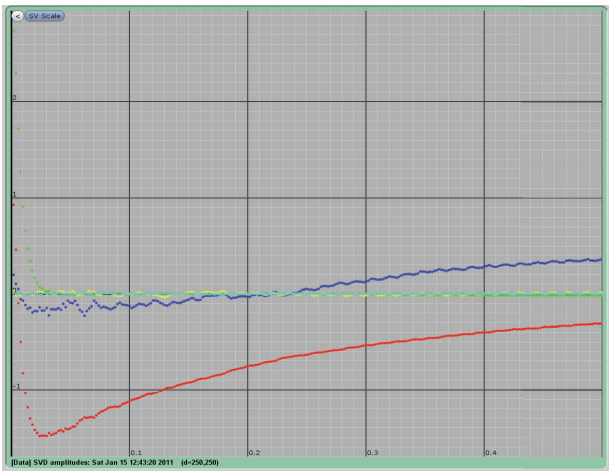
Clear Data — Remove data that have been linked to the current experiment. This is useful to start over in fitting SVD amplitudes to a different number of species.

Export Sim... — Export the results of the simulation to a text file

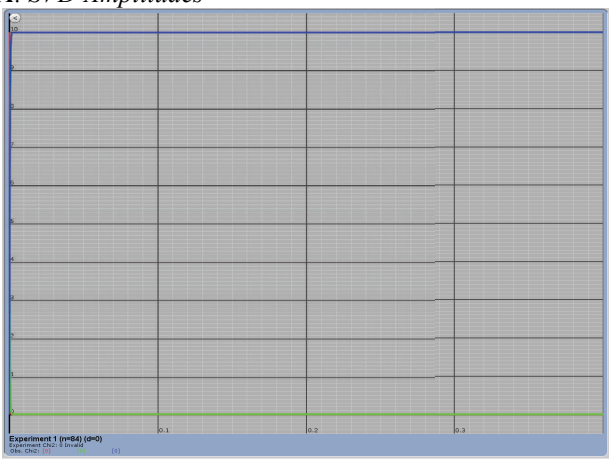
Gen Data... — Generate synthetic data to explore models and kinetics

The final solution depends upon fitting the SVD amplitudes (Figure 10.9A), with arbitrary scaling factors, to derive an appropriate model and set of rate constants. That is, the numerical integration of the rate equation produces the time dependence of individual species in absolute concentration (Figures 10.9B and C), while the SVD amplitude vectors consist of a sum of these species with arbitrary scaling factors. Therefore, we fit the SVD amplitude vectors to the model by deriving the scaling factors as we approach the best fit.

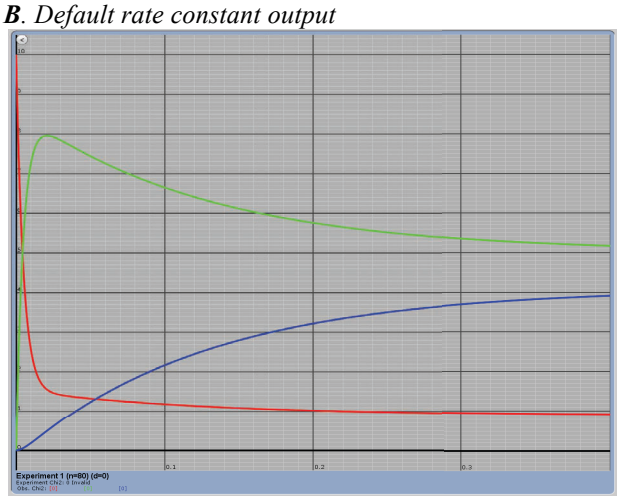
The default kinetic parameters (Figure 10.9B) do not reflect the time dependence seen with the SVD amplitudes. Therefore, we scroll the rate constants until we get a time dependence that is closer to mimicking the data (Figure 10.9C). Here we only attempt to mimic the time constants, not the amplitudes of the data because the amplitudes of the *SVD Amplitude Eigenvectors* are arbitrary. The revised estimates more closely reflect the time dependence of species, but we need the scaling factors, which are computed automatically when the data are link to the model in the next step.



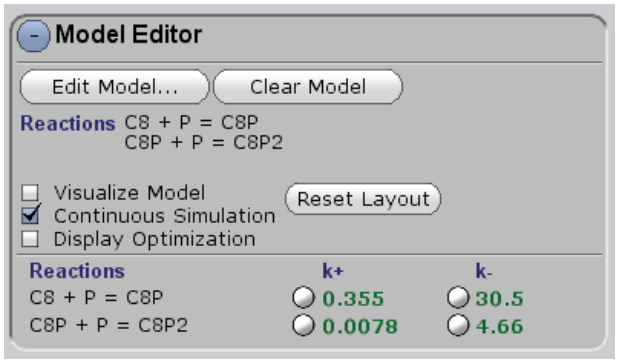
A. *SVD Amplitudes*



B. *Default rate constant output*



C. *Adjusted rates constants*



D. *Estimated rate constants*

Figure 10.9 *Adjusting the model to mimic the SVD Amplitudes*. Rate constants (D) are adjusted so that the time dependence of changes in concentration (B, C) occur on a time scale similar to the transients seen in the SVD Amplitude vectors (A).

Fitting the data is now reduced to finding a solution with appropriate rates of reaction and scaling factors to relate concentrations of species to the four SVD amplitude vectors. This process begins by dragging the data to the appropriate experiment. When data are dragged to an experiment, the scaling factors are computed automatically, a process that is facilitated by having the simulation more closely

mimic the data. Thus, adjusting the rate constants before associating the data with the experiment is helpful, although not essential.

10.5 Link data to an experiment then fit data to the model

Link the data with a given *Experiment* by dragging the *SVD Amplitudes* to the *Observables* defined for that experiment (Figure 10.10). You can click on and drag the set of three amplitude vectors all at once and drop them onto any of the three *Observable* definitions. The most significant amplitude vectors will be selected and assigned to the three outputs. Alternatively, you can select individual traces to link vectors with each species. Because the vectors represent a weighted sum of all four species, the order of assignment does not matter.

NOTE: *If you have known spectra, drag and drop the onto individual species before you drag and drop the amplitude vectors*.

When data are dragged to an *Experiment*, the program automatically computes the scaling factors for the best fit to the model. As shown in Figure 10.10, a new set of *SVD-Fit Observables* is defined and the simulation output is mapped onto the SVD Eigenvectors.

This analysis also produces a preliminary fit, but the time dependence has not yet been fit adequately and the spectra are not quite right yet. The final step is to fit the data directly to the model.

Note that on the *Experiment* display, you can toggle between the display of the *SVD Amplitude* vectors, the *Species* concentrations, or both. In this case, display of both is not very useful because the two are on very different scales.

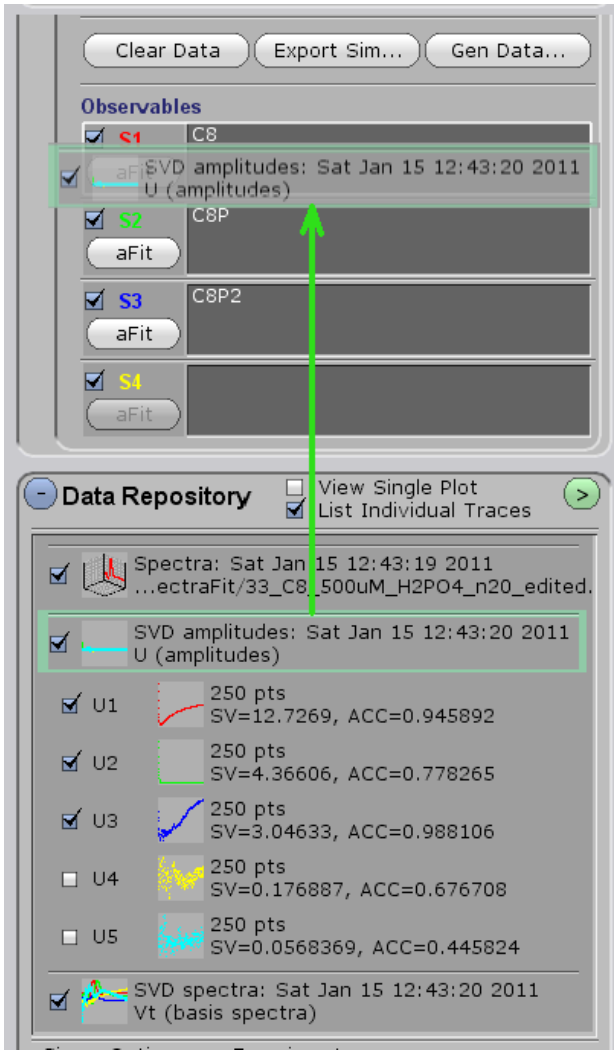
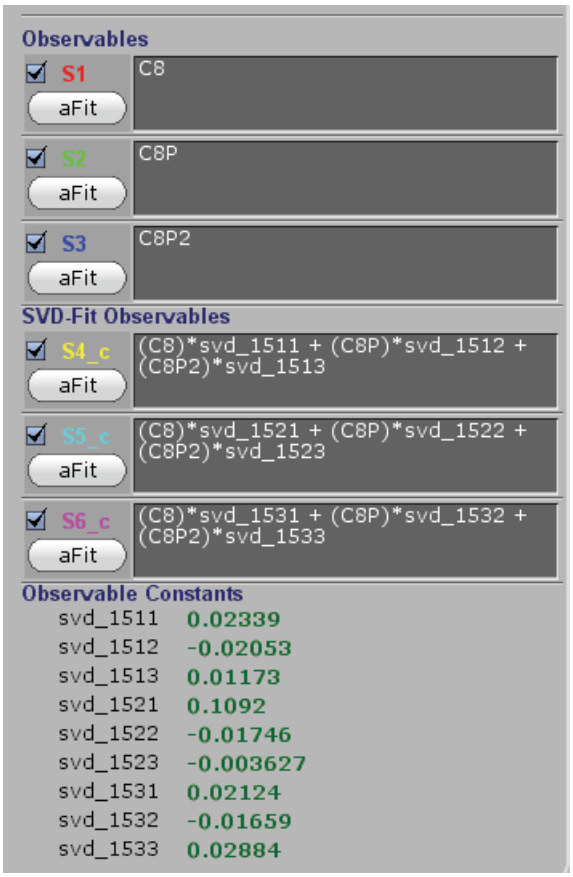


Figure 10.10 Dragging data onto an experiment

The *Experiment* graphic now shows the *SVD amplitude* vectors superimposed on the simulation, derived by solving for the scaling factors using fixed rate constants. In the next step, fitting is achieved by adjusting both rate constants and scaling factors. Setting the rate constants in the right range prior to dragging the data provides a more accurate estimation of the scaling factors.

Use the clear data function if you wish to edit the model or change the number or identity of observable species. After clearing the existing data, you can drag the data onto the newly defined experiment to fit the data to your new model.



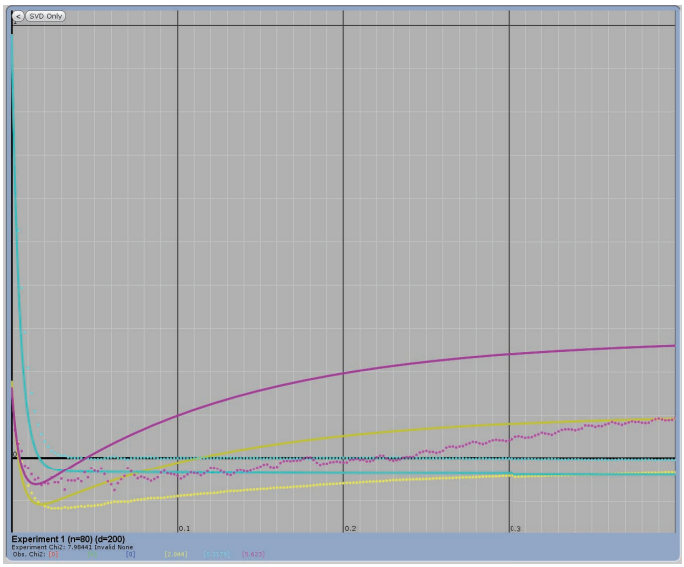


Figure 10.11 *SVD-Fit Observables. Left:* Newly defined output Observables with scaling factors. *Right:* SVD Amplitude vectors.

a. Fit data

Fit the data by pressing *Fit Active Experiment* (or *Fit All Experiments* if fitting multiple experiments globally) under the *Data Fit Editor*. A best fit is derived by nonlinear regression provide a set of rate constants and scaling factors to fit the *SVD Amplitudes*. Various factors and rate constants can be scrolled in searching parameter space to find an optimal set of starting values with which to begin the nonlinear regression and to explore the extent to which the parameters are constrained by the data. A set of arbitrary scaling factors, shown below, allow fitting of the amplitude Eigenvectors to achieve a final solution based on the concentrations of species and their spectra. Like other parameters, you can scroll the values of individual constants or lock them in seeking a final fit to the data.

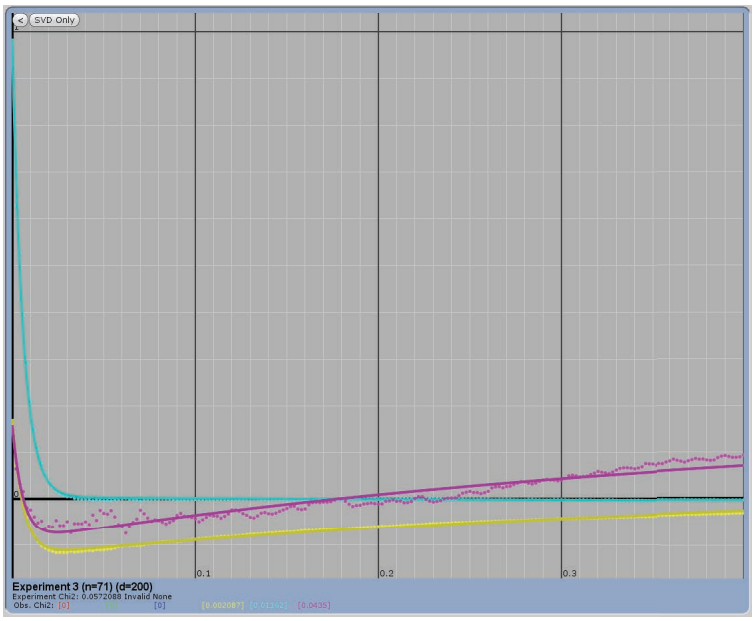


Figure 10.12 *Final fit of amplitude Eigenvectors*. The scaling factors and rate constants are adjusted to give the final fit of the SVD Amplitude vectors shown on the right.

The final fit to the data is shown in Figure 10.13 defining the time dependence of each species and their spectra. As with fitting single wavelength data, one must use caution to avoid over-interpretation and carefully assess the extent to which each parameter is constrained by the data.

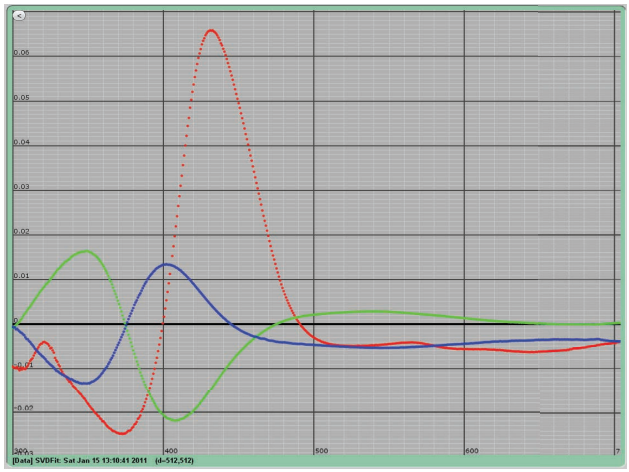


Figure 10.13A Final fit showing spectra of three species.

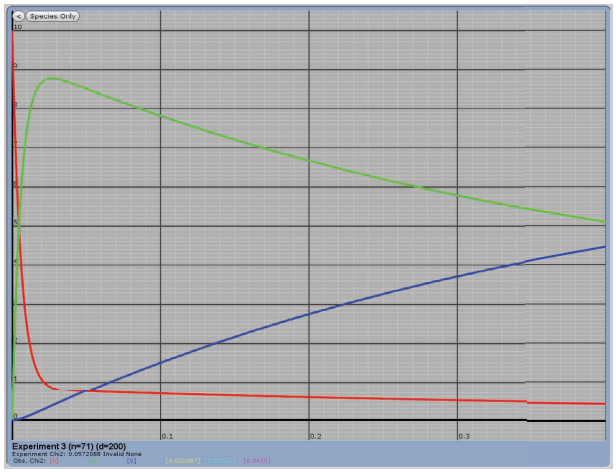


Figure 10.13B Time dependence of species

$$C8 \underset{k_{-1}}{\overset{k_1[P]}{\rightleftharpoons}} C8P \underset{k_{-2}}{\overset{k_2[P]}{\rightleftharpoons}} C8P2$$

Although the SVD analysis has enabled identification of the intermediates and defined their spectra, not all of the rates of reaction are well constrained from this single experiment at one substrate concentration. Therefore, we included in our global fit of the data an experiment performed at a single wavelength but over a range of concentrations. The global fitting of time-resolved spectra obtained at two concentrations (50 and 500 M) plus a full concentration series obtained at 425 nm, led to a complete fit of the model with all four rate constants defined and the spectra of the three distinct species as summarized in Figure 10.14.

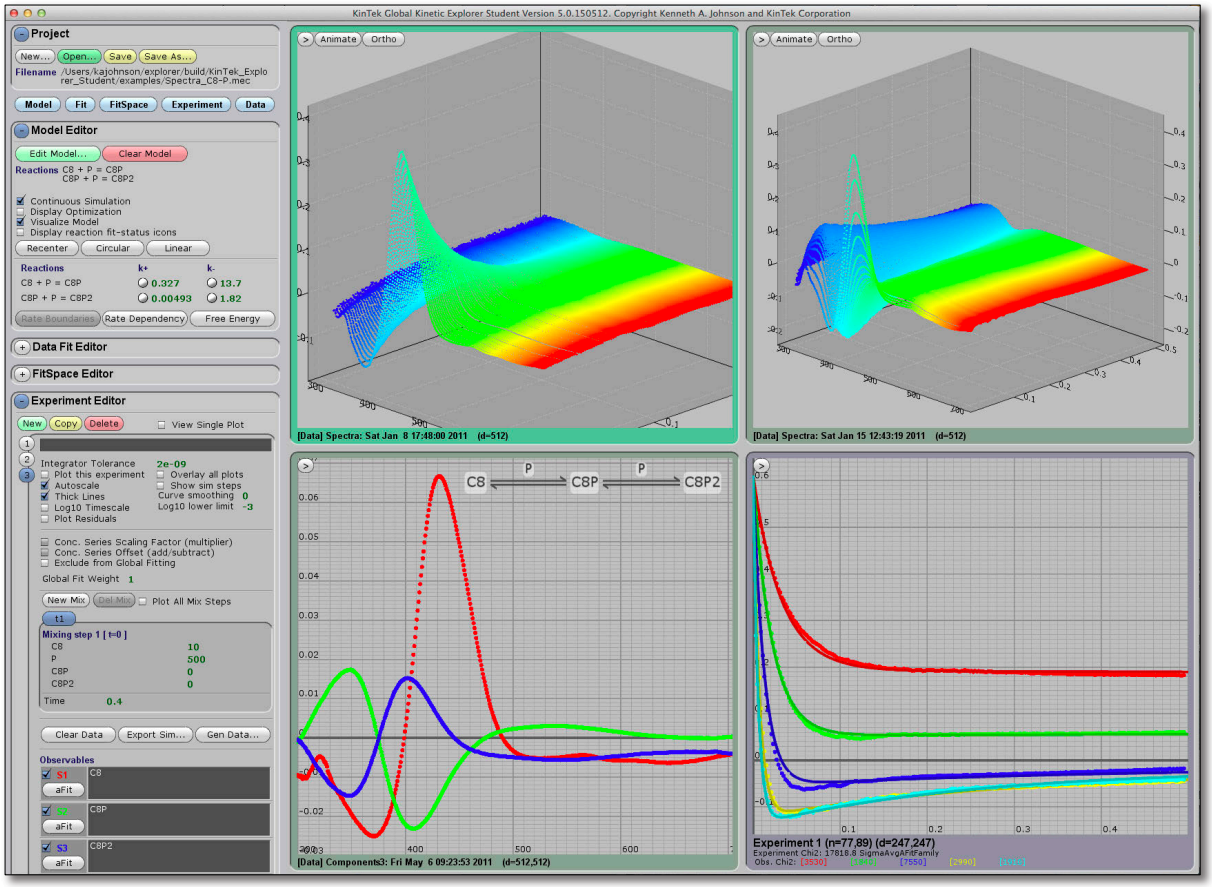


Figure 10.14 *Simultaneous fitting of time-resolved spectra and concentration series*. Three data sets were fit simultaneously to derive the spectra of three species and the four rate constants: *upper left:* spectra at 50 μM phosphate; *upper right:* spectra at 500 μM phosphate; *lower left*: postulated spectra of three species; *lower right:* fit to concentration series collected at 425 nm.

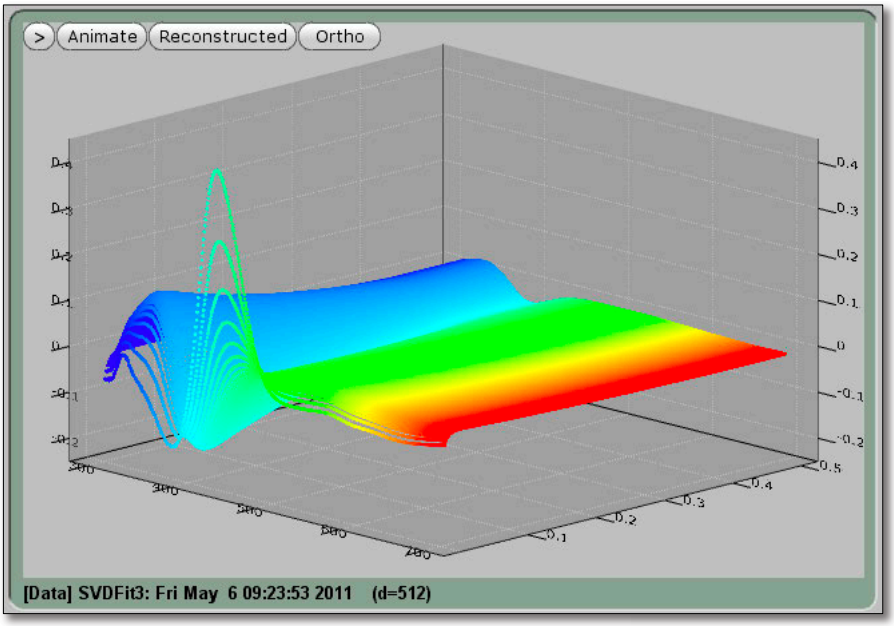


Figure 10.15 *Simulated time-resolved spectra.* The product of the spectra and concentration of species versus time is used to regenerate the time-resolved spectra. Click on *Residuals* to display the difference between the data and the

fitted spectra.

b. Output of simulation and fitting results

Export Under the *Data Repository* you can export any of the data sets as a text file. First click on the desired data set, and then press the *Export* button. Using this function, you can export original *Imported Spectra*, the *SVD Amplitudes*, the *SVD Spectra*, the *Postulated Spectral Components* and the *Reconstructed Spectra*. The exported text files can them be imported into your favorite graphics program for making figures for publication.

Export Sim... In order to export the fit to the amplitude vectors, use the *Export Sim* function under the *Experiment Editor*. The output of text files will depend on the display status: SVD only, Species only, or both. Three files will be generated (where *filename* is the name you give to the file):

filename.dat.txt file contains the data points
filename.sim.txt file contains the simulated curve (the number of points computed is entered)
filename.info.txt file contains indentifying information

Note: In the *documents* directory, we provide a Matlab script that will read the .sim.txt and .dat.txt files and generate a publication quality figure.

c. FitSpace Analysis

We can evaluate the extent to which each of the forward apparent rate constants is determined by the data using the *FitSpace* function to get the confidence contours shown in Figure 10.16.

This analysis shows that each of the rate constants except k_{-2} is well constrained by the data. The data place an upper limit on k_{-2}, but there is no lower limit. Notably, both of the second order rate constants for phosphate binding (k_1 and k_2) are well constrained by inclusion of the concentration dependence at a single wavelength. Fitting multiple experiments simultaneously is a powerful method in that one can design several experiments that complement one another in terms of the information content of the results.

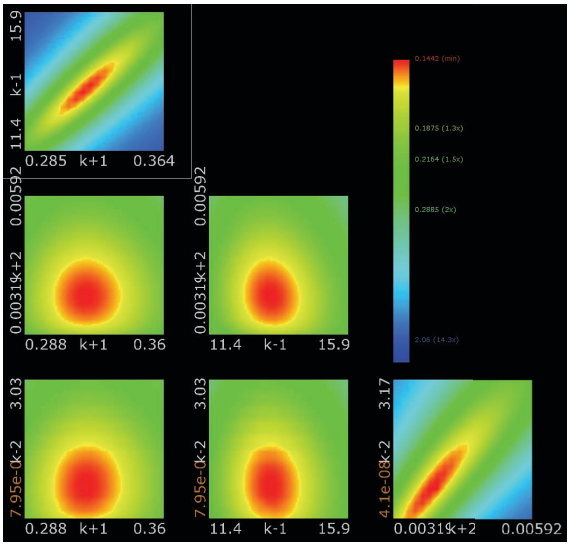


Figure 10.16 *FitSpace Analysis*

d. Fitting multiple time resolved spectra simultaneously

Two types of time resolved spectra are possible. *Absolute spectra* (such as Spectra_DHOD.mec) are collected with an appropriate reference containing only buffer and therefore the absorbance is a linear function of the concentration. Alternatively, it is often easier to measure the reference intensity with the observation cell containing products at the end of the reaction with the result being *difference spectra* (such as the Spectra_C8-P.mec). In this case the absorbance is not linearly related to concentration and it is much more difficult to relate the spectra directly to the concentrations of reactants to calculate absolute spectra.

Multiple *absolute spectra* can be fit simultaneously to account for differences in concentrations of reactants and to derive a single spectrum for the individual species that represents the best average fit to all of the data. The program will automatically detect when the spectra have the qualities expected for *absolute spectra* and then solve simultaneously the linear equations relating the concentrations of species and spectra. If the program recognizes the properties that signify *difference spectra*, such as negative amplitudes, this calculation is not performed.

10.6 Fitting Spectral Changes with an Equilibrium Titration: pH Titration

A pH titration is set up by using the notation: **[@ 0.00003 – 3500, 200]** to cover the range of hydrogen ion concentration from 0.00003 to 3500 μM (pH 10.5 to 2.5), with 200 points. To get points calculated on logarithmic scale to cover the large range in concentration, use the Log10 Timescale checkbox in the Experiment Editor. We no longer use the **exp** term for modeling a logarithmic concentration scale. The brackets specify that the proton concentration does not change during the reaction because the buffer maintains the constant concentration. Otherwise, the SVD analysis and data fitting are identical to the methods used in fitting time-resolve spectra.

Of course dealing with hydrogen ion concentrations is troublesome when we are all used to thinking in terms of pH. Therefore, we have instituted a new syntax to specify pH rather than $[H^+]$. The software will recognize the reserved term pH as part of a model definition and convert pH units to $[H^+]$ so you don't have to. Below are examples of pH titrations of a small molecule mimic of the fluorophore in green fluorescent protein (22). Figure 10.17 shows two files, the first configured for a titration as a function of hydrogen ion concentration (Figure 10.17a) and the second configured for a titration versus pH (Figure 10.17b). The table below provides a comparison of the two methods of entry.

Entry item	[H+] format	pH format
Mechanism	A + H = B B + H = C C + H = D	A + pH = B B + pH = C C + pH = D
Rate constants	Reactions / k+ / k- A + H = B — 10000 — 165000 B + H = C — 10000 — 3.17e+07 C + H = D — 10000 — 2.17e+09	Reactions / k+ / k- A + pH = B — pKa1 — 4.78 B + pH = C — pKa2 — 2.5 C + pH = D — pKa3 — 0.676
Concentrations	Mixing step 1 [t=0] A 1 H [@0.001-680000,200] B 0 C 0 D 0	Mixing step 1 [t=0] A 1 pH [@0.15-9,200] B 0 C 0 D 0

For the [H+] format, we work in μM units with an apparent binding rate of 10000 $\mu M^{-1}s^{-1}$, hold the concentrations fixed as denoted by the brackets and compute 200 points on an exponential scale using

the syntax [@0.001-680000,200]. That same calculation is performed in the pH mode using the syntax [@0.15-9,200]. Simply write your mechanism using pH rather than H and the program will the be set up to accept pH units.

Note: When using the pH syntax you must enter all concentrations in μM because the program uses the proton binding rate of 10000 $\mu M^{-1}s^{-1}$ to compute the proton dissociation rate from the pK_a.

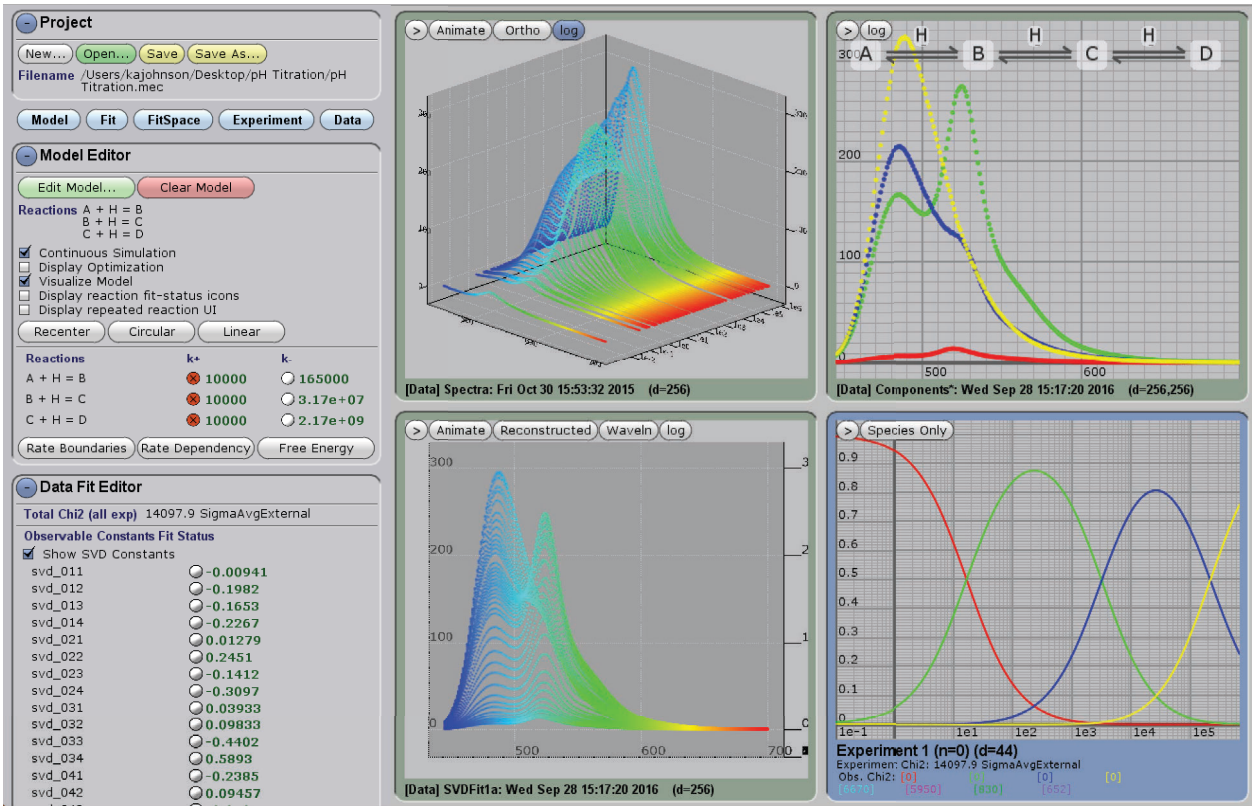


Figure 10.17a. *pH titration in [H+] units from example file "pH Titration_H+ conc.mec."*

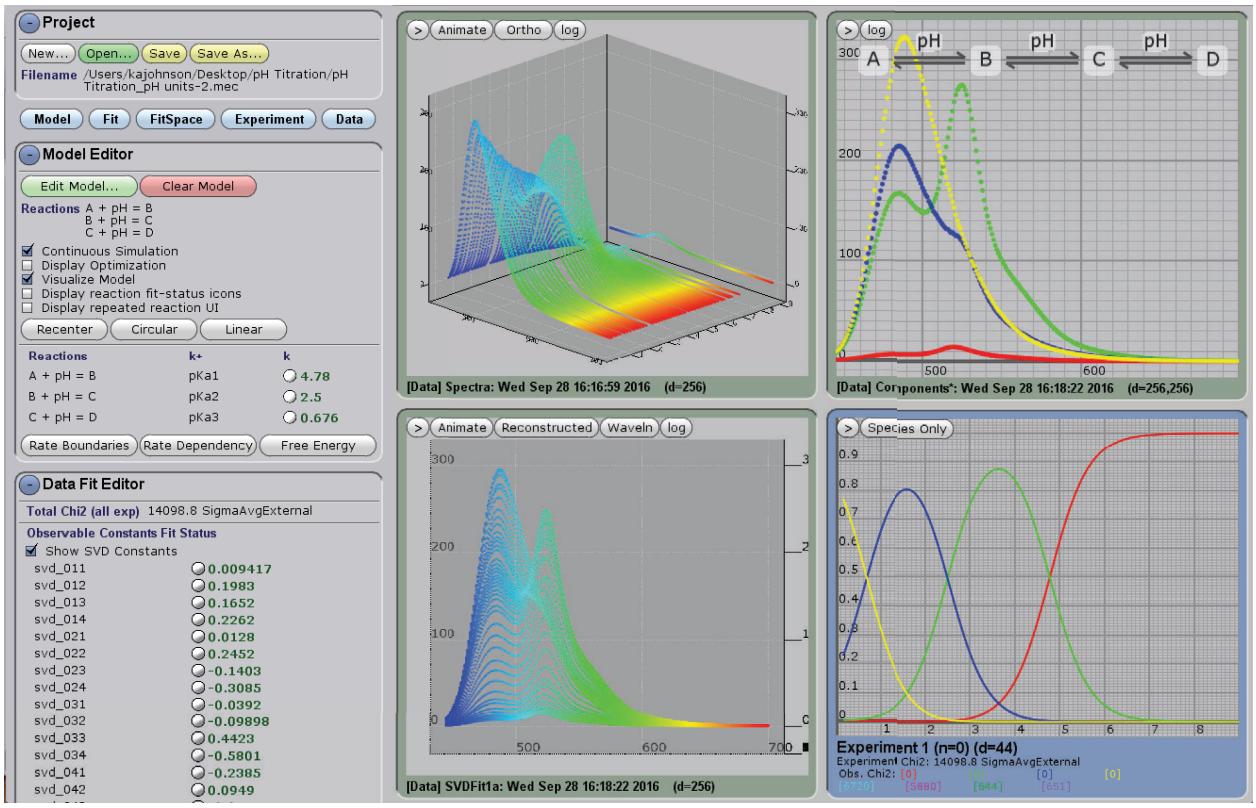


Figure 10.17b. *pH titration in pH units from example file "pH Titration_pH units.mec."*

10.7 CCD/Diode Array Test Reaction

The assay is based upon a paper "The oxidation of ascorbic acid by hexacyanoferrate(III) ion in acidic aqueous media" by L. J. A. Martins and J. B. da Costa (1988) J. Chemical Education 65,176-178. The reaction is pH dependent. We have adapted to the conditions to use pH 4.4 phosphate buffer to give a reaction rate of approximately 1 s^{-1}.

The reaction follows a complex pathway, and can be modeled in its entirety or simplified to single reaction step at a given pH.

$$H_2Asc \overset{K_1}{\rightleftharpoons} H^+ + HAsc^-$$

$$[Fe(CN)_6]^{3-} + HAsc^- \xrightarrow{k_2} [Fe(CN)_6]^{4-} + HAsc^{\bullet}$$

$$HAsc^{\bullet} \overset{Kr_2}{\rightleftharpoons} H + Asc^{\bullet -}$$

$$[Fe(CN)_6]^{3-} + Asc^{\bullet -} \xrightarrow{fast} [Fe(CN)_6]^{4-} + Asc$$

Stock Solutions:

0.1 M KH_2PO_4, pH 4.4. (F.W. = 136 g/mol). Dissolve 13.6 g in 1 L of distilled water. The pH will be approximately 4.4 and requires no adjustment.

0.01 M $K_3[Fe(CN)_6]$ in 0.1 M KH_2PO_4, pH 4.4. (F.W. = 329.26) Dissolve 0.329 g in 100 mL of phosphate buffer. Freshly prepared on the day of the experiment.

0.01 M Ascorbic Acid in 0.1 M KH_2PO_4, pH 4.4. (F.W. = 176.13) Dissolve 0.176 g in 100 mL of phosphate buffer.

Reaction:

Prepare a working solution of 0.001 M $K_3[Fe(CN)_6]$ in 0.1 M KH_2PO_4, pH 4.4 by tenfold dilution of the stock solution prepared above. Use the 0.01 M ascorbic acid solution undiluted.

In the stopped-flow, mix 0.01 M Ascorbic acid with 0.001 M $K_3[Fe(CN)_6]$ in 0.1 M KH_2PO_4, pH 4.4. This gives final concentrations of 0.0005 M $K_3[Fe(CN)_6]$ and 0.005 M Ascorbic acid. A change in absorbance at 418 nm from approximately 0.5 to 0.25 absorbance units should occur at a rate of 1 s^{-1}. The extinction coefficient of $K_3[Fe(CN)_6]$ is 1012 $M^{-1}cm^{-1}$, predicting A418 = 0.506 for a 0.0005 M solution.

Spectra recorded as a function of time can be resolved by SVD as shown in Figure 10.17.

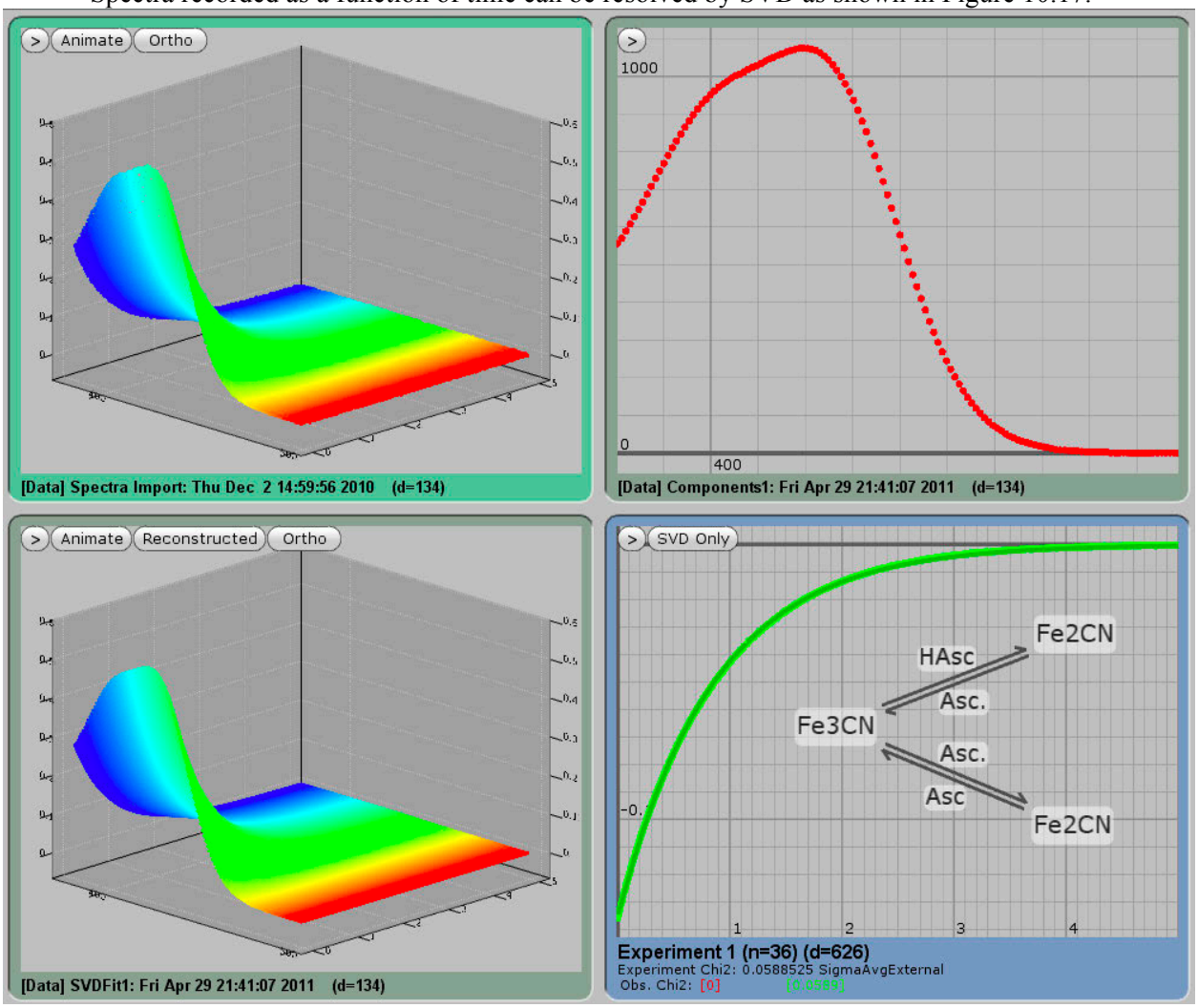


Figure 10.18 *Reduction of ferricyanide*. This figure shows the time resolved spectra resulting from the reaction with ascorbic acid at pH 4.4. *Upper left*: Original data. *Upper Right*: Resolved spectrum. *Lower Left*: Predicted time dependent spectra. *Lower Right*: Fit of amplitude vector to the model. Data were collected by Mike Repasky from KinTek Corporation.

Note: Using KinTek Explorer, we model the actual two step reaction involving the free radical, Asc•

Fe3CN + HAsc = Fe2CN + Asc•

Fe3CN + Asc• = Fe2CN + Asc

Other programs relying upon the simplified A+B→C reaction give the wrong stoichiometry for reaction of $Fe^{+3}(CN)_6$ with ascorbate.

Chapter 11. Output of Publication Quality Figures

Starting with Version 6.0 you can output publication quality .eps and .png files using the Licensed version of the software. This applies to the output of simulations and data fitting, analytical data fitting including concentration dependence of rates and amplitudes, FitSpace 1D and 2D figures, and SVD analysis of time-resolved spectra, including 3D figures. The graphic output is tailored to the requirements of each data type and reflects the current state of the on-screen display. Therefore, controls to output publication quality figures are located in the appropriate sub-menus of the software:

Graphic output	**Location of controls**
Simulation-based data fitting results	Experiment Editor
Free Energy Profile	Model Editor
Analytic equation-based data fitting	Experiment Editor *aFit*
FitSpace confidence contours	FitSpace Editor
SVD analysis	Data Repository

11.1 Export of simulation results

There are three methods to output the results of simulations. First, you can either simply do a screen capture of the simulation results. If you have the Licensed version, you can export the simulations as x,y data that can then be imported into a graphics program of your choice to make publication quality figures. You can also directly generate publication quality figures; this requires the Licensed version and installation of a Python graphics package. See installation instructions.

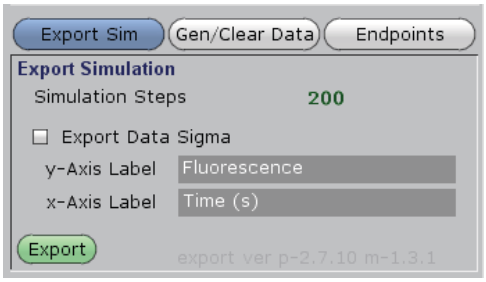
When exporting the results of a simulation you get both a set of publication quality graphs in EPS and PNG format, and a set of text files that you can use to generate your own figures using a graphics program of your choice.

The output of a publication quality figure is a new feature starting with Version 6.0. This feature requires that Python graphics software be installed on your computer. Under Mac OSX, you should not have to do anything since Python is already installed. However, on older Mac computers it may not be installed, in which case you need to follow the directions below. For Windows, you need to install Anaconda Python 2.7. You can either search for it, or go to the following URL: https://www.continuum.io/downloads. Then follow instructions for installation. DO NOT install Python 3.5, which contains changes incompatible with version 2.7; the industry standard is version 2.7.

Click *Export Sim.*. under the *Experiment Editor* to output the results of a simulation.

If Python is installed, you will get the menu at the right after clicking on “Export Sim”.

Enter the axes labels in the menu shown, then click *Export* and select a directory for saving the image files.

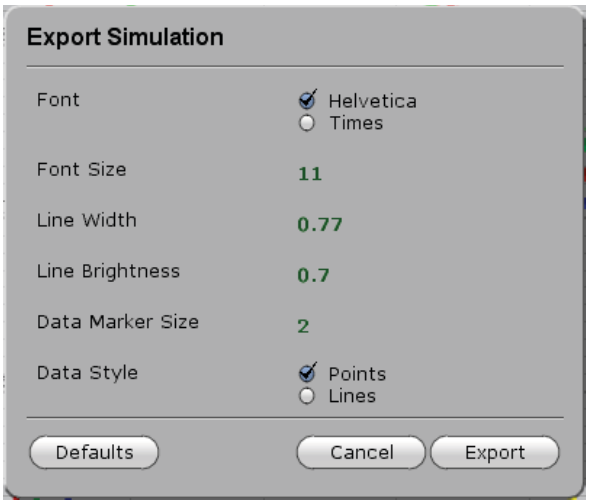


Note: The output is governed by the computer display. If the data are shown on a logarithmic timescale, then the hardcopy figure will also be on a log scale.

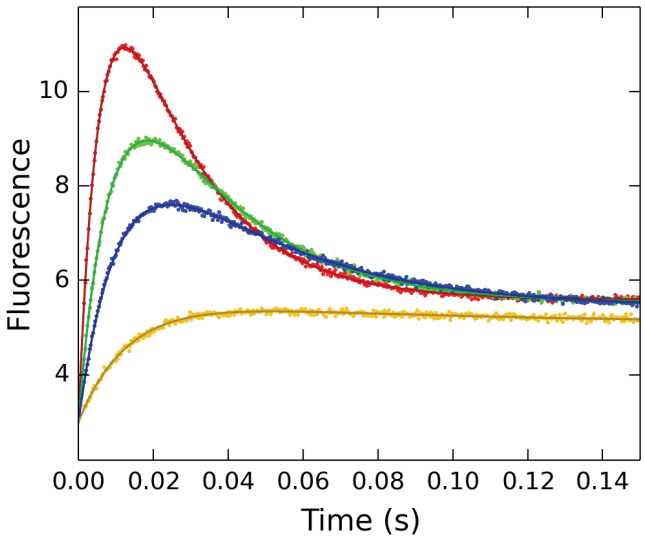
The menu shown here then appears to allow control of the simulation output. The default values will give you a plot that conforms to the standards for publication.

We only allow two fonts, Helvetica and Times Roman, since these are typically allowed by journals. Besides, does one really need 120 different font choices?

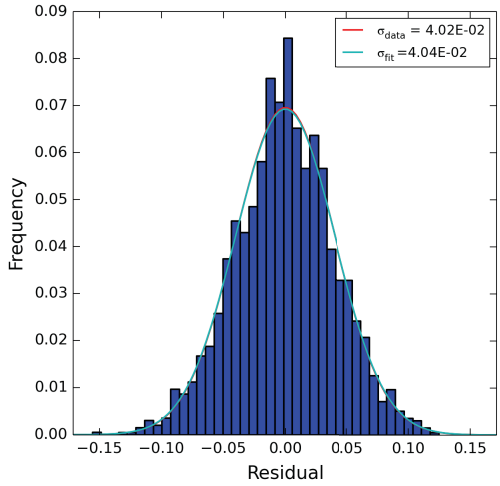
Line Brightness provides control of the color of the plotted line relative to the data. The default value of 0.7 gives the same color, only slightly darker. A value of 0 gives black lines.



You will get outputs as illustrated below as examples from Tryp_synthase.mec, showing the publication quality figures obtained using default parameters. The histogram is a bonus, to show you if the data are uniformly distributed about the fitted curve.



This shows the graphic output from the Tryp_synthase.mec example file.



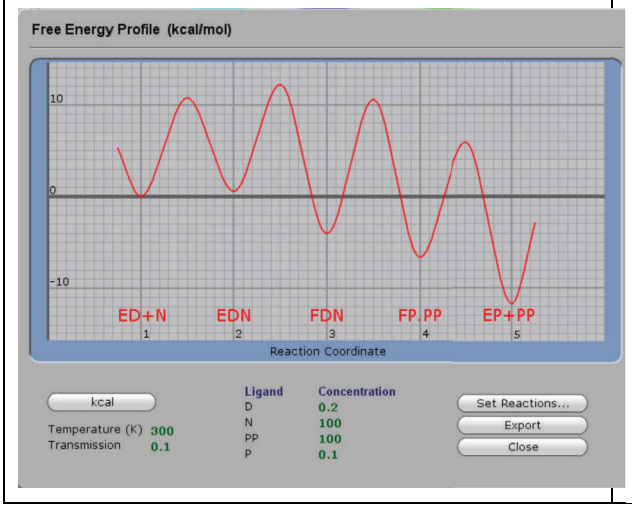
This shows a histogram of the residuals with two lines, red and light blue, showing the distribution expected from the sigma values from the data and from the fitted curves, respectively.

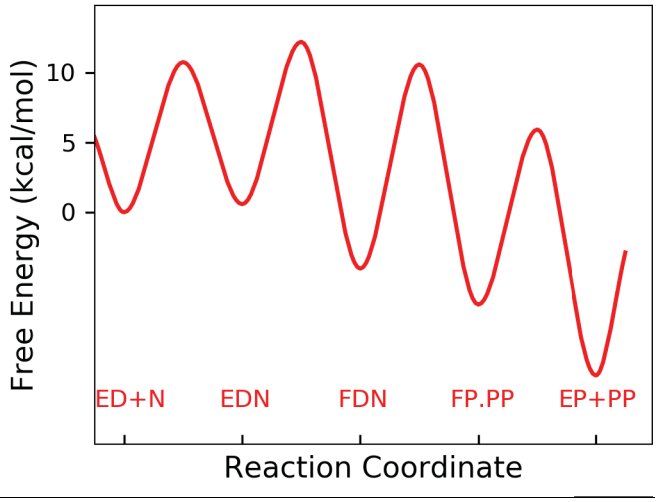
Note the program outputs both .png and .eps files. The .png files are useful for embedding into manuscripts and slides, but the .eps files are vector graphics than can be enlarged to any size and can be edited in Adobe Illustrator, or other graphics programs to change colors or fonts. We offer limited selections of fonts (Helvetica or Times Roman) to conform to journal requirements. If you want a different font, you can edit the .eps file using Adobe Illustrator.

11.2 Free energy profile output

Under the *Model Editor*, click on *Free Energy*, then edit the pathway to get the desired output of the mechanism. Then click Export. Below is an example of the results from HIVRT_fluor_qf.mec.

Onscreen Display | **Ouput of *Export* Function.**

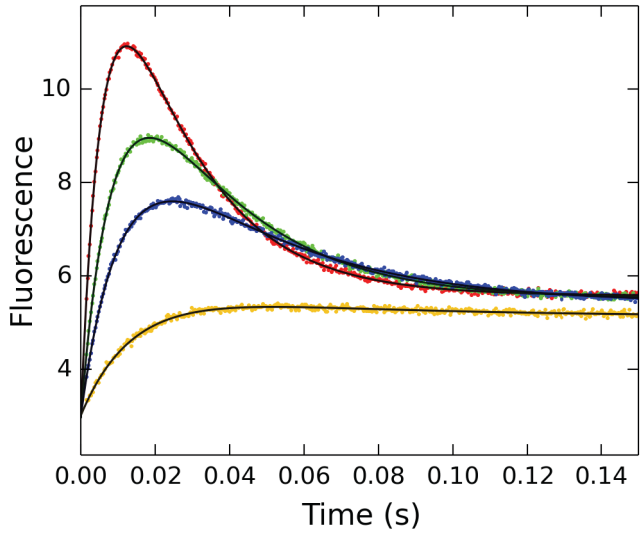


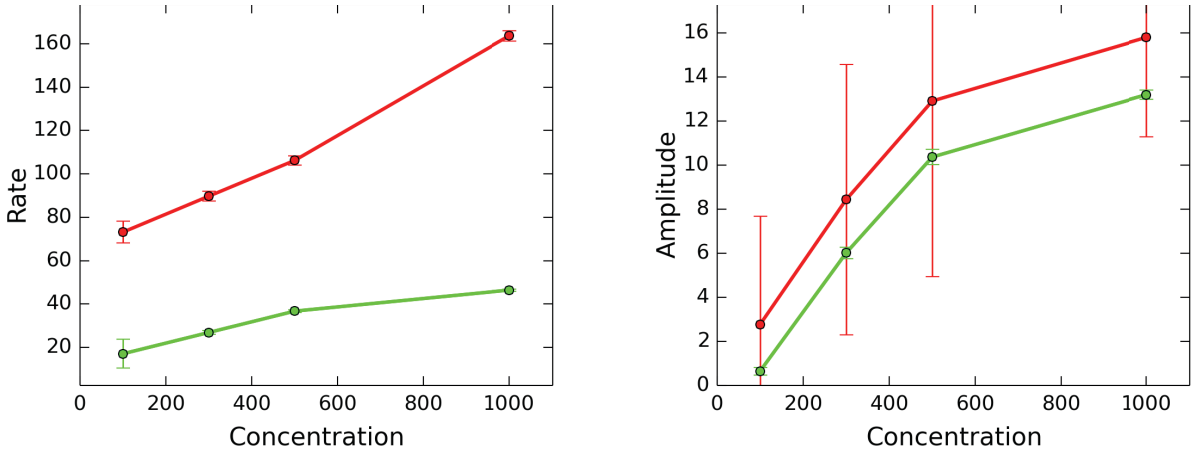


11.3 Exporting analytic equation fit results (aFit)

After fitting data to a equation using the *aFit* function, click on *Export Results* and save as *yourfilename.txt.* The program will save text files as well as publication quality figures under the same root name. For example, after fitting the Tyrp_syntase.mec data to a double exponential function, you will get the figures shown below. Both .png and .eps (vector graphics) files are produced.

The program outputs a plot of the data with the fitted curves overlain in black as shown in the example at the right, in addition to the two curves shown below illustrating the concentration dependence of the rates and amplitudes of the fast and slow phases. The units of concentration are not shown because they are not known by the program. You can edit the .eps file to alter the axes labels.



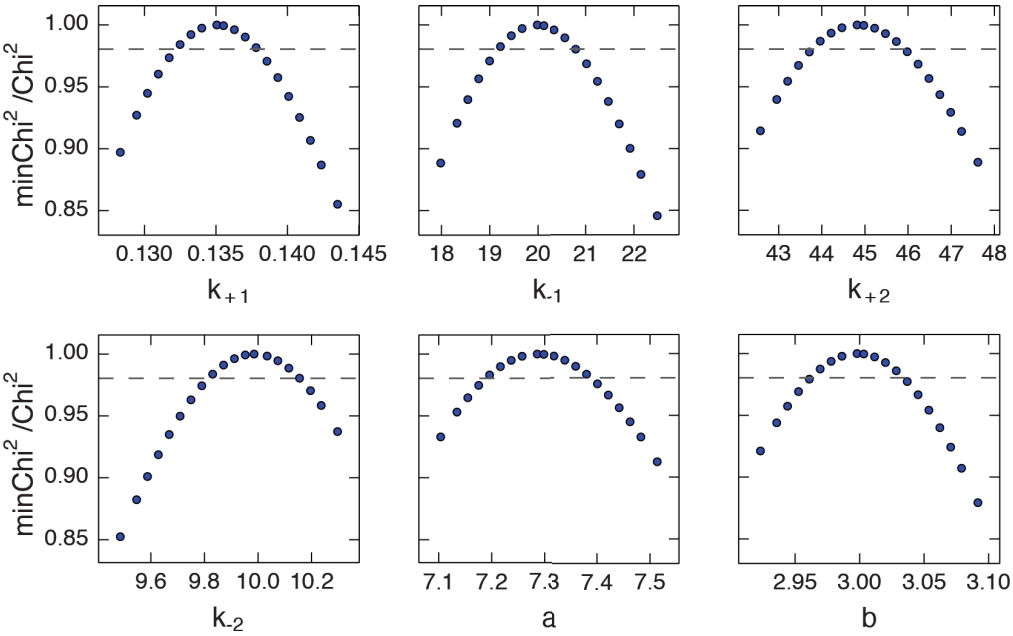


In addition to the graphics files, text files are output that you can use to create figures using other graphics programs. In addition, a text file summarizes the fit results and statistical analysis.

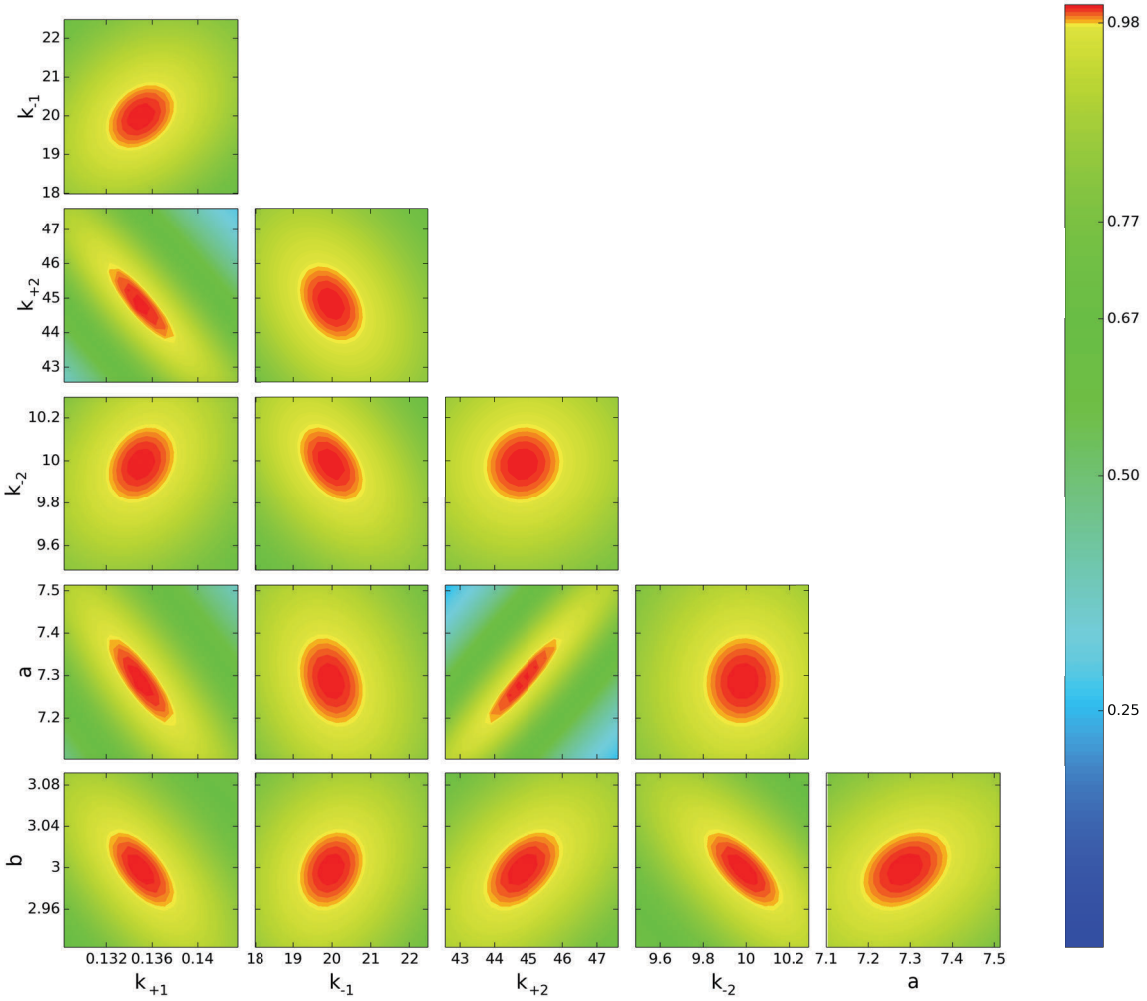
File name extension	Contents
–afit.data.txt	Original data matrix
–afit.fn.txt	Y values computed from the fitted function
–afit.txt	Tables of fit results and statistics

11.4 Exporting FitSpace confidence contours

After completing *FitSpace* confidence contour analysis, click on *Export FitSpace.* Select a directory and filename. By default, the root name will be from the mechanism filename with the extension –fsp.txt. In addition to a text file containing the results, you will get .eps and .png files of publication quality figures. For example, from the Tryp_synthase.mec example file, you will get the figures shown below. Note, if you have only calculated the 1D contours, that is all you will get for the graphic outputs. If you have also calculated the 2D contours, you will get those as well.



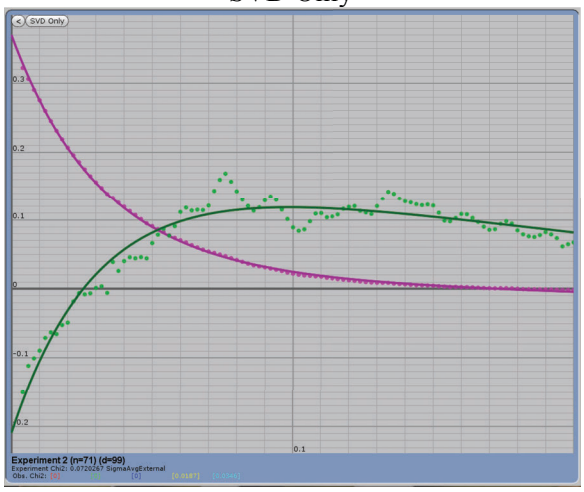
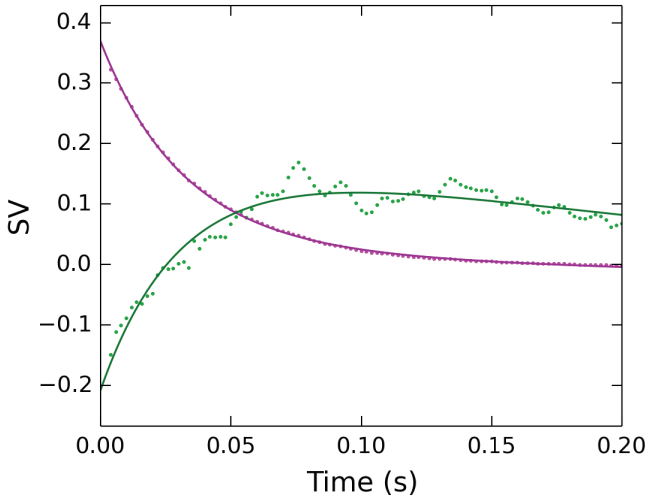
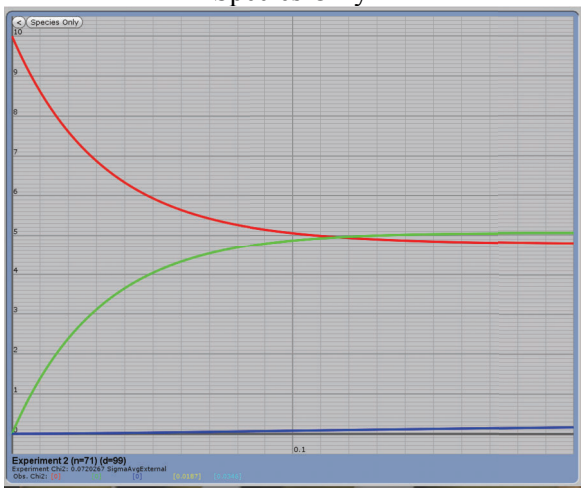
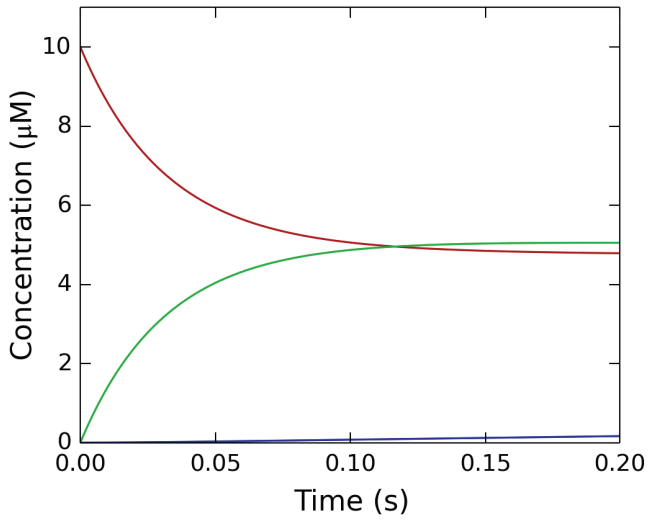
1D confidence contours for Tryp_synthase.mec. The y-axis is the minimum chi^2 divided by the chi^2 value at each parameter value. This inverse function of the normalized chi^2 was chosen because it limits the range of the y-axis from 0 to 1, while emphasizing the values closer to one (best fit).



2D confidence contours for Tryp_synthase.mec

11.5 Exporting SVD analysis of spectra

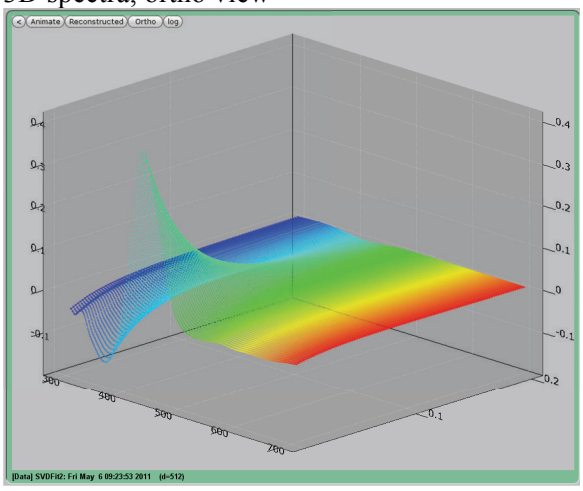
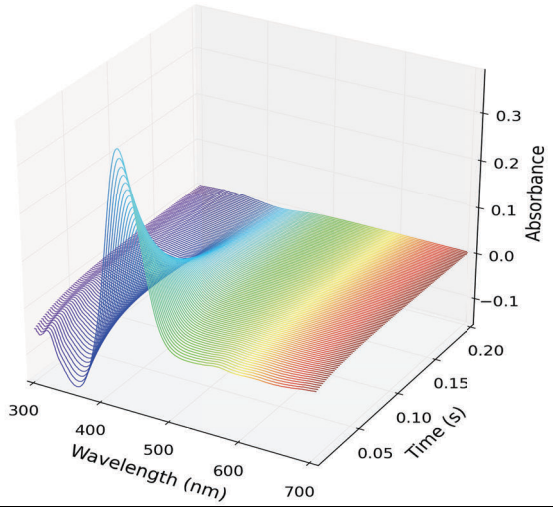
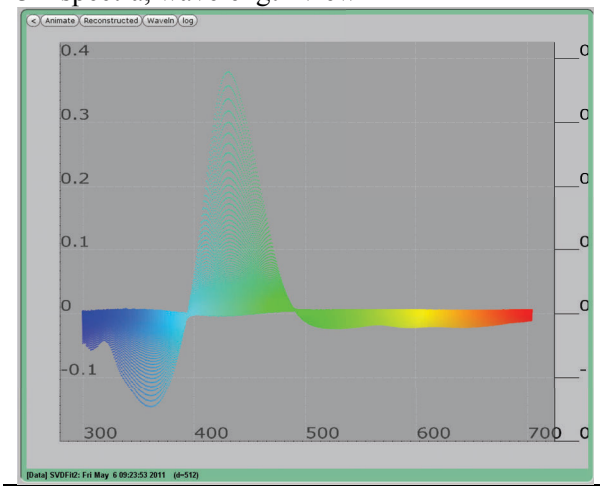
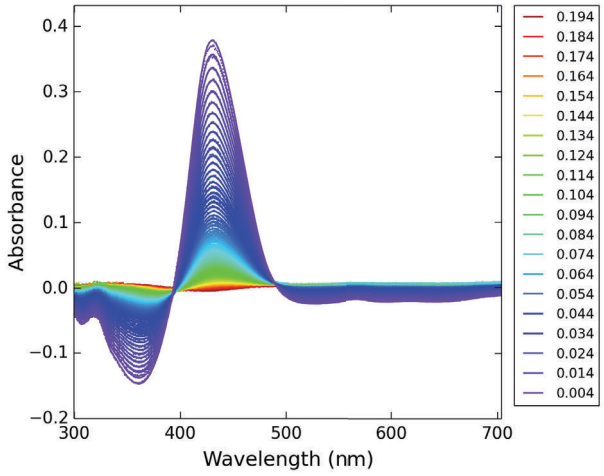
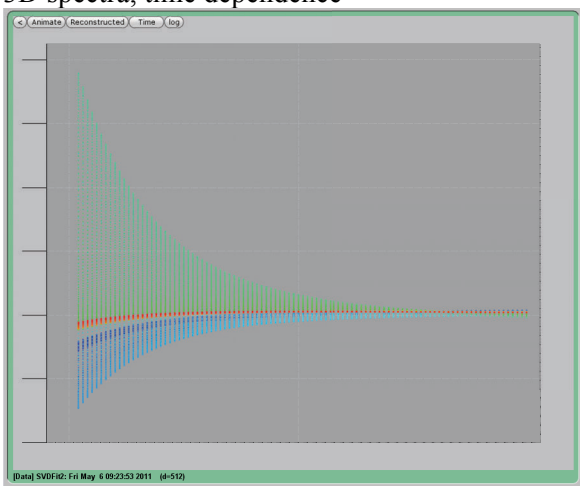
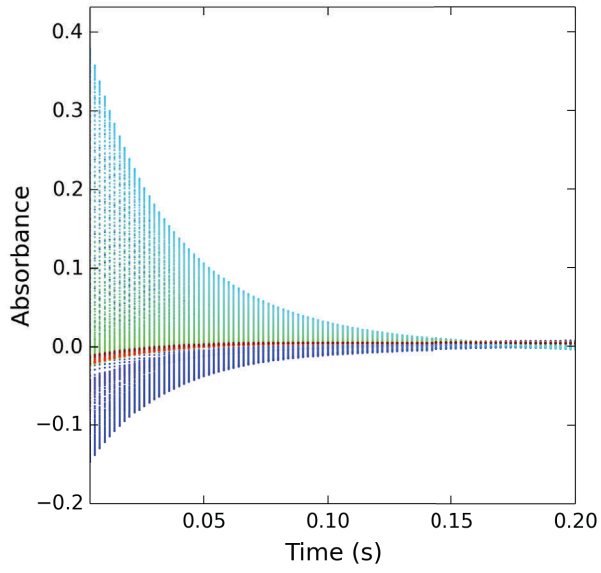
Output of the results from SVD analysis is done in two parts. First, under the experiment editor, you can export the results of the data fitting as described above in *Section 11.1*. However, the output depends on your selection of the onscreen display, where you can show either the calculated species or the fit to the SVD amplitude vectors as illustrated below. After setting the desired output onscreen, then click on *Export* under the *Experiment Editor*. These figures are made using the example file Spectra_C8-P.mec.

Onscreen Display	**Output of Export function**
SVD Only 	
Species Only 	

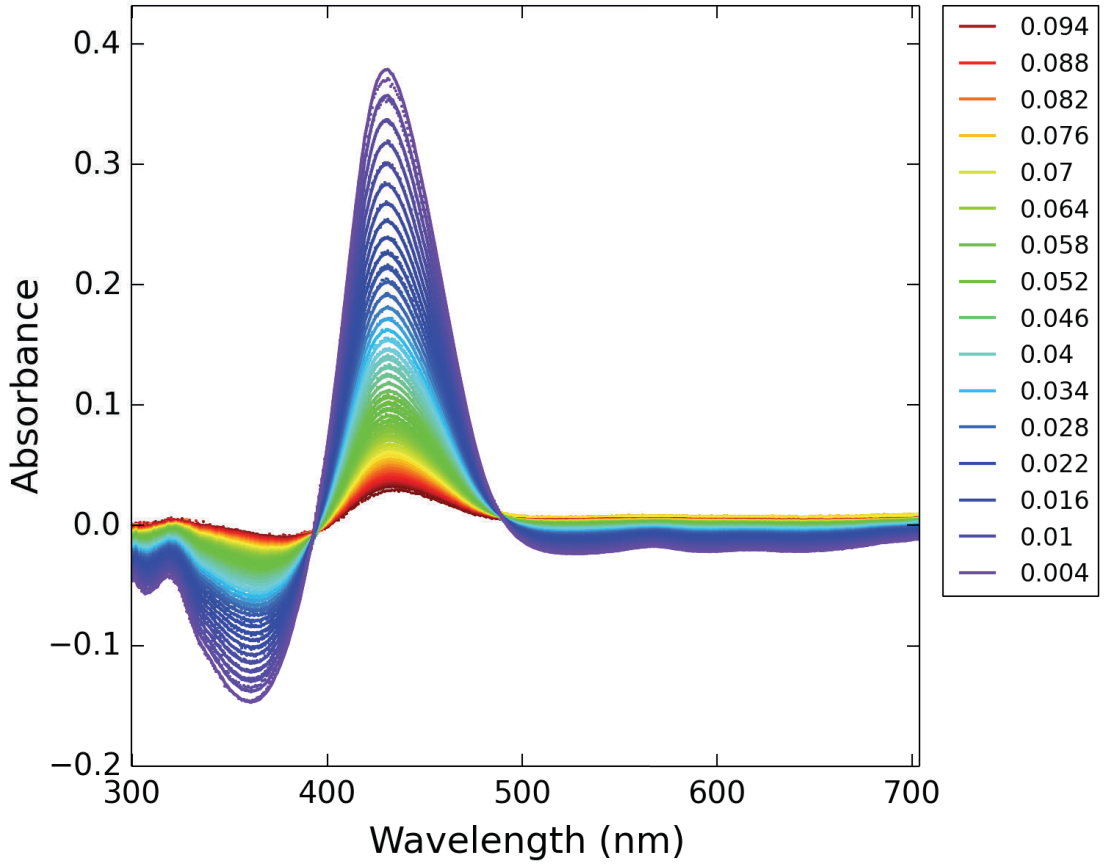
Output of the spectra in various forms is achieved through the Data Repository.

1. Select the figure you want to export
2. Select the desired configuration (see examples on the following page)
3. Click on Export and enter the desired axes labels

NOTE: Be patient when exporting 3D spectra; it takes about a minute to write the file.

Onscreen Display	Output of Export function
3D spectra, ortho view	
	
3D spectra, wavelength view	
	
3D spectra, time dependence	
	

In the ortho display, we use a color to approximate the wavelength of light. In the wavelength display, the colors denote traces collected at different times. Note also that the plot in the wavelength display shows both the data (points) and the calculated spectra (solid lines). You can also limit the maximum time to display so that the colors are more evenly distributed over the range of the largest change, as shown in the example below from Spectra_C8_P.mec where the time was limited to the first 0.1 second to generate the new display.



Chapter 12. Example Files

Here we present example files that illustrate the fitting routines and the use of dynamic simulation to evaluate models. *We welcome additions to this list*. If you have a data set that you would be willing to include, please send the mechanism file to kajohnson@kintekcorp.com along with references that we should include in citing your work.

8oxodG.mec This mechanism file shows data collected on the incorporation of 8-oxo-dGTP by the human mitochondrial DNA polymerase and the fit to a model with slow pyrophosphate release (23).

Actin_Isomerization.mec This mechanism provides a good example to illustrate the extent to which conventional nonlinear regression underestimates errors on parameters. Nonlinear regression returns standard errors indicating approximately 10-18% error on k_1 and k_{-1}. However, FitSpace analysis shows that the data only place a lower limit on $k_1 = 0.002$ and define the ratio $K_d = k_{-1}/k_1 = 990 \pm 40$ µM. You can confirm this by linking k_1 and k_{-1}, then scrolling to see the allowable range of variation in k_1 and k_{-1}.

Actin_polym.mec The kinetics of actin polymerization are modeled as described by Freiden (24) where spontaneous initiation is a function of several weak oligomerization steps. Note that his model is valid only when polymerization is favored; otherwise, the species F becomes an infinite source of actin monomers during depolymerization.

Actomyosin.mec This complex model shows the simulation of several experiments to mimic stopped-flow fluorescence and light scattering data for actomyosin and myosin ATPase (25).

Alcohol_dehydrogenase.mec In this mechanism, de-protonation of horse liver alcohol dehydrogenase-NAD+ complex controls formation of the ternary complex (26). Data are from Figure 5 of the original manuscript. The mechanism and rate constants are from Table 3, with the output factors from Table 4 as described in the Experimental Procedures section or the original paper (for the buffer concentrations and apparent extinction coefficients). Only 9 data sets are included here, but the original fitting included 28 data sets. Special thanks to Bryce Plapp (Univ. of Iowa) for providing this file.

Binding_data1.mec and binding_data2.mec These files provide data that have not yet been fit. As an exercise in data fitting, you will need to enter a model, define the output factors and set up the experiments. Binding_data1.mec: [E] = 1 µM, [S] = 1, 2, 5, 10, 20 µM. Binding_data2.mec: [E] = 1 µM, [S] = 10, 20, 50, 100, 200, 500 µM.

Binding_onestep.mec, Binding_rapidequil.mec, Binding_reversible.mec, Binding_twostep.mec These example files illustrate the various kinetics that one can observe with simple binding reactions.

Burst_data. mec Data resulting from a pre-steady state burst of product formation are provided from an experiment where the concentration of enzyme was 0.5 µM and the substrate concentration was 500 µM. You will need to define observable outputs and experiment setup, then assign the data to the experiment and fit the data to the model. What constants can you reliably get from the data? If the enzyme concentration were 1 µM, how would the constants change? What does this tell you about how much you rely upon knowing the enzyme concentration?

Burst_irrev. mec A simple pre-steady state burst of product formation is shown for the case when the chemistry step is irreversible.

Burst.mec This shows a simple pre-steady state burst of product formation, with each of the species illustrated. A typical burst is given by the sum of EP + P.

Burst_with_Fluorescence.mec In this model, a possible fluorescence trace is displayed as the weighted sum of enzyme species. Note how the shape of the curve changes as output factors *f1*, *f2* and *f3* are changed; although the observable rates of reaction will not change, there are circumstances under which it may be difficult to resolve the two rates. Use the "aFit" function to see what rates would be obtained by conventional fitting to each of the traces.

Butyronitrile.mec. This file shows the thermal decomposition of butyronitrile at 330° C and 472 bar as described by Helfferich in his book "Kinetics of Multistep Reactions" (27) from original data published by Iyer and Klien (28). Note that an additional step was added to account for the lag phase; perhaps the lag could be explained by the time required to reach 330° C. The rigorous data fitting afforded by KinTek Explorer reveals features of the data that were overlooked previously using standard approached in the field of chemical kinetics.

Data1.mec…Data5.mec. Each of these files contains a set of data collected in a fluorescence experiment with 0.2 μM enzyme and 1, 2, 5, 10, 20, 50 μM substrate. Examine each data set, define a model and output observable function, assign the data to the experiment and then fit to derive rate constants. Evaluate which rates are well constrained and which are not.

DHFR-binding.mec The time dependence of a protein fluorescence change observed when binding tetrahydrofolate to dihydrofolate reductase is shown, along with a competition experiment to measure the dissociation rate observed after mixing with methotrexate. The synthetic data were generated to mimic the results reported by Fierke *et al.* (29).

DHFR.mec This is a complex model showing the reaction pathway catalyzed by dihydrofolate reductase and simulation of several experiments as reported by Fierke *et al.* (29).

Dynein_burst.mec Two rapid quench experiments are shown, performed at two different ATP concentration, for the dynein ATPase. The red trace shows the results of a pulse-chase experiment to measure ATP binding, while the blue trace shows the results of an acid quench to measure ATP hydrolysis. Fitting the data simultaneously provides estimates for rates of ATP binding and hydrolysis as well as the reverse of hydrolysis at the active site. Note the lag in the time course of ATP hydrolysis, determined not by the acid quench-date, but required by the time dependence of ATP binding (8).

EPSP.mec This file contains 5 experiments performed on the enzyme, EPSP synthase. In these experiments, rapid quench-flow methods afforded the direct observation of an intermediate formed at the active site of the enzyme. Fitting is constrained by known equilibrium constants for steps 1, 3, 4, and 6 in the reaction sequence. For example, note how k_3 and k_{-3} are linked to give $K_3 = 12$. In addition, the overall equilibrium constant is 180 (30). Use the dynamic simulation to show that the data set a lower limit on k_3 but not an upper limit until all 8 experiments are included (see EPSP_8exp.mec). Data are from Anderson et al (30).

EPSP_8exp.mec This file contains all 8 experiments performed on the enzyme, EPSP synthase. In these experiments, rapid quench-flow methods afforded the direct observation of an intermediate formed at the active site of the enzyme. Fitting is constrained by known equilibrium constants for steps 1, 3, 4, and 6 in the reaction sequence. For example, note how k_3 and k_{-3} are linked to give $K_3 = 12$. In addition, the overall equilibrium constant is 180 (30). Data are from Anderson et al (30).

EPSP_glyphosate_binding.mec This file shows both the glyphosate and S3P concentration dependence of the fluorescence change that occurs after both S3P and glyphosate bind to EPSP synthase.

EPSPS_titration.mec This files shows the simultaneous fitting of equilibrium titration and stopped-flow kinetic data for the binding of S3P and glyphosate to EPSP synthase. Details are given in the legend to Figure 5.6. Data are from (9).

Fluor & Quench Data.mec This file contains data simulated to mimic experiments obtained by stopped-flow fluorescence and rapid quench-flow methods and illustrates the use of normalizing data in fitting data that have markedly different Y-axis scales. If you attempt to Fit All Experiments without normalization, the fit is dominated by the quench-flow data. However, if you check "Normalize Data" then the global fitting weights each experiment equally.

Gleevec.mec and Gleevec_simple.mec. Gleevec is an anticancer drug that targets the ABL kinase but binds the SRC kinase 3000-fold weaker. A dominant theory suggested that conformational selection accounted for the difference in affinity by suggesting that a DFG loop in the in state (Ein) blocked

binding to the SRC kinase while in the out position (Eout), binding to the ABL kinase was permitted. Agafonov measured the kinetics of binding by monitoring protein fluorescence and provided definitive evidence to support an induced-fit mechanism (31). Here we provide synthetic data based upon the authors' model (Gleevec_simple.mec). In Gleevec.mec, we also include the Ein – Eout isomerization preceding binding to Eout.

Global_Fit.mec Here we present data only for a set of four experiments to determine the mechanism of a simple enzyme-catalyzed reaction and inhibition by a slow binding competitive inhibitor. See Global_Fit.pdf in the documents directory to learn of the setup for each experiment.

HCV_protease.mec These data were generated to mimic the results published by Lemke et al., (32) on the slow-onset binding of an inhibitor to HCV protease. This work nicely demonstrates the utility of performing slow-onset inhibition experiments at multiple concentrations of enzymes and fitting all of the data simultaneously in order to extend the range of inhibitor concentrations that can be assayed. This allowed the authors to better assess whether the inhibitor binds in one or two steps.

HIVRT.mec This file contains data describing the fluorescence change observed with MDCC-labeled HIV reverse transcriptase which defines the kinetics of substrate-induced conformational changes, and rapid quench data to define the kinetics of nucleotide incorporation. The data are fit simultaneously to show the correlation between the rise in fluorescence and the nucleotide incorporation. Data are from Kellinger and Johnson (1).

HIV_NNRTI.mec This file contains data showing the effect of nonnucleoside inhibitors (NNRTI) on DNA polymerization catalyzed by HIV reverse transcriptase (33). This file illustrates the use of double mixing where the slow-binding NNRTI is first pre-incubated with the enzyme-DNA complex before adding nucleotide to begin polymerization.

Kinesin-mantADP.mec Kinesin with mantADP bound to two sites was mixed with microtubules at various concentrations of ATP, ranging from 0 to 150 μM. The decrease in fluorescence provides a signal for the release of mantADP from the kinesin. The reaction is biphasic. The fast phase is due to the release of mantADP from the first site and is independent of ATP concentration. ATP binds to the open site, stimulating the rate of release of mantADP from the second site. Simultaneously fitting the family of curves defines the rate of ATP binding and the rate of release of mantADP from the M-K-ATP-mantADP complex. These data provide direct evidence for an alternating site ATPase in which the binding of ATP to one site stimulates the release of product (ADP) from the neighboring site (34,35).

Michaelis-Menten_1913.mec This file shows the fitting of full time course kinetic data originally derived by Michaelis and Menten in their classic 1913 paper (36), which Roger Goody and I recently translated (37). In our translation we explain how Leonor Michaelis and Maud Menten not only fit initial velocity data using the equation that now bears their names, but they also fit their full time course data to the integrated form of the rate equations while accounting for product inhibition. This mechanism file shows their original 1913 data fit to the complete model using constants derived by Michaelis and Menten and with the assumptions that both substrate (sucrose) and products (fructose and glucose) bind in a rapid equilibrium mode and that chemistry is rate-limiting. Given their apparent weak binding to the enzyme, these assumptions are probably valid. They had no way of knowing the enzyme concentration, so we simulated the results using an arbitrarily small concentration. Interestingly, the constant derived by Michaelis and Menten was V_{max}/K_m, not K_m.

Myosin.mec This illustrates the simultaneous fitting of stopped-flow fluorescence and rapid chemical quench-flow data based upon the work by Johnson and Taylor (8). The file contains synthetic data to mimic that shown in the original publication. By simultaneously fitting the two data sets, the correlation between the second fluorescence change and the rate of the chemical reaction is established much more definitively than by simply overlaying the two data sets. In addition, one should note that the first very fast conformational change (k_2) was based, in part, on the loss of signal amplitude in the stopped-flow. Global fitting places a lower limit on k_2 by including the amplitude information.

PABA_thrombin.mec. This file gives an example of a "capture" or "conformational selection" mechanism (in contrast to induced-fit) in which the enzyme is predominantly in an occluded conformation that does not bind substrate and a rate-limiting conformational change to an open state precedes substrate binding. In this example, the observed rate decreases as a function of increasing substrate concentration (38). Note that there is also a significant amount of information in the concentration dependence of the amplitude, which is ignored by conventional fitting. Moreover, the model predicts an initial fast reaction. Synthetic shown here data were derived from the authors' model.

PBP_PPase.mec Data supporting a coupled fluorescence enzyme assay are shown in which pyrophosphatase hydrolyzes pyrophosphate and the resulting phosphate binds to a fluorescently labeled phosphate binding protein to provide a signal (39-41).

Pinene.mec The time dependence of the thermal decomposition of α-pinene is shown with a fit to a simple model (42). These data have provided a challenging example for fitting data by nonlinear regression to analytical functions (11,12), but are handled easily using *KinTek Explorer*. In that example as described in the textbooks, the reaction involving dimerization of alloocimene was assumed to be unimolecular in order to derive an analytical solution to the rate equations! Fitting by numerical integration overcomes this severe limitation and provides a meaningful rate constant for the dimerization reaction. This is described in more detail in our paper on the *KinTek Explorer* simulation program (3).

PNPase-Hs-ImmH.mec and **PNPase-Pf-DADMe-ImmH.mec**. Immucilin-H is a slow tight binding transition-state analog inhibitor of purine nucleoside phosphorylase. These files show the time dependence of product formation after mixing enzyme with substrate and various concentrations of Immucilin-H (43). PNPase-Hs-ImmH.mec shows results for the human enzyme, while PNPase-Pf-DADMe-ImmH.mec shows results for the enzyme from Plasmodium falciparum (malarial parasite). The example files show how relevant parameters can be obtained by global fitting. In fitting the data the challenge is to establish the errors on individual parameters and to define which parameters are well constrained. Note that confidence contour analysis shows that the data do not support a two-step inhibitor binding because the initial binding is too weak to measure relative to the concentration range of the experiment. Compare the global fit to the results obtained by fitting to a burst equation. Immucilin-H is now in clinical trials as a treatment for T-cell leukemia. Special thanks to Andrew S. Murkin and Vern Schramm (Albert Einstein College of Medicine) for providing these data.

Pol_burst_series.mec To study DNA polymerization, an enzyme-DNA (ED25) complex is mixed with nucleoside triphosphate (N) at various concentrations and the burst of product formation is measured (resolving DNA 25 nucleotides in length from DNA 26 nucleotides in length on a polyacrylamide gel). The family of curves can be fit simultaneously to extract the ground-state binding constant for binding dNTP and the maximum rate of polymerization. In this example, the rates of binding and dissociation of each ED complex is included for completeness, but the rates are constrained from known values.

Pol_processive.mec To study processive DNA synthesis by a polymerase, an enzyme DNA complex is mixed with a solution containing 3 of the nucleoside triphosphates, which according to the DNA sequence of the template leads to sequential incorporation of 5 nucleotides. Analysis of the time dependence of the appearance and disappearance of each intermediate species can be fit to obtain estimates for the rate of each elongation step.

Progress_curves.mec This file illustrates synthetic data for a minimal set of experiments to define k_{cat}, K_m and K_i for product inhibition from analysis of a full time course (progress curve) of enzyme catalysis. Note that the substrate and product binding rates (k_1 and k_{-3}, respectively) are fixed at 10 $\mu M^{-1}s^{-1}$ in order to reduce the number of variables to correspond to the information content of the data. The kinetic parameters obtained by fitting allow computation of k_{cat} and K_m in both the forward and reverse directions, but do not allow definition of intrinsic rate constants. Note also that this example shows that a single progress curve is not sufficient to define the mechanism; one trace can be fit adequately to a simple k_{cat}, K_m model with irreversible reactions (ignoring product rebinding and reversal of chemistry) but the concentration series cannot be fit to such a simple model. Thus, when fitting real data and no knowledge of the extent of product inhibition (reaction reversal), one cannot reliably fit to the progress curve obtained at a single concentration of substrate.

Protein folding-unfolding.mec This file shows synthetic data illustrating the folding and unfolding of a protein as a function of the concentration of a denaturant. This new feature available with Version 6 allows data to be fit globally to define the rates of folding and unfolding at zero denaturant concentration, as illustrated in a recent publication from James Bardwell's lab (7).

Question_1.mec, Question_2.mec and Question_3.mec These contain data files corresponding to the sample exam questions given in Sample_Exam.pdf in the /documents directory and reproduced below under *Sample Exam Questions*. Note that the mechanisms are valid, but the output definitions must be altered to fit the data.

Q1_data.mec, Q2_data.mec and Q3_data.mec are files that contain data only for questions 1-2 of the sample exam.

Racemase_3step.mec, Racemase_4step.mec These files contain data simulated according to the rate constants derived by globally fitting progress curves in both the forward and reverse direction of alanine racemase. Use the fitting and the dynamic simulation to ask whether the model is well-determined by the data. A simpler model (Racemase_3step.mec) can fully account for the data. Link k_{-2} and k_3 in a constant ratio and see how far they can be reduced without affecting the fit. Also, edit the model to eliminate the EI intermediate, then fit the data to the new model. What do you learn from this lesson? Synthetic data were created based upon the rate constants reported by (4,5). For a full discussion, see (2,3).

Ribozyme.mec This file illustrates data obtained from the binding of a fluorescent oligonucleotide to the hammerhead ribozyme (not the real data, but a facsimile). The first experiment shows the biphasic binding kinetics, while the second illustrates a double mixing experiment to measure the oligonucleotide dissociation rate by competition with an unlabeled oligonucleotide. Why is the dissociation rate slower than any rate in the dissociation pathway? Synthetic data were created to mimic the results reported by Bevilacqua *et al.* (44).

SlowInhibition.mec This shows a simple model with slow-onset inhibition. See also the two PNPase mechanisms for examples with real data.

Spectra_C8-P.mec These data were kindly provided by Elizabeth Karnas showing data collected in the laboratory of Professor Shunichi Fukuzumi (Osaka University) and published in (15). In this experiment, 10 μM cyclo[8]pyrrole was mixed with 500 μM phosphate at 253 K in CH_3CN. This low temperature experiment allowed resolution of the binding kinetics.

Spectra_DHOD.mec These data were kindly provided by Bruce Palfey (University of Michigan) and are from (21). In this experiment 16 μM enzyme (DHOD, dihydroorotate dehydrogenase) was mixed with 2 mM orotate and the changes in the absorption spectrum of the FMN (flavin mononucleotide) was recorded over time. The zero time point was obtained by mixing enzyme with buffer. The data allow resolution of four distinct species, but only two rate constants, assuming each step is irreversible.

Spectra_ES-EI-EP.mec This file contains synthetic data derived from a hypothetical model where there is a change in absorption spectra upon the conversion of a substrate to an intermediate and then product.

Spectra_FeCN-Asc.mec This file contains data from a test reaction based upon the reduction of $Fe(CN)_6$ by ascorbic acid described in *Section 10.5.*

Steady_State.mec This is a simple mechanism showing how the time dependence of product formation at several concentrations can be fit globally to get steady state kinetic parameters. What are k_{cat} and K_m according to the best fit to the data? Can anything other than k_{cat} and K_m (and k_{cat}/K_m) be derived from fitting these data?

Tryp_synthase.mec This file contains a single trace following changes in fluorescence of the external aldimine formed upon reaction of serine with pyridoxal phosphate at the active site of tryptophan synthase. All four rate constants and the two output factors cannot be determined unambiguously from a single trace. However, when one fits simultaneously several traces obtained at different concentrations, all constants can be determined. Data are simulated to mimic that published in (13).

Tryp_synthase_lamp_error.mec contains synthetic data illustrating fluctuations in lamp intensity that make global fitting difficult. We solve this problem by normalizing data as described under the Experiment Editor. Use this example to understand the method.

Tutorial1.mec, Tutorial2.mec and Tutorial3.mec are solutions to the tutorial problems described in the accompanying *Tutorial.pdf* found in the /documents directory.

VRS.mec The valyl-tRNA synthetases undergoes a proofreading reaction after mischarging $tRNA^{VAL}$ with isoleucine. The data show the time dependence of the formation and hydrolysis of the mischarged tRNA (45). Data were read from the original publication using a micrometer.

Chapter 13. Tutorial problem sets

13.1 Sample problems

1. Create a model for a simple reaction:

$$A + B \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} C$$

Simulate the formation of C when [*A*] = 1 μM and [*B*] = 5 μM, and with $k_1 = 3\ \mu M^{-1}s^{-1}$ and $k_{-1} = 16\ s^{-1}$.

a) What is the observed rate of formation of C?
b) How can you explain the observed rate in terms of the intrinsic rate constants?
c) If you were to design an experiment to measure this rate, what time period should you use?
d) Set up a concentration series, measuring the rate of formation of C at various concentrations of B, then plot rate versus concentration. Explain how the concentration dependence defines k_1 and k_{-1}.
e) If you repeat the experiment at different concentrations of A, what do you see?
f) Simulate the reaction where your signal is defined by 0.1*C + 0.6*A. How does that change the reaction amplitude and rate?

2. Modify the reaction in problem 1 to include a second reaction:

$$A + B \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} C \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} D$$

and simulate the loss of A, and the formation of C and D when [*A*] = 1 μM and [*B*] = 5 μM, with $k_1 = 3\ \mu M^{-1}s^{-1}$, $k_{-1} = 16\ s^{-1}$, $k_2 = 20\ s^{-1}$, $k_{-2} = 2\ s^{-1}$.

a) Determine the observed rates of reaction (fit to a double exponential) of A, C, and D. Are the rates the same for each species, and, if so, why?
b) Repeat the calculation at several concentrations of B and plot rate versus concentration for one of the species. Which species do you chose and why? What do you learn from the concentration dependence of the fast and slow rates?
c) Describe how the curves change in shape as you scroll each of the rate constants. For example, I would say, "As k_1 is increased, the rate of decay of A and formation of C increases, while the observed lag time preceding the formation of D decreases, and the amplitude for formation of C increases." Do you agree? What about the other rate constants?

3. Modify the reaction in problem 2 to make the initial binding a rapid equilibrium with the same equilibrium constant, $k_1 = 300\ \mu M^{-1}s^{-1}$, $k_{-1} = 1600\ s^{-1}$.

a) How does that change the shape of the curves?
b) Measure the rates of each observable reaction (on the ms – s time scale). What is the rate and how can you explain the observed rate based upon the intrinsic rate constants?
c) Repeat the calculation at several concentrations of B and plot rate versus concentration for one of the species. What does the concentration dependence of the observed rate tell you?
d) What equation would you use to fit the concentration dependence of the observed rate?
e) What are the minimum values of k_1 and k_{-1} (kept in a constant ratio) where you could simply the data fitting for the concentration dependence of the rate of formation of D by assuming a rapid equilibrium binding mode? That is, when is a lag no longer noticeable? How is this somewhat arbitrary judgment influenced by time resolution, signal/noise ratio and dead time of the instrument?

4. Open the *example* file "binding_data1.mec" and define an appropriate output function, then fit the data. What are the limits of error on the parameters based upon nonlinear regression? Perform a *FitSpace* calculation and compare the error limits with those obtained by simple nonlinear regression.

5. Open the example file "binding_data2.mec". Examine the data and alter the mechanism as needed and define an appropriate output function to fit the data. What are the limits of error on the parameters based upon nonlinear regression? Perform a *Fitspace* calculation and compare the error limits with those obtained by nonlinear regression. How does the shape of the curve depend upon the output factors? Can you get completely different appearance from the same set of rate constants but a different set of output factors? If you link k_1 and k_{-1} to maintain a constant ratio, can you find a lower limit on k_1?

6. Create a model for the binding of a ligand (S) to an enzyme (E) where there is a 16% increase in protein fluorescence upon substrate binding. Enter the output as a*E+b*ES with a=1 and b=1.16. Enter the rate constants shown:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \qquad k_1 = 5\ \mu M^{-1}s^{-1};\ \ k_{-1} = 15\ s^{-1}$$

a) Simulate an experiment where 2 μM enzyme is mixed with 2, 5, 10, 20, 50, and 100 μM substrate. Click on *Gen Data* under the *Experiment Editor* to generate synthetic data with sigma = 0.002 and 200 data points per trace over an appropriate time scale for the reaction to go to completion (0.1 s). How does the noise on the data change as you change the sigma value? What is sigma (it is a standard term used in statistics)?

b) Use the aFit function to fit the data to an appropriate analytical function. Click on "Rate vs Conc" to display the concentration dependence of the observed rate(s). Explain the observed concentration dependence in terms of the rate constants. Is there curvature in the plot and, if so, what causes the nonlinearity

c) What happens to the shape of the curves as you change k_1 or k_{-1}? What happens to the "Rate vs Conc" plot?

d) Fit the synthetic data globally by nonlinear regression to the model. How close are the rate constants to the values originally entered? What are the standard error estimates on the parameters? Is there sufficient information in the data to define all four rate constants and the required output factors?

e) Open the *FitSpace Editor* and *Compute FitSpace*. How do the error limits derived by Fitspace differ from those estimated from the covariance matrix obtained by nonlinear regression (standard error)?

7. Expand the model to include the formation of product alter the output to show a further increase in fluorescence upon forming EP to a level that is 40% greater than E. Edit the output to be a*E+b*ES+c*EP with a=1, b=1.16 and c=1.40. Change the timescale to 0.3 s and generate new synthetic data as described in Problem 1.

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} EP$$

$k_1 = 5\ \mu M^{-1}s^{-1};\ \ k_{-1} = 15\ s^{-1}$

$k_2 = 100\ s^{-1};\ \ k_{-2} = 25\ s^{-1}$

a) How does the additional step alter the observable kinetics?
b) Use the aFit function to fit the data to an appropriate analytical function. Click on "Rate vs Conc" to display the concentration dependence of the observed rate(s). Explain the observed concentration dependence in terms of the rate constants.
c) Fit the synthetic data globally by nonlinear regression to the model. How close are the rate constants to the values originally entered? What are the limits of error on the parameters? Is there sufficient information in the data to define all four rate constants and the required output factors?
d) Open the *FitSpace Editor* and *Compute FitSpace*. How do the error limits derived by Fitspace differ from those estimated from the covariance matrix obtained by nonlinear regression?
e) Click on "Plot observable traces at FitSpace bounds." This will overlay all possible fitted curves computed at the boundaries of the FitSpace parameter boundaries. What does this tell you?

8. Expand the model from the previous problem to include product release.

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} EP \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} E + P$$

$k_1 = 5\ \mu M^{-1}s^{-1};\ \ k_{-1} = 15\ s^{-1}$

$k_2 = 100\ s^{-1};\ \ k_{-2} = 25\ s^{-1}$

$k_3 = 5\ s^{-1};\ \ k_{-3} = 0.1\ \mu M^{-1}s^{-1}$

Change time timescale to 0.08 s and generate new synthetic data as described in Problem 1.
Now click on *New* under the *Experiment Editor* to add a new experiment and enter the initial concentrations of 2 μM enzyme with 100 μM substrate. Define the output to measure the rate of product formation. In this case the output will be the sum of EP and P. Set the timescale to 0.08 s.

a) How do the kinetics differ in comparing experiment 1 (question 2) to experiment 2 (question 3)?
b) Generate synthetic data to show the burst of product formation with sigma = 0.04 and generating 20 data points.
c) Now fit the data globally including Experiment 1 and Experiment 2. What are the limits of error on the parameters? Is there sufficient information in the data to define all five rate constants and the required output factors?

Compare the standard error based upon nonlinear regression with the limits of error obtained from the FitSpace calculation.

9. Open the file Tryp_synthase_nofit.mec and fit the data.

a) Use dynamic simulation to fit the data by eye by adjusting each parameter. Be patient and work to understand the relationships between the parameters and the observed traces. Keep working until you have a fit that overlaps the data. Write down the parameters you found. Now press "Fit Active Experiment". How close did you get?

b) Use the *aFit* function to fit the data to an appropriate analytical function then press "Rate v Conc". What do you learn from the rate versus concentration plot and how could this plot have helped you with starting estimates for parameters.

c) Return to the results of the global fitting of the data. What are the limits of errors on the parameters? Are these realistic? Perform a *Fitspace* calculation and compare the error limits with those obtained by nonlinear regression.

10. Open the file PNPase-Pf-DADMe-ImmH.mec. The data have already been fit to a model for slow-tight binding inhibition proposing a two-step inhibitor binding.

a) Use dynamic simulation and attempt to understand which of the fitted parameters are constrained by the data. Can you discover correlations between parameters; that is, after changing one parameter can you change another parameter to compensate for the changes; or can you link two parameters and find a range over which the two can vary in constant ratio without altering the fitted curves? What do you learn from investigation of the standard error derived from the fit?

b) Use the *aFit* function to fit the data to the appropriate analytic function. Which function do you chose and why? What do you learn from inspection of the concentration dependence of the rate(s)?

c) Examine the results of the FitSpace analysis. What do you now learn about the number of parameters that can be derived in fitting the data? Which parameters can be derived with confidence. How can you derive appropriate error limits on the most well-defined parameters? Re-do the FitSpace analysis with fewer parameters after choosing those most constrained by the data. What does this tell you?

d) Do the data require a two-step inhibitor binding mechanism, or would a simpler, one-step binding be sufficient?

11. Full progress curve kinetics are useful in that an enzyme reaction can be followed to completion to extract k_{cat} and K_m values. It has been claimed that only one progress curve obtained at one concentration of substrate is sufficient and that the data can be fit to a simplified model with two irreversible steps:

$$E + S \xrightarrow{k_1} ES \xrightarrow{k_2} E + P$$

$$k_{cat} = k_2$$

$$K_m = k_2 / k_1$$

a) Convince yourself that, theoretically at least, this is a valid shortcut under some limited circumstances.

b) Determine when this shortcut is invalid. As one approach, you could open the file "Progress_curves.mec" and edit it to obtain the shortcut. One method is to edit the mechanism file to eliminate the explicit ES to EP conversion. Another is to set k_3 to a very large number and set all reverse rate constants to zero. Now attempt to fit the data. Can you fit one progress curve, but not all at the same time? Why is that? What does this tell you? What does the simplified model assume that may not be true?

13.2 Sample take-home exam questions

In the KinTek Explorer examples folder you will find “Question_1.mec”, “Question_2.mec” and “Question_3.mec” containing the data needed to answer the three questions. Data for question 2 in the form of text files for are in the /rawData sub-directory. Note that for questions 2 and 3 you will need to redefine the output expression.

Question 1. In the KinTek Explorer examples folder, find “Question_1.mec”. The data show the protein fluorescence change following the mixing of an enzyme with an inhibitor. The reaction was initiated by mixing 0.5 μM enzyme with various concentrations of inhibitor (2, 5, 10, 20, 50, 100, 200 μM).

a. Fit the data using the analytical fitting function (aFit) to the appropriate equation. Explain your choice of the equation for fitting the time dependence of the fluorescence change.
b. Plot (in an external program) rate and amplitude versus concentration of inhibitor, then fit the concentration dependence of the rate to an appropriate equation in order to derive relevant parameters governing the inhibitor binding reaction. Explain your choice of equation for fitting the concentration dependence of the rate and what the parameters mean. What information is available in the amplitude plot?
c. Fit the data globally using the KinTek Explorer “Fit Active Experiment” function. You will need to define an appropriate output function for the “Observables” under the Experiment Editor and find reasonable starting estimates. Note that the Observable output just has a letter *a* as a placeholder in the downloaded file. Explain your rationale for defining the output and deriving the scaling factors in your fitting.
d. Use a print screen (or screen capture) function to save and show your fitted curves from KinTek Explorer.
e. Which of the intrinsic rate constants can be derived with confidence from these data? What are the limits of error on the rate constants?
f. What additional experiment would you want to perform to more accurately establish all four rate constants?

Question 2. In the KinTek Explorer examples folder, find "Question_2.mec". In addition, data for each of the four experiments can be found in the /rawData directory: Q2A.data.txt, Q2B.data.txt, Q2C.data.txt and Q2D.data.txt. The data are already in the .mec file so will not need to import the data into the program. Note also that the program has default values for rate constants and simple placeholders for outputs. You will need to enter appropriate output factors and estimates for rate constants.

To fit data to equations, you can use either the aFit functions built into KinTek Explorer, or you can import the data into another program for nonlinear regression. In either case, describe the rationale for the equation you chose for fitting and the meaning of the parameters derived as outlined below. As detailed below, fit the data to derive as many kinetic constants as you can for a minimal reaction scheme:

$$E + S \underset{k_{-1}}{\overset{k_1}{\rightleftharpoons}} ES \underset{k_{-2}}{\overset{k_2}{\rightleftharpoons}} EP \underset{k_{-3}}{\overset{k_3}{\rightleftharpoons}} E + P$$

Experiment 1 (dataset Q2A.data.txt). A small amount of enzyme (0.01 μM) was mixed with various concentrations of substrate (0.5, 1, 2, 5, and 10 μM) and the time dependence of product formation was monitored until the reaction went to near completion.

a. Fit the data to derive k_{cat} and K_m for the substrate by conventional methods measuring the initial velocity.
b. Fit the data to derive k_{cat} and K_m using the full reaction time course based upon simulation. Using the "Fit Active Experiment" function, find a set of rate constants that fit the data in Experiment 1 without regard to fitting the remaining experiments. Use these constants to compute k_{cat} and K_m. How do these numbers compare to those derived in (a)?
c. Describe how you use these constants to guide you in creating estimates of intrinsic kinetic parameters.

Experiment 2 (dataset Q2B.data.txt). The time dependence of changes in protein fluorescence was recorded after mixing 1 μM enzyme with various concentrations of substrate (2, 5, 10, 20, 50 and 100 μM).

a. Fit the data to an appropriate function (using either aFit function in KinTek Explorer or another program capable of nonlinear regression). Explain why you chose the function you did for fitting data.
b. Plot the rate(s) versus concentration in a separate graph and fit the concentration dependence to extract kinetic parameters. What do the kinetic parameters tell you? Explain the rationale you used in choosing an equation.
c. Describe how these data guide you in entering estimates for intrinsic rate constants in the model.

Experiment 3 (dataset Q2C.data.txt). The time dependence product formation was recorded using rapid chemical-quench flow methods by mixing 1 μM enzyme with 200 μM substrate. The amount of product formed was quantified after quenching the reaction with 1 N HCl at various times.

a. Fit the data to an appropriate function (using either the aFit function in KinTek Explorer or another program capable of nonlinear regression). Explain why you chose the function you did for fitting data. Show your fit to the data.
b. What do the kinetic parameters tell you? Explain the rationale you used in choosing an equation.
c. Describe how these data guide you in entering estimates for intrinsic rate constants in the model.

Experiment 4 (dataset Q2D.data.txt). The time dependence product formation was recorded using rapid chemical quench flow methods after mixing 20 μM enzyme with 2 μM substrate. The amount of product formed was quantified after quenching the reaction with 1 N HCl at various times.

a. Fit the data to an appropriate function (using either aFit function in KinTek Explorer or another program capable of nonlinear regression). Explain why you chose the function you did for fitting data.
b. What do the kinetic parameters tell you? Explain the rationale you used in choosing an equation.

c. Describe how these data guide you in entering estimates for intrinsic rate constants in the model.

Finally, fit all of the data globally to obtain a single set of rate constant to explain all data.

a. Use a print screen (or screen capture) function and show your fitted curves from KinTek Explorer.
b. How many rate constants can be determined?
c. What are the limits of error on each of the rate constants?
d. Compute k_{cat} and K_m, k_{cat}/K_m, K_d for S and K_2 (the equilibrium constant for the chemistry step at the active site) from the final model. Create a table comparing the numbers. How do the values compare to those derived in fitting Experiment 1 by each of the methods?
e. What is the fraction of the enzyme in E, ES and EP during steady state turnover in the presence of saturating substrate concentration?
f. Construct a free energy profile for the pathway. What does this tell you about differences in individual steps in the pathway comparing the two enzymes?

Question 3. Open the file "Question 3.mec" which was obtained for a variant of the enzyme described in Question 2. Note that the experiments included in the file are similar to those described in Question 2, but the time allowed for reaction and some of the concentrations of substrate were different, as described under the Experiment Editor for each experiment.

a. Fit the data globally to derive the rate constants.
b. Use a print screen (or screen capture) function and show your fitted curves from KinTek Explorer.
c. Compute k_{cat} and K_m, k_{cat}/K_m, K_d for S and K_2 (the equilibrium constant for the chemistry step at the active site) from the final model. Create a table comparing enzyme from Question 2 with enzyme from Question 3.
d. Calculate the free energy for substrate binding at a physiological substrate concentration of 5 μM for each enzyme.
e. Is this a better or worse enzyme compared to the enzyme in Question 2? Explain your rationale in describing how the two enzymes differ.
f. Construct a free energy profile for the pathway and use it as a basis for comparing the two enzymes.

Literature Cited

1. Kellinger, M. W., and Johnson, K. A. (2010) Nucleotide-dependent conformational change governs specificity and analog discrimination by HIV reverse transcriptase. *Proc. Natl. Acad. Sci. U. S. A.* **107**, 7734-7739
2. Johnson, K. A., Simpson, Z. B., and Blom, T. (2009) FitSpace Explorer: An algorithm to evaluate multidimensional parameter space in fitting kinetic data. *Anal. Biochem.* **387**, 30-41
3. Johnson, K. A., Simpson, Z. B., and Blom, T. (2009) Global Kinetic Explorer: A new computer program for dynamic simulation and fitting of kinetic data. *Anal. Biochem.* **387**, 20-29
4. Spies, M. A., Woodward, J. J., Watnik, M. R., and Toney, M. D. (2004) Alanine racemase free energy profiles from global analyses of progress curves. *J. Am. Chem. Soc.* **126**, 7464-7475
5. Spies, M. A., and Toney, M. D. (2007) Intrinsic primary and secondary hydrogen kinetic isotope effects for alanine racemase from global analysis of progress curves. *J. Am. Chem. Soc.* **129**, 10678-10685
6. Smith-Maxwell, C. J., Ledwell, J. L., and Aldrich, R. W. (1998) Role of the S4 in cooperativity of voltage-dependent potassium channel activation. *The Journal of general physiology* **111**, 399-420
7. Stull, F., Koldewey, P., Humes, J. R., Radford, S. E., and Bardwell, J. C. (2016) Substrate protein folds while it is bound to the ATP-independent chaperone Spy. *Nature structural & molecular biology* **23**, 53-58
8. Johnson, K. A. (1983) The pathway of ATP hydrolysis by dynein. Kinetics of a presteady state phosphate burst. *J.Biol.Chem.* **258**, 13825-13832
9. Anderson, K. S., Sikorski, J. A., and Johnson, K. A. (1988) Evaluation of 5-enolpyruvoylshikimate-3-phosphate synthase substrate and inhibitor binding by stopped-flow and equilibrium fluorescence measurements. *Biochemistry (Mosc).* **27**, 1604-1610
10. Johnson, K. A. (1992) Transient-state kinetic analysis of enzyme reaction pathways. *The.Enzymes.* **XX**, 1-61
11. Bates, D. M., and Watts, D. G. (1988) *Nonlinear Regression Analysis and its Applications*, John Wiley & Sons, New York
12. Seber, G. A. F., and Wild, C. J. (2003) *Nonlinear Regression*, John Wiley & Sons, Hoboken
13. Anderson, K. S., Miles, E. W., and Johnson, K. A. (1991) Serine modulates substrate channeling in tryptophan synthase. A novel intersubunit triggering mechanism. *J.Biol.Chem.* **266**, 8020-8033
14. Press, W. H., Vetterling, W. T., and Flannery, B. P. (2007) *Numerical Recipies in C++*, 3 ed., Cambridge University Press, New York
15. Paradies, G., Petrosillo, G., Paradies, V., and Ruggiero, F. M. (2011) Mitochondrial dysfunction in brain aging: Role of oxidative stress and cardiolipin. *Neurochem. Int.* **58**, 447-457
16. Henry, E. R., and Hofrichter, J. (1992) Singular Value Decomposition: Application to Analysis of Experimental Data. *Methods Enzymol.* **210**, 129-192
17. Maeder, M., and Zuberbuhler, A. D. (1990) Nonlinear Least-Squares Fitting of Multivariate Absorption Data. *Anal. Chem.* **62**, 2220-2224
18. Hendler, R. W., and Shrager, R. I. (1994) Deconvolutions based on singular value decomposition and the pseudoinverse: a guide for beginners. *J. Biochem. and Biophys. Methods* **28**, 1-33
19. Volkov, V. V. (1996) Separation of Additive Mixture Spectra by a Self-Modeling Method. *Applied Spectroscopy* **50**, 1996
20. Malinowski, E. R. (2002) Compenent Analysis. in *Factor Analysis in Chemistry*, 3rd Ed., John Wiley & Sons, New York. pp
21. Park, H. Y., Kim, S. A., Korlach, J., Rhoades, E., Kwok, L. W., Zipfel, W. R., Waxham, M. N., Webb, W. W., and Pollack, L. (2008) Conformational changes of calmodulin upon Ca2+ binding studied with a microfluidic mixer. *Proc. Natl. Acad. Sci. U. S. A.* **105**, 542-547
22. Olsen, S., Baranov, M. S., Baleeva, N. S., Antonova, M. M., Johnson, K. A., and Solntsev, K. M. (2016) pH-Sensitive fluorophores from locked GFP chromophores by a non-alternant analogue of the photochemical meta effect. *Physical chemistry chemical physics : PCCP* **18**, 26703-26711

23. Hanes, J. W., and Johnson, K. A. (2007) A novel mechanism of selectivity against AZT by the human mitochondrial DNA polymerase. *Nucleic Acids Res.* **35**, 6973-6983
24. Frieden, C. (1985) Actin and tubulin polymerization: the use of kinetic methods to determine mechanism. *Annu. Rev. Biophys. Biophys. Chem.* **14**, 189-210
25. Johnson, K. A., and Taylor, E. W. (1978) Intermediate states of subfragment 1 and actosubfragment 1 ATPase: reevaluation of the mechanism. *Biochemistry (Mosc).* **17**, 3432-3442
26. Kovaleva, E. G., and Plapp, B. V. (2005) Deprotonation of the horse liver alcohol dehydrogenase-NAD+ complex controls formation of the ternary complexes. *Biochemistry (Mosc).* **44**, 12797-12808
27. Helfferich, F. G. (2004) *Kinetics of Multistep Reactions*, 2nd ed., Elsevier, Amsterdam
28. Iyer, D. D., and Klein, M. T. (1997). *J. Supercrit. Fluids* **10**
29. Fierke, C. A., Johnson, K. A., and Benkovic, S. J. (1987) Construction and evaluation of the kinetic scheme associated with dihydrofolate reductase from Escherichia coli. *Biochemistry (Mosc).* **26**, 4085-4092
30. Anderson, K. S., Sikorski, J. A., and Johnson, K. A. (1988) A tetrahedral intermediate in the EPSP synthase reaction observed by rapid quench kinetics. *Biochemistry (Mosc).* **27**, 7395-7406
31. Agafonov, R. V., Wilson, C., Otten, R., Buosi, V., and Kern, D. (2014) Energetic dissection of Gleevec's selectivity toward human tyrosine kinases. *Nature structural & molecular biology* **21**, 848-853
32. Lemke, C. T., Goudreau, N., Zhao, S., Hucke, O., Thibeault, D., Llinas-Brunet, M., and White, P. W. (2011) Combined X-ray, NMR, and Kinetic Analyses Reveal Uncommon Binding Characteristics of the Hepatitis C Virus NS3-NS4A Protease Inhibitor BI 201335. *J. Biol. Chem.* **286**, 11434-11443
33. Spence, R. A., Kati, W. M., Anderson, K. S., and Johnson, K. A. (1995) Mechanism of inhibition of HIV-1 reverse transcriptase by nonnucleoside inhibitors. *Science* **267**, 988-993
34. Gilbert, S. P., Moyer, M. L., and Johnson, K. A. (1998) Alternating site mechanism of the kinesin ATPase. *Biochemistry (Mosc).* **37**, 792-799
35. Auerbach, S. D., and Johnson, K. A. (2005) Alternating site ATPase pathway of rat conventional kinesin. *J. Biol. Chem.* **280**, 37048-37060
36. Michaelis, L., and Menten, M. L. (1913) Die Kinetik der Invertinwirkung. *Biochem. Z.* **49**, 333-369
37. Johnson, K. A., and Goody, R. S. (2011) The original Michaelis constant: translation of the 1913 Michaelis-Menten paper. *Biochemistry (Mosc).* **50**, 8264-8269
38. Pozzi, N., Vogt, A. D., Gohara, D. W., and Di Cera, E. (2012) Conformational selection in trypsin-like proteases. *Curr. Opin. Struct. Biol.* **22**, 421-431
39. Hanes, J. W., and Johnson, K. A. (2008) Real-time measurement of pyrophosphate release kinetics. *Anal. Biochem.* **372**, 125-127
40. Brune, M., Hunter, J. L., Corrie, J. E., and Webb, M. R. (1994) Direct, real-time measurement of rapid inorganic phosphate release using a novel fluorescent probe and its application to actomyosin subfragment 1 ATPase. *Biochemistry (Mosc).* **33**, 8262-8271
41. Brune, M., Hunter, J. L., Howell, S. A., Martin, S. R., Hazlett, T. L., Corrie, J. E., and Webb, M. R. (1998) Mechanism of inorganic phosphate interaction with phosphate binding protein from Escherichia coli. *Biochemistry (Mosc).* **37**, 10370-10380
42. Fuguitt, R. E., and Hawkins, J. E. (1947) Rate of thermal isomerization of α-pinene in the liquid phase. *J. Am. Chem. Soc.* **69**, 319-322
43. Kicska, G. A., Tyler, P. C., Evans, G. B., Furneaux, R. H., Kim, K., and Schramm, V. L. (2002) Transition state analogue inhibitors of purine nucleoside phosphorylase from Plasmodium falciparum. *J. Biol. Chem.* **277**, 3219-3225
44. Bevilacqua, P. C., Kierzek, R., Johnson, K. A., and Turner, D. H. (1992) Dynamics of ribozyme binding of substrate revealed by fluorescence-detected stopped-flow methods. *Science* **258**, 1355-1358
45. Fersht, A. R., and Dingwall, C. (1979) An editing mechanism for the methionyl-tRNA synthetase in the selection of amino acids in protein synthesis. *Biochemistry (Mosc).* **18**, 1250-1256

Appendix

Troubleshooting

Error message: This application has failed to start because gsl18_vc6.dll was not found. The *KinTek_Explorer.exe* program must be in the *KinTek_Explorer* directory with the subdirectory containing the dll files. If you want to have an icon to start the program elsewhere on your computer, create a shortcut to *KinTek_Explorer.exe* and then copy the shortcut to a different location. Right-click on the shortcut and select "Properties" to ensure that the program is set to start in the *KinTek_Explorer* folder. On a Mac, simply drag the *KinTek Explorer* icon to the dock.

If you experience a reproducible crash or bug when running the program, download the latest version of the software and see if the crash persists. If so, then send us the mechanism file and a description of what you were doing when the program crashed to: software@kintekcorp.com.

Downloading updates

To download the latest update of the software go to:
http://www.kintekexplorer.com/ and follow the links to *download.*

Check the version number at the top of the program. The last four digits define the latest updates of the current release; for example, Version 7.6.180702 was the version number on July 2, 2018.

Citing KinTek Explorer

KinTek Explorer was conceived by Kenneth A. Johnson and Zachary Booth Simpson, and programmed by Thomas Blom with the sole support of KinTek Corporation. Simpson and Blom are former professional game programmers who now work to create fast and interactive scientific software. *KinTek Explorer* is built on the zlab framework, designed by Mr. Simpson who writes scientifically-based games (see www.mine-control.com). We hope you enjoy playing KinTek Explorer! For more information about this software and KinTek Corp, see www.kintekcorp.com

Two papers describing the use of the *KinTek Explorer* and *FitSpace Explorer* are in press in *Analytical Biochemistry*, where the theory and use of *FitSpace Explorer* are explained in more detail than provided here. *Please reference these papers in citing the use of KinTek Explorer:*

Johnson, K. A., Simpson, Z. B., and Blom, T. (2009) Global Kinetic Explorer: A new computer program for dynamic simulation and fitting of kinetic data. *Analytical Biochemistry* 387, 20-29

Johnson, K. A., Simpson, Z. B., and Blom, T. (2009) FitSpace Explorer: An algorithm to evaluate multi-dimensional parameter space in fitting kinetic data. *Analytical Biochemistry* 387, 30-41.

A paper describing the use of the program to fit full progress curve kinetic traces and slow-onset inhibition data was published in *Methods in Enzymology*.

Johnson, K. A. (2009) Fitting Enzyme Kinetic Data with KinTek Explorer *in Methods in Enzymology*, 467, 601-626.

KinTek Software license agreement

Single User License Grant: KinTek Corporation ("KinTek") and its suppliers grant to Customer ("Customer") a nonexclusive and nontransferable license to use the KinTek software ("Software") in object code form and run solely on a single central processing unit owned or leased by Customer or its employees.

Starting with Version 7.0, we will no longer support the USB hardware key. Rather, the license will be supplied electronically. Each license enables a user to install the software on two computers. In the absence of a valid license, the software can still be run, but new data cannot be imported and the results of fitting cannot be exported. Still the software can be used in "student" mode to model kinetics of open files previously saved.

CUSTOMER SHALL NOT: MODIFY THE SOFTWARE; REVERSE COMPILE OR REVERSE ASSEMBLE ALL OR ANY PORTION OF THE SOFTWARE; OR RENT, LEASE, DISTRIBUTE, SELL, OR CREATE DERIVATIVE WORKS OF THE SOFTWARE.

Customer agrees that aspects of the licensed materials, including the specific design and structure of individual programs, constitute trade secrets and/or copyrighted material of KinTek. Customer agrees not to disclose, provide, or otherwise make available such trade secrets or copyrighted material in any form to any third party without the prior written consent of KinTek. Customer agrees to implement reasonable security measures to protect such trade secrets and copyrighted material. Title to Software and documentation shall remain solely with KinTek.

LIMITED WARRANTY. KinTek warrants that for a period of ninety (90) days from the date of shipment from KinTek: (i) the media on which the Software is furnished will be free of defects in materials and workmanship under normal use; and (ii) the Software substantially conforms to its published specifications.

Except for the foregoing, the Software is provided AS IS. This limited warranty extends only to Customer as the original licensee. Customer's exclusive remedy and the entire liability of KinTek and its suppliers under this limited warranty will be, at KinTek or its service center's option, repair, replacement, or refund of the Software if reported (or, upon request, returned) to the party supplying the Software to Customer. In no event does KinTek warrant that the Software is error free or that Customer will be able to operate the Software without problems or interruptions.

This warranty does not apply if the software has been altered, except by KinTek, or has not been installed, operated, repaired, or maintained in accordance with instructions supplied by KinTek.

DISCLAIMER. EXCEPT AS SPECIFIED IN THIS WARRANTY, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE, ARE HEREBY EXCLUDED TO THE EXTENT ALLOWED BY APPLICABLE LAW.

IN NO EVENT WILL KINTEK OR ITS SUPPLIERS BE LIABLE FOR ANY LOST REVENUE, PROFIT, OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR PUNITIVE DAMAGES HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE EVEN IF KINTEK OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event shall KinTek's or its suppliers' liability to Customer, whether in contract, tort

(including negligence), or otherwise, exceed the price paid by Customer. The foregoing limitations shall apply even if the above-stated warranty fails of its essential purpose.

The above warranty DOES NOT apply to any beta software, any software made available for testing or demonstration purposes, any temporary software modules or any software for which KinTek does not receive a license fee. All such software products are provided AS IS without any warranty whatsoever.

This License is effective until terminated. Customer may terminate this License at any time by destroying all copies of Software including any documentation. This License will terminate immediately without notice from KinTek if Customer fails to comply with any provision of this License. Upon termination, Customer must destroy all copies of Software.

Software, including technical data, is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Customer agrees to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import Software.

This License shall be governed by and construed in accordance with the laws of the State of Texas, United States of America, as if performed wholly within the state and without giving effect to the principles of conflict of law. If any portion hereof is found to be void or unenforceable, the remaining provisions of this License shall remain in full force and effect. This License constitutes the entire License between the parties with respect to the use of the Software.

Restricted Rights - KinTek's software is provided to non-DOD agencies with RESTRICTED RIGHTS and its supporting documentation is provided with LIMITED RIGHTS. Use, duplication, or disclosure by the Government is subject to the restrictions as set forth in subparagraph "C" of the Commercial Computer Software - Restricted Rights clause at FAR 52.227-19. In the event the sale is to a DOD agency, the government's rights in software, supporting documentation, and technical data are governed by the restrictions in the Technical Data Commercial Items clause at DFARS 252.227-7015 and DFARS 227.7202. Manufacturer is KinTek Corporation, 7604 Sandia Loop, Austin, TX 78735.