

Petr Kuzmič

Getting Started with DynaFit

Software Version 4.0

Manual Version 1.1.004

May 2015



BioKin Ltd.

www.biokin.com

Contents

1	Introduction	1
1.1	About DynaFit	1
1.1.1	Biochemical examples	1
1.1.2	Examples from chemistry, pharmacology, and other areas ..	8
1.1.3	Simulations	10
1.1.4	Algebraic fitting models	12
1.2	Installation	12
1.2.1	Microsoft Windows	13
1.2.2	Mac OS X	13
1.2.3	Obtaining the license file	13
1.2.4	Installing the license file	13
1.3	Example problems	16
1.4	Support	16
2	User Interface	17
2.1	Graphical User Interface (GUI)	17
2.1.1	File menu	17
2.1.2	Edit menu	19
2.1.3	View menu	19
2.2	Command-line interface	21
2.2.1	Microsoft Windows	21
2.2.2	Mac OS X	21
2.2.3	Batch script automation	22
3	Input files	23
3.1	Data files	23
3.1.1	ASCII text format	23
3.1.2	Required format of experimental data sets	25
3.1.3	“Sheet” format	26
3.1.4	Comment and header lines	26
3.1.5	Number of independent variables	27

3.1.6	Number format	32
3.2	Script files	33
3.2.1	Sections	34
3.2.2	Keywords	35
3.2.3	Delimiters and special characters	36
3.2.4	Comprehensive scripting manual	38
3.3	Initialization files	38
3.3.1	Master initialization file	38
3.3.2	Overriding default settings	41
4	Output files	43
4.1	HTML files	43
4.2	Plain text files	45
4.3	GIF graphics files	46
5	Step-by-step tutorial	49
5.1	Input data files	49
5.1.1	Create new project directory and data directory	50
5.1.2	Create data files on the disk	51
5.2	Script files	52
5.2.1	Fitting the control experiment	52
5.2.2	Fitting the competition experiment	59
5.2.3	Summary	62
5.3	Output files: Publication quality graphics	62
5.4	Note to Mac OS X users	63
5.4.1	Copying plain-text (ASCII) data from MS Excel	64
5.4.2	Pure plain text (ASCII) editors	65
	References	67
	Index	69

Chapter 1

Introduction

1.1 About DynaFit

The primary purpose of the DynaFit [1, 2] software package is to fit experimental data arising in chemistry, biochemistry, pharmacology, biophysics, and other related disciplines, to a mathematical model corresponding to the postulated reaction mechanism. The key advantage is that the mathematical models are derived automatically, from symbolic input.

1.1.1 Biochemical examples

Most uses of DynaFit published in the journal literature are from biochemistry or molecular biology (for review cf. [2]). The experimental data files associated with all examples described below and the required DynaFit input files (“script” files) are always installed with the program, in the `examples` directory.

The distributed `examples` directory contains many more illustrative examples than are those described here. A very good approach to getting familiar with the program is to work through the entire set of distributed examples.

“Slow, tight” enzyme inhibition kinetics

The enzyme 5α -ketosteroid reductase is involved in the enlargement of the prostate gland known as benign prostatic hyperplasia. The enzyme is inhibited by a certain chemical analog of testosterone. We [3] studied the “slow, tight” kinetics of this inhibition. It is called “slow” inhibition, because the binding of the inhibitor to the enzyme is not instantaneous under the particular conditions of the *in vitro* assay we used. It is also called “tight” inhibition, because the concentrations of the enzyme

and the inhibitor are comparable in the experiment. The experimental data are shown in *Figure 1.1*.

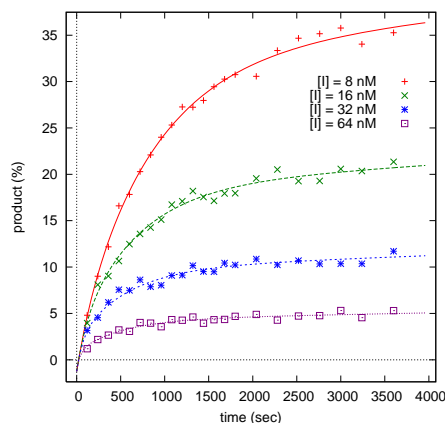
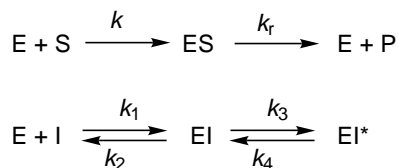


Fig. 1.1 “Slow, tight” inhibition of 5α -ketosteroid reductase, using testosterone as substrate. The appearance of the reaction product (dihydrotestosterone) over time was monitored by HPLC at different concentrations of the inhibitor indicated in the inset. See ref. [3] for details.



Scheme 1.1

The mathematical models for “slow, tight” enzyme inhibition kinetics are relatively complex and, in general, consists of sets of simultaneous differential equations [4]. Such differential equations must be solved numerically (iteratively), as opposed to by explicit algebraic methods. For example, the mathematical model for a two-step inhibition mechanism shown in *Scheme 1.1* is a system of seven first-order Ordinary Differential Equations (ODEs) for the concentrations of reactants.

Importantly, the biochemical researcher employing DynaFit for data analysis does *not* have to specify the differential equations derived from the reaction mechanism in *Scheme 1.1*. Instead, DynaFit automatically derived the mathematical model

for “slow, tight” inhibition of 5 α -ketosteroid reductase from the following symbolic input¹:

Listing 1.1

```
[task]

data = progress
task = fit
model = two-step-rapid ?

[mechanism]

E + S <==> ES      :          kaS      kdS
ES ---> E + P      :          kdP
E + I <==> EI      :          kaI      kdI
EI <==> EJ         :          kij      kji

[constants]

kaS = 0.1
kdS = 1 ?
kdP = 1 ?
kaI = 0.1          ; fixed!
kdI = 1 ?
kij = 0.001 ?
kji = 0.0001 ?
```

When DynaFit was given the symbolic description of the reaction mechanism and the available experimental data, as shown in *Listing 1.1*, it automatically derived the mathematical model, and then fit the data to this model to extract the best-fit values of rate constants for the particular inhibitor. This allowed the full biochemical characterization of the inhibitor and a comparison of the relevant rate constants with other similar steroid reductase inhibitors.

Competitive ligand displacement

Cyclosporin A (CsA) is a naturally-occurring immunosuppressant frequently used to treat organ-rejection after transplantations. We [5] have found that CsA binds to various proteins including cyclophilin, Hsp70, and actin. One task before us was the determine the strength of the binding interaction, that is, measure the equilibrium dissociation constant of the CsA–protein complexes. The difficulty was that CsA is impossible to observe by the usual physicochemical methods. For example, CsA itself is invisible in fluorescent spectroscopy.

¹ See directory `./examples/5alpha-reductase/` distributed with the program.

To study the binding of such a spectroscopically invisible molecule, we designed a *competitive binding assay* [6]. First we prepared a strongly fluorescent chemical analog of CsA, and easily observed its binding to various proteins. Subsequently, we added CsA itself to the mixture and observed how CsA and its fluorescent analog competed for protein binding. In a way the fluorescent analog acted as a “reporter molecule”. Representative experimental data are shown in *Figure 1.2*.

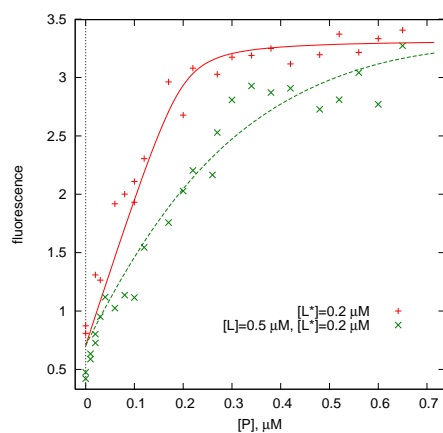


Fig. 1.2 The binding of cyclophilin protein (P) to a fluorescent analog of CsA (L*) alone or to a mixture of this reporter molecule plus a spectroscopically invisible CsA itself (L). The spectroscopically “dark” or “silent” CsA molecule displaced the fluorescent analog to a certain degree. See ref. [6] for details.

The mathematical model for the ligand displacement assays is surprisingly complex, given that only three interacting molecules are involved (two ligands competing for binding to the same protein target). Wang [7] derived an algebraic equation to analyze experimental data similar to those shown in *Figure 1.2*. When printed, Wang’s equation would occupy at least half this page. In contrast, when DynaFit is used for the data analysis, the binding mechanism can be specified in a symbolic form shown in *Listing 1.2*.

Listing 1.2

```
[task]

data = equilibria
task = fit

[mechanism]

; P ... protein (cyclophilin)
; L* ... fluorescently labeled ligand with a known Kd*
; L ... competing ligand (cyclosporin) with an unknown Kd
```



```

P + L* <==> PL*   :   Kd*   dissoc
P + L  <==> PL     :   Kd     dissoc

[constants]

Kd* = 0.005
Kd  = 0.02 ?

```

When DynaFit is presented with the reaction mechanism shown in *Listing 1.2* and with the experimental data shown in *Figure 1.2*, it automatically derives the appropriate mathematical model and performs the fit to extract the most probable values of the equilibrium constants K_d and K_d^* . This is much more convenient than having to specify an exceedingly complicated algebraic equation [7].

NOTE

Traditional algebraic methods to analyze competitive binding data impose severe and – as it turns out – completely unnecessary restrictions on the experimental design.

A typical example is presented by Goodrich & Kugel in their laboratory handbook [8, p. 34]. According to these authors' recipe, it is required that one of the two competing ligands be always present at concentrations *very much higher* than the corresponding dissociation constant. Depending on which molecular species is directly observed in the experiment, another rather severe requirement (unfortunately not made sufficiently clear in the text) is that the target protein must be present at a *negligibly small* concentration when compared to both competing ligands.

In contrast, in *Figure 1.2* we see that no such restrictions are at all necessary when the mathematical model – automatically derived by DynaFit – is sufficiently general. When using DynaFit to analyze the competitive binding data, there are *no restrictions* on concentrations of any species (a) relative to the equilibrium constants and (b) relative to each other.

Initial rate enzyme kinetics

Our final biochemical example is from initial rate enzyme kinetics, where we observe the rate of product formation at the very beginning of the enzyme assay in dependence on the initial concentration of various reactants (substrates, inhibitors, activators, or the enzyme itself).

The Lethal Factor (LF) protease from *Bacillus anthracis* is involved in the pathogenesis of anthrax disease. The protease is inhibited by aminoglycosides such as Neomycin B (*Figure 1.3*). Fridman *et al.* [9] used a semiquantitative (graphical) assessment of a Lineweaver-Burk plot to conclude that Neomycin B is a “competitive” inhibitor of LF.

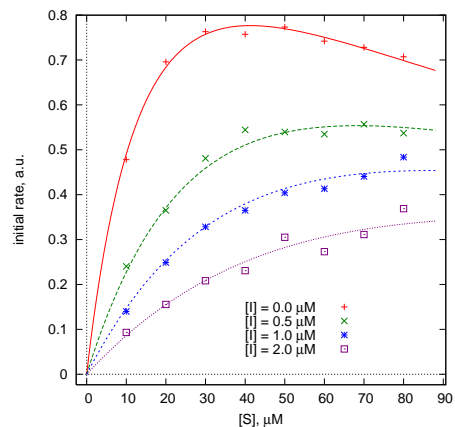


Fig. 1.3 Inhibition of the Lethal Factor protease from *B. anthracis*. Initial reaction rate of LF assays in the presence of increasing concentrations of Neomycin B (“I”). See ref. [10] for details.

We [10] have used DynaFit to determine the inhibition mechanism. To accomplish that task, DynaFit was presented with an input script partially shown in *Listing 1.3*.

Listing 1.3

```
[task]

task = fit
data = rates
approx = rapid-equilibrium
model = Mixed type noncompetitive ?

[mechanism]

E + S <====> ES      : Ks   dissociation
ES + S <====> ESS    : Ks2  dissociation
ES ---> E + P      : kcat
E + I <====> EI     : Ki   dissociation
ES + I <====> ESI    : Kis  dissociation

...

[task]

task = fit
data = rates
approx = rapid-equilibrium
model = Competitive ?

[mechanism]
```

```

    E + S <====> ES      : Ks   dissoc
    ES + S <====> ESS     : Ks2  dissoc
        ES  ----> E + P   : kcat
    E + I <====> EI      : Ki   dissoc

...

[task]

task   = fit
data   = rates
approx = rapid-equilibrium
model  = Uncompetitive ?

[mechanism]

    E + S <====> ES      : Ks   dissoc
    ES + S <====> ESS     : Ks2  dissoc
        ES  ----> E + P   : kcat
    ES + I <====> ESI    : Kis  dissoc

...

[task]

task   = fit
data   = rates
approx = rapid-equilibrium
model  = Noncompetitive ?

[mechanism]

    E + S <====> ES      : Ks   dissoc
    ES + S <====> ESS     : Ks2  dissoc
        ES  ----> E + P   : kcat
    E + I <====> EI      : Ki   dissoc ; only one Ki
    ES + I <====> ESI    : Ki   dissoc ; only one Ki

```

The script fragment in *Listing 1.3* contains multiple alternate models or mechanisms. The model names are completely arbitrary. Importantly, those fitting models that are marked by the question mark will be compared for suitability using several model-discrimination criteria, including the second-order Akaike Information Criterion (AIC) [11, 12, 13]. Given this notation, DynaFit makes an attempt to identify, if possible, the “best model” out of several candidates.

```

    model = Mixed type noncompetitive ?
...
    model = Competitive ?

```

```
...  
model = Uncompetitive ?  
...  
model = Noncompetitive ?
```

In this particular case, the mixed type noncompetitive mechanism was chosen as the most plausible model [10]. This contradicts an earlier report [9], which used a less advanced method of kinetic analysis, apparently relying on a visual examination of the double-reciprocal Lineweaver-Burk plot.

Fridman *et al.* [9] do not specify how exactly they arrived at the conclusion that “all compounds tested were found to be competitive inhibitors.” They do state that the inhibition constant values “were estimated from double-reciprocal plots”, so we can speculate that these authors used double-reciprocal (Lineweaver-Burk) plots to determine the inhibition mechanism also. One serious problem with this simplistic approach is made clear upon visual inspection of *Figure 1.3*. The plot of substrate concentration vs. initial rate in the absence of inhibitor shows a prominent maximum on the saturation curve (“substrate inhibition”). Thus, the Lineweaver-Burk plot necessarily must be *nonlinear* and therefore unsuitable for the determination of the inhibition mechanism by graphical methods.[14]

1.1.2 Examples from chemistry, pharmacology, and other areas

DynaFit [1, 2] has been used not only in biochemistry and molecular biology, but also in biophysics (protein folding [15]), organic chemistry (organic reaction mechanisms [16]), physical chemistry (guest–host complexation equilibria [17]), food chemistry (fermentation dynamics [18]), chemical engineering (bio-reactor design [19]), environmental science (bio-sensors for heavy metals [20]), and in other areas. The common link is the ability to cast the mechanism for the process under investigation in terms of the *mass action law*.

For example, *Figure 1.4* shows the experimental data on the thermal decomposition of the organic chemical α -pinene, which appeared in several classic statistical textbooks [21, 22]. The postulated reaction mechanism for the thermolysis of α -pinene is shown in *Figure 1.5*.

The goal of the data analysis is to extract the most probable values of rate constants k_1 through k_5 from the experimental data shown in *Figure 1.4*. The mathematical model is formulated as a set of simultaneous first-order Ordinary Differential Equations. Importantly, the user of DynaFit does not have to specify those equations. Instead, the software automatically derives the requisite mathematical model from symbolic input shown in *Listing 1.4*.²

Listing 1.4

² See distributed example files in the directory `./examples/chemistry/pinene`.

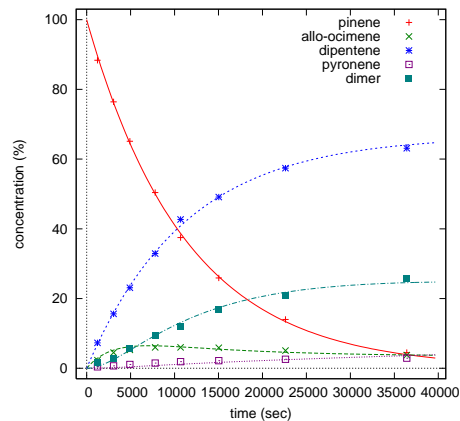


Fig. 1.4 Experimental data for the thermolysis kinetics of α -pinene. See ref. [21, p. 551] for details.

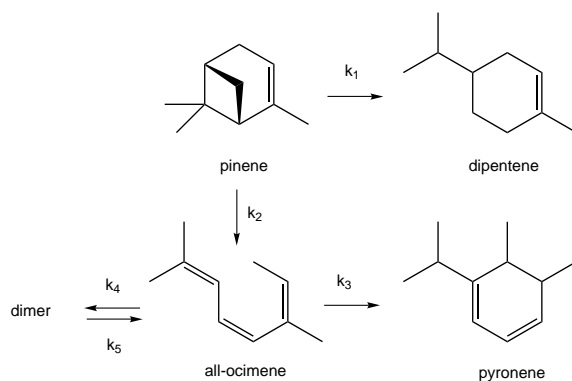


Fig. 1.5 Postulated molecular mechanism for the thermolysis of α -pinene.

[task]

```
data = progress
task = fit
model = pinene
```

[mechanism]

```
pinene ---> dipentene      : k1
pinene ---> allo-ocimene   : k2
allo-ocimene ---> pyronene : k3
allo-ocimene <==> dimer    : k4 k5
```

[constants]

```
k1 = 0.0001 ?, k2 = 0.0001 ?, k3 = 0.0001 ?
```

```

k4 = 0.0001 ?, k5 = 0.0001 ?

[concentrations]

pinene = 100

```

1.1.3 Simulations

Aside from data fitting, DynaFit can be used for heuristic simulations. Both the format and contents of the DynaFit input files stay exactly the same, except a single keyword change. For data fitting, use the `fit` keyword:

```

[task]
task = fit
data = ...

```

For simulations, use the `simulate` keyword instead:

```

[task]
task = simulate
data = ...

```

Oscillatory kinetics under product inhibition

Prigogine *et al.* [23] predicted that oscillations should be observed in a flow-through enzyme system, if and when the reaction product inhibits the enzyme (“negative feedback” inhibition) *and* if the inhibitory activity of the product is sufficiently high. The DynaFit script file partially shown in *Listing 1.5* was used to verify this prediction.

Listing 1.5

```

[task]

data = progress
task = simulate

[mechanism]

S1 + E <===> S1.E      :   k      ks1

```

```

      S1.E ----> E + S2      :   kr1
      S2 + E <==== S2.E      :   k       ks2

      ----> S1                :   v1
      S2 ---->                :   v2

[constants]

k      = 10
ks1    = 10
kr1    = 10
ks2    = 0.05

v1     = 1, v2 = 1

[concentrations]

E      = 1
S1     = 10, S2 = 10

```

In *Listing 1.5*, S_1 is the substrate and S_2 is the product of an enzyme reaction catalyzed by enzyme E . The initial concentrations of both S_1 and S_2 are 10 arbitrary concentration units (e.g., micromoles per liter), as specified in the [concentration] section. The substrate is supplied into the open system at a constant rate v_1 , and the product is removed at the same rate. At time zero, the enzyme is introduced at a unit concentration ($E = 1$).

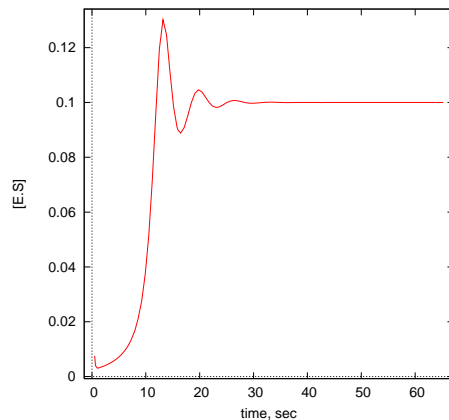


Fig. 1.6 Damped oscillations in a single-enzyme, single-substrate system induced by strong product inhibition as predicted by Prigogine *et al.* [23].

Figure 1.6 shows the transient concentration of enzyme–substrate complex $E.S_1$ simulated by DynaFit. The simulated concentration displays several periods of damped oscillations, as predicted by the general theory [23]. Importantly, the oscillations disappear if the dissociation rate constant k_{s2} in the above input file is made sufficiently high:

```
[constants]
    k      = 10
    ks1    = 10
    kr1    = 10
    ks2    = 1      ; ... no oscillations!
    ...
```

Thus, the oscillatory system exhibits a threshold behavior. It should be noted that oscillations can only be seen if the simulated system is *open*, in the sense that there is a constant flow of mass. The substrate S_1 must be continuously added and the product S_2 must be continuously removed. Only under those circumstances, non-equilibrium thermodynamics can lead to oscillatory behavior.

1.1.4 Algebraic fitting models

In addition to the fitting models represented by systems of differential equations (used in mechanistic analysis of reaction time course) or by the systems of simultaneous nonlinear algebraic equations (used for modeling complex equilibria), DynaFit offers several types of algebraic models.

- **Arbitrary algebraic equations.** The user can specify an arbitrary algebraic equation, in a way that is similar to other data-analysis software packages.
- **Single or multiple exponentials.** As a special case of an algebraic model, DynaFit encodes the single- or multiple-exponential model according to the Legendre polynomial method [24, 25].
- **Piecewise polynomial fit.** This algorithm allows the fitting of data sets containing many data points (typically a densely spaced progress curves) to series of straight lines..

1.2 Installation

The DynaFit package can be downloaded from <http://www.biokin.com/dynafit>. The installation procedure consists of extracting a ZIP archive into the appropriate directory.

1. Point your web browser to <http://www.biokin.com/dynafit>.
2. Navigate to download the appropriate version of DynaFit (MS Win or Mac OS X).
3. Extract the ZIP archive into any directory where you have full (read and write) user permissions.

1.2.1 Microsoft Windows

There is no “Windows Installer” program to run. This avoids issues with user vs. administrator permissions, especially in corporate environments, where end users frequently are not permitted to install software packages.

Extracting the ZIP archive file `dynafit4-win.zip` will create the directory named DynaFit4. It is very important to extract the the ZIP package into a directory where the user has full (read and write) permissions. On Windows XP this is typically the directory called My Documents. On Windows 8 the default destination is typically Libraries / Documents.

The important point is that during DynaFit execution the software will automatically create files on the disk. Therefore DynaFit needs to be executed either with “Run as Administrator”, or it needs to run in a disk area where the user has automatic file creation privileges.

1.2.2 Mac OS X

Extracting the downloaded ZIP archive file `dynafit4-macosx.zip` will create the directory named DynaFit4. There are two executable files, DynaFitC and DynaFit. DynaFitC is the console or command line version, for use in automation. DynaFit is the GUI version, for general use.

1.2.3 Obtaining the license file

...

1.2.4 Installing the license file

When freshly downloaded from the Internet, DynaFit is enabled to process only the example files distributed with it. In order to process any arbitrary data files, it is necessary to install a valid license file. Indeed when DynaFit is first started, imme-

diately after downloading the program from the Internet, the welcome screen will appear similar to what is shown in Figure 1.7. Please note the notification “License file could not be opened”.

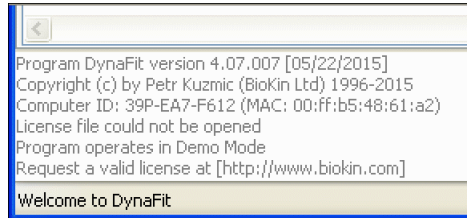


Fig. 1.7 The DynaFit status message area (lower left corner of the main window) on first execution of DynaFit, before the license file is installed. Your “Computer ID” information will vary.

To install the license file, select menu item **File — Install license...** and navigate to the text file `license.txt` received from the BioKin.COM server (see Figure 1.8). After selecting the license file, distributed as an email attachment, it is very important to exit DynaFit and restart it. After restarting the program, the welcome screen should look similar to what is shown in Figure 1.9

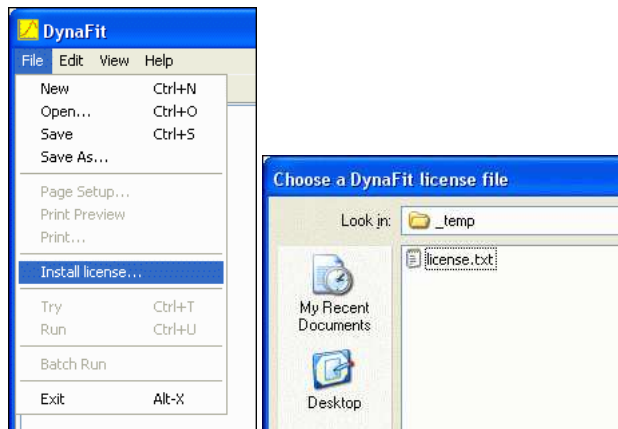


Fig. 1.8 Installing the license file. After selecting the license file it is important to exit DynaFit and restart it.

Academic licenses

DynaFit users affiliated with educational and non-profit institutions can obtain license files that are not tied to any particular computer and do not require activation

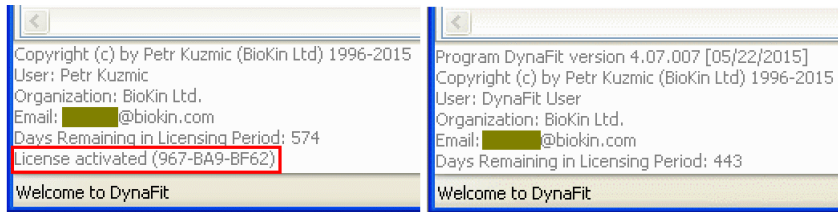


Fig. 1.9 The DynaFit welcome screen after the license file has been installed. Your Computer ID information will vary. *Right-Hand Panel:* Academic license files do not utilize the Computer ID.

(see below). The BioKin.COM server will auto-generate, and send via automated email, an academic license file to any user affiliated with educational institutions. To request an academic license file visit www.biokin.com and navigate to the academic license request form.

A full text of an example academic license is listed immediately below. Please note the the special encoding `ComputerID = "n/a"`. This signifies that the educational license file can be installed on any number of computers simultaneously, and is not tied to any particular machine.

```

BioKin-License-Begin
Program           = "DynaFit"
Version          = "4"
User             = "DynaFit User"
Email            = "dynafit.user@brandeis.edu"
Organization     = "Brandeis University"
Department       = "n/a"
ComputerID       = "n/a"
LicenseType      = "Academic"
Seat             = "1"
Expiration       = "Wed Jan 13 12:00:00 2016"
LicenseCode      = "f30a7be066903e58655d9240323a4ed4966b3b036c435b91"
BioKin-License-End

```

Commercial licenses: Computer ID and activation

DynaFit users affiliated with for-profit commercial ventures need to obtain an “activated” license file tied to a particular computer. To generate the requisite Computer ID, select menu **View — Computer ID...**, as is shown in Figure 1.10. The information displayed in the Input window, in particular the Computer ID information highlighted by the red rectangle, needs to be submitted via email to licensing2@biokin.com. The activated license file will be supplied by return email.

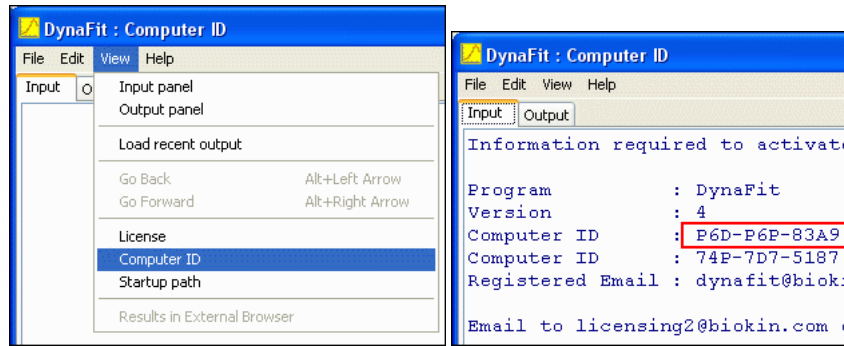


Fig. 1.10 Generating the Computer ID required for commercial license activation.

1.3 Example problems

DynaFit is distributed with a collection of typical example problems contained in the directory `examples`. It is very highly recommended to begin working with DynaFit by first examining these example problems. The DynaFit script files associated with the distributed examples can be easily modified to suit each new user's particular research needs.

1.4 Support

DynaFit users are encouraged to contact the program's author, Dr. Petr Kuzmic, with any questions about setting up the data analysis, or about correctly interpreting the results. The most recent contact information is always available at the BioKin website, <http://www.biokin.com>. Dr. Kuzmic is occasionally available to help craft journal publications utilizing DynaFit in manner that can withstand even the most rigorous peer review. A number of research publications using the DynaFit software came to life in this manner.

Chapter 2

User Interface

The DynaFit software can be used in one two distinct application modes, utilizing two distinct user interfaces:

1. Interactive mode, using the graphical user interface (GUI).
2. Batch mode, using the command-line interface.

2.1 Graphical User Interface (GUI)

In its simplicity and sparseness, the graphical user interface (GUI) to DynaFit might look unfamiliar to users of most Windows or Mac OS X software. When DynaFit starts up, it presents a very simple text editor window shown in [Figure 2.1](#)

2.1.1 *File menu*

The File menu serves principally to manipulate the DynaFit script files. Aside from the usual **New**, **Open**, **Save** and **Save As...** menu item, the two most important selections are **Try** and **Run** (see [Figure 2.2](#)).

The **File | Try** menu selection will perform a simulation utilizing the initial estimates of model parameters, such as rate constants, initial reactant concentrations, and molar response coefficients. In the Output tab, the simulated model curves will be overlaid on the experimental data. The purpose is to make sure that the initial estimates of model parameters are sufficiently close to the true values.

After the initial estimates of model parameters are adjusted in the Input tab, such that the simulated model curve and the experimental data are sufficiently close, the **File | Run** menu selection will perform the actual data analysis. Importantly, the **File | Try** and **File | Run** menu items are available when the focus is on the Input

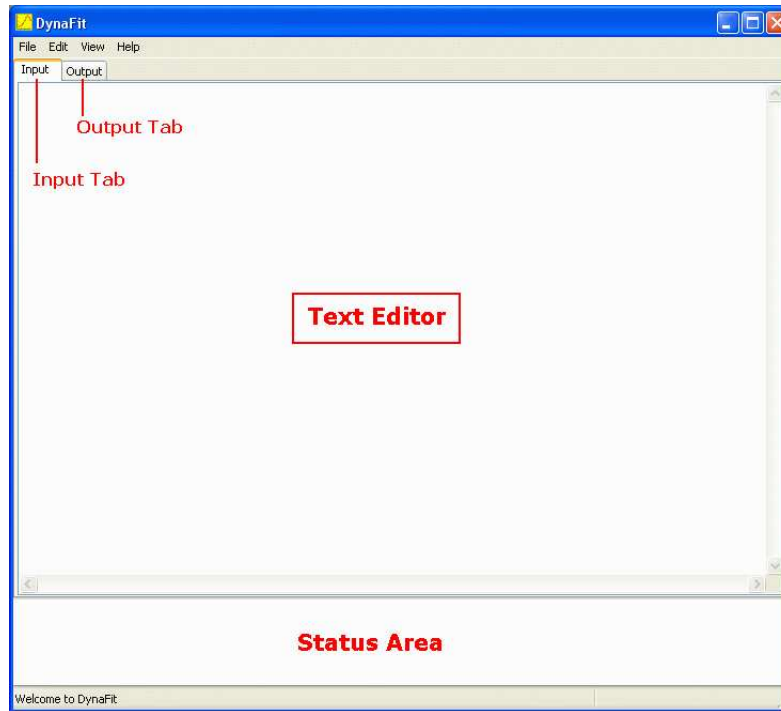


Fig. 2.1 Graphical user interface (GUI) to DynaFit.

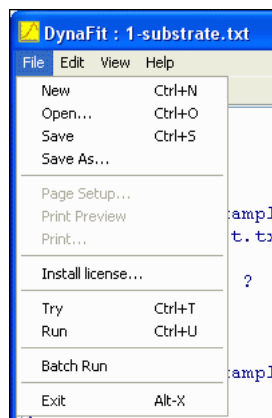


Fig. 2.2 Graphical user interface (GUI) to DynaFit: File menu

window. Therefore, before “trying” or “running” the model, one must click on the Input tab in the main window or select **View | Input panel** from the main menu.

How close is “sufficiently close”, in terms of the overlay of the experimental data and the initial model estimate, depends on each particular case. In the most

unfavorable cases the microscopic rate constants have to be within one order of magnitude of their “true” values (which, of course, are not known in advance).

The menu item **Batch Run** serves to execute multiple script files in a single run. For example, to execute in a single run the scripts `substrate.txt` and `models.txt` located in the 5α -ketosteroid reductase directory, one can list the script file names in the input window (Figure 2.3) and then select menu **File | Batch Run**.

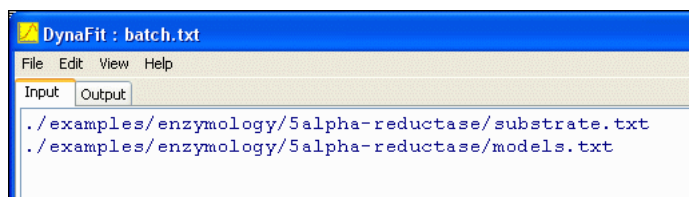


Fig. 2.3 Input text for executing multiple script files in a single run.

The menu item **File | Install license...** is used to install the license file, as is explained in section 1.2.4.

2.1.2 Edit menu

The **Edit** menu items are used to edit the script file located in the Input tab. The menu contains the conventional **Cut**, **Copy**, **Paste** and **Undo** items, along with the corresponding shortcuts (for example **Ctrl+Z** for undoing the most recent editing change).

Probably the most useful menu item is **Edit | Insert...** and its corresponding submenus. Selecting any of the choices in the **Insert...** submenu tree will insert the corresponding text into the text editor (Figure 2.4).

Importantly, *all* existing DynaFit keywords are gathered together in the **Insert...** menu tree. This means that the user does not have to remember any keywords and, if necessary, can interactively compile the input script almost exclusively by inserting text by utilizing the **Edit** menu.

2.1.3 View menu

The **View** menu can be used to navigate between the Input and Output panels in the main window (see the two top menu entries in Figure 2.5). In the Input mode, selecting **View | License** will insert the text of the current license file into the text edit area. This is useful mostly to check the license expiration date.

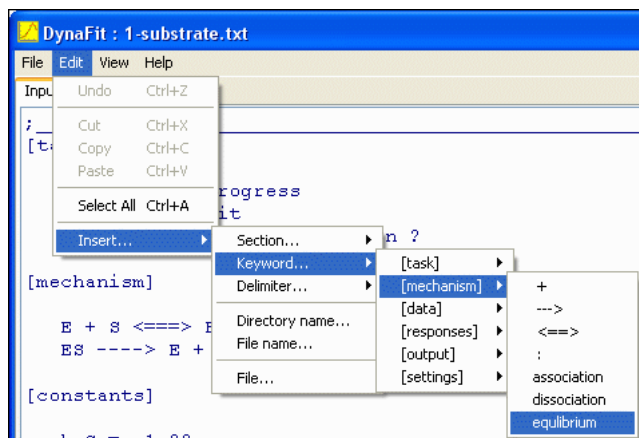


Fig. 2.4 Inserting recognized keywords and section names into the Input editable area.

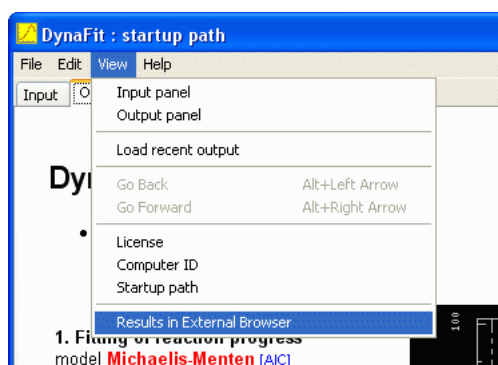


Fig. 2.5 Triggering the default web browser to view results of fit.

Selecting **View | Computer ID** will insert into the editable Input area the Computer ID that is required to activate a commercial license, as is explained in section 1.2.4. This information is not relevant for academic, educational licenses.

Selecting **View | Startup Path** will insert into the editable Input area the full path name of the DynaFit startup directory. This can be useful in compiling the [output] or [data] sections of the input script file.

Probably the most useful item in the **View** menu is the **Results in External Browser** selection (see Figure 2.5). This selection will trigger the default web browser installed on the user's computer and display within it the most recent DynaFit output. The main benefit is that full fledged web browsers such as Firefox or Opera will display the DynaFit-generated output with richer formatting. The very simple browser built into DynaFit has very limited capacity to process CSS (Cascading Style Sheets) and other advanced HTML features.

2.2 Command-line interface

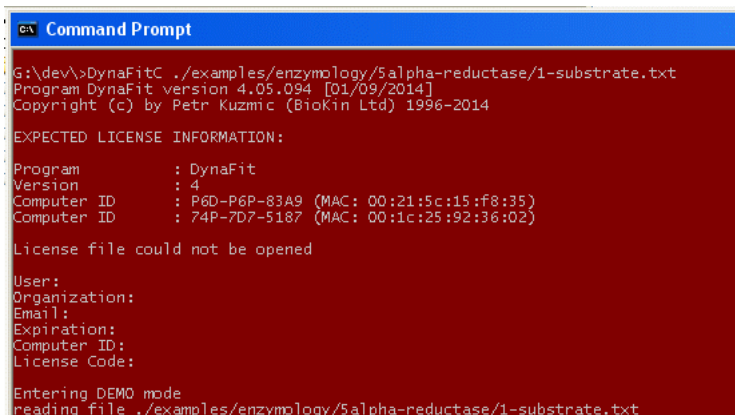
DynaFit is distributed as two separate executable files. The DynaFit.exe (MS Windows) or DynaFit.app (Mac OS X) binary file presents the graphical user interface described in section 2.1. In contrast, the DynaFitC.exe (MS Windows) or DynaFitC (Mac OS X) binary files can be used from the command line. This is most useful in automating routine tasks.

2.2.1 Microsoft Windows

The command line syntax is

```
DynaFitC SCRIPTFILENAME
```

where SCRIPTFILENAME stands for the name of the DynaFit script file to be processed. See Figure 2.6 for an example.



```
Command Prompt
G:\dev\>DynaFitC ./examples/enzymology/5alpha-reductase/1-substrate.txt
Program DynaFit version 4.05.094 [01/09/2014]
Copyright (c) by Petr Kuzmic (BioKin Ltd) 1996-2014

EXPECTED LICENSE INFORMATION:
Program       : DynaFit
Version      : 4
Computer ID   : P6D-P6P-83A9 (MAC: 00:21:5c:15:f8:35)
Computer ID   : 74P-7D7-5187 (MAC: 00:1c:25:92:36:02)

License file could not be opened

User:
Organization:
Email:
Expiration:
Computer ID:
License Code:

Entering DEMO mode
reading file ./examples/enzymology/5alpha-reductase/1-substrate.txt
```

Fig. 2.6 DynaFit command line interface under MS Windows.

2.2.2 Mac OS X

The command line syntax is

```
./DynaFitC SCRIPTFILENAME
```

where SCRIPTFILENAME again stands for the name of the DynaFit script file to be processed. See Figure 2.7 for an example.

```

Terminal Shell Edit View Window Help
Terminal — bash — 150x53
MacBook1:DynaFit4 pkuzmic$ ./DynaFitC ./examples/enzymology/5alpha-reductase/1-substrate.txt
Program DynaFit version 4.05.095 [01/18/2014]
Copyright (c) by Petr Kuzmic (BioKin Ltd) 1996-2014

EXPECTED LICENSE INFORMATION:

Program       : DynaFit
Version      : 4
Computer ID   : FE8-1ED-9E79 (MAC: 00:1b:63:ba:2b:4c)

License file could not be opened

User:
Organization:
Email:
Expiration:
Computer ID:
License Code:

Entering DEMO mode
reading file ./examples/enzymology/5alpha-reductase/1-substrate.txt

```

Fig. 2.7 DynaFit command line interface under Mac OS X.

2.2.3 Batch script automation

The command line interface is very convenient in automating multiple consecutive runs. This is most useful in performing very lengthy calculations, such as those involving the Differential Evolution algorithm [26].

As an example, let us assume that we wish to execute three separate script files located in the directory `./examples/enzymology/5alpha-reductase`. To arrange for this we could create a batch file called `three-scripts.bat` in the same directory. The listing of such a batch file is given below:

```

cd ..\..\..\
DynaFitC ./examples/enzymology/5alpha-reductase/substrate.txt
DynaFitC ./examples/enzymology/5alpha-reductase/substrate-ci-kaS.txt
DynaFitC ./examples/enzymology/5alpha-reductase/substrate-ci-kdS.txt

pause

```

In the above listing, the first line (**cd**) changes the current directory into the main DynaFit working directory, where the binary file `DynaFitC.exe` is located. The subsequent three lines trigger DynaFit execution, processing the three substrate-related scripts one at a time. The final line (**pause**) prompts the user to dismiss the console window.

The command-line automation scripts are particularly useful in performing many repeated kinetic analyses in the context of high-throughput screening, in the pharmaceutical or biotechnology industries. An important advantage is that all output files generated by DynaFit in the unsupervised batch mode are ASCII text files. This makes it possible to create completely integrated scripting systems, using Perl or Python scripting languages.

Chapter 3

Input files

All input data utilized by DynaFit are plain text (ASCII) files. There are three kinds of such input files:

1. Experimental data files
2. DynaFit script files
3. DynaFit initialization (settings) files

It is required that all DynaFit script files are named with the file extension TXT. This is also highly recommended for all DynaFit data files, although other file extensions such as DAT can also be used. DynaFit initialization files typically have the file extension TXT although the extension INI can also be used.

3.1 Data files

DynaFit can be used either to fit existing experimental data, or to simulate artificial data. Data fitting is probably the more important, and also more frequent, use of the software. In order to fit existing experimental data, we must first format the data properly. This subsection explains how the formatting is to be accomplished.

3.1.1 ASCII text format

All experimental data for DynaFit processing are formatted as ASCII text (“plain text”). DynaFit cannot directly process any other files, such as Microsoft Excel files. Therefore, to prepare a data set for processing with DynaFit, the data set usually must be exported or copied from another source.

Depending on the particular software application, in which your data are preprocessed or originally generated, different procedures might be necessary to convert

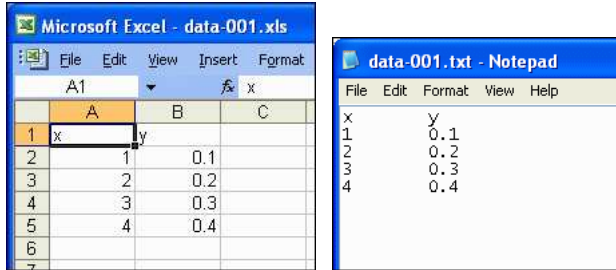


Fig. 3.1 Excel data file and a corresponding ASCII data file.

or export the data into ASCII format. For example, to produce ASCII data from an Excel or OpenOffice file, we have two different choices:

- Export a particular data sheet from the Excel or OpenOffice file.
- Copy and paste the data through the system clipboard.

Exporting ASCII data files

Figure 3.2 illustrates the first method of generating data files readable by DynaFit, namely, exporting an entire data sheet in ASCII text format.

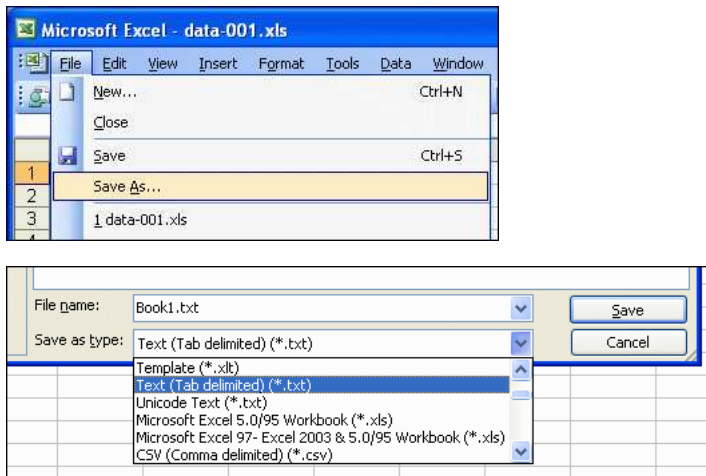


Fig. 3.2 Converting Excel data file into an ASCII data file.

1. Prepare your data in a spreadsheet program such as Excel.
2. Select an appropriate Excel sheet to be exported.
3. Choose from the main menu **File | Save As ...**

4. Choose from the drop-down menu **Save as Type: | Text (Tab delimited) (*.txt)**.
5. Type a file name for the exported file, ending with the file extension *.TXT.
6. Click [OK] so save only the single selected sheet.

Probably the best way to ensure that a given data file is readable by DynaFit is to open the file in a plain-text editor, such as Microsoft Notepad. Double-click on the file name in Windows Explorer, or in another file-system navigator. If the data file opens successfully and is readable, the ASCII export succeeded.

Copying ASCII text through system clipboard

Alternately, one could use the copy-and-paste procedure, as follows:

1. Open an Excel or OpenOffice file containing the experimental data.
2. Create a new (blank) text file, for example, in Notepad.
3. Highlight the data in Excel, select **Edit | Copy**.
4. Place the cursor into the blank Notepad file, select **Edit | Paste**.
5. Save the newly created Notepad file.

ASCII text representing experimental data can be pasted not only into external files, using Notepad and similar text editors, but also directly into DynaFit script files, explained elsewhere in this manual.

3.1.2 Required format of experimental data sets

The ASCII text files holding experimental data must be formatted as follows:

- Numerical items on each line must be separated by blank spaces or tabs (comma-separated data files are not recognized).
- The first column always holds the independent variable, such as time or concentration of the variable reactant.
- Multiple independent variables are allowed. For example, in an enzyme-inhibition study, the first column could contain the substrate concentration and the second column could contain the inhibitor concentration.
- Following the independent variable value(s), DynaFit expects the numerical value of the experimentally observed quantity.
- Optionally, each line may end with a numerical value indicating the experimental measurement error, such as the standard deviation from replicates. This feature is only relevant for a single independent variable.
- Each line must contain the same number of numerical items.
- If a line of text does not begin with a recognizable number, the entire line will be ignored as a “comment” or a header.
- Blank lines are ignored.

3.1.3 “Sheet” format

Data files for DynaFit can be optionally arranged in “spreadsheet” format, such that the first column contains the independent variable (for example, time) and the subsequent columns contain data traces from multiple separate experiments. The relevant keywords in the DynaFit scripting language are `sheet` and `column`. For details see the separate “Scripting Manual”.

3.1.4 Comment and header lines

The listing below shows the full text of a typical DynaFit data file. The first six lines are header lines, three of which are blank lines included for readability.

```
Rate of an enzyme reaction vs. the concentration of substrate.  
Data presumably conform to the Michaelis-Menten kinetic model.  
From: Bates & Watts (1988) 'Nonlinear Regression', page 236.
```

```
conc. rate
```

```
0.02 76  
0.02 47  
0.06 97  
0.06 107  
0.11 123  
0.11 139  
0.22 159  
0.22 152  
0.56 191  
0.56 201  
1.10 207  
1.10 200
```

A very good use of header lines is to keep meta-data together with the experimental data. Such meta-data typically contain detailed information about the experimental conditions (such as concentrations of reactants) associated with each data set. Detailed concentration information, including the approximate molar concentration of enzymes in biochemical assays, is critically important for a successful analysis in DynaFit.

If a line of text does not begin with a recognizable number, the entire line will be ignored. This can be used to temporarily or permanently exclude grossly outlying data points from analysis, while at the same time keeping a readable record of such exclusion. In the data file fragment below, the fourth line (holding the fourth data point) was excluded by starting the line with the asterisk (*).

```
0.11    139
0.22    159
0.22    152
***0.56 123456 ... this data point was excluded from analysis
0.56    201
1.10    207
1.10    200
```

Because the text `***` does not represent a valid number, the entire line is ignored by DynaFit. Excluding experimental data points from analysis must be used only in very exceptional circumstances, when the experimenter has strong evidence that something went truly awry in the particular measurement (equipment failure or human error). Otherwise “cherry-picking” experimental data can lead to undesirable bias in the final results.

After DynaFit successfully reads the appropriate number of numerical items from each line of input, the rest of the text is ignored. This could be used to make informative notation about particular data points.

```
0.11    139    only one replicate collected
0.22    159
0.22    152
0.56    191    careful: possible instrument failure
0.56    201
1.10    207
1.10    200
```

3.1.5 *Number of independent variables*

The main use of DynaFit is for the analysis of one the following three types of experiments in chemistry, biochemistry, or experimental biology:

1. **Reaction progress**

A mixture of reactants is followed over time, while recording the value of some physically observable quantity (fluorescence, absorbance, chromatographic peak area).

2. **Equilibrium composition**

A mixture of reactants is first brought to full equilibrium, when the composition no longer changes. Then the value of some physically observable quantity is recorded.

3. Initial reaction rates (enzymology)

The enzyme catalyst is added to a mixture of substrates, inhibitors, or activators. The ensuing change in some physically observable quantity is recorded over time (primary data), but the information actually used in DynaFit analysis is the rate of change in the experimental signal at time zero (“initial rate”).

Progress curve analysis

In the first type of experiment, progress curve analysis, there is only one independent variable – the reaction time. The experimental data will always contain exactly two columns. The first column will contain the readings of time, typically in seconds although other time-measurement units (minutes, hours, or milliseconds) are also acceptable. The second column will contain the readings of the experimentally observable physical quantity (absorbance, fluorescence, and the like).

It is highly recommended to always use seconds (as opposed to minutes, hours, or other units) as the units of time in all DynaFit data files. The reason is that (a) most published rate constants for chemical or biochemical processes are reported in seconds; and (b) the time unit used in the data file must be the same as the time unit used to define rate constants elsewhere in DynaFit input.

Equilibrium binding data

In most equilibrium binding experiments, the concentration of one particular reactant is systematically varied, whereas the concentrations of the remaining reactants are held constant. Under these circumstances, the DynaFit data file will contain only two columns (or three columns, if the optional experimental error is also included). The first column holds the concentration of the variable reactant (for example, the total protein concentration in a protein–DNA binding experiment), and the second column holds the observable experimental signal (for example, fluorescence intensity).

If the concentrations of more than one reactant were systematically varied in a particular experiment, we have two different ways to arrange the data, depending on the pattern of variation. This is best explained on practical examples.

EXAMPLE: Regular grid pattern in experiment design

In a protein–ligand physical binding study, using fluorescence as the detection method, the total concentrations of the protein (P) and the ligand (L) were varied in a grid pattern.

First, the protein concentration was held at $0.1 \mu\text{M}$, and the ligand concentration was varied from $1 \mu\text{M}$ to $16 \mu\text{M}$ stepping by a factor of two (1, 2, 4, 8, $16 \mu\text{M}$).

Subsequently the protein concentration was raised to 0.2 μM , and the ligand concentration was varied exactly as before (1, 2, 4, 8, 16 μM).

The same sequence of binding experiments was performed at protein concentrations 0.4, 0.8, and 1.6 μM . Thus the overall experimental design involves a regular 5×5 grid.

There are two different ways to arrange such regularly laid-out data for DynaFit analysis. The first and strongly preferred method is to group the 25 measurements into groups of five. Each group of data points will be stored in a separate data file. For example:

```
Total P = 0.1 uM (file P01.txt)
L,uM  fluorescence
1      1.23
2      4.56
4      7.89
8      12.3
16     45.6
```

A separate data file would then contain the experimental data collected at 0.2 μM total protein:

```
Total P = 0.2 uM (file P02.txt)
L,uM  fluorescence
1      4.56
2      7.89
4      12.3
8      45.6
16     78.9
```

Similar separate data files would be present for protein concentrations 0.4, 0.8 and 1.6 μM .

Another alternate way to arrange the example data would be to use three columns instead of two. In that case, the first column would contain the ligand (or protein) concentration; the second column would contain the concentration of the other component; and the third column would contain the observed experimental signal:

```
L,uM  P,uM  fluorescence
```

1	0.1	1.23
2	0.1	4.56
4	0.1	7.89
8	0.1	12.3
16	0.1	45.6

1	0.2	4.56
2	0.2	7.89
4	0.2	12.3
8	0.2	45.6
16	0.2	78.9

similarly for remaining 15 data points in the 5 x 5 grid

The multi-column arrangement presents significant difficulties in graphical display of the experimental data and/or the best-fit theoretical model. If the experiment does represent a regular grid (as is the case in our example), it is in principle possible to use a “three-dimensional” plot or a contour plot. However, DynaFit does not have an ability to generate such plots.

Therefore, the multi-column input files should be reserved for only three somewhat exceptional circumstances:

1. continuous serial dilution (“titration”) experiments;
2. Job plot (varied mole-fraction) experiments; and
3. irregular experimental designs (no “grid” pattern in concentrations).

An excerpt from a DynaFit script file for the analysis of an NMR chemical shift protein-ligand titration experiment (raw data supplied by courtesy of Dr. Klára Briknarová, University of Montana) is shown below:

Listing 6

```
[mechanism]
  P + L <==> P.L      :      Kd      dissociation
...
[data]
  variable L, P
  plot      titration
  set      Proton-A
...
[set:Proton-A]

L,mM      P,mM      shift
0.00      0.1250    8.232
0.0332    0.1249    8.219
0.0663    0.1247    8.204
0.1324    0.1245    8.184
```

```

0.2636  0.1240  8.151
0.5230  0.1230  8.140
;1.029  0.1210  8.146
...
similarly for other proton and nitrogen chemical shifts

```

Note that the data block in subsection [set] contains the ligand concentration in the first column, the protein concentration in the second column, and the observed chemical shift in the third column. Such data would normally be difficult to display in the usual two-dimensional plot.

However, because the script also contains the reserved keywords `titration`, `DynaFit`, in this special case, will construct a titration plot. The titration plot takes into account that the same sample of protein (initially 0.125 mM) was being progressively diluted by added ligand. Only the total concentration of the varied ligand is shown on the horizontal axis, whereas the changes in the protein concentration are only implied (see *Fig. 3.3*).

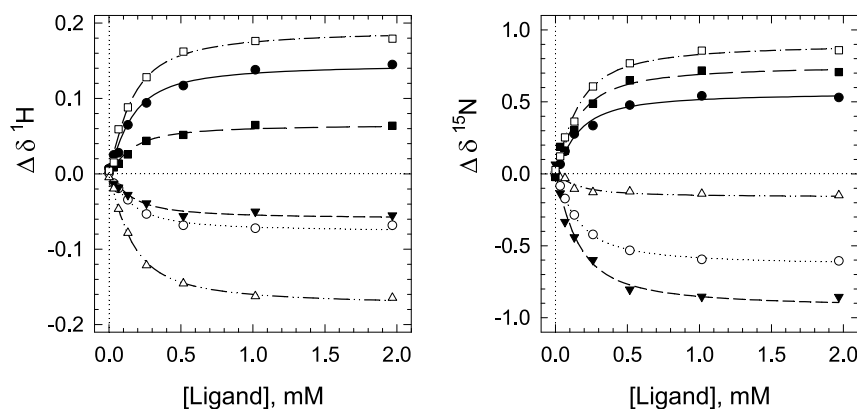


Fig. 3.3 NMR chemical shift titration experiment to determine a protein–ligand dissociation constant (raw data courtesy of Dr. Klára Briknarová, University of Montana). See *Listing 6* for required data format.

NOTE

Results of titration experiments such as those depicted in *Fig. 3.3* should be viewed with a certain amount of scepticism, because the successive experiments are not statistically independent as is required by the theory of least-squares regression.

For example, if any of the serial addition of ligand to the same sample of protein happens to be affected by an unusually large experimental error, due to a human operator or equipment failure, then all subsequent data points also will be similarly affected.

Never-the-less, such serial dilution experiments are used quite frequently, because they obviously conserve a very significant amount of often very precious protein. If every individual sample were prepared independently, as required by theory, the amount of available material might never suffice to complete even a single titration curve.

Aside from highly irregular experimental designs, the only other situation where multi-column data files come into play is the special case of the Job plot experiment. In this type of experiment, the concentrations of exactly two interacting components are varied such that the total amount of both is kept constant. Thus, only the mole fraction, x , of either component is varied, typically over the entire interval from $x = 0$ to $x = 1$. In this special case, DynaFit will automatically convert the three-column data file such that the independent variable will be the mole fraction of the first varied component. This is very convenient for visualizing the final results. Please see the description of keyword `mole-fraction` elsewhere in this manual.

3.1.6 Number format

DynaFit can read only fractional numbers written down by using the decimal-point notation (English, American) as opposed to the decimal-comma notation (German, French, Czech, etc.). No “thousand separator” is allowed. Consider the following five ways to represent the number *one thousand two hundred thirty four point fifty six*:

English	1,234.56
German, Czech	1.234,56
French	1 234,56
DynaFit	1234.56
DynaFit	1.23456e+3

Thus, in the English or U.S. context, comma is the optional “thousand separator” and period is the decimal (radix) separator. However, in the German context and elsewhere the situation is exactly reversed. In the French cultural context, the optional “thousand separator” is the space character, in addition to decimal comma (no decimal point). DynaFit can only interpret numerical values shown in the last two rows in the above table: (a) using a decimal point with no “thousand separator”, or (b) using the scientific notation.

NOTE

Such number-formatting issues have been known to cause difficulties from time to time – when DynaFit users working in various countries export numerical values from spreadsheet programs, while maintaining the default regional settings.

3.2 Script files

DynaFit is driven by case-sensitive script files that resemble small computer programs, in that there is a vocabulary of recognized key words and a required syntax.

The miniature scripting language recognized by DynaFit is sufficiently complex to deserve a separate “Scripting Manual”. In this “Getting Started” manual we will only give a brief overview and a few illustrative examples. As an introductory example let us examine *Listing 3.7*.

Listing 3.7

```
[task]
  data = progress
  task = fit

[mechanism]
  E + S <===> ES      :      kaS      kdS
  ES ----> E + P      :      kdP

[constants]
  kaS = 1, kdS = 10 ?, kdP = 1 ?

[concentrations]
  S = 31, E = 0.05

[responses]
  P = 3.21 ?

[data]
  directory ./examples/5alpha-reductase/data
  sheet     sheet.txt
  column    2
  offset    auto ?

[output]
  directory ./examples/5alpha-reductase/output/substrate

[end]
```

In *Listing 3.7* the lines containing single keywords enclosed in square brackets (for example, [data], [mechanism], [constants], etc.) represent *sections*

of the script. Within each section we find a certain particular *keywords*. For example the section [data] contains the keywords `directory`, `sheet`, `column`, etc.

All recognized keywords and sections can be automatically inserted into the DynaFit script being compiled, by using the menu selection **Edit | Insert...**, as is described in section 2.1.2. Thus the user does not have to remember all the various keywords and section names. The distributed example problems also serve as a useful guide.

3.2.1 Sections

DynaFit scripts are organized into sections. Individual section names are enclosed in square brackets. Each given section has associated with it a certain collection of recognized keywords. For example the section [data] typically contains one or more instances of the the keyword `file`, which specifies the location of data files. The complete list of recognized section names is shown in Table 3.1

section	purpose
[task]	Major task to be performed (e.g., simulation of data fitting) and the type of data to analyze.
[mechanism]	Reaction mechanism for the bio/chemical process under investigation.
[model]	Algebraic model for simulation or data fitting.
[constants]	Numerical values of rate constants or equilibrium constants appearing in the [mechanism] section.
[concentrations]	Total or initial concentration of molecular species appearing in the [mechanism] section.
[responses]	Specific molar responses of molecular species appearing in the [mechanism] section.
[parameters]	Arbitrary model parameters.
[correlations]	List of model parameters to construct Monte-Carlo correlation plot.
[data]	Pointers to experimental data residing in external files or embedded in the script.
[set:...]	Data sets embedded in the script.
[output]	Location of the output directory.
[settings]	Control settings for various internal algorithms.
[end]	End of script file. Any text past this line can serve as optional comments.

Table 3.1 Recognized sections of DynaFit script files.

3.2.2 Keywords

The alphabetical list of all keywords recognized in DynaFit scripts is shown in Table 3.2.

algorithm	extension	plot
approximation	file	poisson
association	fit	power
auto	from	progress
column	generic	proportional
concentration	graph	quadratic
confidence	incubate	rapid-equilibrium
constant	king-altman	rates
data	levenberg-marquardt	response
delay	linear	search
design	logarithmic	set
differential-evolution	mesh	sheet
dilute	model	simulate
directory	mole-fraction	step
dissociation	monte-carlo	task
equilibrate	none	time
equilibria	offset	titration
equilibrium	parameter	to
error	percent	variable
exponential	piecewise-linear	

Table 3.2 Alphabetical list of keywords that can appear in DynaFit script files.

Numerous combinations of keywords can only appear in certain groupings, with a fixed order. For example, let us assume that we wish to describe an experiment, in which the enzyme E (10 nM) and the inhibitor I (10 nM) were pre-incubated for one hour. After the preincubation period the substrate S was added such that the final concentration of substrate were 3 μM . The preincubated mixtures was deleted 1:10 upon the addition of the last component. The appropriate DynaFit notation is in *Listing 3.8*.

Listing 3.8

```

; concentration units: micromolar
; time units: seconds

incubate E = 0.01, I = 0.01, dilute 0.1, time 3600
concentration S = 3

```

In this script file fragment, the keywords `incubate`, `dilute`, and `time` must appear on the same line, in this particular order. This illustrates that although the

vocabulary contains 59 distinct keywords (see Table 3.2), certain groupings of keywords must appear as complete “phrases” in the scripting language. These “phrases” are available for automatic insertion in the script file editor, as is illustrated in Figure 3.4

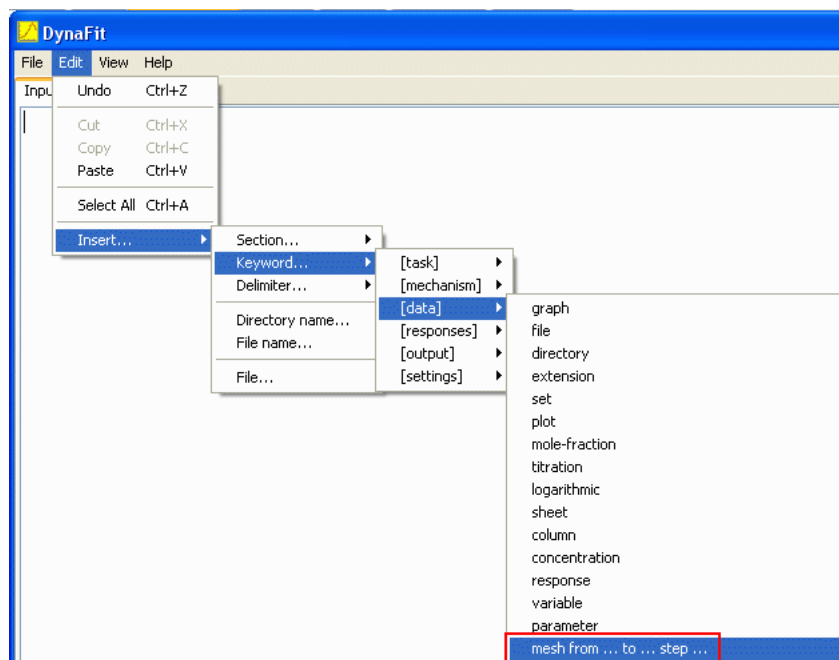


Fig. 3.4 Inserting recognized “phrases” (groupings of multiple keywords) into the Input editable area.

3.2.3 Delimiters and special characters

In addition to section names and keywords, DynaFit scripts almost always contain certain particular delimiters and special characters. The complete list of such special characters is shown in Table 3.3.

As in the case of keywords, certain groups of delimiters must appear together in recognized “phrases”. For example, to specify that a particular rate constant k_1 must be optimized within the range of 1 to 100, we can use the following notation:

```
[constants]
k1 = 10 ? (1 .. 100)
```


delimiter	purpose
+	Reaction mechanisms.
--->	Reaction mechanisms: irreversible step.
<---	Reaction mechanisms: irreversible step.
<==>	Reaction mechanisms: reversible step.
:	Reaction mechanisms: assign rate or equilibrium constants.
?	Identify an adjustable model parameter.
??	Request confidence interval estimation.
,	Separate lists of assignments.
..	Specify range of values for constrained optimization.
	Line break.
(Specify range of values for constrained optimization.
)	Specify range of values for constrained optimization.
{	Specify a set of values for initial estimate scan.
}	Specify a set of values for initial estimate scan.
*	Multiplication symbol.
/	Division symbol (thermodynamic cycles).
;	Comment.
!	Fixed concentration, for example pH.

Table 3.3 Recognized delimiters and special characters.

Thus the delimiters (, .., and) must appear together in this particular order. The **Edit | Insert...** menu selection facilitates this “phrasing” as is illustrated in Figure 3.5.

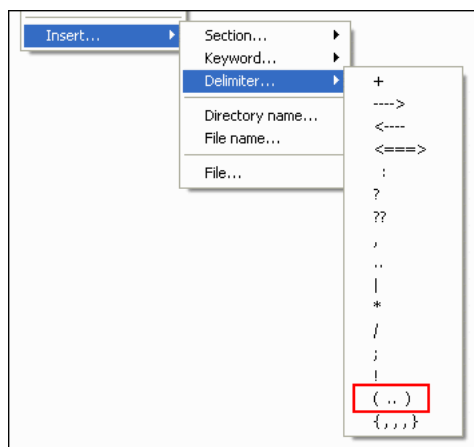


Fig. 3.5 Inserting recognized groupings of multiple delimiters into the Input editable area.

Comments and free-format annotations

In DynaFit scripts any text following the semicolon character (;) is ignored to the end of the given line. This makes it possible to annotate the input script by various comments, for example regarding concentration units or time units used throughout the analysis. Similarly it is possible to “uncomment” steps in the reaction mechanism without having to actually delete the given line.

Any text appearing before the first [task] section, or after the [end] marker line, is ignored by the DynaFit script interpreter. This is exceptionally useful for stating the purpose of the given analysis or for a summary of the results. Working with DynaFit often consists of interactive sessions, whereby the given input script is progressively modified numerous times. Free-format comments at the very beginning or at the very end of each script file represents the only effective way to rerecord what was accomplished in the given analysis.

3.2.4 Comprehensive scripting manual

The syntax of DynaFit scripting language is fully documented in the accompanying “Scripting Manual”, which contains numerous worked examples. Another useful resource are DynaFit scripts distributed in the `examples` directory. It is highly recommended that all new users of DynaFit work through the distributed examples by examining closely the script files.

The separate comprehensive scripting manual also contains detailed explanations of all the internal settings (algorithm controls) briefly described in the immediately following section 3.3.

3.3 Initialization files

The inner workings of the DynaFit software package rely of many diverse algorithms, some of which can be permanently fine-tuned by modifying the master initialization file `settings.txt` located in the subdirectory `system/DynaFit`. DynaFit internal settings can also be modified on a script-by-script basis, by inserting appropriate code in the [settings] section of the particular script.

3.3.1 Master initialization file

Every time DynaFit starts up, it reads the initialization file `./system/DynaFit/settings.txt`. The full text of this special file is shown in *Listing 3.9*. Please note that the individ-

ual sections of the master initialization file are delimited by curly brackets (. . .), as distinct from the square brackets ([. . .]) used elsewhere in the input scripts.

Listing 3.9

```

{DynaFit}
  RandomizationSeed          = 4357 ; | 0 for system time

{ODESolver}
  Iterations                 = 5000
  AbsoluteError              = 1.e-15
  RelativeError              = 1.e-9
  Interpolate                = n
  Increment                  = 0.1
  InterpolateAuto            = y
  Fraction                   = 0.1

{Marquardt}
  IterationsPerParameter    = 100
  Subiterations              = 30
  LambdaMax                  = 1000000000
  StopParameters             = 0.00001
  StopParametersAdd         = 0.0000001
  StopSquares                = 0.000001
  StopSquaresAdd            = 0.00000001
  MaxBoundViolations        = 0
  RestartPerturbation        = 0.1
  Restarts                   = 2
  RestartsConfidence        = 1
  RestartSeed                = 1234
  RobustFit                  = n
  RobustIterations           = 100
  RobustFitStop              = 0.000001
  EqualizeDatasets          = n
  AllowLimitingValues        = all           ; | some | none
  AllowLimitingStep          = 10
  WeightedFit                = y
  IntermediateWeights        = n
  CheckRedundantParameters  = y
  FixRedundantParameters    = y
  ParameterErrorsWeighted   = y

{ConfidenceIntervals}
  LevelPercent               = 95
  OnlyConstants              = y
  InferenceBand              = confidence   ; | prediction
  JointProbability           = n
  MaxSteps                   = 50
  MaxFirstTau                = 2
  StopSlope                  = 0 ; unused (was 0.00001)
  StopTau                    = 0.01
  StopTauRelative            = 0.001
  SmartSearch                = n
  FirstStep                  = 1.1

```

```

Acceleration = 1.1
DataParametersRatio = 0
SquaresIncreasePercent = 0

{DifferentialEvolution}
PopulationSizeFixed = 0
PopulationSizeMinimal = 300
PopulationSizePerParameter = 5
PopulationSizePerOrderOfMag = 3
MinimumGenerationsPerParameter = 5
MaximumGenerationsPerParameter = 100
MaximumEvolutions = 4
MinimumEvolutions = 1
RandomSeed = 1234
TestParameterRange = y
TestParameterRangeAll = y
TestParameterRangeFull = n
StopParameterRange = 0.01
TestCostFunctionRange = y
StopCostFunctionRange = 0.000001
TestCostFunctionChange = y
StopCostFunctionChange = 0.000001
TestCostFunctionChangeCount = 10

{Constraints}
Constants = 1000000
Responses = 1000000
Concentrations = 1000
AllParametersConstrained = y
IdenticalRelativeBounds = y
AllParametersRelativeBound = 1000000

{Filter}
PointsPerDataset = 0
ExponentialSpacing = n
ReadEveryNthPoint = 0
SkipFirstNPoints = 0
XMin = 0
XMax = 0
XShift = 0
XFirstMesh = 0.00001
TimeInitialRate = 1
PrintInitialRate = y
SmoothData = n
SmoothingMethod = savitzky-golay ; | average
SavitzkyGolayWindow = 10
SavitzkyGolayDegree = 4
ExtrapolationMethod = quadratic ; | linear
SmoothingPasses = 4
AverageReplicates = n
ZeroBaselineSignal = n

{PiecewiseLinearFit}
Points = 0

```

```

Segments = 4
Time = 0
Overlap = n

{Output}
UseDefaultDirectory = y
Autocorrelations = n
InferenceBands = n
StdDevInflationFactor = n
WriteTXT = n
WriteEPS = n
WriteTeX = n
XAxisLabel =
YAxisLabel =
BlackBackground = y
IncludeYZero = n
InitialRateDigits = 4

{Residuals}
AnalyzeEveryNthPoint = 1

{MonteCarlo}
PerformInitialFit = y
Runs = 1000
RandomizationMethod = simulate ; | shuffle | shift
Distribution = normal ; | cauchy | logistic | uniform
StandardDeviationSource = fit ; | data | explicit
StandardDeviation = 1.0
SignificantDigits = 4
HistogramBuckets = 20
TruncateMeanPercent = 5
ColorOutput = y
RandomizationSeed = 1267
ConcentrationErrorPercent = 0
OriginalEstimates = n

{EstimateScan}
ReportSizeMax = 1000
RefineMarquardt = 10

```

3.3.2 *Overriding default settings*

In most cases the default settings for DynaFit, determined by the contents of the master initialization file, will be appropriate for the task at hand. However, if necessary, the default settings for DynaFit can be overridden on the script-by-script basis, using the special section [settings].

For example, by default DynaFit generates only the best-fit model curves, but not the corresponding model inference bands. Also by default, DynaFit produces graphics with black background and brightly colored curves and symbols. If we

wished to modify these setting inside a particular script, we could achieve this in the given script file as is shown in *Listing 3.10*.

Listing 3.10

```
...
[settings]
{Output}
  InferenceBands = y ; default value = 'n'
  BlackBackground = n ; default value = 'y'
...
```

Great care must be exercised when modifying the contents of the master initialization file, or when overriding the default settings by way of the `[settings]` section of the given script file. The reason is that DynaFit has only limited capability for error-checking the particular values assigned to the initialization variables.

The precise meaning of each particular setting is explained in the detailed “Scripting Manual”.

Chapter 4

Output files

When DynaFit processes a given input script, it automatically writes all generated output files to the disk. This allows the program to be optionally executed in the unsupervised batch or automation mode. The output files generated by DynaFit can be grouped into three separate categories:

1. HTML files (plain text, formatted).
2. Tab-delimited ASCII files (plain text, unformatted).
3. GIF graphic files embedded into the auto-generated HTML code.

The various kinds of output files are auto-generated in appropriate subdirectories (htm, txt, or gif, respectively) of the main output directory. The main output directory associated with each script is specified in the [output] section of the input script file. For example, the following notation will direct all output into a directory ./project/kinase/output/fit-progress:

```
[output]
  directory ./project/kinase/output/fit-progress
```

In the above notation the period character (.) stands for “current working directory”, i.e., the particular directory where the DynaFit binary files are installed.

If the input script does not contain the [output] section, all output is written into the directory ./output. However, this is not recommended because all subsequent data-analytic sections would always overwrite pre-existing output files. It is much preferred to always specify a particular [output] directory associated with each script file. This will allow archiving the results of fit separately for each DynaFit run.

4.1 HTML files

When DynaFit processes a given script the focus in the main window changes from the Input tab to the Output tab, which displays the generated output files arranged

as a small “web site”, i.e., a collection of HTML files with embedded GIF graphics (see Figure 4.1).

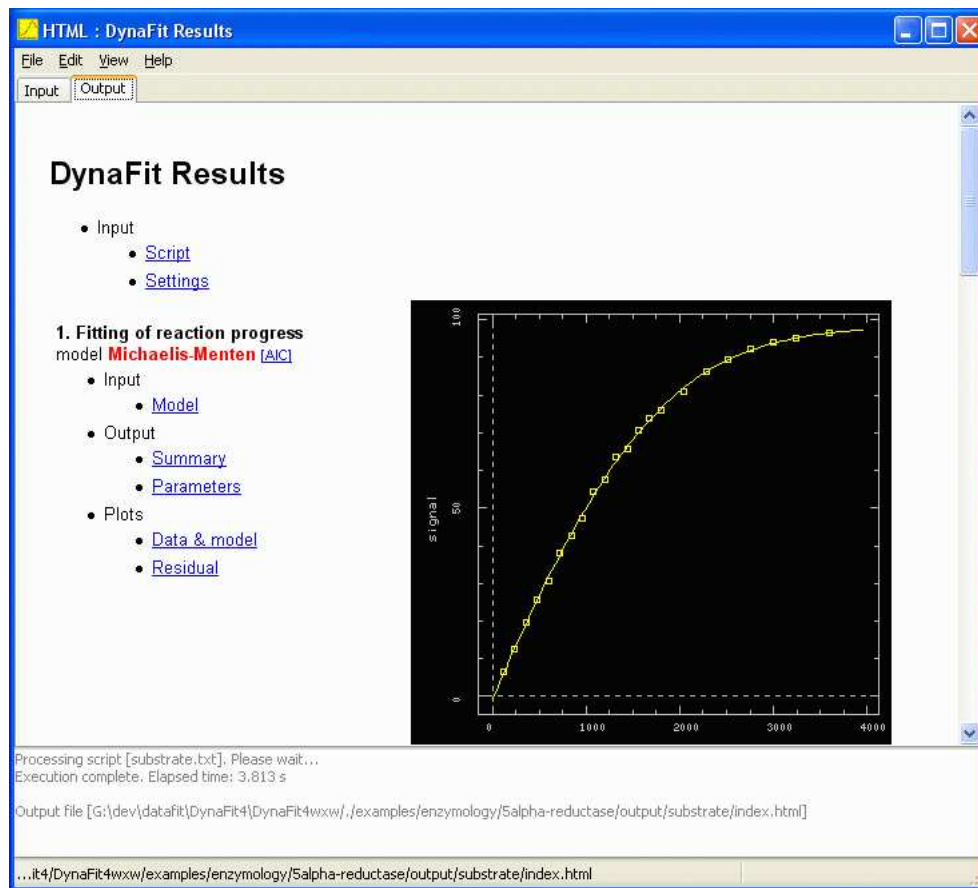


Fig. 4.1 Example output screen after executing the script `./examples/5alpha-reductase/substrate.txt`.

The DynaFit graphical user interface has a very simple built-in web browser, which allows navigation through the interlinked output files by clicking on various links. Pressing the **Alt-Left Arrow** or **Alt-Right Arrow** key combinations will bring forward a previously or subsequently browsed page.

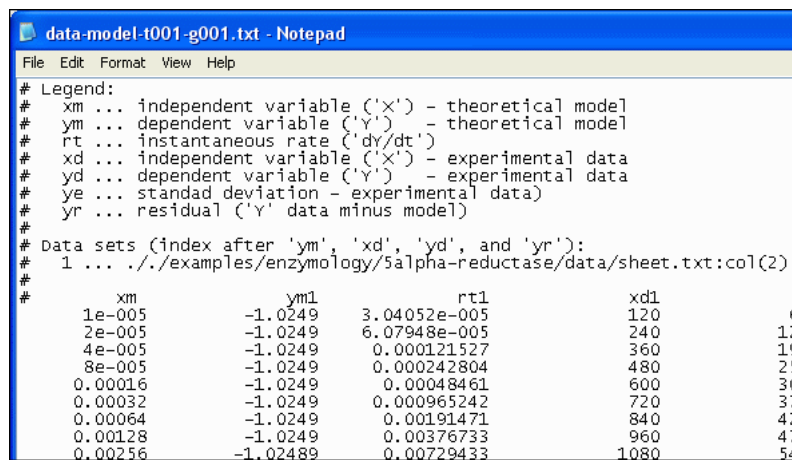
The simple web browser built into the DynaFit GUI does not have the capability to display richly formatted pages. For example, it ignores all Cascading Style Sheet (CSS) codings. To display the generated output files with greater depth of formatting, it is recommended to open the main output page in a full-scale external web browser. This is accomplished by selecting the menu item **View | Results in External Browser**.

All HTML output files are created in the subdirectory `htm` of the main output directory.

4.2 Plain text files

In addition to the formatted HTML files, DynaFit generates tab-delimited ASCII text files that can be used to generate publication-quality graphics in software packages such as Microsoft Excel, SigmaPlot, or other plotting programs. All text files are automatically generated in the subdirectory `txt` of the main output directory.

For example, Figure 4.2 shows the contents of file `data-model-t001-g001.txt` generated in the directory `./examples/5alpha-reductase/output/substrate`. The coding `t001` in the file name corresponds to the first “task” defined in the script; the coding `g001` signifies the first graph auto-generated by DynaFit.



```

# Legend:
# xm ... independent variable ('X') - theoretical model
# ym ... dependent variable ('Y') - theoretical model
# rt ... instantaneous rate ('dy/dt')
# xd ... independent variable ('X') - experimental data
# yd ... dependent variable ('Y') - experimental data
# ye ... standad deviation - experimental data)
# yr ... residual ('Y' data minus model)
#
# Data sets (index after 'ym', 'xd', 'yd', and 'yr'):
# 1 ... ./examples/enzymology/5alpha-reductase/data/sheet.txt:col(2)
#
#
#      xm          ym1          rt1          xd1          6
# 1e-005          -1.0249          3.04052e-005          120          12
# 2e-005          -1.0249          6.07948e-005          240          19
# 4e-005          -1.0249          0.000121527          360          25
# 8e-005          -1.0249          0.000242804          480          30
# 0.00016          -1.0249          0.00048461          600          37
# 0.00032          -1.0249          0.000965242          720          42
# 0.00064          -1.0249          0.00191471          840          47
# 0.00128          -1.0249          0.00376733          960          54
# 0.00256          -1.02489          0.00729433          1080

```

Fig. 4.2 Example output text file generated after executing the script `./examples/5alpha-reductase/substrate.txt`.

There are two ways to access the text files for import into third-party graphing programs. One possibility is utilize the requisite file manager (“Explorer” under MS Windows or “Finder” under Mac OS X) and open the given text file directly. Another possibility is to click on the **Export** link under each particular image, as is shown in Figure 4.3.

Among the plain text output files generated by DynaFit are those that contain the following information:

- Data for generating graphic images in external programs.
- Detailed list of all model parameters, both fixed and optimized.
- Detailed information about the progress of least-squares minimization.

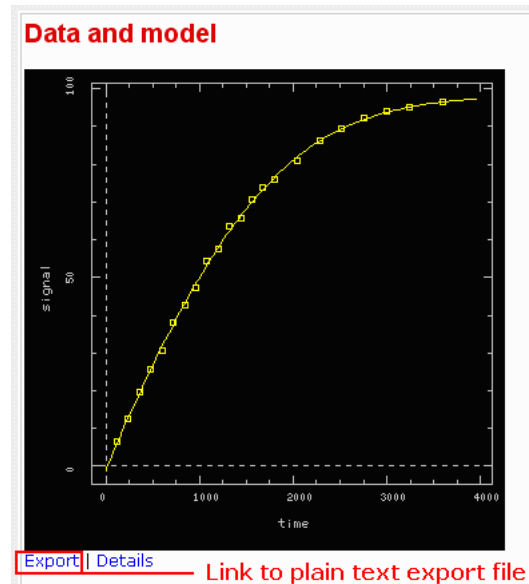


Fig. 4.3 Hyperlink to a plain text export file containing all information necessary to recreate the given image in an external graphics program.

- Detailed information about confidence interval search history.
- Distribution of population members during Differential Evolution global optimization.
- Histogram data from Monte-Carlo confidence interval estimations.

New users of DynaFit are encouraged to explore the contents of the subdirectory `txt` auto-generated in each output directory.

4.3 GIF graphics files

By default the GIF graphic images auto-generated by DynaFit contain black background and brightly colored foreground. This can be changed in the `[settings]` section of the input script, by setting `BlackBackground = n` in the `Output` section. DynaFit will then generate ALL GIF images with a white background, suitable for a quick copy-and-paste into a report document. It is also possible to alter the default axis labels, as is shown in *Listing 4.11*.

Listing 4.11

```
{Output}
  BlackBackground = n
  XAxisLabel = time, sec
```

```
YAxisLabel = % Product
```

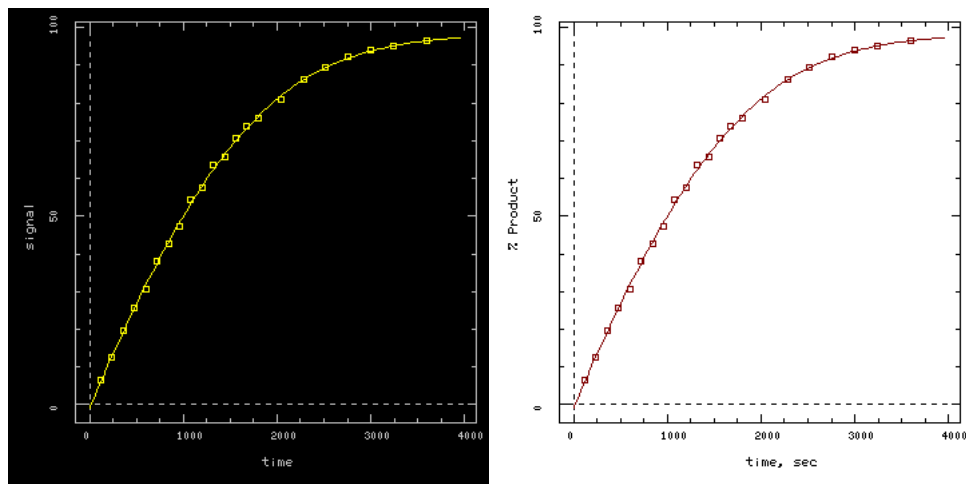


Fig. 4.4 Give graphics generated with default settings (*left-hand panel*) or with altered settings as shown in *Listing 4.11* (*right-hand panel*).

To quickly insert the auto-generated GIF graphics into an external report document such as a Microsoft Word file, you may follow these steps:

1. Select menu **View | Results in External Browser**.
2. Right-click on the given image.
3. Select **Copy image** from the drop-down menu.
4. Switch to a suitable report document, such as a Word, Excel, or PowerPoint.
5. Press **Ctrl+V** to paste the image into the report document.

To repeat for emphasis, the GIF graphic files automatically generated by DynaFit are not necessarily intended for journal publication purposes, except for quick laboratory reports in electronic notebooks and the like. To generate publication quality graphics as required by most research journals, it is necessary to import plain text files (see section 4.2) into sophisticated third-party plotting programs, such as SigmaPlot, and generate high resolution graphic images by utilizing those specialized software packages.

Chapter 5

Step-by-step tutorial

This chapter describes the step-by-step procedure required to analyze a typical set of biophysical data. The purpose of the experiment was to determine the equilibrium binding constant for Cyclosporin A (CsA), a known immunosuppressive drug, interacting with recombinant human cyclophilin protein.

CsA itself is not visible in fluorescence spectroscopy. For this reason we prepared a fluorescently labeled analog of CsA that functioned as a *spectroscopic probe* [5, 6]. In one set of equilibrium binding measurements, the concentration of the labeled analog (CsA-L) was held fixed at 200 nM, while the protein concentration was varied from zero to 650 nM. This resulted in an overall increase in fluorescence intensity, because the cyclophilin/CsA-L molecular complex has greater propensity to fluoresce than does the CsA-L labeled ligand alone. The excitation wavelength (350 nm) was chosen such that the intrinsic protein fluorescence of the protein was not involved.

In another set of equilibrium binding measurements, the concentration of the labeled ligand, CsA-L, was again held fixed at 200 nM while the protein concentration was varied, but this time the unlabeled ligand, CsA, was also present at 500 nM concentration. The presence of the unlabeled ligand resulted in less protein being available to form the cyclophilin/CsA-L molecular complex and therefore in less increase in fluorescence intensity. From the observed difference in fluorescence intensities we can infer the binding constant of the spectroscopically “invisible” ligand.

5.1 Input data files

This section describes how to prepare the input data for DynaFit. Let us assume that the raw data come to us as an Excel file. The numerical values are shown in Table 5.1. The second and fourth columns show the concentration of cyclophilin protein, $[P]$, in micromolar units. The third and fifth column show the associated fluorescence intensity, F , in relative fluorescence units (RFU). In the first experi-

ment (columns 2 and 3) the concentration of the unlabeled ligand, UL , was zero. In the second experiment (columns 4 and 5) the concentration of the unlabeled ligand, UL , was $0.5 \mu\text{M}$. In both experiments the concentration of the labeled ligand, LL , was $0.2 \mu\text{M}$.

No.	$[UL] = 0$		$[UL] = 0.5 \mu\text{M}$	
	$[P], \mu\text{M}$	F, RFU	$[P], \mu\text{M}$	F, RFU
1	0.00	0.873	0.00	0.420
2	0.00	0.809	0.00	0.476
3	0.02	1.309	0.01	0.588
4	0.03	1.265	0.01	0.634
5	0.06	1.918	0.02	0.727
6	0.08	2.001	0.02	0.802
7	0.10	1.932	0.03	0.951
8	0.10	2.109	0.04	1.119
9	0.12	2.305	0.06	1.025
10	0.17	2.962	0.08	1.136
11	0.20	2.678	0.10	1.116
12	0.22	3.080	0.12	1.545
13	0.27	3.027	0.17	1.758
14	0.30	3.174	0.20	2.028
15	0.34	3.189	0.22	2.204
16	0.38	3.248	0.26	2.166
17	0.42	3.116	0.27	2.529
18	0.48	3.195	0.30	2.808
19	0.52	3.371	0.34	2.928
20	0.56	3.215	0.38	2.871
21	0.60	3.333	0.42	2.907
22	0.65	3.406	0.48	2.727
23			0.52	2.810
24			0.56	3.042
25			0.60	2.770
26			0.65	3.272

1	A	B	C	D	E
2	No.	P, μM	F:L=0	P, μM	F:L=0.5 μM
3	2	0.00	0.873	0.00	0.420
4	3	0.02	1.309	0.01	0.588
5	4	0.03	1.265	0.01	0.634
6	5	0.06	1.918	0.02	0.727
7	6	0.08	2.001	0.02	0.802
8	7	0.10	1.932	0.03	0.951
9	8	0.10	2.109	0.04	1.119
10	9	0.12	2.305	0.06	1.025
11	10	0.17	2.962	0.08	1.136

Fluorescence in the absence of unlabeled ligand Fluorescence in the presence of unlabeled ligand

Table 5.1 Raw data for the CsA / cyclophilin competitive ligand displacement experiment. The concentration of the labeled ligand was $[LL] = 0.2 \mu\text{M}$.

To prepare the experimental data for analysis with DynaFit, we must first create plain text (ASCII) data files and place them into appropriate location on the disk. We will end up with two data files, each corresponding to one of the two experiments summarized in Table 5.1. The procedure is as follows.

5.1.1 Create new project directory and data directory

1. Locate the directory where the DynaFit binary files are installed. Under the Windows operating system, this is typically My Documents/DynaFit4.

2. Within the main DynaFit working directory, create a new subdirectory called **tutorial**.
3. Within this subdirectory, create another subdirectory called **data**.

The final directory layout is shown in Figure 5.1

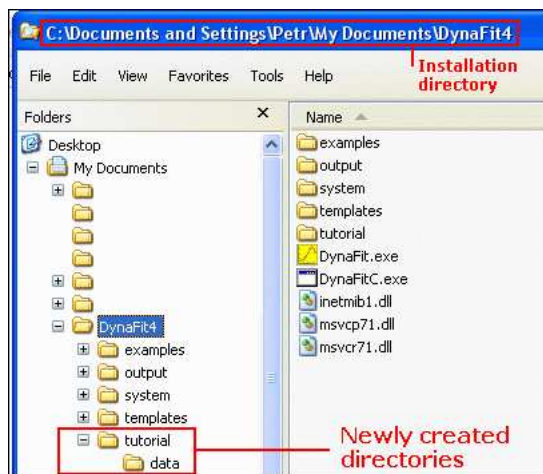


Fig. 5.1 Creating a new project directory and data directory for the tutorial example problem.

5.1.2 Create data files on the disk

1. Within the newly created directory `./tutorial/data`, create a new plain text file named `LL.txt`. This file will hold experimental data from the first experiment, involving labeled ligand (hence, *LL*) only.
2. Copy columns B and C from the Excel file holding the raw experimental data (see Table 5.1) into the system clipboard: (a) select the data contained in both columns; (b) press **Ctrl+C**.
3. Paste the raw experimental data into the newly created data file: (a) switch focus to the file `LL.txt`; (b) press **Ctrl+V**; (c) save the modified data file to disk.
4. Within the directory `./tutorial/data`, create a new plain text file named `LL+UL.txt`. This file will hold experimental data from the second experiment, involving both the labeled ligand (*LL*) and the unlabeled ligand (*UL*).
5. Copy columns D and E from the Excel file holding the raw experimental data (see Table 5.1) into the system clipboard: (a) select the data contained in both columns; (b) press **Ctrl+C**.
6. Paste the raw experimental data into the newly created data file: (a) switch focus to the file `LL+UL.txt`; (b) press **Ctrl+V**; (c) save the modified data file to disk.

The final directory layout is shown in Figure 5.2

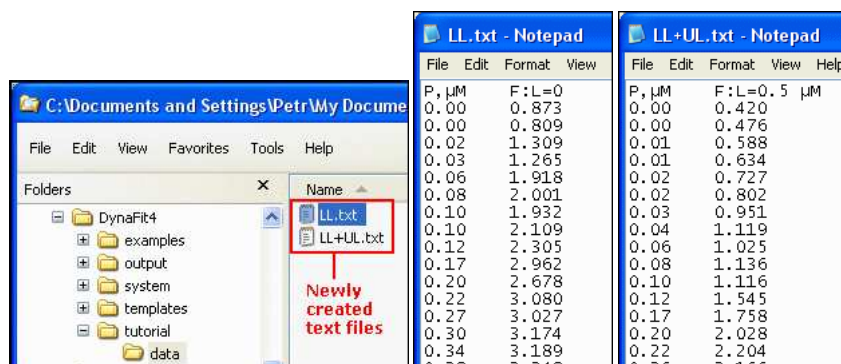


Fig. 5.2 Creating ASCII text data files for the tutorial example problem.

5.2 Script files

This section describes step-by-step how to create DynaFit script files to perform the following data analyses:

1. Fit the control experiment (labeled ligand only) to determine the binding constant for the labeled ligand.
2. Fit the competition experiment (mixtures of labeled and unlabeled ligand) to determine the binding constant for the unlabeled ligand, while keeping the previously determined binding constant for the labeled ligand as a fixed parameter in the fitting model.

5.2.1 Fitting the control experiment

Prepare the script file

The “control” experiment is one where only the labeled ligand (LL) is present, whereas the unlabeled ligand (L) is absent (the concentration is $[L] = 0$). The purpose of this experiment is to determine the equilibrium binding constant K_{dL} .

1. Within the newly created directory `./tutorial`, create a new plain text file named `fit-LL.txt`.
2. Follow the steps described below in detail, to compose the script file.
3. Save the modified script file to the disk.

Listing 5.12 shows the completed script file. The remainder of this section describes in detail how and why the individual sections were arranged.

Listing 5.12

```
Fit control experiment (labeled ligand only).
[task]
  task = fit
  data = equilibria
[mechanism]
  P + LL <=> P.LL      :      KdL      dissociation
[constants]
  KdL = 0.1 ?
[concentrations]
  LL = 0.2
[responses]
  LL = 4.25 ?
  P.LL = 16.5 ?
[data]
  variable P
  file ./tutorial/data/LL.txt
[output]
  directory ./tutorial/output/fit-LL
[end]
```

We begin the construction of the script file by specifying the particular task (in this case, data fitting) and the particular data type (in this case, equilibrium binding). This is accomplished by typing the following text at the beginning of the script file `fit-LL.txt`:

```
Fit control experiment (labeled ligand only).
[task]
  task = fit
  data = equilibria
```

The first line above is a “comment” that reminds us of the purpose of this particular script. Such informative comments are often crucially important in working with DynaFit.

Next we need to specify the molecular mechanism for the binding process. Let us assume a simple 1:1 binding stoichiometry, with one molecule of the labeled ligand (*LL*) binding to one molecule of the protein (*P*). We shall label the corresponding equilibrium dissociation constant as K_{dL} . These assumptions will be represented in the DynaFit script by the following notation:

```
[mechanism]
```



The choice of molecular species names (in this case P , LL and $P.LL$) is entirely arbitrary. We could have chosen labels such as `Protein`, `LabeledLigand`, and `LabeledComplex`, or any other convenient names that make sense to us. The same applies to the name or label of the equilibrium dissociation constant. We could have chosen equivalently something like $K_{d(1)}$, $K_{d(\text{labeled})}$ or anything similar that makes sense to us.

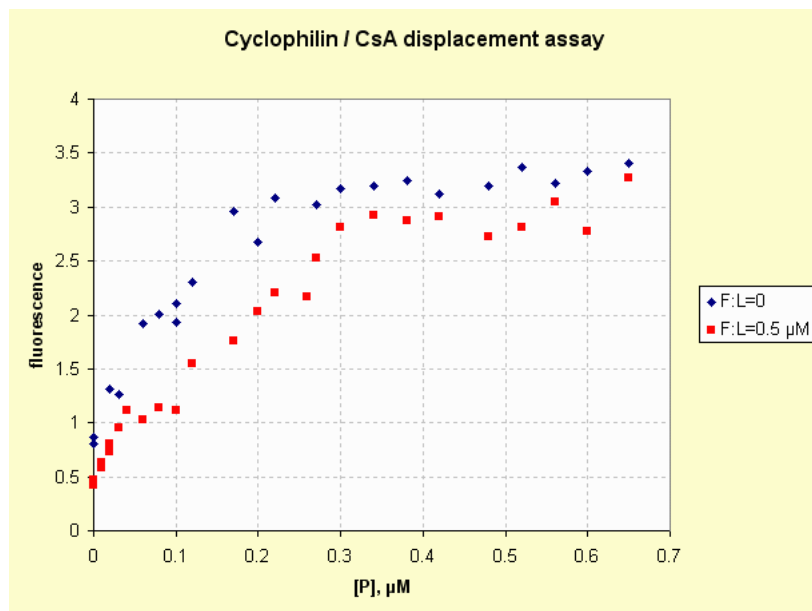


Fig. 5.3 Plot of raw experimental data in Excel.

In the next step we need to choose the *initial value* of the dissociation constant K_{dL} , to be optimized in the least-squares fit of the experimental data. To accomplish this task it will be useful to examine Figure 5.3. This Figure shows a plot of the raw experimental data that is included in the practice data file `csa-displacement-data.xls`.

We see that already at approximately 100 nM protein concentration (0.1 μM) the abscissa in Figure 5.3 the fluorescence signal changes significantly. Recall from basic physical chemistry that the dissociation equilibrium constant has the same dimension as concentrations (i.e., moles per liter); and that significant changes in the composition occur at total reactant concentrations that are commensurate with the equilibrium constant. Thus, it seems plausible that K_{dL} should be commensurate with approximately 100 nM.

In working with DynaFit, it is not possible to formulate a general rule of thumb regarding how close is close enough when making initial estimates of model parameters such as equilibrium constants. In the case of simple molecular mechanisms, involving only two binding equilibria as is the case here, an initial estimate that is within two orders of magnitude for the equilibrium constant might suffice. In any case, the overlay of theoretical model curves simulated with the initial estimates of all model parameters should produce a plot where the model curves and the data points are approximately in scale. In practical terms, this means that after selecting **File ... Try**, we should not see either the model curve or the data points visually coinciding with horizontal axis.

Our experimental data are given in micromolar concentrations, and therefore the equilibrium dissociation constant must be given in the same concentration units (micromoles per liter). *The concentration units used for the experimental data must always be the same as the concentration units used to define all equilibrium constants appearing in the given binding mechanism.* Thus the most appropriate numerical value for the dissociation constant is approximately $K_{dL} = 0.1 \mu\text{M}$. This is reflected in the DynaFit script as follows:

```
[constants]
KdL = 0.1 ?
```

The question mark after the numerical value signifies that we wish to optimize the initial estimate in the least-squares regression to be performed by DynaFit.

In the next step we must specify the concentrations of those molecular species that are unchanging in the given experiment. In this case it is the concentration of the labeled ligand, which is held at 200 nM throughout the binding experiment. The notation used for the labeled ligand in the [mechanism] section is LL. Again we must use the same concentration units that are implied by the experimental data, in this case micromolar. This leads to the following section of the DynaFit script:

```
[concentrations]
LL = 0.2
```

Next we come to the notion of *specific molar responses* that link molecular species concentrations to the observable experimental signal. Each response coefficient is a number that specifies how many instrument units (for example, fluorescence, absorbance, chromatographic peak area, radioactive counts, etc.) will be produced by one concentration unit of the given molecular species. Recall that in this case we have chosen to represent all concentrations in micromolar units, so the question is how many fluorescence units correspond to one micromole per liter of the labeled ligand LL, and correspondingly how many fluorescence units correspond to one micromole per liter of the protein–ligand complex P.LL.

According to the data in Table 5.1, in the absence of protein and at 0.2 μM labeled ligand, the observed fluorescence intensity is approximately $F = 0.85$ RFU.

Thus, if $0.2 \mu\text{M}$ produces 0.85 relative fluorescence units (RFU) then $1 \mu\text{M}$ of LL (the unit concentration amount) would produce $0.85/0.2 = 4.25$ RFU. This will be reflected in the DynaFit script as

```
[responses]
  LL = 4.25 ?
  ...
```

Please note that the molar response coefficient of LL is to be treated as one of the adjustable model parameters, as is specified by the question mark after the numerical value immediately above.

However, the labeled ligand LL is not the only fluorescent species in this experiment. In fact, we know with certainty that the complex $P.LL$ must show specific molar fluorescence intensity that is higher than the response coefficient of LL , otherwise there would be no increase of fluorescence upon protein–ligand binding. Thus we need to come up with the initial estimate for the molar response coefficient of $P.LL$.

Let us assume that at the highest protein concentration listed in Table 5.1, the system is already at saturation, in the sense that the maximum possible amount of the protein–ligand complex has already been formed. This is suggested by the fact that between $0.3 \mu\text{M}$ and $0.65 \mu\text{M}$ total protein the fluorescence intensity (approximately 3.3 RFU) is changing very little. The maximum amount of protein–ligand complex formed in this binding experiment obviously must be equal to the amount of labeled ligand, namely, $[P.LL]_{\text{max}} = 0.2 \mu\text{M}$. Thus the specific molar response coefficient must be approximately equal to $(3.3 \text{ RFU})/(0.2 \mu\text{M}) = 16.5 \text{ RFU}/\mu\text{M}$. This allows us to complete the `[responses]` section of the script file as follows:

```
[responses]
  LL = 4.25 ?
  P.LL = 16.5 ?
```

Again, note that both specific molar response coefficients are treated as adjustable parameter in the least-squares fitting model.

In the next step we need to specify the location of the experimental data files to be analyzed, as well as which molecular species appearing in the molecular mechanism is was varied in the experiment, as the *independent variable*. In this case the variable component was the total concentration of the protein, P . This is specified by the following notation in the script file:

```
[data]
  variable P
  file ./tutorial/data/LL.txt
```

Note that the period `.` at the start of the file name means “current working directory”, in which the DynaFit binary files are installed (see Figure 5.2 in section 5.1). The forward slashes `/` signify hierarchical nesting of subdirectories. Also note that all DynaFit keywords are *case sensitive*, so that `File` in place of `file` would not work.

Finally, we must specify the location of the output directory, in which DynaFit should create the output files. This is accomplished by the following notation:

```
[output]
  directory ./tutorial/output/fit-LL
```

This notation will lead to the automatic creation of the subdirectory `output/fit-LL` inside the project directory `./tutorial`. Listing 5.12 shows the full text of the completed script file `./tutorial/fit-LL.txt`. All DynaFit script files must end with the special notation `[end]` standing on separate line.

Important Note

The treatment of the molar response coefficients as one of the adjustable model parameters (see notation `LL = 4.25 ?` above) brings up a serious problem of *over-parametrization*. This problem has not been sufficiently recognized among biochemical data analysts in general. The consequence of over-parametrization (i.e., having too many adjustable parameters relative to the information content of the experimental data) is that the model parameters such as equilibrium constants are in the end affected by very high uncertainty. One might legitimately ask, why not determine the molar response of the labeled ligand in an independent experiment and subsequently (in the competition experiment) treat it as a fixed constant? The reader is invited to explore this issue by setting the labeled ligand response coefficient to a fixed value (`LL = 4.25`, without the question mark after “4.25”).

Perform the analysis to determine K_{dL}

The following procedure is used to ascertain if our initial estimates of model parameters are sufficiently accurate.

1. Start DynaFit, if it is not already running.
2. Select menu **File** | **Open...**
3. Navigate to select the script file `./tutorial/fit-LL.txt`.
4. Select menu **File** | **Try**.

Selecting menu **File** | **Try** caused DynaFit to create a plot of the initial estimate of the fitting model overlaid on the experimental data, as is shown in Figure 5.4. The purpose is to make sure that the plot of the initial estimate for the model curve is sufficiently close to the experimental data.

It is difficult to formulate hard-and-fast rules to specify what is a “sufficiently good” initial estimate of the fitting model. Probably the best we can say is that the plot of the experimental data and the initially estimated model curves must be in the same scale. This means that neither the model curve plot, nor the experimental data plot, is allowed to essentially coincide with the horizontal axis.

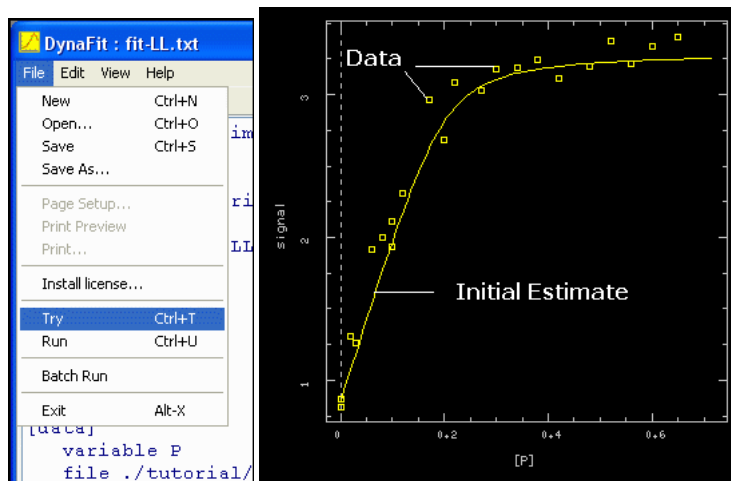


Fig. 5.4 Comparing the initial estimate of the fitting model with the experimental data.

In this particular case (see Figure 5.4) the experimental data and the initially estimated model curve are approximately in the same scale, so we can be reasonably confident that the initial estimate of the adjustable model parameters – the equilibrium constant K_{dL} and the molar response coefficients – are reasonably close to their true values. This being the case, we can proceed to finally perform the fit.

1. Switch focus to the Input tab in the main DynaFit window.
2. Select menu **File** | **Run** (see Figure 5.5).
3. Wait several seconds, while the computation is proceeding.
4. Select menu **View** | **Results in External Browser**.
5. Navigate in the generated output files. For example, click on the **Parameters** link to examine the best-fit value of the equilibrium constant K_{dL} .

Clicking on the **Parameters** link on the main output page reveals the best-fit values of the adjustable model parameters:

Optimized Parameters

No.	Par#Set	Initial	Final	Std. Error	CV (%)	Note
#1	KdL	0.1	0.00864319	0.00514116	59.48	
#2	r(LL)	4.25	4.92917	0.311364	6.32	
#3	r(P,LL)	16.5	16.7136	0.396418	2.37	

Thus the best-fit value of the equilibrium dissociation constant is $K_{dL} = (8.6 \pm 5.1)$ nM. (Recall that the concentration units used throughout this analysis are micromolar, which means that the value of K_{dL} listed in the **Final** column above need

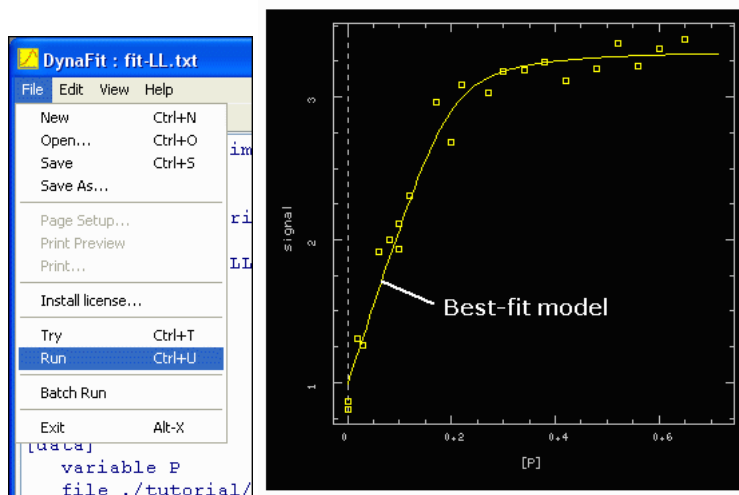


Fig. 5.5 Performing the least squares fit.

to be multiplied by 1000 to scale the results to nanomoles per liter.) In the next round of analysis, this best-fit value of K_{dL} will be treated as a fixed constant.

The user is invited to experiment with the initial estimate of K_{dL} in the input script file. As was mentioned above, in the case of a simple competition model where one of the equilibrium constants is treated as a fixed parameter, the initial estimate of the remaining equilibrium constant could be “off” by many orders of magnitude and yet DynaFit would converge on the correct fit.

5.2.2 Fitting the competition experiment

The purpose of this round of analysis is to determine the equilibrium dissociation constant for the spectroscopically “invisible” ligand, Cyclosporin A. The raw experimental data are listed in columns 4 and 5 of of Table 5.1.

Prepare the script file

The completed script file for fitting the competition experiment is shown in *Listing 5.13*. The script is very similar to the *Listing 5.12*. The modified script should be saved with a *different file name*, fit-LL+UL.txt, so that the original script, fit-LL.txt is not overwritten and lost. The differences between the two scripts are discussed below in detail.

Listing 5.13

Fit the competition experiment (labeled + unlabeled ligand) while treating KdL as a fixed parameter.

```
[task]
  task = fit
  data = equilibria
[mechanism]
  P + LL <==> P.LL      :      KdL      dissociation
  P + UL <==> P.UL      :      KdU      dissociation
[constants]
  KdL = 0.00864319      ; fixed !
  KdU = 0.1 ?
[concentrations]
  LL = 0.2
  UL = 0.5
[responses]
  LL = 4.25 ?
  P.LL = 16.5 ?
[data]
  variable P
  file ./tutorial/data/LL+UL.txt
[output]
  directory ./tutorial/output/fit-LL+UL
[end]
```

Important Note

It is extremely beneficial to keep track of different versions of the DynaFit script files as they emerge during a particular data fitting session. The recommended procedure is to save each variant with a different file name, for example fit-001.txt, fit-002.txt, fit-003.txt etc. Each of these script files should produce the output files in a *separate output directory*. In this way the collection of DynaFit input and output files can serve as an effective electronic notebook.

Whereas in the case of the control experiment the [mechanism] section referred to a single binding interaction, the section now describes to the competition between the labeled ligand (*LL*) and the unlabeled ligand (*UL*) for binding to the protein *P*:

```
[mechanism]
  P + LL <==> P.LL      :      KdL      dissociation
  P + UL <==> P.UL      :      KdU      dissociation
```

The “competitive” nature of the two binding interactions is expressed by the fact the protein *P* binds *either* the labeled ligand *or* the unlabeled ligand, but never both ligands at the same time. This assumes that the cyclophilin protein possess only a single, unique binding site for either of the two ligands (i.e., Cyclosporin A proper and the fluorescent analog of Cyclosporin A).

The [constants] section of the script now shows two dissociation equilibrium constants, rather than just one:


```
[constants]
  KdL = 0.00864319    ; fixed !
  KdU = 0.1 ?
```

The dissociation equilibrium constant for the labeled ligand, K_{dL} , is treated as a fixed parameter, set to the numerical values obtained in the control experiment. This is expressed by the fact that the numerical value $K_{dL} = 0.00864319 \mu\text{M}$ is *not* followed by the question mark symbol. In contrast, the dissociation equilibrium constant for the unlabeled ligand, K_{dU} , is treated as an adjustable model parameter.

The modified [concentration] section now correctly reflects the fact that both the labeled and the unlabeled ligand are present in the experiment. As before, the concentration units are micromoles per liter, because that is the concentration unit used in the experimental data files:

```
[concentrations]
  LL = 0.2
  UL = 0.5
```

The [data] section of the script now refers to the competition data file LL+UL.txt, as opposed to the control data file LL.txt:

```
[data]
  variable P
  file ./tutorial/data/LL+UL.txt
```

Finally, the output directory has been changed to direct all output files such that they do not overwrite the output generated in the analysis of the control experiment above:

```
[output]
  directory ./tutorial/output/fit-LL+UL
```

Perform the analysis to determine K_{dU}

The procedure to fit the competition binding data is essentially identical to one that was used to fit the control experiment:

1. Start DynaFit, if it is not already running.
2. Select menu **File** | **Open...**
3. Navigate to select the script file ./tutorial/fit-LL+UL.txt.
4. Select menu **File** | **Try**. Verify that the initial estimate is sufficiently close to the experimental data.
5. Select menu **File** | **Run**.
6. Wait several seconds, while the computation is proceeding.
7. Select menu **View** | **Results in External Browser**.

Optimized Parameters

No.	Par#Set	Initial	Final	Std. Error	CV (%)	Note
#1	KdU	0.1	0.0709144	0.0236251	33.32	
#2	r(LL)	4.25	2.5175	0.313285	12.44	
#3	r(P,LL)	16.5	16.1934	0.533532	3.29	

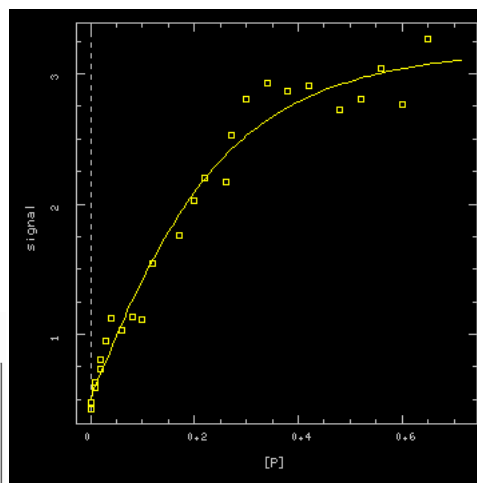


Fig. 5.6 Results of fit of the competition experiment.

- Navigate in the generated output files. For example, click on the **Parameters** link to examine the best-fit value of the equilibrium constant K_{dL} .

The final results are illustrated in Figure 5.6. Thus, we can conclude that the dissociation equilibrium constant of the unlabeled ligand is $K_{dU} = (71 \pm 23)$ nM. This completes the determination of the binding affinity of Cyclosporin A toward human recombinant cyclophilin.

5.2.3 Summary

In this section we have learned how to construct a DynaFit script file used to determine the binding constant of an unlabeled, spectroscopically invisible ligand binding to a protein molecule. A labeled ligand served as a fluorescent probe. The `examples` directory distributed with DynaFit contains useful templates to achieve other research goals. The user is invited to modify these templates to suit his or her particular research needs.

5.3 Output files: Publication quality graphics

This section describes how the user can generate a particular illustration for publication purposes, utilizing the plain text ASCII files generated by DynaFit. Let us assume that we wish to construct an illustration combining the results of fit from

the control experiment (section 5.2.1) and from the competition experiment (section 5.2.2).

1. Create a new Microsoft Excel file called `plot.xls` in subdirectory `./tutorial/results`.
2. Label the first sheet as `control`; the second sheet as `competition`; and the third sheet as `plot`.
3. Navigate to the auto-generated output directory `./tutorial/output/fit-LL/txt`.
4. Open the auto-generated text file `data-model-t001-g001.txt`.
5. Select the entire file contents and copy it into the system clipboard (**Ctrl+A**, **Ctrl+C**).
6. Paste the numerical data into the sheet `control` in the newly created Excel file `plot.xls`.
7. Navigate to the auto-generated output directory `./tutorial/output/fit-LL+UL/txt`.
8. Open the auto-generated text file `data-model-t001-g001.txt`.
9. Select the entire file contents and copy it into the system clipboard (**Ctrl+A**, **Ctrl+C**).
10. Paste the numerical data into the sheet `competition` in the newly created Excel file `plot.xls`.
11. Copy the columns labeled `xm` and `ym1` from the sheet `control` into the sheet `plot`. This represents the first model curve, from the control experiment.
12. Copy the column labeled `ym1` from the sheet `competition` into the sheet `plot`. This represents the second model curve, from the competition experiment.
13. Copy the columns labeled `xd1` and `yd1` from the sheet `control` into the sheet `plot`. This represents the first experimental data set, from the control experiment.
14. Copy the columns labeled `xd1` and `yd1` from the sheet `competition` into the sheet `plot`. This represents the second experimental data set, from the competition experiment.
15. Label the columns in the Excel sheet `plot`, for example as is shown in Figure 5.7.
16. Use the numerical data in the Excel sheet `plot` to generate a publication-quality illustration, as is shown in Figure 5.8.

5.4 Note to Mac OS X users

When using DynaFit under the Apple Mac OS X operating system, one needs to be aware of two idiosyncrasies that are specific to that software platform. The first has to do with copying data out of MS Excel files in “plain text” (ASCII) format. The second issue arises in producing purely “plain text” files as such. Both issues have a direct impact on the success of working through the tutorial described in this chapter.

	A	B	C	D	E	F	G
1	x:model	y:model-1	y:model-2	x:data-1	y:data-1	x:data-2	y:data-2
2	0.0000	0.9858	0.5035	0.00	0.873	0.00	0.420
3	0.0072	1.0665	0.5756	0.00	0.809	0.00	0.476
4	0.0143	1.1469	0.6468	0.02	1.309	0.01	0.588
5	0.0215	1.2270	0.7170	0.03	1.265	0.01	0.634
6	0.0286	1.3068	0.7862	0.06	1.918	0.02	0.727
7	0.0358	1.3863	0.8545	0.08	2.001	0.02	0.802
8	0.0429	1.4654	0.9217	0.10	1.932	0.03	0.951
9	0.0501	1.5440	0.9878	0.10	2.109	0.04	1.119
10	0.0572	1.6222	1.0528	0.12	2.305	0.06	1.025
11	0.0644	1.6999	1.1167	0.17	2.962	0.08	1.136
12	0.0715	1.7770	1.1795	0.20	2.678	0.10	1.116
13	0.0787	1.8534	1.2410	0.22	3.080	0.12	1.545
14	0.0858	1.9290	1.3014	0.27	3.027	0.17	1.758

Fig. 5.7 Collecting numerical data generated by DynaFit in ASCII format, for a publication quality graph.

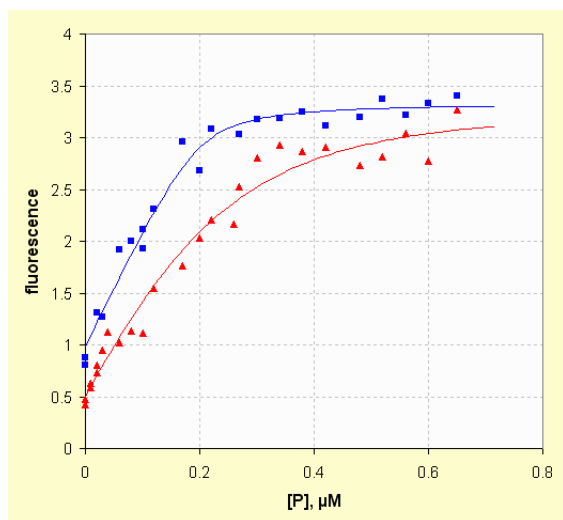


Fig. 5.8 Publication-quality graph in Excel, compiled from multiple ASCII files generated by DynaFit.

5.4.1 Copying plain-text (ASCII) data from MS Excel

Depending on a particular version of MS Excel for Mac, copying a block of cells from an Excel spreadsheet into the system clipboard (**Command+C**) and subsequently pasting (**Command+V**) the clipboard contents into a text file may fail. The

reason is that Excel apparently puts into the system clipboard data that are not pure ASCII but do contain formatting information.

A successful work-around has been to install and utilize [OpenOffice](http://www.openoffice.org)¹ as the open-source alternative to MS Office. OpenOffice does have the ability to open, save, and otherwise manipulate MS Excel files. Importantly, copying numerical data from OpenOffice and pasting the data into a text file does produce pure ASCII code.

5.4.2 *Pure plain text (ASCII) editors*

It appears that it is somewhat challenging to identify a “pure” ASCII editor for the Mac OS X operating system, or at least one that does not require special effort to produce “pure” plain text by default. Older version of TextEdit distributed with the system prior to version 10.8 did indeed produce “pure” plain text files automatically. However, the most recent version of TextEdit requires a change of default settings:

1. Start TextEdit
2. Choose *Preferences* from the TextEdit menu.
3. Under *Format*, select the *Plain text* radio-button rather than the default *Rich text*.

As of this writing, a suitable alternative to TextEdit appears to be the freeware package TextWrangler made available by Bare Bones Software Inc. This package does produce “pure” ASCII text files by default. It is therefore quite suitable for the handling of experimental data files to be processed by DynaFit, or for the handling of DynaFit script files manipulated outside the DynaFit script editor window.

¹ <http://www.openoffice.org>

References

1. Kuzmič, P. (1996) Program DYNAFIT for the analysis of enzyme kinetic data: Application to HIV proteinase. *Anal. Biochem.* **237**, 260–273.
2. Kuzmič, P. (2009) DynaFit – A software package for enzymology. *Meth. Enzymol.* **467**, 247–280.
3. Moss, M. L., Kuzmič, P., Stuart, J. D., Tian, G., Peranteau, A. G., Frye, S. V., Kadwell, S. H., Kost, T. A., Overton, L. K., and Patel, I. R. (1996) Inhibition of human steroid 5 α reductases type I and II by 6-aza-steroids: Structural determinants of one-step vs two-step mechanism. *Biochemistry* **35**, 3457–3464.
4. Szedlacsek, S. and Duggleby, R. G. (1995) Kinetics of slow and tight-binding inhibitors. *Meth. Enzymol.* **249**, 144–180.
5. Moss, M. L., Palmer, R. E., Kuzmič, P., Dunlap, B. E., Henzel, W., Kofron, J. L., Mellon, W. S., Royer, C. A., and Rich, D. H. (1992) Identification of actin and HSP 70 as Cyclosporin A binding proteins by photoaffinity labeling and fluorescence displacement assays. *J. Biol. Chem.* **267**, 22054–59.
6. Kuzmič, P., Moss, M. L., Kofron, J. L., and Rich, D. H. (1992) Fluorescence displacement method for the determination of receptor–ligand binding constants. *Anal. Biochem.* **205**, 65–69.
7. Wang, Z.-X. (1995) An exact mathematical expression for describing competitive binding of two different ligands to a protein molecule. *FEBS Lett.* **360**, 111–114.
8. Goodrich, J. A. and Kugel, J. F. (2007) *Binding and Kinetics for Molecular Biologists*. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.
9. Fridman, M., Belakhov, V., Lee, L. V., Liang, F.-S., Wong, C.-H., and Baasov, T. (2005) Dual effect of synthetic aminoglycosides: Antibacterial activity against *bacillus anthracis* and inhibition of anthrax Lethal Factor. *Angew. Chem. Int. Ed.* **44**, 447–452.
10. Kuzmič, P., Cregar, L., Millis, S. Z., and Goldman, M. (2006) Mixed-type noncompetitive inhibition of anthrax lethal factor protease by aminoglycosides. *FEBS J.* **273**, 30543062.
11. Burnham, K. B. and Anderson, D. R. (2002) *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*. Springer-Verlag, New York, 2nd edition.
12. Myung, J. I. and Pitt, M. A. (2004) Model comparison methods. *Meth. Enzymol.* **383**, 351–366.
13. Myung, J. I., Tang, Y., and Pitt, M. A. (2009) Evaluation and comparison of computational models. *Meth. Enzymol.* **454**, 287–304.
14. Segel, I. H. (1975) *Enzyme Kinetics*. Wiley, New York.
15. Bosco, G., Baxa, M., and Sosnick, T. (2009) Metal binding kinetics of Bi-Histidine sites used in ψ analysis: Evidence of high-energy protein folding intermediates. *Biochemistry* **48**, 2950–2959.

16. Storme, T., Deroussent, A., Mercier, L., Prost, E., Re, M., Munier, F., Martens, T., Bourget, P., Vassal, G., Royer, J., and Paci, A. (2009) New ifosfamide analogs designed for lower associated neurotoxicity and nephrotoxicity with modified alkylating kinetics leading to enhanced in vitro anticancer activity. *J. Pharmacol. Exp. Ther.* **328**, 598–609.
17. Gasa, T., Spruell, J., Dichtel, W., Srensen, T., Philp, D., Stoddart, J., and Kuzmič, P. (2009) Complexation between methyl viologen (paraquat) bis(hexafluorophosphate) and dibenzo[24]crown-8 revisited. *Chem. Eur. J.* **15**, 106–116.
18. Van Boekel, M. (2000) Kinetic modelling in food science: A case study on chlorophyll degradation in olives. *J. Sci. Food Agric.* **80**, 3–9.
19. Von Weymarn, N., Kiviharju, K., and Leisola, M. (2002) High-level production of D-mannitol with membrane cell-recycle bioreactor. *J. Ind. Microbiol. Biotechnol.* **29**, 44–49.
20. Le Clainche, L. and Vita, C. (2006) Selective binding of uranyl cation by a novel calmodulin peptide. *Environ. Chem. Lett.* **4**, 45–49.
21. Seber, G. A. F. and Wild, C. J. (1989) *Nonlinear Regression*. Wiley, New York.
22. Bates, D. M. and Watts, D. G. (1988) *Nonlinear Regression Analysis and its Applications*. Wiley, New York.
23. Prigogine, I., Lefever, R., Goldbeter, A., and Herschkowitz-Kaufman, M. (1969) Symmetry breaking instabilities in biological systems. *Nature* **223**, 913–916.
24. Martin, J. L. and Maconochie, D. J. (1989) The direct fitting of a differential equation to experimental data. *J. Physiol.* **418**, 7.
25. Martin, J. L., Maconochie, D. J., and Knight, D. R. (1994) A novel use of differential equations to fit exponential functions to experimental data. *J. Neurosci. Meth.* **51**, 135–146.
26. Price, K. V., Storn, R. M., and Lampinen, J. A. (2005) *Differential Evolution: A Practical Approach to Global Optimization*. Springer Verlag, New York.

Index

A

- actin, 3
- Akaike Information Criterion, 7
- algebraic methods, 5
 - experimental design, 5
- aminoglycosides, 5
- anthrax, 5
- ASCII
 - output files, 45
- ASCII files, 23
 - Microsoft Excel, 24

B

- Bacillus anthracis, 5
- benign prostatic hyperplasia, 1
- binding constant, 49
 - tutorial, 49
- biophysics, 8
- black background, 41, 46
 - override, 41

C

- case sensitivity
 - script files, 33
- command-line interface, 21, 22
 - automation, 22
 - batch run, 22
 - high-throughput screening, 22
- comments
 - data files, 26
 - script files, 38
- competition experiment, 49
- competitive inhibition, 5
- competitive ligand binding, 49, 60

- competitive ligand displacement, 3
- complex equilibria, 3
 - algebraic methods, 5
 - mathematical model, 5
- Computer ID, 15
- concentrations, 55
 - tutorial, 55
- cyclophilin, 3, 49
- Cyclosporin A, 3, 49

D

- data files, 23, 25
 - columns, 25
 - comments, 26
 - equilibrium binding, 28
 - grid pattern, 28
 - header lines, 26
 - Job plot, 32
 - meta-data, 26
 - multi-variate data, 27
 - multiple independent variables, 30
 - NMR chemical shift, 30
 - number format, 32
 - localization, 32
 - reaction progress, 28
 - required format, 25
 - sheet format, 26
 - titration plot, 30
 - tutorial, 49, 56
- delimiters, 36
 - groupings, 36
- dissociation equilibrium constant, 49
 - tutorial, 49
- DynaFit
 - download, 12
 - example problems, 15

- examples, 1
- executable files, 21
 - DynaFit.exe, 21
 - DynaFitC.exe, 21
- for Mac OS X, 13
- for Microsoft Windows, 13
- installation, 12
- purpose, 1
- script
 - keywords, 19
 - sections, 19
- support, 16

E

- enzyme inhibition, 1
 - competitive, 7
 - inhibition by substrate, 7
 - mixed type, 7
 - noncompetitive, 7
 - slow, tight, 1
 - uncompetitive, 7
- equilibrium binding
 - data files, 28

F

- file permissions
 - administrator, 13
 - user, 13
- fluorescence, 49
 - displacement assay, 3

G

- GIF files, 47
- Graphical User Interface, 17
- graphics, 46
 - publication quality, 62
 - text files, 62
 - third-party software, 45

H

- high-throughput screening, 22
- Hsp70, 3
- HTML, 43
 - built-in browser, 44
 - external browsers, 44

I

- inference bands, 41

- initial estimate, 17, 57
- initial rate enzyme kinetics, 5
- initialization, 38
 - master file, 38
 - overriding defaults, 41

J

- Job plot
 - data files, 32

K

- ketosteroid reductase, 1
- keywords
 - groupings, 35

L

- Lethal Factor protease, 5
- license file, 13
 - academic, 13
 - full text, 14
 - request, 13
 - activation, 15
 - commercial, 15
 - Computer ID, 15
 - installation, 13
- localization
 - number format, 32
 - American, 32
 - French, 32
 - German, 32

M

- Mac OS X
 - command-line interface, 21
- mathematical model
 - algebraic, 12
 - exponential, 12
 - piecewise linear, 12
- menu
 - Edit, 19
 - insert, 19
 - File, 17
 - Install License..., 19
 - Run, 19
 - Try, 17
 - View, 19
 - Computer ID, 19
 - Results in External Browser, 20
 - Startup Path, 20

Microsoft Excel
 ASCII files, 24
Microsoft Windows
 command-line interface, 21
 installer program, 13
model discrimination analysis, 7
molar response coefficients, 55
 tutorial, 55
molecular biology, 8

N

Neomycin B, 5
NMR chemical shift
 data files, 30
noncompetitive inhibition, 7
number format, 32

O

ordinary differential equations, 2
 automatic derivation, 2
organic chemistry, 8
output directory
 tutorial, 57
output files, 46
 ASCII, 45
 GIF, 47
 graphics, 46
 black background, 46
 HTML, 43
 plain text, 45

P

physical chemistry, 8
pinene
 decomposition kinetics, 8
plain text
 output files, 45
protein folding, 8

R

reaction progress
 data files, 28
responses, 55

S

script files, 33
 case sensitivity, 33
 comments, 38
 free format, 38
 delimiters, 36
 keywords, 34, 35
 alphabetical list of, 35
 groupings, 35
 Scripting Manual, 38
 sections, 33, 34
 comprehensive list of, 34
 special characters, 36
 tutorial, 52
settings, 38
simulations, 10
 negative feedback inhibition, 10
 oscillatory kinetics, 10
special characters, 36
substrate inhibition, 7

T

text editor, 17
text files
 graphics, 62
titration plot
 data files, 30
tutorial, 49
 data files, 49

U

uncompetitive inhibition, 7

Z

ZIP archive, 13