

Mathematical simulation and analysis of cellular metabolism and regulation

Igor Goryanin^{1,2}, T. Charles Hodgman² and Evgeni Selkov^{1,3}

¹Laboratory for Metabolic Modeling and Bioinformatics, The Russian Academy of Sciences, 142292 Pushchino, Moscow Region, Russia, ²Advanced Technology and Informatics Unit, GlaxoWellcome Medicines Research Centre, Gunnels Wood Road, Stevenage, Herts. SG1 2NY, UK and ³Mathematics and Computer Science Division, Argonne National Laboratory, 7400 S. Cass Avenue, Argonne, IL 60439-4844, USA

Received on September 11, 1998; revised and accepted on April 19, 1999

Abstract

Motivation: A better understanding of the biological phenomena observed in cells requires the creation and analysis of mathematical models of cellular metabolism and physiology. The formulation and study of such models must also be simplified as far as possible to cope with the increasing complexity demanded and exponential accumulation of the metabolic reconstructions computed from sequenced genomes.

Results: A mathematical simulation workbench, DBsolve, has been developed to simplify the derivation and analysis of mathematical models. It combines: (i) derivation of large-scale mathematical models from metabolic reconstructions and other data sources; (ii) solving and parameter continuation of non-linear algebraic equations (NAEs), including metabolic control analysis; (iii) solving the non-linear stiff systems of ordinary differential equations (ODEs); (iv) bifurcation analysis of ODEs; (v) parameter fitting to experimental data or functional criteria based on constrained optimization. The workbench has been successfully used for dynamic metabolic modeling of some typical biochemical networks (Dolgacheva *et al.*, *Biochemistry (Moscow)*, **6**, 1063–1068, 1996; Goldstein and Goryanin, *Mol. Biol. (Moscow)*, **30**, 976–983, 1996), including microbial glycolytic pathways, signal transduction pathways and receptor–ligand interactions.

Availability: DBsolve 5.00 is freely available from <http://websites.ntl.com/~igor.goryanin>.

Contact: gzz78923@ggr.co.uk

Introduction

The number of sequenced genomes is growing exponentially and will probably exceed 100 by the turn of the century. One first step toward understanding the processes of life is metabolic reconstruction from the specified sequenced genome (<http://wit.mcs.anl.gov>; Overbeek *et al.*, 1997). Each reconstruction represents a skeleton model of the cellular organization consistent with the sequence data and

available biochemical knowledge stored in databases such as EMP (<http://wit.mcs.anl.gov/EMP>; Selkov *et al.*, 1996) and MPW (<http://wit.mcs.anl.gov/MPW>; Selkov *et al.*, 1997). The mathematical simulation and analysis of such reconstructions is the next logical step toward a deeper understanding of cellular behaviour.

A number of computer programs have been designed to assist biochemists in describing aspects of enzymology and metabolic networks. MetaModel (Cornish-Bowden and Hofmeyr, 1991) and SCAMP (Sauro, 1993) deal with steady states and their metabolic control analysis (MCA) coefficients. Gepasi (Mendes, 1993, 1997) calculates steady states as well as reaction time behaviour, and characterizes the steady state with MCA elasticity, flux- and concentration-control coefficients. SIMFIT (Holzhutter and Colosimo, 1990) is an MS-DOS program that integrates the simulation of dynamic models and fitting them to experimental data. MIST (Ehlde and Zacchi, 1995) is a Windows-based simulator that can calculate reaction progress, steady states and use MCA for characterization of the steady states.

Regarding the mathematics that is required for metabolic modelling, two principal software packages incorporating the parameter continuation techniques and bifurcation analysis have been developed (Khibnik *et al.*, 1993; Doedel *et al.*, 1994). Khibnik and co-workers have designed a numerical technique and software package for the analysis of local bifurcations, which are based on the continuation of structurally unstable invariant sets in a suitable phase-parameter space.

We believe that both the exponential increase in the number of sequenced genomes and the increasing demand for models of biochemical and physiological processes necessitate the development of a comprehensive software suite which excludes all the time-consuming manual operations involved in formulating, debugging and analysis of mathematical models. Moreover, this requires a tight integration of all appropriate simulation tools and available

information resources into one integrated development environment. From the programmer's perspective, such an environment dictates its own internal language, built-in editors and on-line compiler. DBsolve is an attempt to realize these ideas.

System and methods

DBsolve 5.00 is written in C++. The 32-bit version is freely available for Windows 95/NT together with example models covering all DBsolve features. The system requires a Pentium PC with a minimum 120 MB free hard disk space and 16 MB RAM. A UNIX-compatible command-line version is also available on request, which has been made accessible through a web-cgi interface. The workbench has an object-oriented structure containing a range of methods and associated data that can be viewed from the main application window. DBsolve has an internal language for building and storing models, and an internal compiler for processing the models and running simulations. A derived model can be saved and/or transferred to other computer systems as an ASCII text file (with the suffix .SLV) which contains all the necessary information about the mathematical model: the system schema, the experimental parameter values, information about the components and reactions/processes, and links to external databases records. SLV files have a platform (and version)-independent, object-oriented structure which is described in the Appendix.

Algorithm

For the purposes of this work, a model is a description of a (biological) system which includes a set of components with attributes that include their quantitative amount, activity and interactions. A simulation is an electronic experiment with a model to discover the consequences of imposing specified initial conditions. Often, a series of specified parameters may be examined to discover which best approximate to a set of observations from the biological system itself, in which case this procedure is known as fitting. The system components are usually small molecules (in the case of metabolic modelling), but may equally be macromolecules, complexes or molecules in different cellular compartments.

The DBsolve architecture and method of use are outlined in Figure 1. Data can be imported from pre-existing SLV files, MPW records or entered manually. These data are then processed to derive, solve, fit and analyse mathematical models with the aid of the range of tools outlined below. Each of the DBsolve methods has its own page on the main window, and these will be described in turn.

Model Designer

Figure 2 shows the main application window with the 'Model Designer' tabbed page in view. Pre-existing SLV files can be entered using the Open button. Upon selecting the file, the appropriate parameters are put into the grid (the stoichiometric matrix) on the Model Designer page, after which modifications to the model can be made. Predefined pathways have been defined and can be viewed through the WIT and MPW web servers. This pathway data can be used in DBsolve by saving the MPW record (as html) on a local disk, and importing it by clicking on the 'New from MPW' option in the File menu and selecting the desired file.

If starting from scratch, the user must select one of the four alternatives from the 'Model type' menu, which offers skeleton models for enzymes, metabolic pathways, receptor-ligand interactions or some general process. Clicking the 'New' button allows the creation of the SLV file into which all the subsequent details will be stored. A window then appears which provides a set of instructions that explain the steps that need to be taken to finish creating the model. Clicking the OK button results in the Model Designer grid being populated with a skeleton set of parameters.

The user can develop this skeleton model either by editing this grid, or clicking first on the 'View as chart' button and then on the 'Edit' button which appears on the graphic editor. If using the latter, the 'Save as matrix' button should be clicked once the graphic editing is complete. On the top row of the grid, the grey, numbered buttons pop up windows for entering details of the specified system component. Likewise, the grey buttons in the first column open a separate window that permits input of details of each reaction process, which include enzyme classification codes and sequence database identifiers (if appropriate), the type of reaction mechanism and confidence in the details provided. This window also contains a button, named 'View original map', which launches a web browser that enables access to other details such as the enzyme reaction mechanism from EMP.

Each reaction is named k_i where i is an integer (see Appendix). The numbers on each row refer to the number of the components, defined by the column, participating in reaction k_i . The negative and positive values indicate whether the components are on the left- or right-hand side of the reaction equation, respectively. The model is then saved by pressing the 'Create Model' button.

The numerical procedures used require the Jacobian matrix to be inverted. However, the Jacobian may have singularities owing to the presence of conserved cycles—which are usually present in metabolic networks. To identify these cycles, DBsolve reduces the stoichiometric matrix and sets up a vector of independent variables using systematic procedures for finding conservation laws from

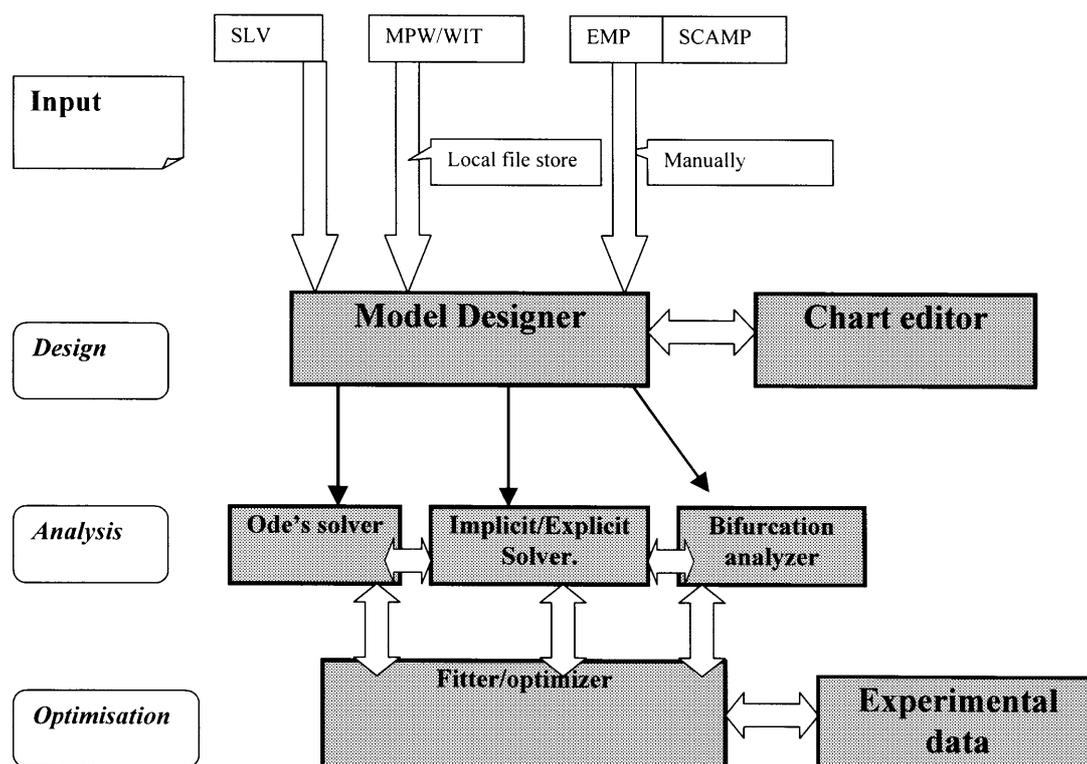


Fig. 1. DBsolve 5.00 architecture. The four stages of DBsolve use are shown on the left-hand side of the page. Data for designing a model may be input from various sources, although the method of input (direct, via the local file store or manually) depends upon the source. Software objects within DBsolve 5.00 are shown as shaded rectangles. The optimization phase employs one object which is used either for optimizing the model to achieve certain specified criteria or for fitting to a set of experimental data.

chemical (Bezdeneniyh, 1973) and metabolic (Reder, 1988) kinetics.

ODE Solver

DBsolve uses numerical procedures for integration of ordinary differential equations (ODEs), or mixed ODE/non-linear algebraic equations (NAEs), to describe the dynamics of these models. In particular, it has an original implicit integration algorithm, with the Newton prediction-correction procedure for every integration step, similar to that of Gear and Petzold (1984). The algorithm has a step size control and a popular LSODE algorithm (Hindmarsh, 1983) has also been implemented. Both methods have special subroutines for getting output for user-defined time points, which is essential for fitting algorithms.

To get the time evolution of the system, the user must go to the 'ODEs' page, define the variable to plot, time limit and accuracy, and press the 'Solve' button. The results appear in the 'Plot' window. By running the solver over long time periods, the model can reach a steady state, which is desirable

for providing the approximate set of initial values that should be used by the explicit and implicit solvers. These steady-state values (labelled as $X[j]=$) can be seen at the bottom of the Messages window, and should be saved either by cutting these lines from the Messages window and pasting them into the bottom of the Initial values window and then closing this window (saving changes), or by checking the 'Change Initial Point' box on the Options page.

Explicit Solver

Users may have their own particular equation which they require solving and wish to be applied to a set of experimental data. DBsolve offers the facility to encode and solve such 'explicitly stated' formulae. They should be typed at the bottom of the RHS window and then solved by turning to the 'Explicit Solver' tabbed page, where the appropriate parameters can be entered and the method executed by pressing the 'Solve' button.

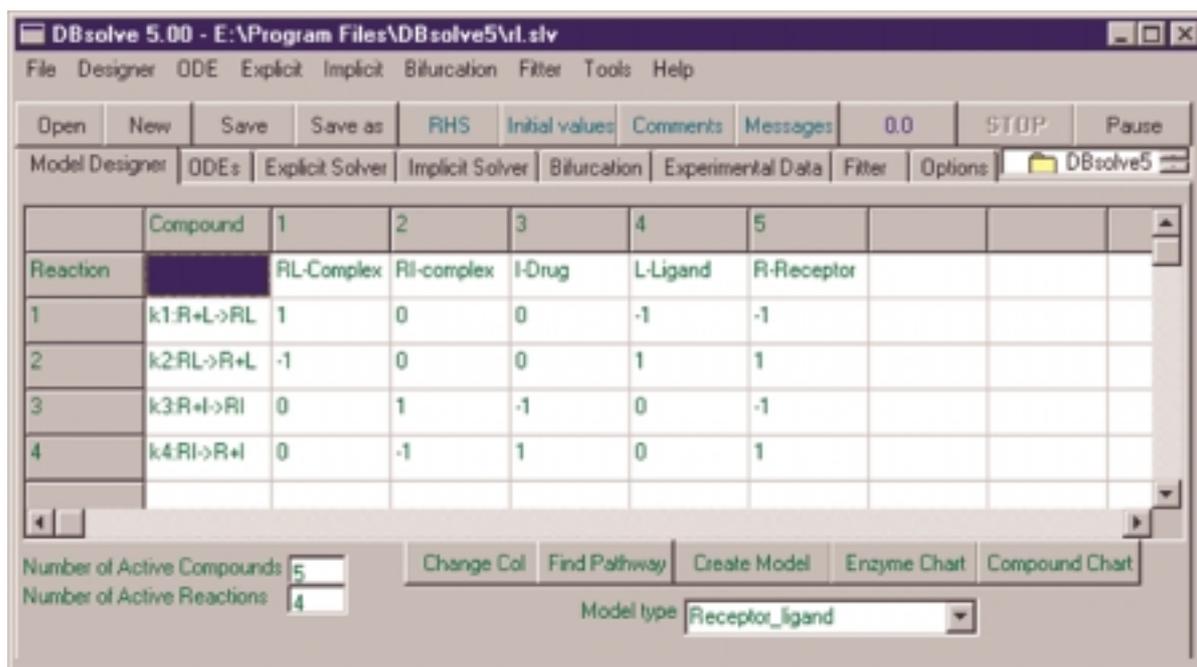


Fig. 2. The main DBsolve 5.00 application window. The menu items across the top mostly execute the particular methods named. The next row of buttons pertain to items from the File menu (in black), the Right Hand Sides and Initial Values which comprise the model (in green), parameter comments and output messages (also in green), the run time conditions (dark blue) and run conditions (Stop in red and Pause in grey). Below these buttons are the tabbed pages which contain the parameters and method-execution buttons for each of the object modules named on the tab. The Model Designer page appears by default.

Implicit Solver

This method allows the user to trace the changes in the system as a result of changing one or more of its parameters. This procedure is very useful for determining any functional dependencies (such as overall steady-state flux, control coefficients, product concentration) against any external (substrate concentrations) or internal (enzyme concentrations) model parameters. It is especially useful in the case of non-linear algebraic systems which have no explicit solution or have multiple or unstable solutions. DBsolve includes a general continuation procedure, based on a tangent predictor continuation scheme (Khibnik *et al.*, 1993). A modified Newton corrector is employed which makes adaptive step sizes on the basis of estimates from the current tangent and secant vectors. This minimizes the possibility of jumping from one branch of a curve to another, and allows the user to optimize the next step size according to computed points on the curve.

The Implicit Solver page contains two columns of parameter boxes: the left hand for the Solver parameters and the right hand for graphical display. It is usual to have the same value on the x -axis as is being applied by the Solver. Clicking the Solve button generates the desired plot.

Bifurcation Analyser

Bifurcation theory is a more systematic and general theory of non-linear systems than the standard, steady-state analysis of metabolic pathways. Computation of one- or two-parameter bifurcation diagrams can quickly inform the user about what is possible for, or prohibited by, a particular type of non-linear model (Guckenheimer and Holmes, 1983; Glass and Mackey, 1988). Bifurcation analysis is useful in the development and evaluation of models in many areas of biology, such as early development, growth, chronobiology, metabolic control, cellular sensors and the theory of enzyme kinetics (Hocker, 1994). These bifurcations may add restrictions to what can be a realistic model of the system under investigation. There are few reliable software products for bifurcation analysis, so we have used numerical methods similar to LOCBIF (Khibnik *et al.*, 1993). All algorithms have been rewritten in C and modified to integrate with the DBsolve object-oriented environment.

One- and two-parameter diagrams of Equilibrium, Fold, Hopf, Flip and Focus-node bifurcations can be generated (Guckenheimer and Holmes, 1983). The Bifurcation Analyser uses the same numerical continuation code as the Implicit Solver, but it is expanded with routines for the

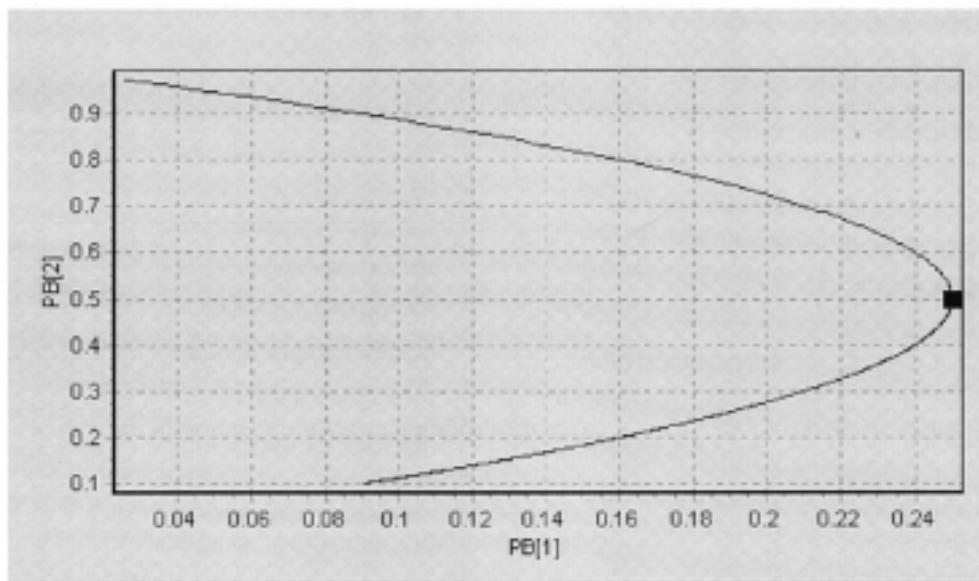


Fig. 3. A Hopf bifurcational analysis plot. A generalized Lotka model was produced (Schulmeister and Selkov, 1977) which provides a description for various regulatory processes including feedback inhibition, autocatalytic activation and energy metabolism. The system scheme involves two equations: $F[1] = 1 - PB[1] \cdot X[1] - X[1] \cdot X[2] \cdot X[2]$ and $F[2] = PB[2] \cdot (X[1] \cdot X[2] \cdot X[2] - X[2])$. The curve shows the line of neutrality, on which the matrix trace of the linearized system passes zero, so that a focus changes its stability. In this example, when $PB[1] = 0.16$, the value of $PB[2] = 0.2$ or 0.8 . Biological examples corresponding to $PB[1]$ and $PB[2]$ are provided by Schulmeister and Selkov (1977). The black square shows where another bifurcation occurs.

evaluation of bifurcation functions and calculating eigenvectors. Bifurcations have been found at points where black rectangles are drawn on the plot (see Figure 3). Further details are put in the Message window.

Experimental data

Where pertinent data are available, DBsolve can use them for refining the model. The Experimental Data page (Figure 4A) contains a grid into which the data can be entered, either by typing or cutting-and-pasting from other sources. There are four columns per experiment and currently up to eight experiments can be entered simultaneously. The first row contains the information on which method (implicit, explicit or ODE as described above) should be applied to the model for comparison with the observed data points. A 1 or 0 on the second row determines whether or not that particular dataset is to be included in the study. The first and second column of each dataset (on the third row) contain the names of the model's parameters that have been determined in the experiment, with the actual data supplied below. The third and fourth column of each dataset contain the name and value of any other parameter which was defined as part of the experiment. These extra details impose helpful constraints for fitting the model to the observed data. Pressing any of the four grey buttons at the top of a dataset will plot the output

from the model (using the specified method) as a solid black line, against which the experimental data points are depicted as red squares (Figure 4B).

Fitter

This method can be used either to fit a model to experimental data (thereby discovering the values with appropriate error margins of the model's parameters under the conditions of the experiment), or to optimize a model to meet specified criteria (to discover how and which parameters should be changed to produce the specified optimum). For fitting, the method of generating the theoretical curves is specified on the Experimental Data page and the 'Use Experimental Data' box is checked. Constraints can be imposed by pressing the Parameters button and defining the initial, boundary and step values for specified parameters. The fitting/optimization employs either a zero-order (Hooke and Jeeves, 1961) or first-order (Levenberg, 1944; Marquardt, 1963) algorithm. Fitting procedures always face difficulties caused by multiple minima, which may be a particular problem when many parameters are involved. The 'best' fit might not be easily found; however, to check the quality of the procedure, the standard deviation and confidence intervals for every active parameter as well as an ANOVA table are shown in the Message window to help users make their assessment.

A

	X#1	Y#1	Vars #1	Values #1	X#2	Y#2	Vars #2	Values #2	X#3
Method	Implicit				ODE				Expli
Include	1				1				0
Variables	X[3]	RLTotNSB			X[3]	RLTotal			X[3]
	4.398	85.1	tmp	10	120	70	tmp	0	4.398
	3.796	87.1	tmp	0.000328	240	130	tmp	0	3.796
	3.194	92.1	tmp	5.001	480	195	tmp	0	3.194

B

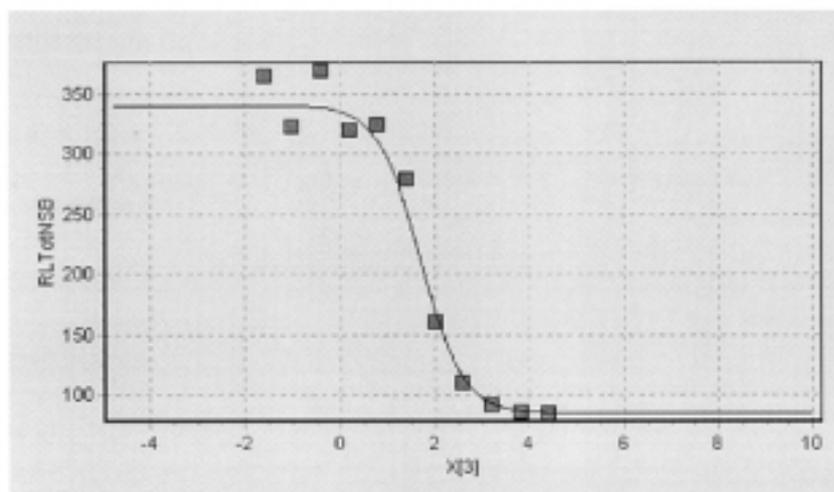


Fig. 4. (A) The Experimental Data tabbed page. The four columns per set of experimental results are labelled X#n, Y#n, Vars#n and Values#n, where X is the independent variable resulting in the observed value of Y, Vars are extra model parameters having the Values defined, and n is the experiment number. In the Vars# column, tmp refers to an unspecified temporary variable that is not used for fitting. (B) The plot generated for the first set of experimental data shown in (A). The squares plot the observed values and the line shows the theoretical curve determined using the Implicit Solver after prior fitting. X[3] and RLTotalNSB refer, respectively, to the log[Inhibitor] and the observed binding in the assay (which is the sum of the specific Receptor-Ligand interactions and Non-Specific Binding).

When fitting to laboratory data, the residual function between theoretical and experimental points is calculated according to a least square or absolute value (modulus) formula. The Absolute Fitting box is checked if all the component concentrations are similar, otherwise relative fitting is applied. These are defined by the following equations:

$$F[0]=\text{SUM}(Y_t(i)-Y_e(i))^2$$

$$F[0]=\text{SUM fabs}(Y_t(i)-Y_e(i))$$

$$F[0]=\text{SUM}(Y_t(i)-Y_e(i))^2/Y_e(i)$$

$$F[0]=\text{SUM fabs}(Y_t(i)-Y_e(i))/Y_e(i)$$

where Y_t and Y_e are the theoretical and experimental values, respectively.

For optimization of some system component or activity, experimental data are not considered, so that box is left unchecked. The implicit or ODE Solver must be chosen (for looking at steady-state or integration over time, respectively) and an objective function ($F[0]=\dots$) appended to the Right Hand Sides page.

Options

This page enables the user to carry out a variety of activities (defined by the check boxes on the left-hand side of the page) associated with other methods in the application. Data for up to 12 parameters can be exported from the workbench by checking the box, specifying the filename and adding the parameters to the list on the right-hand side. Most of the activities are obvious from the checkbox names. However, it is possible to generate a family of curves on the same plot by checking the corresponding box, and supplying the details along the bottom row: which 'Variable' to change, from what initial 'Value', by which arithmetic 'Sign' [+ , - , * or /], of what 'Step' size and for how many 'Times'.

Discussion

DBsolve is a workbench which enables the user to generate mathematical models of intermolecular processes of particular interest to biologists. These processes obviously

include, but are not limited to, metabolic networks, signal transduction and receptor–ligand–inhibitor interactions.

The use of a stoichiometric matrix, in which each row defines a specific process step in the system and each column the component involved, enables powerful mathematical techniques to be brought to bear and profoundly simplifies the modelling of biological processes. For example, when two metabolic pathways have molecules in common (such as ATP), the two pathways can be merged simply by matrix addition—the 'Add from MPW' option in the File menu offers this facility. Hence, it is now feasible to generate a complete metabolic model where complete genome data are available. Likewise, it is straightforward from this matrix to evaluate the overall stoichiometry of the model's input substrate(s) and output product(s). Gene knockouts or repression can be modelled simply by eliminating the appropriate rows from the matrix or assigning extremely low activities to the enzyme(s) concerned.

DBsolve is also able to create and solve the ODEs and NAEs implicitly defined by the matrix. This makes it possible to discover the steady state of the system, and changes in pool sizes from defined initial conditions as the system approaches the steady state. The workbench's ability to fit a model to experimental data provides a mechanism for imposing constraints upon the model and determining parameter values within the system which cannot be assayed directly (e.g. binding or catalytic constants; see Appendix). The Implicit and Explicit Solvers allow the user to determine which parameter changes will result in desired perturbations to the system (such as maximal flux, and overproduction or depletion of specified system components or molecules), and bifurcational analysis provides a window on which components have the most effect on the model, or which parts and circumstances of the process may start to oscillate. For a comprehensive description of the use of ODEs and NAEs for biochemical pathway optimization and metabolic engineering, readers should refer to a paper that was published during revision of this work (Mendes and Kell, 1998).

Table 1. A comparison of DBsolve 5.00 with similar software tools

Software	Direct user access to mathematical model	Interactive plot and model charteditors	Steady-state and MCA analysis	Continuation and bifurcation analysis	Fitter and optimizer	Simultaneous fitting to NAE and ODE	Internet ready distributed object model
DBsolve 5.0	+	+	+	+	+	+	+
Gepasi 3.2	–	–	+	–	+	–	–
SCAMP 2.6	+	–	+	–	–	–	–
DynaFit 2.7	–	–	–	–	+	+	–

A comparison of DBsolve 5.00 with similar software tools (Table 1) shows that it offers the broadest set of facilities, which include the ability to access and modify the mathematical equations (in the SLV file), plot and chart editors, steady-state and metabolic control analysis, continuation and bifurcational analysis, general fitting/optimization, and simultaneous fitting to NAEs and ODEs. There is also a version that can be used across the Internet (through the unix-cgi version) and, at the time of writing, a public web site is being sought for the Internet version.

Data for DBsolve may be obtained from a variety of sources, including WIT, MPW and EMP. An Internet-aware version of the workbench is currently under development so that it can obtain data (including predesigned models) directly from specified web sites. Parsers are also being written to import data and models from other sources, such as SCAMP. The ease with which models can now be made also raises the issue of producing a database for storing and making them available to the scientific community.

Acknowledgement

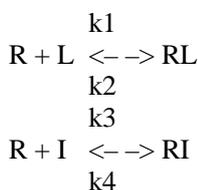
Part of the above work was carried out by I.G. under the auspices of a post-doctoral fellowship funded by Oxford Molecular Ltd.

Note added in proof

Since completing this work, a parser has been added to DBsolve for converting SCAMP (arguably the most widely used metabolic control analysis package) files into SLV format. Other enhancements are in progress with new releases anticipated every 6 months.

Appendix: SLV format

The SLV file format shall be outlined by describing a model of the effect of inhibitors upon receptor–ligand interactions, summarized by the following equations:



This model has been used with data from cell-binding inhibition assays (in which the absolute numbers/concentrations of receptor and ligand were unknown) for K_D and K_I determination. DBsolve's applicability to enzyme or other systems is entirely analogous. The model contains four processes, which are the forward and reverse reactions of the above equations, and five components (RL, RI, I, L, R).

These correspond, respectively, to the rows and columns of the stoichiometric matrix, which can be seen in Figure 2 and the SLV file below.

Every DBsolve object is named (shown in bold font below) and their values come after a '#'. The order in which the objects appear is crucial, as new or revised objects are appended so that a history of work on the model is maintained. Short internal comments on specified lines can appear after a semicolon, although there is a separate **Comments** object for extended free text information. The **Right Hand Sides** object contains all the equations generated or used by DBsolve. A user-defined 'explicit' formula, in this case the Motulsky equation (Motulsky and Mahan, 1984), is appended to this object.

In the **Right Hand Sides** and **Initial Values** objects, DBsolve also has a reserved set of parameters (where i and j correspond, respectively, to the row and column of the stoichiometric matrix) which are alphabetically defined as follows:

F[0]—the objective function for minimization during fitting or optimization.
 F[i]—the array of time derivatives (velocities) $dX[i]/dt$ of system variables.
 J[i]—the array of fluxes in a metabolic system.
 k_i —the rate constant for reaction/process i .
 P[k]—the array of system parameters for fitting or control analysis.
 Pool[m]—the array of component concentrations (each array element corresponds to a dependent variable X_j).
 RJ[i][j]—control coefficients of the flux J[i] of metabolic system upon parameter P[j].
 RV[i][j]—response coefficients of the velocity V[i] of the enzyme reaction upon parameter P[j].
 RX[i][j]—response coefficients of the concentration X[i] upon parameter P[j].
 V[i]—array of velocities (rate laws) of elementary reactions of the system.
 X[0]—system time.
 X[i]—array of variables for compound/entity concentrations in the ODE-NAE system.
 X_j —the dependent variable corresponding to component j of the stoichiometric matrix.

In most models, the number of independent variables (X[]) and pools (Pool[]) equals the number of system components (j). However, in this model, the users prefer to depict the results on a log scale, so an extra independent variable was created corresponding to the log drug concentration ($X[3] = \log(\text{Pool}[2])$). The SLV file in its entirety now follows:

DBsolve 5.00 1 November 98

Number of rows in matrix (Number of activities) #4
Number of cols in matrix (Number of compounds) #5
Initial System dimension #2
Selected Variable(Always TIME for ODE) #3

Maximum Value of the Selected Variable #1.000000e+04
Minimum Value of the Selected Variable #-1.000000e+01
Initial Step #1.000000e-03
Method Accuracy #1.000000e-05
Maximum Number of Method Steps #10000
1st Variable for plotting #X[0]
2nd Variable for plotting #RLTotal
Step for Numerical derivation #1.000000e-05
(Internal) Dbsolve Step WWW #401
Elementary Constant #1
FileOutput #0
TextOutput #1
DrawOutput #1
TransferLast #0
Stoichiometric Matrix
1 0 0 -1 -1
-1 0 0 1 1
0 1 -1 0 -1
0 -1 1 0 1
&
Elementary Constants
#1.581402e-05 1.657051e-05 1.926045e-05 1.153910e-05
1.091716e-05 1.071796e-05
Initial Concentrations
#1.000000e-05 1.000000e-05 2.000000e+00 1.000000e-06
1.000000e+00
Variables for FileOutput
#X[3] RLeq RL RL100 logI logIb tmp tmp tmp tmp tmp tmp
Output File Name
#extended1.txt
Experimental Data File Name
#r1.dat
Right Hand Sides

k4=10^(P[4]);Logarithm parameter has been added for fitting
k3=10^(P[3]);Logarithm parameter has been added for fitting
k2=10^(P[2]);Logarithm parameter has been added for fitting
k1=10^(P[1]);Logarithm parameter has been added for fitting
;k2=k1*Kd;if we would fix Kd from saturation experiments
Pool[2]=10^(X[3]);Due to Implicit Solver new logarithmic
independent variable has been added
X5=-X[1]-X[2]+Pool[1];=2.000020e+00 for R-Receptor
X3=-X[2]+Pool[2];=1.100000e-05 for I-Drug
X4=-X[1]+Pool[3];=1.000010e+00 for L-Ligand
V[1]=k1*X4*X5
V[2]=k2*X[1]
V[3]=k3*X3*X5
V[4]=k4*X[2]
V[5]=k5
F[1]=V[1]-V[2]
F[2]=V[3]-V[4]
F[3]=-V[3]+V[4]
F[4]=-V[1]+V[2]

F[5]=-V[1]+V[2]-V[3]+V[4]
RLTotal = X[1]*RTotal;Due to RL Total Model
RLTotNSB = X[1]*RTotal+NSB;Due to RL Total Model
; Motulsky explicit formula;
Ki=k4/k3
Kd=k2/k1
RLeq=RTotal*Pool[1]/
((Kd*(1+Pool[2]/Ki+Pool[1]/Kd)))+NSB;
Initial Values

X[1]=1.000000e-05;RL-Complex
X[2]=1.000000e-05;RI-complex
X3=1.000000e-06;Compound I-Drug has been assigned as
parameter X3
X4=1.000000e+00;Compound L-Ligand has been assigned
as parameter X4
X5=2.000000e+00;Compound R-Receptor has been as-
signed as parameter X5
k1=1.581402e-05;k1:R+L->RL
P[1]=log(k1);k1:R+L->RL
k2=1.657051e-05;k2:RL->R+L
P[2]=log(k2);k2:RL->R+L
k3=1.926045e-05;R3:R+I->RI
P[3]=log(k3);R3:R+I->RI
k4=1.153910e-05;k4:RI->R+I
P[4]=log(k4);k4:RI->R+I
Pool[1]=2.000020e+00;R-Receptor
Pool[2]=1.100000e-05;I-Drug
Pool[3]=1.000010e+00;L-Ligand
;Additional Parameters for models;
RTotal=400;Total Receptor
NSB=85;From saturation experiments
Kd=1.047837e+00;
; k2/k1 or From saturation experiments
X[3]=log(Pool[2]);Due to Implicit Solver new independent
variable has been added
P[1]=-3.022958e+00;;
P[2]=-3.135664e+00;;
P[3]=-5.833334e+00;;
P[4]=-4.721828e+00;;
Comments

The Number of Active P[i] for fitting
#4
Tolerance for fitting
#1.000000e-02
Step for fitting
#1.000000e-02
Use Implicit Solving Method Flag
#1
Use Experimental Data Flag
#1
Pathway Name

#Receptor-Ligand Interactions

Chart file Name

#rl.bmp

Title for X Axes

#log(k1)

Title for Y Axes

#log(k2)

Use Sensitivity coefficients

#0

>Reaction Names

#k1:R+L->RL

#k2:RL->R+L

#k3:R+I->RI

#k4:RI->R+I

>Compound Names

#RL-Complex

#RI-complex

#I-Drug

#L-Ligand

#R-Receptor

References

- Bezdeneynyh,A. (1973) Inzenernye metody sostavleniya uravnenii skorostei reakzii i rascheta kineticheskikh konstant. Chimija, Leningradskoe otdelenie.
- Comish-Bowden,A. and Hofmeyr,J.H. (1991) METAMODEL—A program for modelling and control analysis of metabolic pathways on the IBM PC and compatibles. *Comput. Appl. Biosci.*, **7**, 89–93.
- Doedel,E., Wang,X. and Fairgrieve,T. (1994) *AUTO94: Software for Continuation Problems in Ordinary Differential Equations*. Applied Mathematics Report, California Institute of Technology, Pasadena, CA.
- Dolgacheva,L.P., Goldstein,B.N., Goryanin,I.I. and Zinchenko,V.P. (1996) Influence Ca²⁺ ions on activity of pyruvate dehydrogenase complex in Ehrlich ascites tumor cells. *Biochemistry (Moscow)*, **6**, 1063–1068.
- Ehlde,M. and Zacchi,G. (1995) MIST: a user-friendly metabolic simulator. *Comput. Appl. Biosci.*, **11**, 201–207.
- Gear,C.W. and Petzold,L.R. (1984) ODE methods for the solution of differential algebraic systems. *SIAM J. Num. Anal.*, **21**, 716–728.
- Glass,L. and Mackey,M.C. (1988) *From Clocks to Chaos: The Rhythms of Life*. Princeton University Press, Princeton, NJ.
- Goldstein,B.N. and Goryanin,I.I. (1996) Modeling of oscillations for two activities of 6-phosphofructo-2-kinase/fructose-2,6-bisphosphatase in rat liver. *Mol. Biol. (Moscow)*, **30**, 976–983.
- Guckenheimer,J. and Holmes,P. (1983) *Nonlinear Oscillations, Dynamical Systems, Bifurcations of Vector Fields*. Springer-Verlag, Berlin.
- Hindmarsh,A.C. (1983) A systematized collection of ODE solvers. In Stepleman,R.S. *et al.* (eds), *Scientific Computing*. North Holland, Amsterdam, pp. 55–64. Also, <http://www.codiciel.fr/netlib/ode-pack/doc>.
- Hocker,C.J. (1994) Applying bifurcation theory to enzyme kinetics. *Methods Enzymol.*, **240**, 240–253.
- Holzhtutter,H.G. and Colosimo,A. (1990) SIMFIT: a microcomputer software toolkit for modelistic studies in biochemistry. *Comput. Appl. Biosci.*, **6**, 23–28.
- Hooke,R. and Jeeves,T.A. (1961) 'Direct search solution' of numerical and statistical problems. *J. Ass. Comput. Mach.*, **8**, 212–229.
- Khibnik,A., Kuznetsov,Y., Levitin,V. and Nikolaev E. (1993) Continuation techniques and interactive software for bifurcation analysis of ODEs and iterated maps. *Physica D*, **62**, 360–370.
- Levenberg,K. (1944) A method of solution of certain nonlinear problems in least squares. *Q. Appl. Math.*, **2**, 164–168.
- Marquardt,D.W. (1963) An algorithm for least square estimation of non-linear parameters. *SIAM J.*, **11**, 431–441.
- Mendes,P. (1993) GEPASI: a software package for modeling the dynamics, steady states and control of biochemical and other systems. *Comput. Appl. Biosci.*, **9**, 563–571.
- Mendes,P. (1997) Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. *Trends Biochem. Sci.*, **22**, 361–363.
- Mendes,P. and Kell D.B. (1998) Non-linear optimisation of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics*, **14**, 869–883.
- Motulsky,H.J. and Mahan,L.C. (1984) The kinetics of competitive radioligand binding predicted by the law of mass action. *Mol. Pharmacol.*, **25**, 1–9.
- Overbeek,R., Larsen,N., Smith,W., Maltsev,N. and Selkov,E. (1997) Representation of function: the next step. *Gene*, **191**, GC1–9.
- Reder,C. (1988) Metabolic control theory: a structural approach. *J. Theor. Biol.*, **135**, 175–201.
- Reich,J.G. and Selkov,E. (1981) *Energy Metabolism of the Cell. A Theoretical Treatise*. Academic Press, London.
- Sauro,H.M. (1993) SCAMP: a general-purpose simulator and metabolic control analysis program. *Comput. Applic. Biosci.*, **9**, 441–450.
- Schulmeister,T. and Selkov,E. (1977) Dynamic properties of a generalised Lotka-model. *Stud. Biophys.*, **65**, 121–123.
- Selkov,E. *et al.* (1996) The metabolic pathway collection from EMP: enzymes and metabolic pathways database. *Nucleic Acids Res.*, **24**, 26–29.
- Selkov,E. *et al.* (1997) The metabolic pathway collection: an update. *Nucleic Acids Res.*, **25**, 37–38.