<u>**I**</u>nsilico <u>**B**</u>iochemical <u>**RE**</u>action <u>**N**</u>etwork <u>**A**</u>nalysis

IBRENA

Manual for IBRENA Version 1.0

By

Gang Liu and Sriram Neelamegham

Department of Chemical and Biological Engineering

State University of New York, Buffalo, NY 14260, USA.

8 February 2008

Table of Content

1. Preface

1.1 Introduction

Novel high-throughput genomic and proteomic tools are allowing the integration of information from a range of biological assays into a single conceptual framework. This framework is often described as a network of biochemical reactions. In such an approach, independent rate constants for a given reaction network are determined using a range of biochemical assays. Algorithms are then designed and implemented in computer programs to integrate and analyze these data. Such an approach has been applied in diverse studies in the fields of signal transduction, protein glycosylation, blood coagulation etc.

To aid the analysis of complex biological reaction networks, we have developed a user-friendly program called "Insilico Biochemical REaction Network Analysis (IBRENA)". This program is written in Matlab^{TM} (The Mathworks, Inc., Natick, MA) with a Graphical User Interface (GUI). In order to reduce computation time, in some cases, ordinary differential equations (ODEs) are solved using Visual FORTRAN (HP-Compaq V6.6, Palo Alto, CA) with integrated IMSL library (Visual Numerics Inc., Houston, TX). This reduces the computational time required for time-consuming sensitivity analysis (described in Chapter 4) by ~5-fold. IBRENA is available from http://www.eng.buffalo.edu/~neel/IBRENA.htm. This application can be deployed on a PC using Windows.

IBRENA can be installed either as a stand-alone application or it can be used in the MATLAB environment. Simulation results in both cases can be graphed using IBRENA's built-in plot functions, or these can be exported in comma delimited files for analysis in Matlab or elsewhere.

- **<u>STAND ALONE APPLICATION:</u>** The stand alone application is easy to install and use. This does not require installation of either the SBML or SUNDIAL toolbox. All inputs for the stand-alone application are provided using .csv (comma separated value) input files. This partially installed version of the software can handle elemental reactions but not more complex reactions.
- **<u>FULL VERSION OF THE APPLICATION:</u>** The complete version of IBRENA that is run in the MATLAB environment can handle any general reaction network that is described by SBML (Systems Biology Markup Language) type input file and that can be processed using the SUNDIAL package. This then requires installation of the SBML and SUNDIAL toolbox to solve arbitrary reactions.

Detailed differences between the stand-alone and full-versions are listed in this chapter and also

mentioned in specific individual chapters of this manual.

Overall, IBRENA enables:

* Conversion of .csv format input files to SBML format output.

* *Insilico* dynamic simulation of reaction networks.

* Identification of critical rate-limiting or regulatory reactions (or species) using sensitivity analysis. Both standard forward sensitivity analysis and adjoint or backward sensitivity analysis can be performed using IBRENA.

* Further analysis of dynamic simulation or sensitivity coefficient data to decipher the connectivity between the reactions or species. Such analysis is performed using Principal Component Analysis (PCA) and Singular Value Decomposition (SVD).

* Model reduction by combining sensitivity analysis with flux analysis.

The schematic in Fig.1 illustrates the software implementation scheme. Input for the program can be provided using files in either SBML2.0 (Systems Biology Markup Language) or Excel Comma Separated Value (.csv) format. In the latter case, three independent files are uploaded including the: i) rate expression file, ii) initial concentration file, and iii) rate constant file. Detailed conventions used for these files are discussed in Chapter 2, and sample files are provided online. Once these files are loaded, the user can run either "Dynamic Simulation" or perform "Sensitivity Analysis". The former is designed to simulate the dynamic behavior of the reaction network in terms of changes in reactant and intermediate concentrations (C_i) with time. The latter calculates the scaled sensitivity coefficients, W_{ij}. C_i and W_{ij} data can be plotted using the software's built-in plot functions or they can be exported in comma delimited file format for analysis elsewhere. Since results from "Dynamic Simulation" (C_i) and "Sensitivity Analysis" (W_{ij}) can produce large data sets depending on the simulation conditions and time-scales considered, and since the concentrations of individual components may vary over several orders of magnitude depending on their natural abundance, analysis strategies are required to reduce the dimensionality of the data/matrix space. Thus, SVD and PCA analysis methods are implemented in the software with the goal of capturing the underlying patterns covered by the reaction network. As discussed below, while the focus of the program is on dry/simulated biochemical networks, SVD and PCA analysis methods in the current program can also be applied to analyze wet/experimental data sets, for example DNA microarray and protein-array based assays. Results from such multivariate analysis can either be plotted by our software or can be exported to local files. To reduce the complexity of the large reaction network, automated model reduction is also implemented in this software.

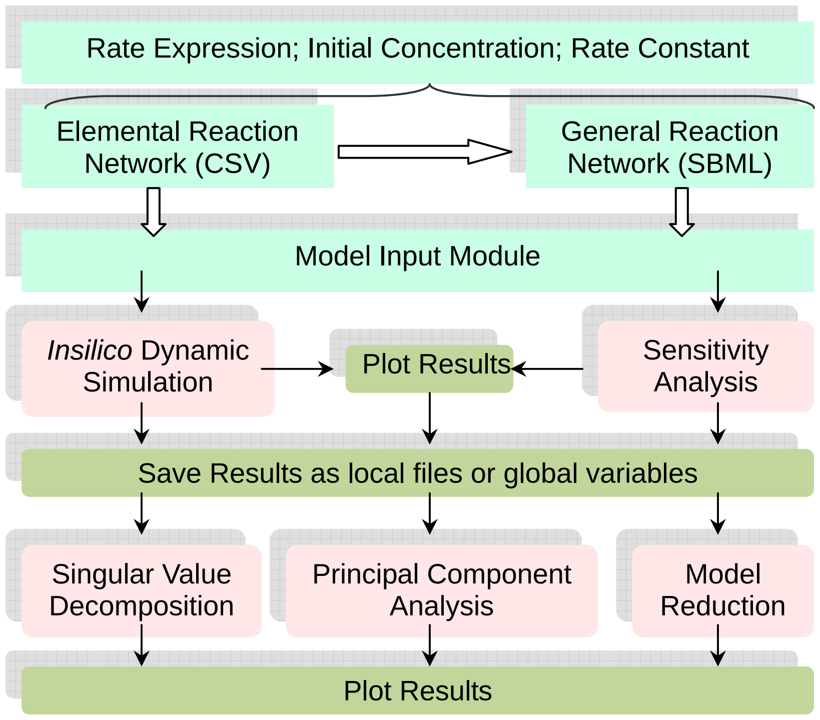

Rate Expression; Initial Concentration; Rate Constant
Elemental Reaction Network (CSV)
General Reaction Network (SBML)
Model Input Module
Insilico Dynamic Simulation
Plot Results
Sensitivity Analysis
Save Results as local files or global variables
Singular Value Decomposition
Principal Component Analysis
Model Reduction
Plot Results

Fig.1 Software implementation flow chart

1.2 Differences between the stand-alone and full versions of the application

The following chapters provide both the theory and details for implementation of the following operations: a) Loading input data (Chapter 2), b) Performing dynamic simulations (Chapter 3), c) Sensitivity analysis (Chapter 4), PCA analysis (Chapter 5), e) SVD analysis (Chapter 6) and f) Model reduction (Chapter 7). Sample data used for illustration of this program are from various sources in literature(Alter, Brown et al. 2000; Schoeberl, Eichler-Jonsson et al. 2002; Liu, Swihart et al. 2005).

A comprehensive list of differences between the stand-alone and full-versions of the IBRENA application as it pertains to individual chapters/functions are given below. This information is reiterated in specific sections of the manual in the individual chapters. These differences between the stand-alone and full versions of IBRENA arise from two features of the full version that are not compatible with the MATLAB compiler: i) The full version requires the use of symbolic math toolbox; ii) The full version requires generation and calling of m-files during individual computations, and this is not handled by the stand-alone application.

- **Model Input (Chapter 2)**: All functions can be fully implemented in both the stand-alone and full versions of the software. This includes the conversion of .csv files to SBML format.
- **Dynamic simulation (Chapter 3)**: Stand alone application will run .csv input only. Full-version will simulate both .csv and SBML inputs. All plot options are implemented identically in both versions of the software.
- **Sensitivity analysis (Chapter 4)**: Forward sensitivity analysis (long and express calculations) can be run in both versions of the software except that the stand-alone application is limited to .csv input files similar to Chapter 3. In addition, m-files for adjoint sensitivity analysis can only be generated using the full version of the software. All plot options are implemented identically in both versions of the software.
- **Principal Component Analysis (Chapter 5):** All functions and options specified here are implemented identically in both the stand-alone and full-versions of the software.
- **Singular value decomposition (Chapter 6):** All functions and options specified here are implemented identically in both the stand-alone and full-versions of the software.
- **Model reduction (Chapter 7):** This is currently only possible in the full version of the software.

1.3 Installation Instructions

The following sections provide instructions for installation of the stand-alone application (Section 1.3.1) and full versions (Section 1.3.2) of the software. Depending on user needs, one of the two protocols may be followed.

1.3.1 For stand-alone application

1. Self Extract IBRENAStandAlone.exe file in the desired application directory. Three folders will appear: (1) IBRENA (2) MCR (3) DLL. Example files are located within the IBRENA folder.

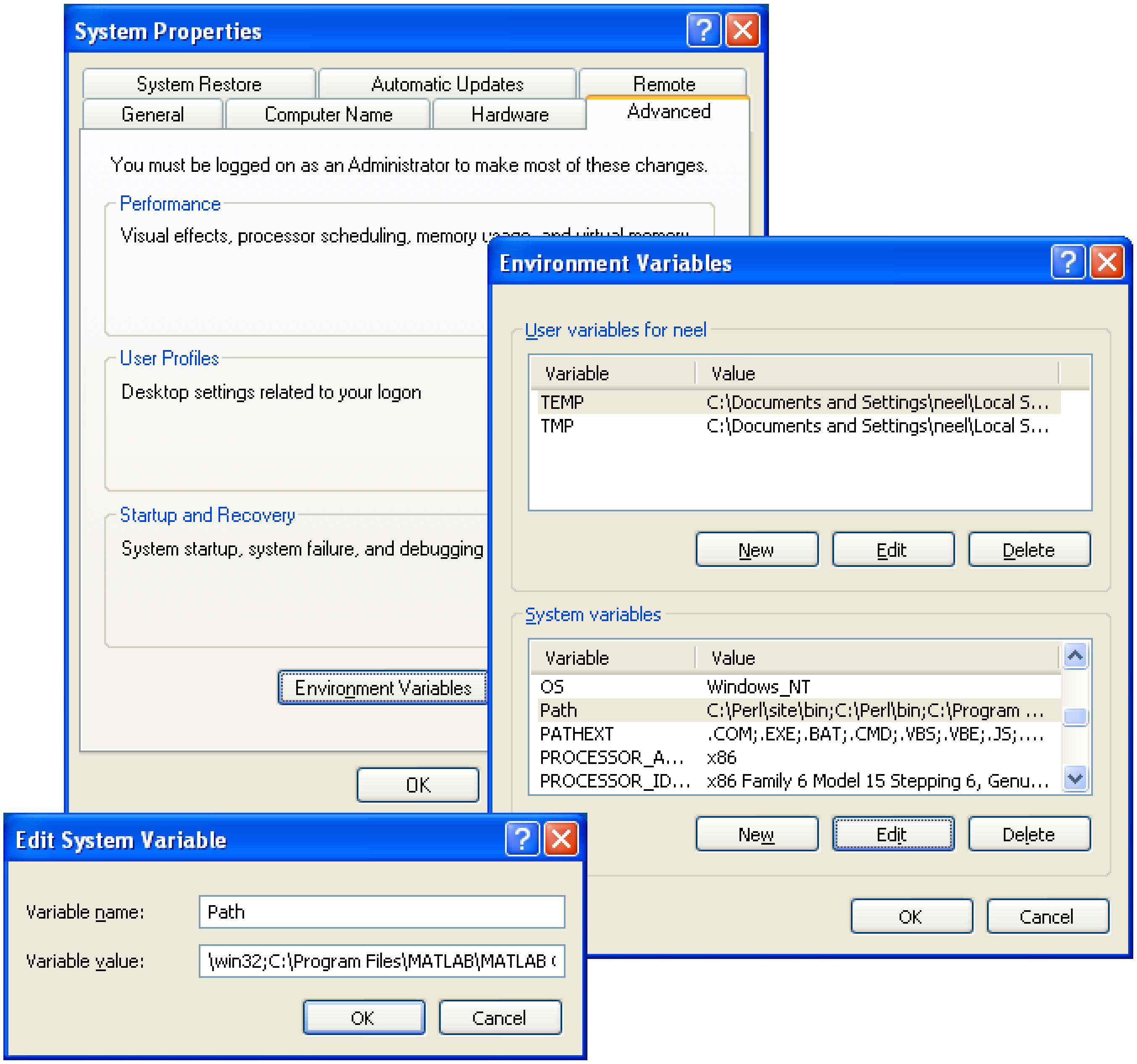
2. Install the MCR (MATLAB Component Runtime) by running the MCRInstaller which is located in the MCR directory. This can be done by either double-clicking on the MCRInstaller icon, or by using an equivalent command at the dos-prompt. Software installation will take a few minutes. For more information on running the MCR Installer utility, see www.mathworks.com website or MCRInstaller.txt which is located in the same directory.

3. Typically, installation of MCR automatically modifies the dynamic path to include its host-directory say: "C:\Program Files\MATLAB\MATLAB Component Runtime\v74\runtime\win32". The inclusion of this path can be verified by typing "path" at the dos-prompt. If, however, it is observed that this path is not included during installation, it may be necessary to include this manually as described next.

4. Include another directory "C:\Program Files\MATLAB\MATLAB Component Runtime\v74\bin\win32" also to the system path. This is done by right-clicking the “My Computer” icon on the desktop and choosing “Properties”. Under the “Advanced” option, choose “Environment Variables”. Then “edit” the “path” command in the “system variable” portion. In this window, which is shown below, add a semi-colon at the end of the path command followed by the additional path, i.e type “; C:\Program Files\MATLAB\MATLAB Component Runtime\v74\bin\win32” (see figure below for additional details, if necessary).

5. Copy all .dll files in the extracted "DLL" folder into c:\windows\system32

6. Run IBRENA.exe to begin software implementation. Program initiation can take a minute or two when loaded the first time. Subsequent runs will take less time. Note: A DOS-window appears in the background along with the GUI. The user should leave this DOS-window open while running IBRENA.



System Properties

You must be logged on as an Administrator to make most of these changes.

Performance
Visual effects, processor scheduling, memory usage, and virtual memory

User Profiles
Desktop settings related to your logon

Startup and Recovery
System startup, system failure, and debugging

Environment Variables

User variables for neel

Variable	Value
TEMP	C:\Documents and Settings\neel\Local S...
TMP	C:\Documents and Settings\neel\Local S...

System variables

Variable	Value
OS	Windows_NT
Path	C:\Perl\site\bin;C:\Perl\bin;C:\Program ...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86
PROCESSOR_ID...	x86 Family 6 Model 15 Stepping 6, Genu...

Edit System Variable

Variable name: Path

Variable value: \win32;C:\Program Files\MATLAB\MATLAB (

1.3.2 For full version of the application

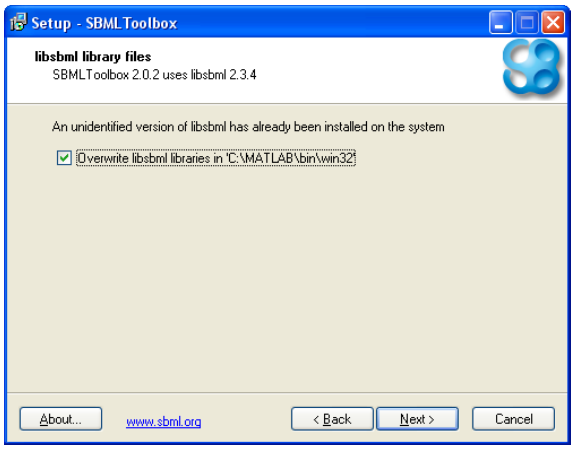
1.3.2.1 Self-extract all files in the package by double-clicking IBRENAFullVersion.exe to user-desired application directory (say c:\IBRENAfullversion). Four folders will appear: 1) IBRENA; 2) win32; 3) toolbox; 4) Examples. Three other files also appear: “Readme.txt”, “Manual.doc” and “install_IBRENA.m” file.

1.3.2.2 Start MATLAB.

1.3.2.3 Install SBML toolbox 2.02 (Keating, Bornstein et al. 2006) and Sundial toolbox 2.3.0 (Hindmarsh, Brown et al. 2005): These two toolboxes are provided with IBRENA (Files “SBMLToolbox-2.0.2-setup-win32.exe” and “Sundials-2.3.0.tar.gz”) for the convenience of the user. IBRENA1.0 has been tested using these toolboxes only. Other versions of the toolbox may be downloaded from http://sbml.org/software/sbmltoolbox and http://www.llnl.gov/CASC/sundials/download/download.html. However, software compatibility cannot be guaranteed for these other versions. We gratefully acknowledge the groups from University of Hertfordshire, Hatfield, UK (Keating S.M. and colleagues), and Lawrence Livermore National Laboratory (Hindmarsh A.C. and colleagues) for graciously providing these open-source codes for the benefit of the community. A brief introduction for installation of each toolbox is given as below. More details can be found at the aforementioned websites.

1.3.2.3.1 Installation of SBML toolbox 2.0.2 in MATLAB

i) Double-click the executable file “SBMLToolbox-2.0.2-setup-win32.exe” and install SBMLToolbox to the user-desired directory (for example: “<MATLABToolboxDirectory>\SBML” which may have the path say c:\Matlab\toolbox\SBMLToolbox-2.0.2). Please note SBML toolbox 2.0.2 uses libSBML 2.3.4. This earlier version is not compatible with the pre-released libSBML 3.0. Choose the option “write libsbml libraries to c:\windows\system32” during installation. If you previously installed libSBML 3.0, during installation process please choose to “overwrite libsbml libraries in <MATLABInstallationDirectory>\bin\win32” as illustrated in the figure below.


Setup - SBMLToolbox

libsbml library files
SBMLToolbox 2.0.2 uses libsbml 2.3.4

An unidentified version of libsbml has already been installed on the system

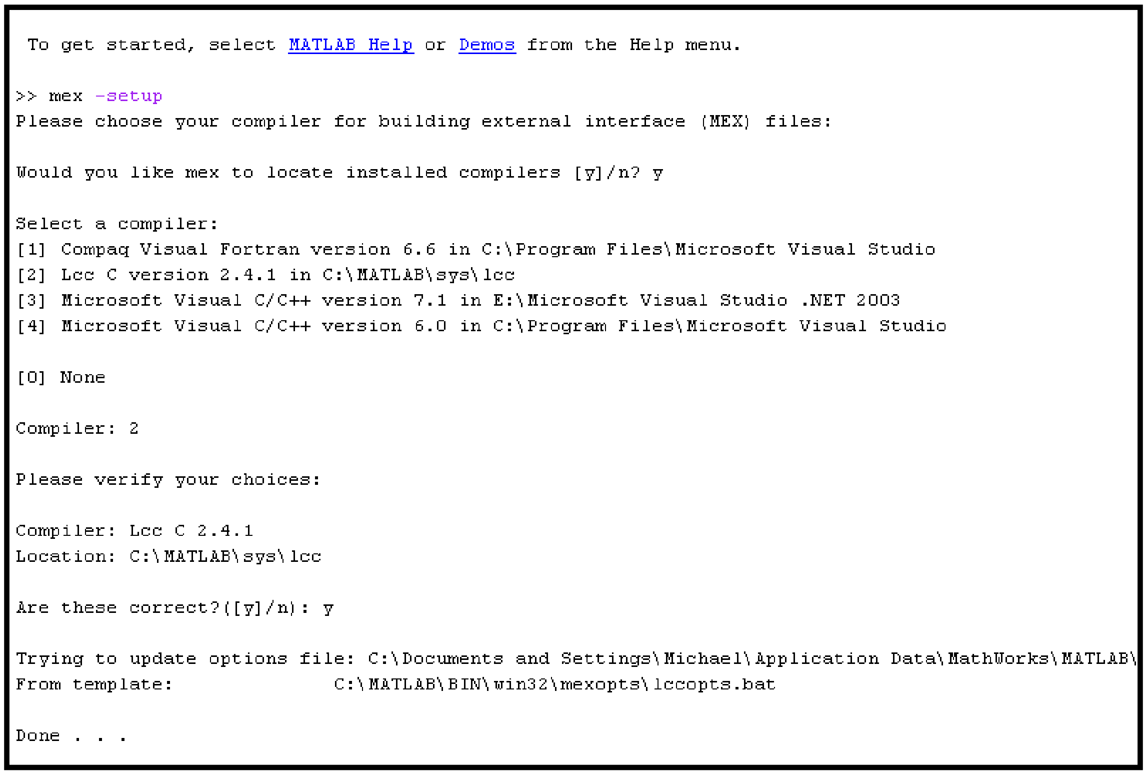
☑ Overwrite libsbml libraries in 'C:\MATLAB\bin\win32'

This step copies the .dll files and m-files to desired directory.

ii) Change MATLAB current path to "<SBMLToolboxInstallationDirectory>\ SBMLToolbox-2.0.2 \toolbox". This can be . Type "install" in the command window and press enter to start installation. Briefly, this step adds the SBML directory to the MATLAB search path. Once the installation succeeds, the "Install successful" message will display in the MATLAB command window. At this point, type 'y' to close MATLAB, or 'n' otherwise to continue with MATLAB.

1.3.2.3.2 Installation of Sundial toolbox 2.3.0 in MATLAB

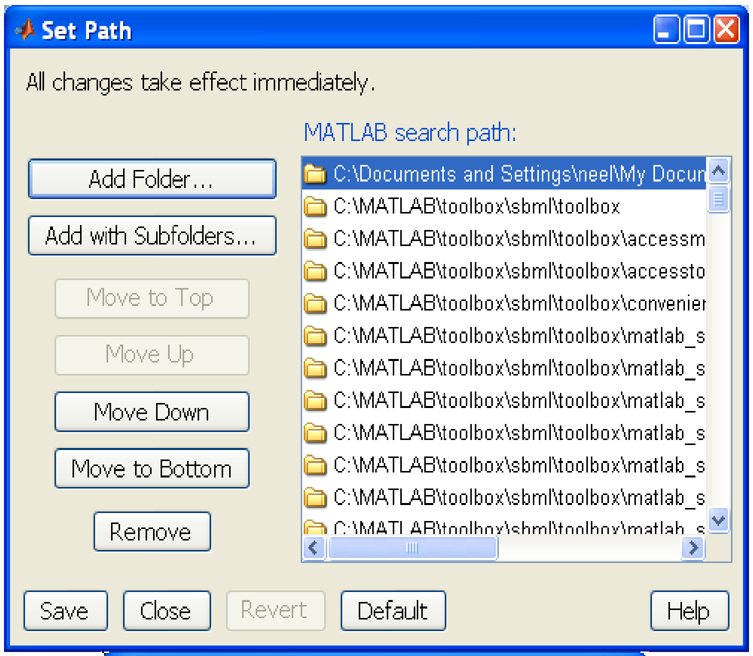
i) Extract all the files from Sundials-2.3.0.tar.gz (using WINRAR, WINZIP or other suitable software) to the user-defined directory (for example, c:\SUNDIAL).


```
 To get started, select MATLAB Help or Demos from the Help menu.

>> mex -setup
Please choose your compiler for building external interface (MEX) files:

Would you like mex to locate installed compilers [y]/n? y

Select a compiler:
[1] Compaq Visual Fortran version 6.6 in C:\Program Files\Microsoft Visual Studio
[2] Lcc C version 2.4.1 in C:\MATLAB\sys\lcc
[3] Microsoft Visual C/C++ version 7.1 in E:\Microsoft Visual Studio .NET 2003
[4] Microsoft Visual C/C++ version 6.0 in C:\Program Files\Microsoft Visual Studio

[0] None

Compiler: 2

Please verify your choices:

Compiler: Lcc C 2.4.1
Location: C:\MATLAB\sys\lcc

Are these correct?([y]/n): y

Trying to update options file: C:\Documents and Settings\Michael\Application Data\MathWorks\MATLAB\
From template:              C:\MATLAB\BIN\win32\mexopts\lccopts.bat

Done . . .
```

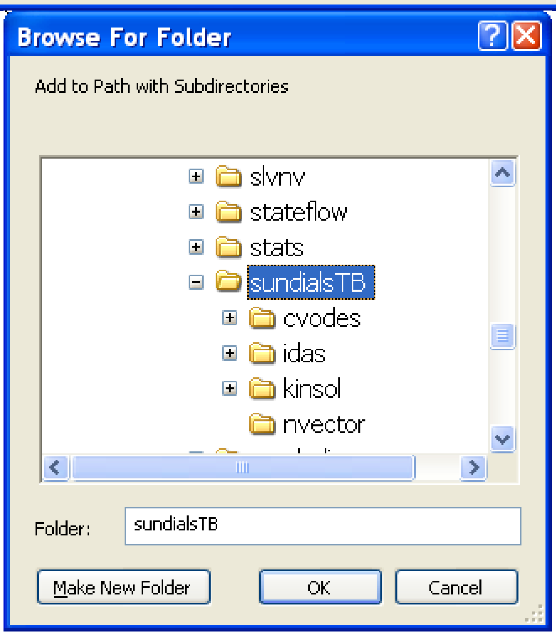
ii) Change MATLAB current path to “c:\SUNDIAL\sundialsTB” in the MATLAB window. Type “mex -setup” at the command line. This allows the user to choose the compiler prior to SUNDIAL installation. Using this option choose the LCC compiler as illustrated above.

Following this, type “install_STB” in the command window and press enter to start installation. The first step of this installation process involves the compilation of Sundial C source codes into .dll (dynamic link library) files. This requires the LCC compiler provided by MATLAB or another C/C++ compilers installed in MATLAB by the user. Proper setting up of the MEX options in MATLAB (as discussed above) is necessary for smooth installation. The second step of the installation process involves the copying of all m-files and dll files of the Sundial Toolbox in MATLAB to the user-defined directory. This step is initiated with the prompt display “Install toolbox?(y/n)” in the command window. Note here that only the input “y” (case-sensitive) from the user will start the process of file transfer while other inputs from the user will abort installation. At this step, the user is requested to input the location of the Toolbox installation directory. For convenience we suggest the use of the preexisting MATLAB toolbox directory (typically c:\MATLAB\toolbox).

iii) Upon completion of the installation procedure, the MATLAB command window including the following message: “The installation directory was installed in <User-defined directory>”. Following this, the user needs to add the toolbox installation directory in the MATLAB search path. This is done in one of two ways: i) By typing ‘ToolboxPath = path(genpath(‘c:\matlab\toolbox\sundialsTB’),path);’ followed by ‘addpath(ToolboxPath)’ and then followed by the command ‘savepath’ to save this setting. ii) Alternatively, the user can choose the setpath window from the ‘File’ pull-down menu in MATLAB (see below). Following this select the <User-defined directory> (e.g. C:\MATLAB\toolbox\sundialsTB”) using the “Add with Subfolders…” option and browse to the desired directory. Finally save the new path/settings.

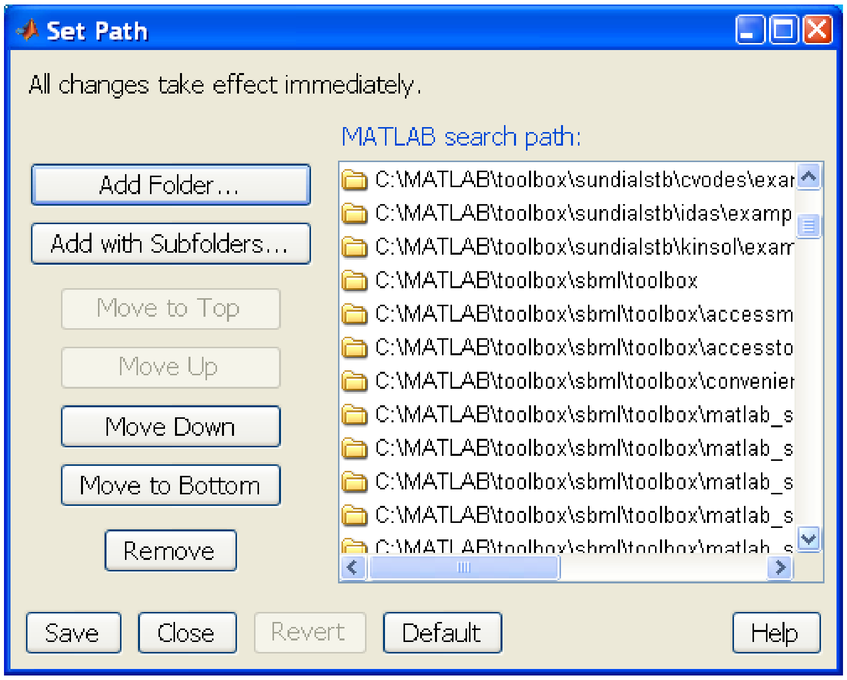


```
C:\Documents and Settings\neel\My Docun
C:\MATLAB\toolbox\sbml\toolbox
C:\MATLAB\toolbox\sbml\toolbox\accessm
C:\MATLAB\toolbox\sbml\toolbox\accessto
C:\MATLAB\toolbox\sbml\toolbox\convenier
C:\MATLAB\toolbox\sbml\toolbox\matlab_s
C:\MATLAB\toolbox\sbml\toolbox\matlab_s
C:\MATLAB\toolbox\sbml\toolbox\matlab_s
C:\MATLAB\toolbox\sbml\toolbox\matlab_s
C:\MATLAB\toolbox\sbml\toolbox\matlab_s
C:\MATLAB\toolbox\sbml\toolbox\matlab_s
```



Folder: sundialsTB

iv) Restart MATLAB to complete installation. Upon restarting MATLAB and viewing the search paths, please verify that both sbml and subdials toolbox subdirectories are listed as shown below:



C:\MATLAB\toolbox\sundialstb\cvodes\exar
C:\MATLAB\toolbox\sundialstb\idas\examp
C:\MATLAB\toolbox\sundialstb\kinsol\exam
C:\MATLAB\toolbox\sbml\toolbox
C:\MATLAB\toolbox\sbml\toolbox\accessm
C:\MATLAB\toolbox\sbml\toolbox\accessto
C:\MATLAB\toolbox\sbml\toolbox\convenier
C:\MATLAB\toolbox\sbml\toolbox\matlab_s
C:\MATLAB\toolbox\sbml\toolbox\matlab_s
C:\MATLAB\toolbox\sbml\toolbox\matlab_s
C:\MATLAB\toolbox\sbml\toolbox\matlab_s
C:\MATLAB\toolbox\sbml\toolbox\matlab_s

1.3.2.4 Change MATLAB current path to the “<IBRENAInstallationDirectory>” in the above step (e.g c:\IBRENAfullversion). Then type “install_IBRENA” in MATLAB command window. This initiates the following steps: i) Removes all variable from MATLAB workspace and closes all figures; ii) Checks for SBML and SUNDIAL toolbox installation; iii) Adds all m-files to MATLAB search path. Following this, the GUI interface will appear shortly, and it will be fully functional. To start the program at a later time, after software installation, simply types “IBRENA” at the MATLAB command prompt.

1.4 Example data sets

Several sample data sets are provided with the program. These data files are located in the "Examples" folder extracted from the IBRENA.exe file. A few of these input files are highlighted below:

The first data set is a reaction network model of EGF induced activation with 125 reversible reactions and 94 species (Schoeberl, Eichler-Jonsson et al. 2002; Liu, Swihart et al. 2005). Appendix 1 located at the end of this manual provides the schematic of this reaction network and tables, with list of model parameters and initial conditions. These data provided in the figure and tables are written into both .csv format files and SBML2.0 format files ("EGFNetwork.xml") in the "Examples" subfolders "CSVFile" and "SBMLFile". Sensitivity analysis results for this model are also provided in the folder "SSC_EGFModel".

A second .xml file included in the "Examples\SBMLFile" folder called "EGFNetwork_dbconstraint.xml" is identical to "EGFNetwork.xml" only some of the reaction rate parameters and initial conditions are changed according to Table below. These changes have been made so that the network model parameters are constrained by the detailed balance equations (Kholodenko, Demin et al. 1999; Yang, Bruno et al. 2006; Ederer and Gilles 2007). Placing these constraints does not markedly affect the conclusion of our previous paper (Liu, Swihart et al. 2005) or results presented in the remainder of this manual, which uses "EGFNetwork.xml".

Rate constant	EGFNetwork_dbconstraint.xml*	EGFNetwork.xml*
k_r (reactions 16 and 63)	0.165 s^{-1}	0.275 s^{-1}
k_f (reaction 33)	0.1 s^{-1}	0.2 s^{-1}
k_r (reactions 41, 83)	0.1716 s^{-1}	0.0429 s^{-1}
k_r (reaction 35)	1e5 $nM^{-1}.s^{-1}$	4.5e6 $nM^{-1}.s^{-1}$

Initial Condition	EGFNetwork_dbconstraint.xml	EGFNetwork.xml
Grb2-SOS	31219 #/cell	35867 #/cell
SOS	4781 #/cell	132 #/cell
Grb2	58781 #/cell	54133 #/cell

* In SBML files provided with the package, the units of the first and second order reaction rate constants are min^{-1} and $(\#/cell)^{-1}.min^{-1}$ respectively. These values are derived from data presented in the above table.

The second data set contains modified gene expression data from the studies by Alter et al. (Alter, Brown et al. 2000). These are also located in the "Alter'sGene…" folder.

The third data set contains the MAPK reaction Network, "MAPK.xml" (Machne, Finney et al. 2006). This file is used as a test case to confirm compatibility of this application with SBML 1.0 format files.

1.5 Limitation of IBRENA

IBRENA supports most features in SBML Level 1 and Level 2. Features that are currently not supported are listed below, as they pertain to specific features/functions of the software:

i) Dynamic Simulation

- Unit definitions in SBML Level 1 and Level 2 format can not be handled by IBRENA. This is a common problem with most other software that are available to the community. This problem will be overcome after further development of the language and upon incorporation of additional toolboxes.

ii) Sensitivity Analysis/Model Reduction

- Unit definitions in SBML Level 1 and Level 2 format can not be handled, as discussed above.
- Events, assignment rules and functions in SBML Level 1 and Level 2 format can not be handled. A warning message box will appear and then the run of sensitivity analysis will stop.
- Multi-compartmental model defined in SBML Level 1 and Level2 is not supported.

Note: Every effort has been taken to provide user friendly pop-up messages for errors and warnings. Additional messages will be incorporated in future versions of the program.

2 Starting program and loading model input data

2.1 Theory

Two types of reaction networks are handled by the software. These include the elemental reaction network (section 2.1.1) and the general reaction network (section 2.1.2). While it is true that the elemental network is but a subset of the general network, we present this in two separate sections for the sake of clarity. The elemental reaction networks described next demonstrate how input to IBRENA can be loaded using .csv files. Section 2.1.2, general reaction network, details methods used to upload SBML input files.

2.1.1 Elemental Reaction Network

Elemental reaction networks are represented by Eq. 1 below. In such networks, the user generally arbitrarily specifies an index for each species and reaction based on convenience, typically with the starting reactants/reactions having the lowest number and species/reactions closer to the system output having larger numbers. In IBRENA, as currently implemented, a maximum of three reactants are allowed to react in each elemental reaction to form up to three products. Such flexibility should be sufficient for most biological systems. In Eq. 1, the velocity of the i^{th} reaction (v_i) is shown to be the net sum of the i^{th} forward and reverse reaction, r_i^f and r_i^r. k_i^f and k_i^r correspond to the rate constants for these reactions, C_l is the concentration of the l^{th} species, and μ_{il}^f and μ_{il}^r refer to the reaction order of the i^{th} reaction with respect to the l^{th} species.

$$v_i = r_i^f - r_i^r = k_i^f \prod_{l=1}^{m} C_l^{\mu_{il}^f} - k_i^r \prod_{l=1}^{m} C_l^{\mu_{il}^r} \tag{1}$$

Using Eq. 1 as a starting point, the data on reaction velocities can also be specified in terms of the rates of formation or consumption of the l^{th} species (C_l). Such information can be written in matrix format (**bold lettering**) as shown in Eq. 2a. Here, the stoichiometric coefficients for the reaction network are represented by $\boldsymbol{\alpha}^T$. $\boldsymbol{C_0}$ is an array containing the initial concentrations of all the reactant species starting with species 1. $\boldsymbol{k^f}$, $\boldsymbol{k^r}$ and $\boldsymbol{v}$ contain data on rate constants and velocities.

$$\begin{cases} \dfrac{d\mathbf{C}}{dt} = \mathrm{f}(\mathbf{k^f}, \mathbf{k^r}, \mathbf{C}) = \boldsymbol{\alpha}^{\mathbf{T}} \cdot \mathbf{v} \\ \mathbf{C}(t=0) = \mathbf{C_0} \end{cases} \tag{2a}$$

2.1.2 General Reaction Network

Besides elemental reaction network, the reaction velocity can be described by other type of enzyme kinetics, such as Michaelis–Menten kinetics, allosteric enzyme kinetics *etc.* Eqn.2a can thus be written in more general format (Eq. 2b). Such input to IBRENA is provided using SBML2.0/1.0 format input files.

$$\begin{cases} \frac{d\mathbf{C}}{dt} = \mathrm{f}(\mathbf{k}, \mathbf{C}) = \boldsymbol{\alpha}^{\mathbf{T}} \cdot \mathbf{v} \\ \mathbf{C}\ (t = 0) = \mathbf{C}_{\mathbf{0}} \\ \mathrm{v} = \mathrm{g(k,\ C)} \end{cases} \tag{2b}$$

In the above expression, ***k*** refers to all the enzymatic parameters in the enzyme velocity expression g(k,C).

2.2 Input File Format

2.2.1 CSV Format (Elemental Reaction Network)

In the current software, the reactants and products in each elemental reaction are specified in the "rate expression file". This file essentially specifies μ_{il}^{f} and μ_{il}^{r} in Eq. 1 above. Similarly, the "rate constant file" specifies k_i^{f} and k_i^{r} values, and the "Initial Concentration" file specifies $\boldsymbol{C_0}$. All files are created using Microsoft Excel and saved in Comma Separated Value (.csv) format.

Below are sample lines from the rate expression file. Here, starting with the second row, the first three columns specify the reactant species while the next three are the products. For example, in the first reaction, species 1 and 2 combine to form product 3:

Species (1) + Species (2) $\rightarrow$ Species (3)

In the second reaction, two units of reactant 3 combine to form products 4 and 5:

Species (3) + Species (3) $\rightarrow$ Species (4) + Species (5)

As seen, if a reaction only has 2 reactants or forms less than 3 products, the user must specify these reactants and products, while leaving the remaining cells either blank or writing in the number zero. A sample rate expression file for the EGF (Epidermal Growth Factor) network from (Schoeberl, Eichler-Jonsson et al. 2002; Liu, Swihart et al. 2005) is provided with the software. This reaction network has 125 reversible reactions and 94 reactants/species.

REACTANT 1	REACTANT 2	REACTANT 3	PRODUCT 1	PRODUCT 2	PRODUCT 3
1	2		3	0	0
3	3		4	5	0

The “initial concentration file” lists all species name and their initial concentrations. This file has two columns, with the first column containing the species name (starting with species 1 onwards), and the second column containing the species initial concentration. Here too, similar to the rate expression file above, IBRENA reads data starting with the second row with the first row being a comment statement. A sample initial concentration file for the EGF network is provided. If the cell in “protein initial concentration” column is blank, the software will assume the initial concentration is zero.

Protein Name	Protein Initial Conc
EGF	4.82E+06
EGFR	36888
EGF-EGFR	0

In the “rate constant file”, k_i^f data appears in the first column and k_i^r in the second (see below). The first row is a comment statement while each subsequent row provides rate data for individual elemental reactions starting with the first reaction in the second row, proceeding to the second reaction in the third row and so on. A sample rate constant file for the EGF network is provided with the software.

k_f	kr
2.99e-6	0.2304
0.001	6
60	0.6

Note that the units of protein concentration in the above example are *molecules/cell*, first order rate constants are in */min*, and the second order rate constants are in *(molecules/cell)$^{-1}$/min*.

2.2.1 SBML Format (General Reaction Network)

IBRENA supports input files in the SBML 1.0/2.0 (System Biology Markup Language, level 2, version 1) input format. In general, SBML defines standard rules for writing systems of biochemical reactions using XML, the eXtensible Markup Language. The structure and facilities for model definitions using SBML can be found at the website http://sbml.org/index.psp. The SBML file for the EGF-induced signaling network model has been provided as a test case for SBML 2.0. MAPK.xml file is the test case for SBML 1.0.

While we have validated the input file (EGFNetwork.xml) used by IBRENA using the online tools provided at sbml.org, we also note that our software as currently implemented does not exploit

the full functionality of SBML Level 2.0 and Level 1. Specifically, reaction networks are currently modeled as a single compartment model. Further, we do not account for different species units in the program as currently implemented. These features will be incorporated in future versions of the software. The other limitations of IBRENA are listed in Chapter 1.5.

2.3 Notes on program output

The units of the output of the program depend on the input units. For example, if the concentration of input parameters is in *molecules/cell*, output concentrations following dynamic simulation (Chapter 3) are also in *molecules/cell*. All results of Sensitivity Analysis, PCA and SVD (Chapters 4-6) are presented in dimensionless form.

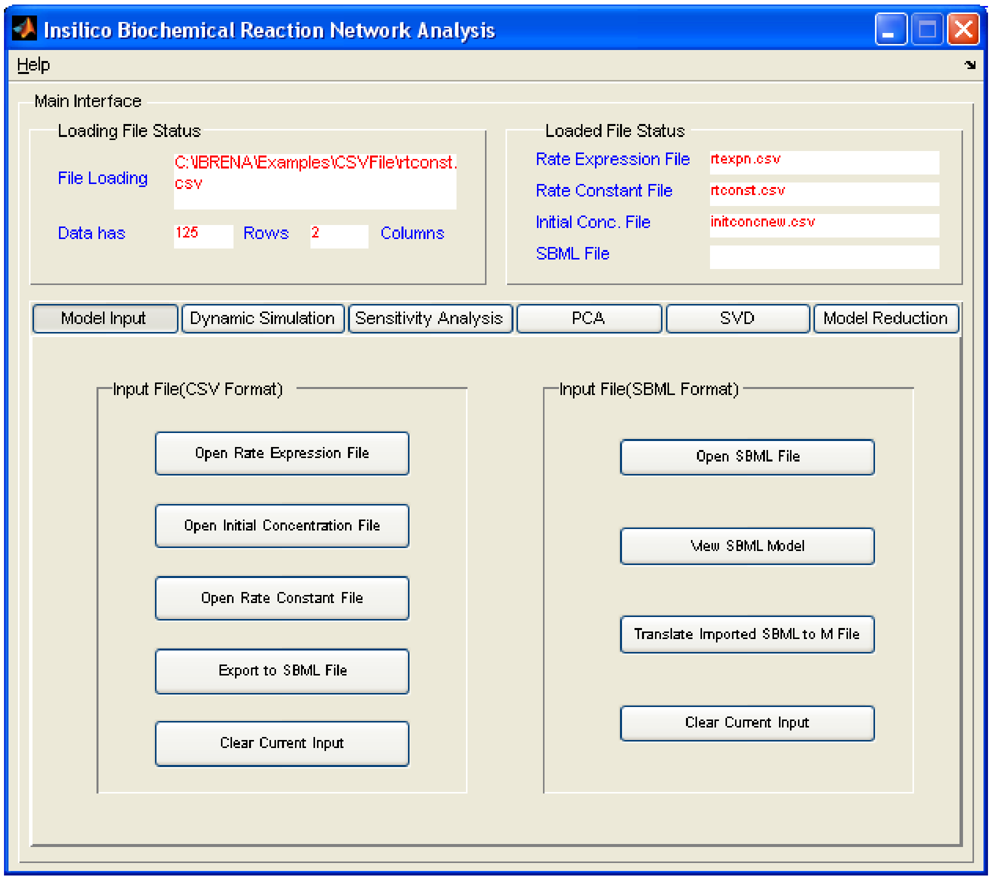
Plots are generated in this software in Matlab format files. These can be used at a later time in Matlab. These figures can also be saved in a variety of other file formats including .jpg, .pdf etc. (using the "save as" function) for archiving purposes.

2.4 Implementation

Once the program starts, the software interface will appears as below. Both the stand alone application and MATLAB versions have identical graphical user interfaces and all functions specified in the "Model input" tab can be fully implemented in both versions of the software.

The Help menu leads to a link that contains this manual.

The standalone version of IBRENA is suitable for analysis of elemental reaction networks. This is the most common representation used in biological systems. Computations for this component are performed efficiently and this does not require installation of either MATLAB or other toolboxes on the PC. Use of SBML2.0 type inputs requires the installation of MATLAB and associated toolboxes (SBML, SUNDIALS) as discussed in Chapter 1.


Insilico Biochemical Reaction Network Analysis

Main Interface

Loading File Status

File Loading: C:\IBRENA\Examples\CSVFile\rtconst.csv

Data has 125 Rows 2 Columns

Loaded File Status

Rate Expression File: rtexpn.csv

Rate Constant File: rtconst.csv

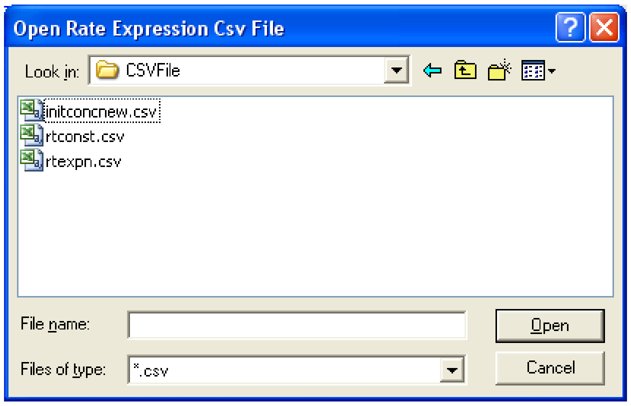
Initial Conc. File: initconcnew.csv

SBML File:

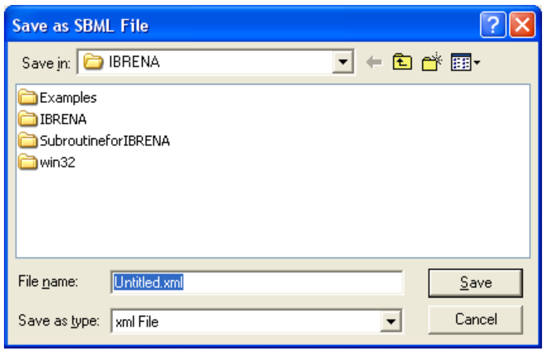
The top half of the interface is termed the “Main interface” and it provides data on the status of the input files that have been read. Details on the last file that was read is provided in the left half of this interface along with information on the size of this files i.e. number of rows and columns. For example, when the sample “rtexpn.csv file (located in EGF Model Data folder)” for the EGF network is read, the number of columns is specified to be 125 which correspond to the number of reactions in this network. There are 6 columns in this file which corresponds to the three reactants and three products. Similarly, reading “rtconst.csv” yields 125 rows and 2 columns, and the “initconc.csv” has data in a single column for all 94 species initial concentrations initially. A sample screen when reading the “initconc.csv” file is shown above.

The bottom half of the interface contains tabs that allow the user to provide inputs to the model or to perform various analysis steps.

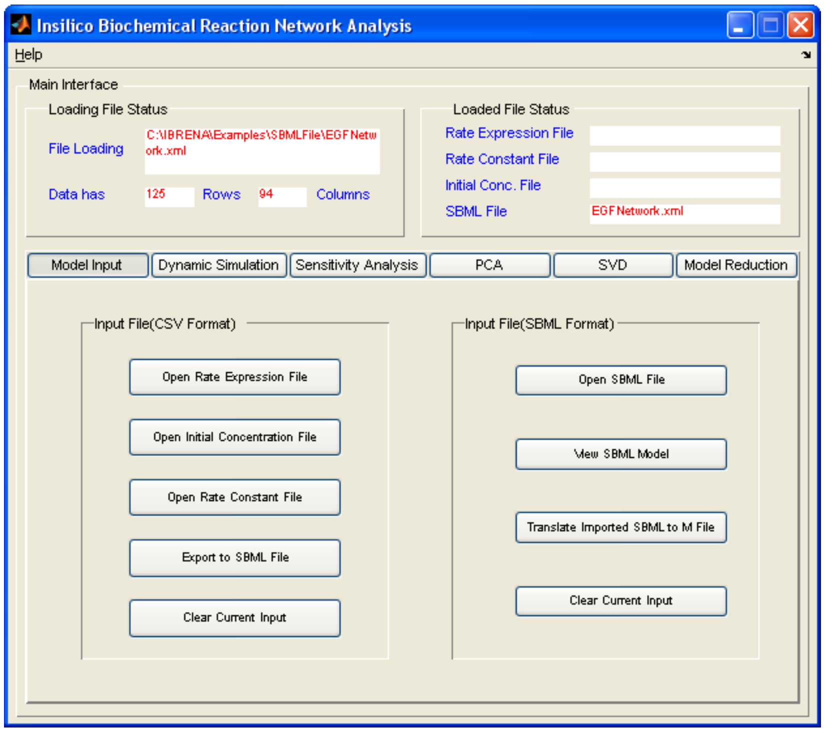
Under “Input files” the three data files of interest can be uploaded. An additional button is also provided to clear the loaded data if necessary. When any of the input file buttons is selected in order to load a data file, a pop up window (see below) appears that allows the user to browse to the desired file.



IBRENA provide another function that exports the .csv file to SBML 2.0 file format. This function can be implemented once all the CSV files are loaded. Upon clicking on the "Export to SBML File" button, after a few seconds, the software will ask the user to select a directory and filename to save .xml file. The pop up window is shown below.



Besides reading .csv format files, IBRENA also supports SBML format files for biological reaction networks. Upon pressing the "Open SBML File" button on the right-hand side of the "Model Input" window, SBML format files can be loaded via the Windows popup feature shown above. The number of rows and columns in the "Loading File Status" window now correspond to the number of reactions and reactants respectively, and the "Loaded File Status" window displays the name of the input file. A sample SBML format input file is provided with this software. This file ("EGFNetwork.xml") contains the input for the EGF signaling network model (Appendix 1). The figure below shows the screen-shot after input of the latter SBML file.

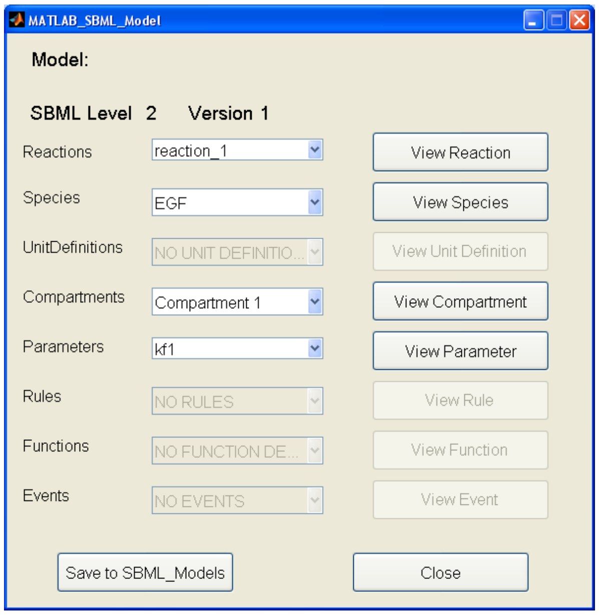


File Loading: C:\IBRENA\Examples\SBMLFile\EGFNetwork.xml

Data has 125 Rows 94 Columns

SBML File: EGFNetwork.xml

Upon input of SBML file, we can click on the "View SBML Model" to view SBML model structure as shown in Figure below. This allows the user to verify the data in the input SBML file including reaction rate constants, reaction pathways, initial concentrations etc. This function was originally created in the SBML toolbox (Keating, Bornstein et al. 2006), and full documentation for this aspect can be found at http://sbml.org/software/sbmltoolbox/.

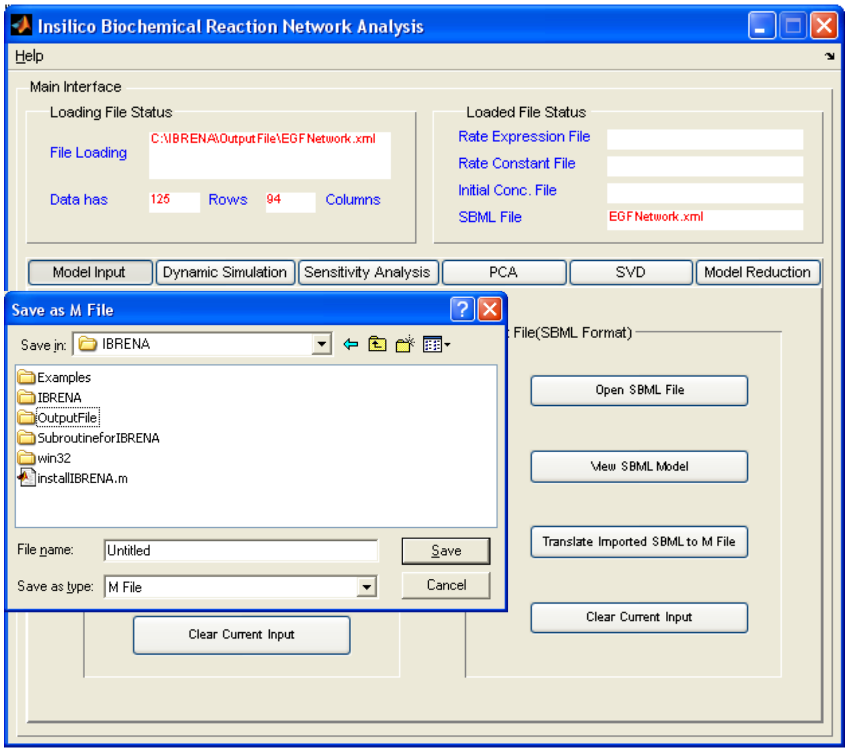


Model:

SBML Level 2 Version 1

Reactions	reaction_1
Species	EGF
UnitDefinitions	NO UNIT DEFINITIO...
Compartments	Compartment 1
Parameters	kf1
Rules	NO RULES
Functions	NO FUNCTION DE...
Events	NO EVENTS

IBRENA allows translation of SBML files to m-files using the button “Translate SBML to M-file”. This function was originally implemented in the SBML toolbox. As explained in the toolbox documentation, such m-files can be directly used with MATLAB ODE solvers.



C:\IBRENA\OutputFile\EGFNetwork.xml

Data has 125 Rows 94 Columns

SBML File: EGFNetwork.xml

Save in: IBRENA

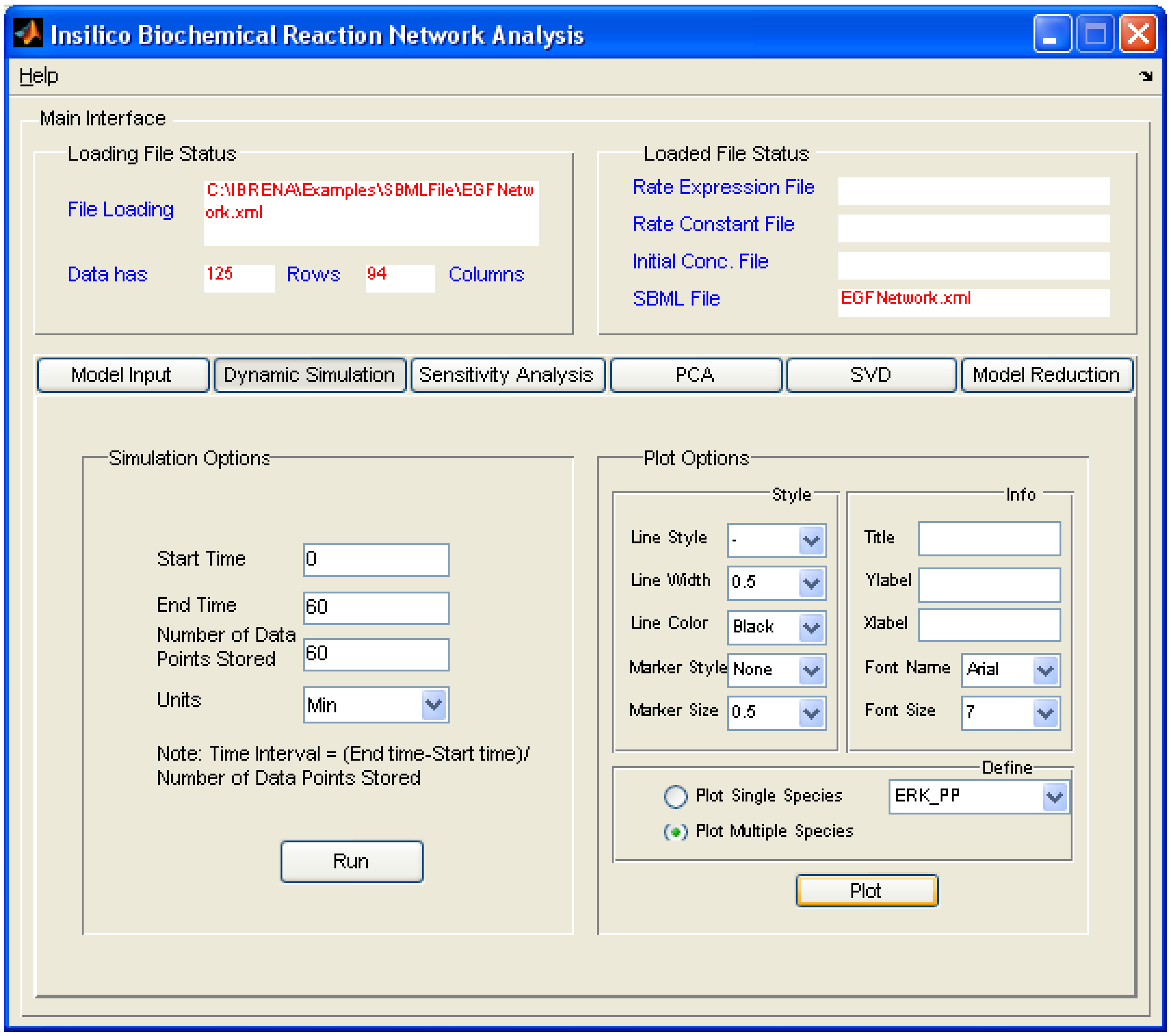
File name: Untitled

Save as type: M File

3. Dynamic Simulation

3.1 Theory

The second tab "Dynamic Simulation" is used to simulate and plot the time-dependent changes in species concentration (see below). This can be run after the three inputs (rate expressions, rate constants and initial conditions) are specified as discussed in Chapter 2 using .csv or SBML format files. The left panel "Simulation Options" in used to specify the parameters for the calculation while the right "Plot Options" panel graphs the results.



File Loading: C:\IBRENA\Examples\SBMLFile\EGFNetwork.xml

Data has 125 Rows 94 Columns

Rate Expression File

Rate Constant File

Initial Conc. File

SBML File: EGFNetwork.xml

Start Time: 0

End Time: 60

Number of Data Points Stored: 60

Units: Min

Note: Time Interval = (End time-Start time)/ Number of Data Points Stored

Line Style: -

Line Width: 0.5

Line Color: Black

Marker Style: None

Marker Size: 0.5

Title:

Ylabel:

Xlabel:

Font Name: Arial

Font Size: 7

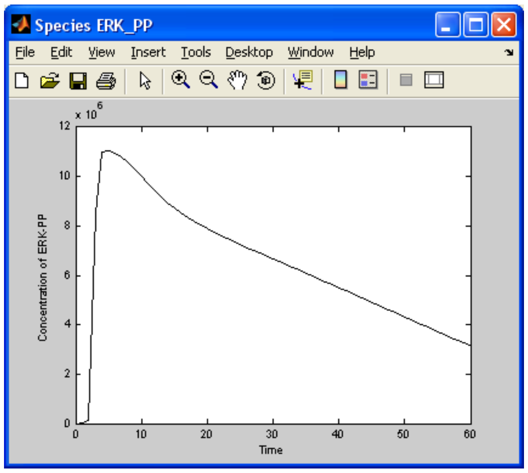
Plot Single Species [✓ Plot Multiple Species]

ERK_PP

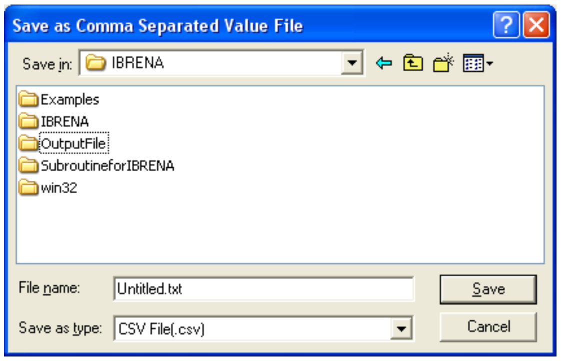
3.2 Implementation

During dynamic simulations, the stand-alone application will run .csv input files only. The full-version will simulate both .csv and SBML inputs. All plot options are implemented identically in both versions of the software.

For both the stand-alone and full versions of the software, under "Simulation Options", the user specifies the "units" of time, the "start time" and "end time" for simulating the reaction network. The "number of data points", which specifies the resolution with which data points are stored for further analysis is also specified. For example, if the number of data points to be collected between 0 and 60 min is 60, then the software will store 60 points at intervals of 1 min. in addition to the starting value at 0 min. Note that the number of data points specified only affects the format of the output data stored and not the computations themselves, since accuracy of the calculations is controlled by the FORTRAN IMSL ODE solver (for .csv input and stand-alone application) or MATLAB ODE solver (for SBML input and full application). The figure below shows a sample simulation for the EGF network between 0 and 60min., with data points being collected at 1 min. intervals.



Upon completion of the above simulation, a window appears (see below) providing the user an option of saving simulation results in a user-defined local file. These data are stored in comma delimited format with the concentration of individual species appearing in columns and the number of rows corresponding to the number of data points specified. This file can be opened using Excel or other suitable software.


Save as Comma Separated Value File

Save in: IBRENA

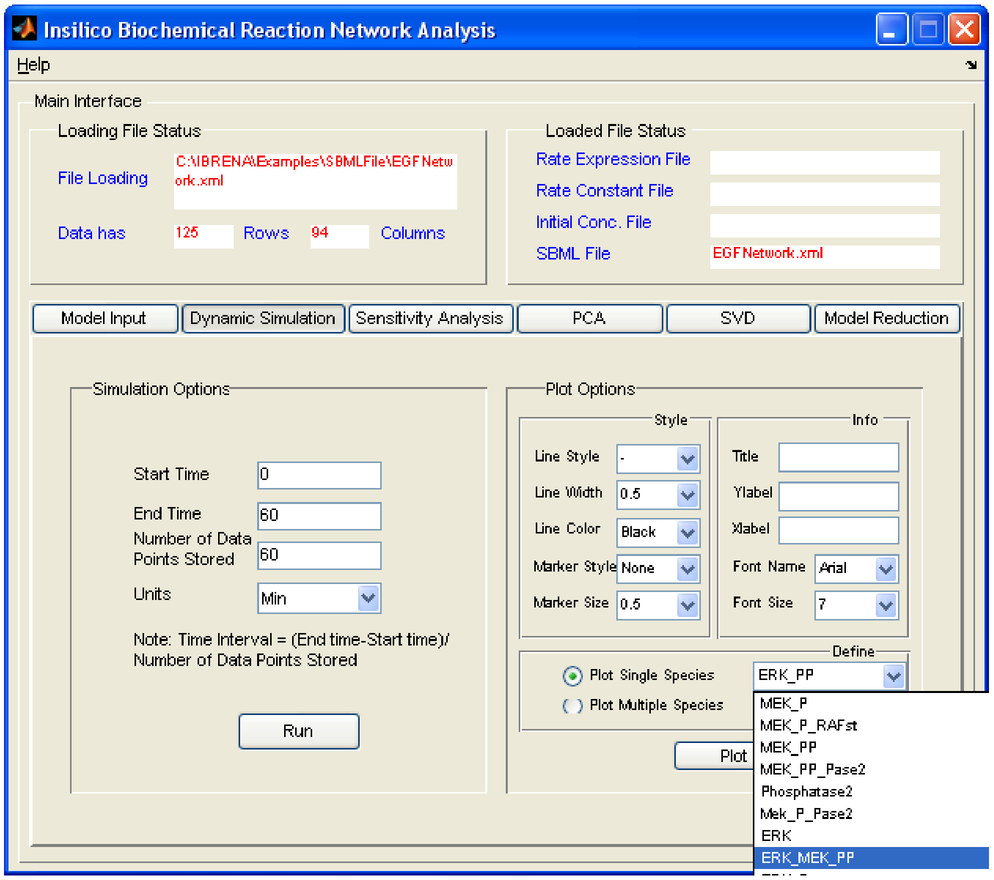
Examples
IBRENA
OutputFile
SubroutineforIBRENA
win32

File name: Untitled.txt

Save as type: CSV File(.csv)

Once the simulation is completed, "Plot Options" in the right panel can be used to visualize time-dependent changes in species concentration. For such work, line style can be specified along with axis and chart labels. Single plots can be made for a given species, for example species ERK-PP in the EGF model above, which corresponds to dual phosphorylated ERK. The species can be chosen from the species list in the model list-box, as shown below.

Once the species name in the Plot Options and "Plot Single Species" are specified, clicking the "Plot" button generates the plot. In the plots generated by the current software, the units of time (X-axis values) always appear in minutes regardless of the time scale of simulation.


Insilico Biochemical Reaction Network Analysis

Help

Main Interface

Loading File Status

File Loading: C:\IBRENA\Examples\SBMLFile\EGFNetwork.xml

Data has 125 Rows 94 Columns

Loaded File Status

Rate Expression File:

Rate Constant File:

Initial Conc. File:

SBML File: EGFNetwork.xml

Simulation Options

Start Time: 0

End Time: 60

Number of Data Points Stored: 60

Units: Min

Note: Time Interval = (End time-Start time)/ Number of Data Points Stored

Plot Options

Style

Line Style: -

Line Width: 0.5

Line Color: Black

Marker Style: None

Marker Size: 0.5

Info

Title:

Ylabel:

Xlabel:

Font Name: Arial

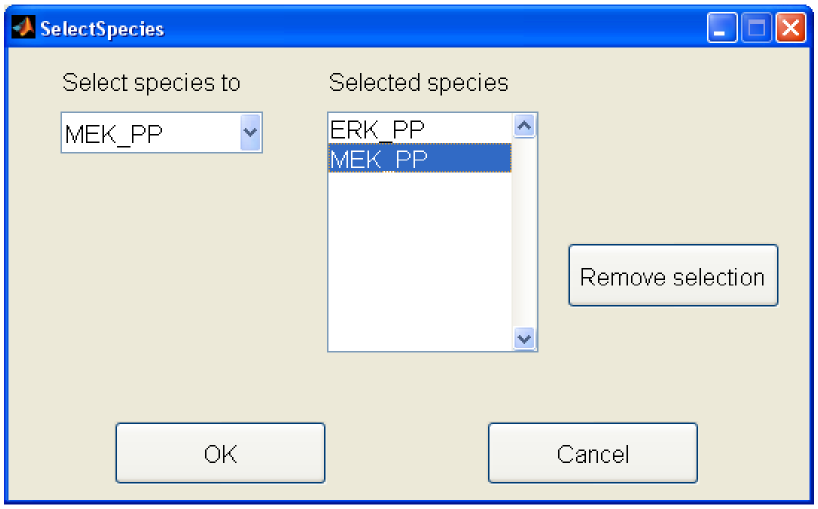
Font Size: 7

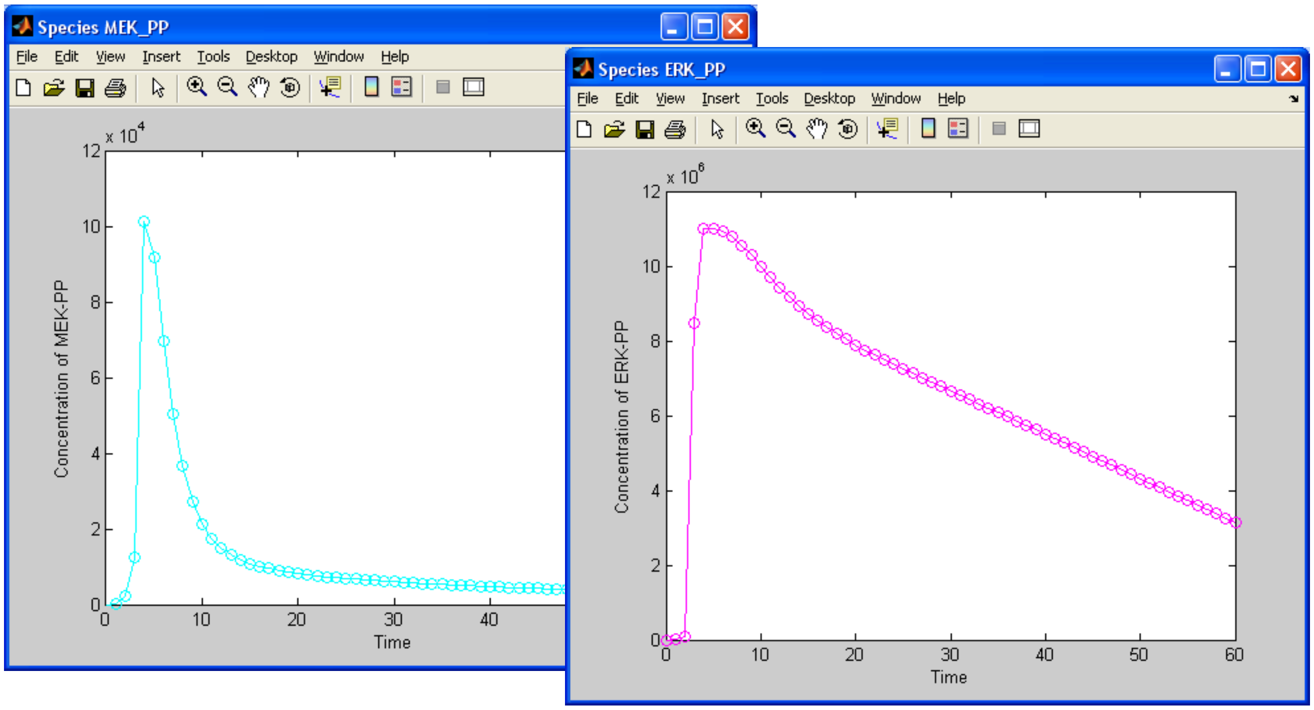
Define

[✓ Plot Single Species] ERK_PP

Plot Multiple Species

Similar to above, multiple plots for different species can also be rapidly generated by choosing the "Plot Multiple Species" option. Upon clicking on this option, the "Select Species" user inteface appears (see below). Here, the user can choose selected species that are of interest one at a time using the "Choose species" list-drop box. These are added to the "selected species" window. Once all desired species are selected, clicking "ok" generates the desired plot time-coursees as shown below.




Species MEK_PP
File Edit View Insert Tools Desktop Window Help
x 10^4
Concentration of MEK-PP
Time
Species ERK_PP
File Edit View Insert Tools Desktop Window Help
x 10^6
Concentration of ERK-PP
Time

All figures can be individually saved in Matlab or other format files.

4 Sensitivity Analysis

4.1 Theory

Sensitivity analysis provides a simple, systematic and powerful methodology for analysis of biological reaction networks, especially non-steady state phenomena. This is a linear perturbation method that is applied to study the effect of infinitesimal changes in system parameters on system variables. Examples of system parameters include the individual reaction rate constants (k_j^f and k_j^r), the pathway structure and the initial conditions of the system. System variables include individual species concentration (C_i) and reaction velocity (v_i).

Two approaches have been developed to perform sensitivity analysis. These are termed forward and adjoint sensitivity analysis. In forward sensitivity analysis, the sensitivities can be described in terms of ordinary differential equations (ODE) that are called sensitivity equations. These equations result from the differentiation of kinetic ODE equations (Eq.2b in Chapter 2) with respect to the system parameters. By thus simultaneously solving both the sensitivity and kinetic ODE equations, sensitivity coefficients can be obtained. In the case of reaction networks with a large number of parameters and variables, the computation of sensitivity coefficients becomes computationally expensive since it requires the calculation of the sensitivities of all variables. Therefore, adjoint method have been developed to reduce computation time. Here, sensitivity coefficients are computed by solving the solution of adjoint sensitivity ODE equations associated with the kinetic equation. The details of each method is described below.

This chapter describes three methods used by IBRENA to evaluate sensitivity coefficients. The input file format and the need for adjoint/forward sensitivity analysis determines the choice of analysis strategies. The three methods include: i) Forward sensitivity analysis of reaction network when input file are in .csv file format; ii) Forward sensitivity analysis of reaction network when input is in SBML format; iii) Adjoint sensitivity analysis of reaction network with SBML input file format. The details follow.

4.1.2 Forward sensitivity analysis of elemental reaction network (.csv input files)

In the elemental reaction network, the system variable considered is C_i and system parameter is either k_j^f or k_j^r. Sensitivity coefficients (Z_{ij}) are derived for both forward and reverse rate constants independently. For forward rates this is:

$$Z_{ij}^f \equiv \partial C_i / \partial k_j^f \qquad (3)$$

These coefficients can be made dimensionless and this yields the scaled sensitivity coefficients, W_{ij}:

$$W_{ij} \equiv \frac{k_j^f}{C_i}\frac{\partial C_i}{\partial k_j^f} \text{ or } \frac{k_j^r}{C_i}\frac{\partial C_i}{\partial k_j^r} \tag{4}$$

When .csv files are used as input, in our computations, the direct differential method is employed for the systematic evaluation of scaled sensitivity coefficients in reaction networks (Liu, Swihart et al. 2005). First, the partial derivative of Eq.3 is taken. Differentiating the above equation with respect to time and substituting $\partial C_i / \partial t = f_i$ in Eq. 2a (chapter 2), we obtain:

$$\frac{dZ_{ij}^f}{dt} = \frac{\partial f_i}{\partial k_j^f} + \sum_{l=1}^{m} \frac{\partial f_i}{\partial C_l}\frac{\partial C_l}{\partial k_j^f} \tag{5}$$

The last partial derivative on the right side is, by definition, Z_{ij}^f. Rewriting Eq.5 in matrix format and substituting $\boldsymbol{f} = \boldsymbol{\alpha}^{\mathrm{T}}\boldsymbol{v}$, we have:

$$\frac{d\,\boldsymbol{Z}^f}{dt} = \boldsymbol{\alpha}^T\left(\frac{\partial \boldsymbol{v}}{\partial \boldsymbol{k}^f} + \frac{\partial \boldsymbol{v}}{\partial \boldsymbol{C}} \cdot \boldsymbol{Z}^f\right) \tag{6}$$

Here, $\partial \boldsymbol{v} / \partial \boldsymbol{k}^f$ and $\partial \boldsymbol{v} / \partial \boldsymbol{C}$ are $n\times n$ and $n\times m$ Jacobian matrices in which the i,j element contains the partial derivative of v_i with respect to k_j^f and C_j, respectively. Upon differentiating the rate expression (Eq.1) with respect to k_j^f and C_j, it follows that:

$$\frac{\partial v_i}{\partial k_j^f} = \begin{cases} \prod_{l=1}^{m} C_l^{\mu_{il}^f} = \frac{r_i^f}{k_j^f} & \text{i = j} \\ 0 & \text{i} \neq \text{j} \end{cases} \tag{7}$$

$$\frac{\partial v_i}{\partial C_j} = \frac{\mu_{ij}^f k_i^f \prod_{l=1}^{n} C_l^{\mu_{il}^f} - \mu_{ij}^r k_i^r \prod_{l=1}^{n} C_l^{\mu_{il}^r}}{C_j} = \frac{\mu_{ij}^f r_i^f - \mu_{ij}^r r_i^r}{C_j} \tag{8}$$

Combining Eq. 6-8 and writing the result in matrix format we get the following equation:

$$\frac{d\mathbf{Z}^{\mathbf{f}}}{dt} = \boldsymbol{\alpha}^{\mathbf{T}} \cdot \left[\mathbf{r}^{\mathbf{f}} \cdot \mathbf{k}^{\mathbf{f}^{-1}} + (\mathbf{r}^{\mathbf{f}} \cdot \boldsymbol{\mu}^{\mathbf{f}} \cdot \mathbf{C}^{-1} - \mathbf{r}^{\mathbf{r}} \cdot \boldsymbol{\mu}^{\mathbf{r}} \cdot \mathbf{C}^{-1}) \cdot \mathbf{Z}^{\mathbf{f}}\right] \tag{9A}$$

Similar to the above, for the reverse rate, we obtain Eq. 9B:

$$\frac{d\mathbf{Z}^{\mathbf{r}}}{dt} = \boldsymbol{\alpha}^{\mathbf{T}} \cdot \left[-\mathbf{r}^{\mathbf{r}} \cdot \mathbf{k}^{\mathbf{r}^{-1}} + (\mathbf{r}^{\mathbf{f}} \cdot \boldsymbol{\mu}^{\mathbf{f}} \cdot \mathbf{C}^{-1} - \mathbf{r}^{\mathbf{r}} \cdot \boldsymbol{\mu}^{\mathbf{r}} \cdot \mathbf{C}^{-1}) \cdot \mathbf{Z}^{\mathbf{r}}\right] \tag{9B}$$

The IMSL ODE solver IVPAG is used to solve Eq. 9 and Eq.2a. The reaction rates, rate constants, reaction orders and initial reactant concentration were read previously from .csv format files, as described in Chapter 2. Once Z_{ij}^{f} and Z_{ij}^{r} were obtained, we normalize the result to obtain the scaled sensitivity coefficients, W_{ij} (Eq. 4).

4.1.2 Forward sensitivity analysis of general reaction network (SBML format input files)

In the general reaction network, the system variable considered is C_i and system parameter is k_j. Sensitivity coefficients (Z_{ij}) are derived as:

$$Z_{ij}^{f} \equiv \partial C_i / \partial k_j \tag{10}$$

These coefficients can be made dimensionless and this yields the scaled sensitivity coefficients, W_{ij} similar to the above section:

$$W_{ij} \equiv \frac{k_j}{C_i} \frac{\partial C_i}{\partial k_j} \tag{11}$$

Given the reaction network described by the general kinetic equation (Eq.2b), the sensitivity coefficients can be described by solving:

$$\frac{d\mathbf{Z}}{dt} = \frac{\partial \mathbf{f}}{\partial \mathbf{C}} \cdot \mathbf{Z} + \frac{\partial \mathbf{f}}{\partial \mathbf{K}} \tag{12}$$

Here f describes the right-hand-side (R.H.S.) of Eq.2b, and it is a function of reactant concentration c and kinetic parameter k. Sensitivity analysis now requires the solution of the forward sensitivity equation (Eq.12) in conjunction with the kinetic equation (Eq.2b in Chapter 2). Two separate approaches are used in our program to perform this computation in the case of “long calculation” (required for the systematic evaluation of all sensitivity coefficients, see section 4.2 below) and “express calculation” (required for the evaluation of selected coefficients only, see section 4.2 below).

For long calculations, we use the "diff" command in the MATLAB symbolic math toolbox to derive analytical expressions for $\frac{\partial \mathbf{f}}{\partial \mathbf{C}}$ and $\frac{\partial \mathbf{f}}{\partial \mathbf{K}}$. Following this, the CVODE command in the Sundials toolbox is used to solve both the sensitivity (Eq. 12) and kinetic ODE equations (Eq. 2b), and to calculate Z_{ij}. These are normalized using the expression in Eq. 11 to obtain the normalized forward sensitivity coefficients, W_{ij}.

During express calculations, finite difference methods are applied for evaluation of the right hand side function in Eq. 12 and thus analytical expressions are not required for this calculation.

4.1.3 Adjoint sensitivity analysis of general reaction network (SBML format input files)

Adjoint sensitivity analysis has advantages over forward sensitivity analysis in some special cases: i) This is a computationally efficient method when large-scale reactions are used since calculation of undesired sensitivity coefficients is not required here; ii) It allows definition of complex system variables (e.g. the sum of all phosphorylated ERK species, $C_{ERK\text{-}P}$ (mono-phosporylated) + $C_{ERK\text{-}PP}$ (dual-phosporylated) in the EGF example), unlike conventional forward sensitivity analysis which is typically limited to system variables that are single species concentration only (e.g. concentration of dual-phosphorylated ERK alone, $C_{ERK\text{-}PP}$). Definition of such complex or arbitrary system variables, thus, may allow biochemical reaction network analysis in a more biological/chemical meaningful manner.

In the general reaction network C_i is considered to be the system variable and k_j is the system parameter. In matrix form, these are represented by ***C*** and ***K.*** Upon defining an arbitrary system variable $g(\mathrm{C},t,K)$, the corresponding integral function $G(\mathrm{C},t_1,K)$ is (see (Cao, Li et al. 2003) and www.llnl.gov/CASC/sundials/documentation/cvs_examples/node4.html for details):

$$G(\mathrm{C},t_1,K) = \int_0^{t1} g\ (C\ ,t\ ,K\ )\,dt \tag{13}$$

The sensitivity coefficient Z_j is then defined as:

$$Z_j = \frac{dG}{dk_j} \tag{14}$$

Normalization of Z_j yields W_j as defined below:

$$W_j \equiv \frac{k_j}{G}\frac{\partial G}{\partial k_j} \tag{15}$$

$\frac{dG}{dK}$ can be written as $\frac{dG}{dK} = \int_0^{t1} (g_K + \lambda^T f_K)\,dt$ (16)

Where *f* is the right hand side function in Eq.2b, f_K is the partial derivative of *f* with respect to *K*, g_K is the partial derivative of g with respect to the *K* and λ is the solution of the adjoint problem ((Cao, Li et al. 2003) see Eq. 17 below):

$$\begin{cases} \dot{\lambda} = -f_C{}^T \lambda - g_C{}^T \\ \lambda(t_1) = 0 \end{cases} \tag{17}$$

In the above expression f_C and g_C is the partial derivatives of *f* and *g* with respect to *C*. *T* is the symbol of transposition. In addition, the quadrature equation (Eq.18) may be used to facilitate calculation of dG/d**K**:

$$\begin{cases} \dot{\xi} = g_K^T + f_K{}^T \lambda \\ \xi(t_1) = 0 \end{cases} \tag{18}$$

Based on the above, the adjoint sensitivity coefficients can be computed using the following equation:

$$\frac{dG}{dK} = -\xi^T(0) \tag{19}$$

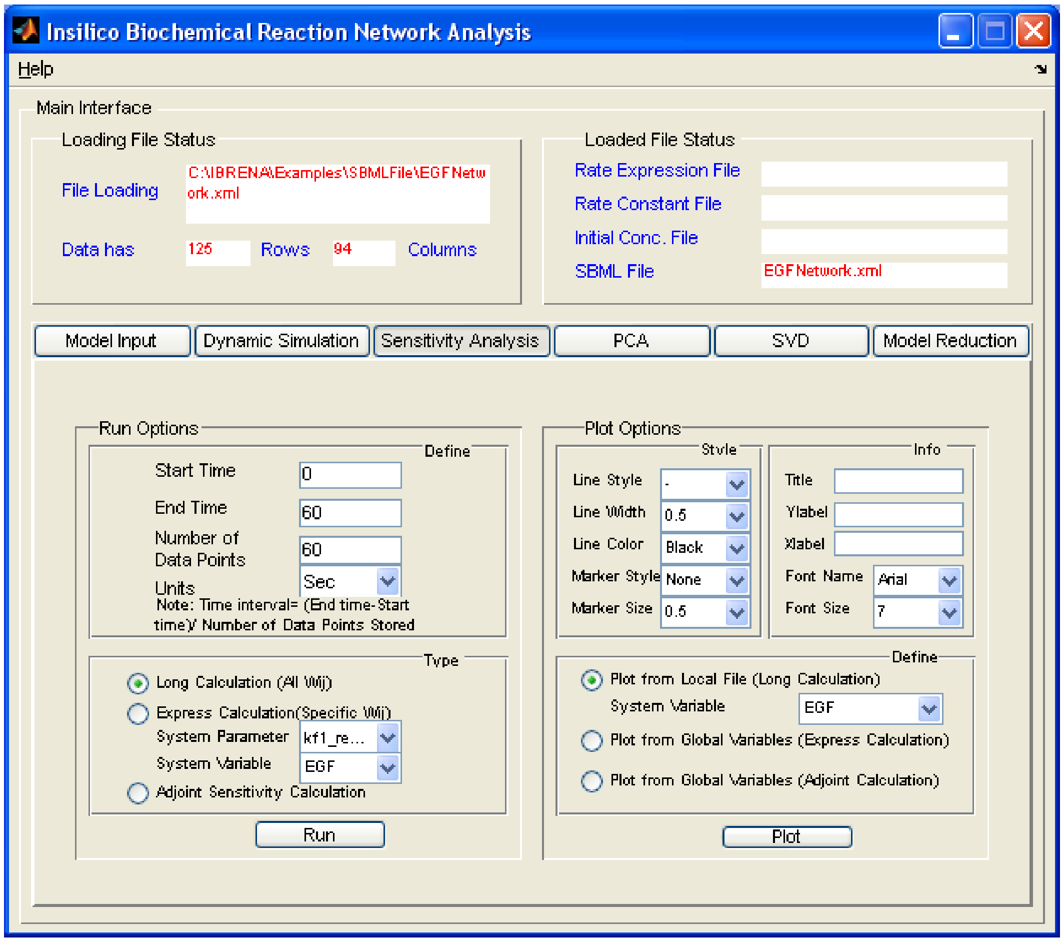
We use the CVODEB command in the Sundials toolbox to perform the adjoint sensitivity calculations. To run the CVODEB command, four functions are required. These include: I) the kinetic ODE equations (R.H.S. of Eq 2b) to define *f*; II) the function *g*. III) the R.H.S. of the adjoint sensitivity equations (Eq.17); and IV) the R.H.S. of the quadrature equations (Eq.18); In these calculations I) and II) are applied in III) to solve the adjoint problem and to evaluate λ. This is then used in Eq. 18 and 19 to evaluate the sensitivity coefficients. We then evaluate the scaled/adjoint sensitivity coefficient (W_{ij}) based on Eq. 15.

4.2 Implementation

The section below presents details used for evaluation of the scaled sensitivity coefficient, W_{ij}. As discussed this parameter varies with time in a manner that is dependent on the reaction network being considered. While sensitivity coefficient remains zero for some reactions, it may decrease or increase in other cases. Negative coefficient implies that the signal output decreases with increasing rate constant and vice versa. The absolute value of sensitivity coefficient greater than one suggests that changes in reaction rate and corresponding reactant concentrations in one region may have an amplified effect on the measured system variable.

IBRENA can be used for three types of W_{ij} computations. In the first option, all W_{ij} values are computed in a given time period, i.e. all possible combinations of C_i and k_j are considered. This sort of calculation takes considerable computation time[1], and thus these are called “Long Calculation”. The second option is to compute one specific $W_{ij,}$ i..e we only consider the effect of one specific parameter (system parameter) on one specific reactant/product (system variable). Due to the reduced time for this calculation, this is termed “Express Calculation”. The third method employs adjoint sensitivity analysis as discussed above.

The “Sensitivity Analysis” interface is shown below. The left panel contains the “Run Options”. Here, the user specifies the “start time”, “end time”, “number of data points” and time “units” in a manner identical to that in Chapter 3. These must be specified for both Long and Express Calculations.



Loading File Status

File Loading: C:\IBRENA\Examples\SBMLFile\EGFNetwork.xml

Data has 125 Rows 94 Columns

Loaded File Status

Rate Expression File:

Rate Constant File:

Initial Conc. File:

SBML File: EGFNetwork.xml

Run Options

Define

Start Time: 0

End Time: 60

Number of Data Points: 60

Units: Sec

Note: Time interval= (End time-Start time)/ Number of Data Points Stored

Type

[✓ Long Calculation (All Wij)]

Express Calculation(Specific Wij)

System Parameter: kf1_re...

System Variable: EGF

Adjoint Sensitivity Calculation

Plot Options

Style

Line Style: -

Line Width: 0.5

Line Color: Black

Marker Style: None

Marker Size: 0.5

Info

Title:

Ylabel:

Xlabel:

Font Name: Arial

Font Size: 7

Define

[✓ Plot from Local File (Long Calculation)]

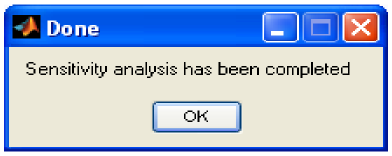
System Variable: EGF

Plot from Global Variables (Express Calculation)

Plot from Global Variables (Adjoint Calculation)

[1] The EGF network example “Long Calculation” takes around 40min. on a 2.66GHz Pentium4 PC with 512MB RAM when the relative error tolerance for ODE solver is set to 10^{-4} (default setting), while “Express Calculations” take 11.8s on the same machine. Usage of FORTRAN to solve ODEs reduced the time for these computations by ~5-fold compared to implementation of these same calculations using MATLAB ODE solver ODE15s. Long calculations using the full version of the software runs ~20% faster than the stand alone application, as benchmarked using the EGF network.

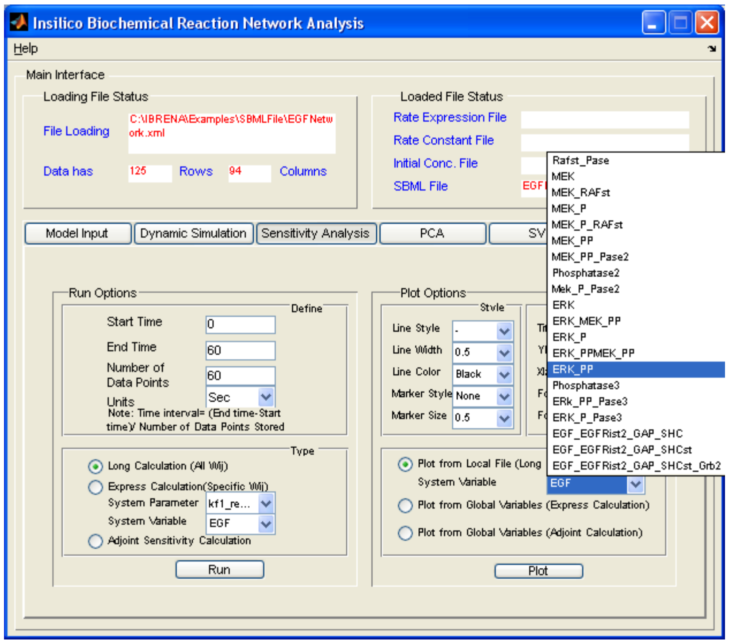
<u>Long Calculations</u> are initiated by choosing the button next to the option and by pressing "Run". The user is then prompted to save results in a comma delimited text file as discussed in Chapter 3 and following this, the "Please wait…" window appears. Once the calculations are completed, a prompt window appears notifying the user that "Sensitivity analysis has been Completed" (see below). The computational time for Long Calculations varies with network size, and it also depends on whether the stand alone or full versions of the software are used (see footnote 1). The data files created from such analysis can be large. The number of columns in the output file equals the "Number of data points stored" plus one, while the number of rows equals the product of reaction rates and species number. For the EGF network example, this file has 61 columns (assuming 60 data points are stored) and 23501 row (=250*94+1, with the first line storing the time points). More details on the format of the output file are provided in a comment statement that appears at the start of this file.


Done

Sensitivity analysis has been completed

Once the Long Calculations are completed, results are plotted in terms of the dependence of scaled sensitivity coefficient on time. The top portion of "Plot Options" is identical to that in Chapter 3 and it is used to format the graphs. Following Long Calculations, the user must re-read the data from the local file created earlier since the memory requirements required to store the entire W_{ij} matrix is large. For this, the user specifies which system parameter and variable are to be plotted. Multiple plots can be generated by specifying the appropriate system parameters and variables in the "Define" window in a comma delimited fashion. The following example illustrates the three steps required to plot data from a local file:

(I) *The program requests that the user specify the system parameter listed in the popup menu defined in sensitivity coefficients. In our example, we choose "ERK-PP"*


Insilico Biochemical Reaction Network Analysis

Main Interface

Loading File Status

File Loading: C:\IBRENA\Examples\SBMLFile\EGFNetwork.xml

Data has 125 Rows 94 Columns

Loaded File Status

Rate Expression File

Rate Constant File

Initial Conc. File

SBML File: EGF

Run Options

Define

Start Time: 0

End Time: 60

Number of Data Points: 60

Units: Sec

Note: Time interval= (End time-Start time)/ Number of Data Points Stored

Type

[✓ Long Calculation (All Wij)]

Express Calculation(Specific Wij)

System Parameter: kf1_re...

System Variable: EGF

Adjoint Sensitivity Calculation

Plot Options

Style

Line Style: -

Line Width: 0.5

Line Color: Black

Marker Style: None

Marker Size: 0.5

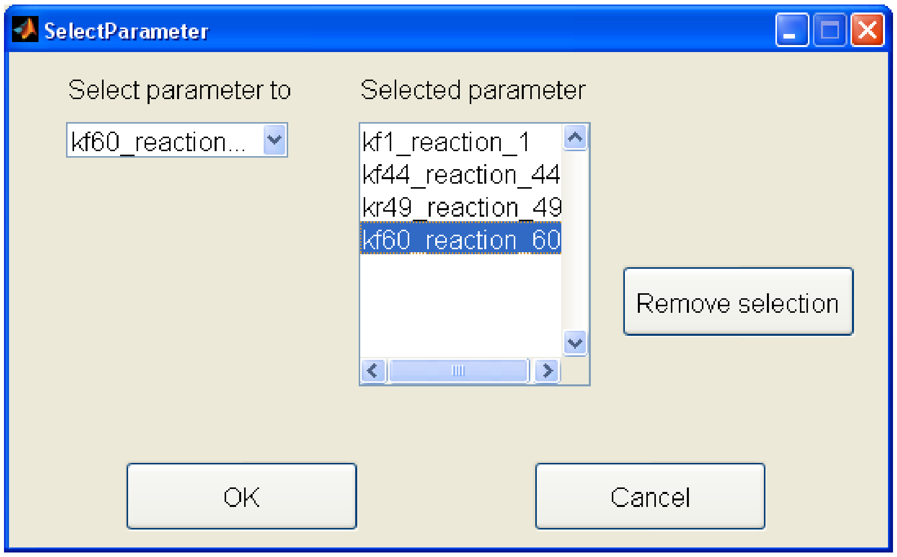
[✓ Plot from Local File (Long]

System Variable: EGF

Plot from Global Variables (Express Calculation)

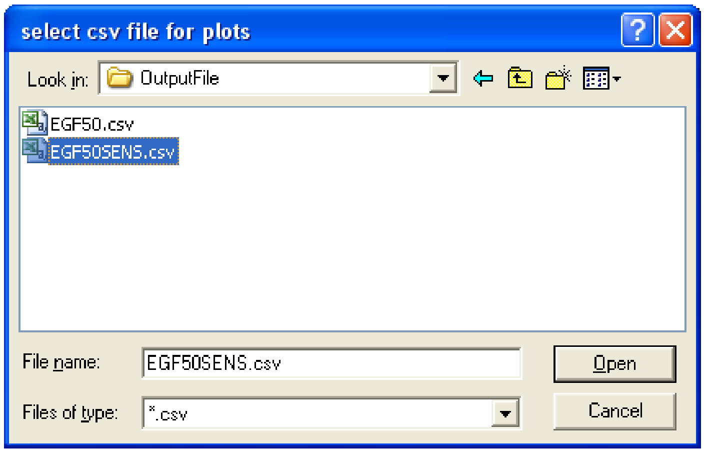
Plot from Global Variables (Adjoint Calculation)

(II) *Upon pressing on the "plot", the "Select Parameter" interface will appear for the user to select single or multiple user-desired parameters. In our example, we choose 4 parameters as shown below. These parameter are kf1,kf44,kr49,kf60.*



kf1_reaction_1
kf44_reaction_44
kr49_reaction_49
kf60_reaction_60

(III) Once the user press on "OK" button in the interface above, IBRENA will request the user for the data file that contains *the output from sensitivity analysis.*

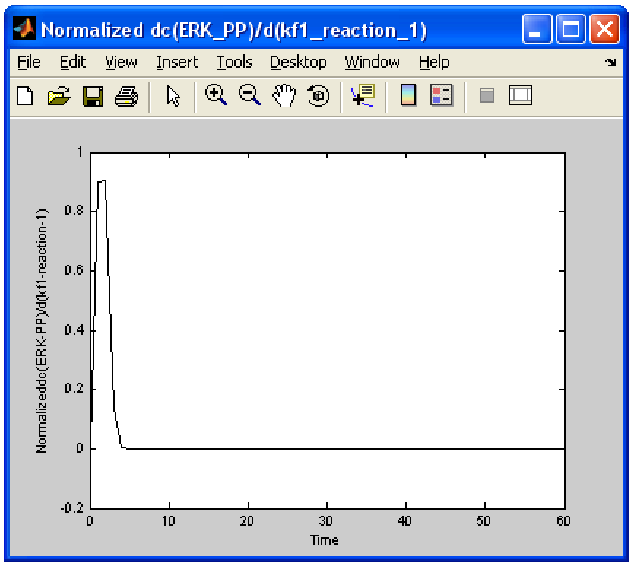


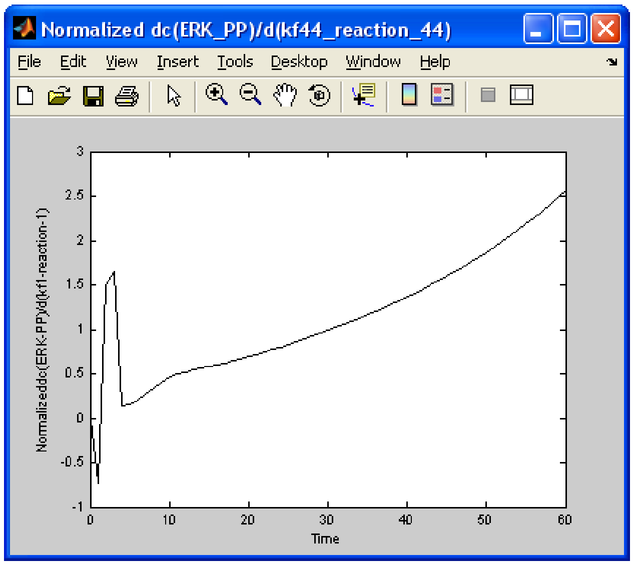
EGF50.csv
EGF50SENS.csv

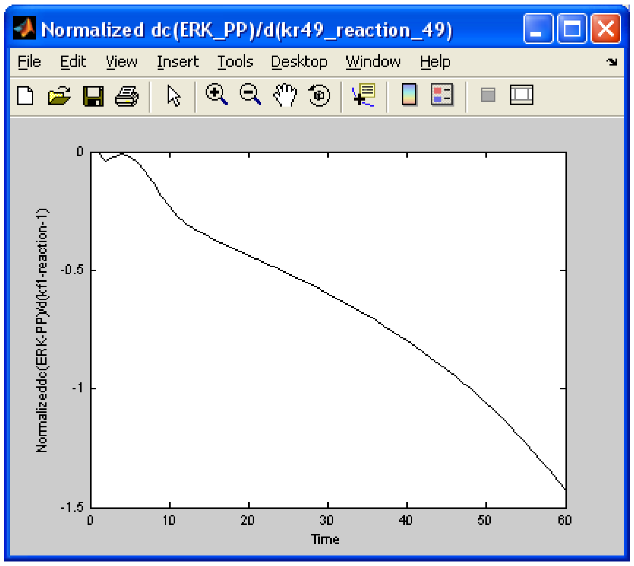
File name: EGF50SENS.csv

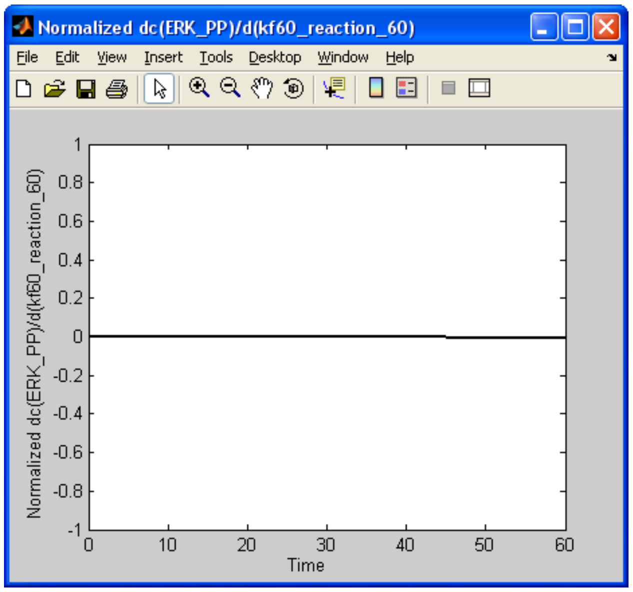
Files of type: *.csv

(IV) After the sensitivity data file is read by the program, time course of user-specified scaled sensitivity coefficients will be plotted. In our example, 4 plots of scaled sensitivity coefficients including dc(ERK-PP)/dkf1, dc(ERK-PP)/dkf44, dc(ERK-PP)/dkr49 and dc(ERK-PP)/dkf60 are shown as below, The definition of scaled sensitivity is shown in the left upper corner of the figure.

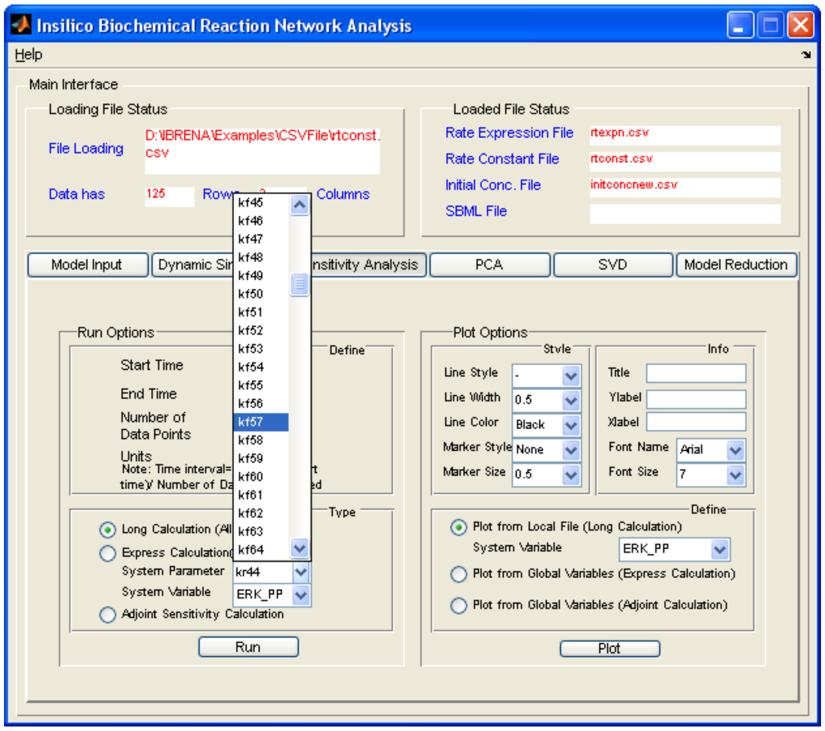








<u>Express calculations</u> compute specific scaled sensitivity coefficients only. In this case, the user specifies the system parameter and variable of interest from the popup menu. Upon pressing the "Run" button calculations are initiated. Similar to above, the user is notified by a pop-up window when sensitivity analysis is completed. Data from express calculations are stored in the computer memory. These can be displayed by clicking the Express Plot option and then the "Plot" button. An example is illustrated below, where the response of the ERK-PP to perturbations of the 1st forward reaction is displayed


Insilico Biochemical Reaction Network Analysis

Help

Main Interface

Loading File Status

File Loading: D:\IBRENA\Examples\CSVFile\rtconst.csv

Data has 125 Rows 3 Columns

Loaded File Status

Rate Expression File: rtexpn.csv

Rate Constant File: rtconst.csv

Initial Conc. File: initconcnew.csv

SBML File:

Model Input | Dynamic Simulation | Sensitivity Analysis | PCA | SVD | Model Reduction

Run Options

Define: Start Time; End Time; Number of Data Points; Units. Note: Time interval=(End time-Start time)/ Number of Data Points

Type

[✓ Long Calculation (All)]

Express Calculation

System Parameter: kr44 (list: kf45, kf46, kf47, kf48, kf49, kf50, kf51, kf52, kf53, kf54, kf55, kf56, kf57, kf58, kf59, kf60, kf61, kf62, kf63, kf64)

System Variable: ERK_PP

Adjoint Sensitivity Calculation

Run

Plot Options

Style: Line Style -; Line Width 0.5; Line Color Black; Marker Style None; Marker Size 0.5

Info: Title; Ylabel; Xlabel; Font Name Arial; Font Size 7

Define

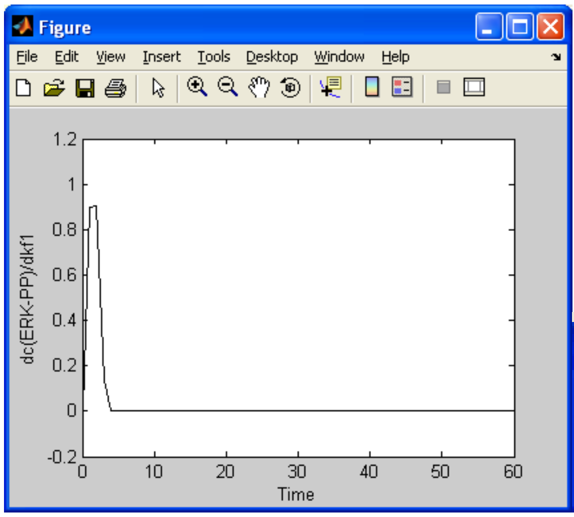
[✓ Plot from Local File (Long Calculation)]

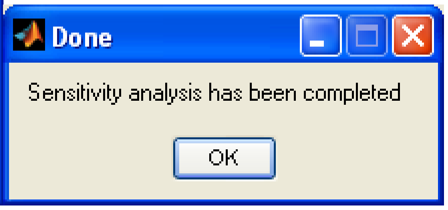
System Variable: ERK_PP

Plot from Global Variables (Express Calculation)

Plot from Global Variables (Adjoint Calculation)

Plot

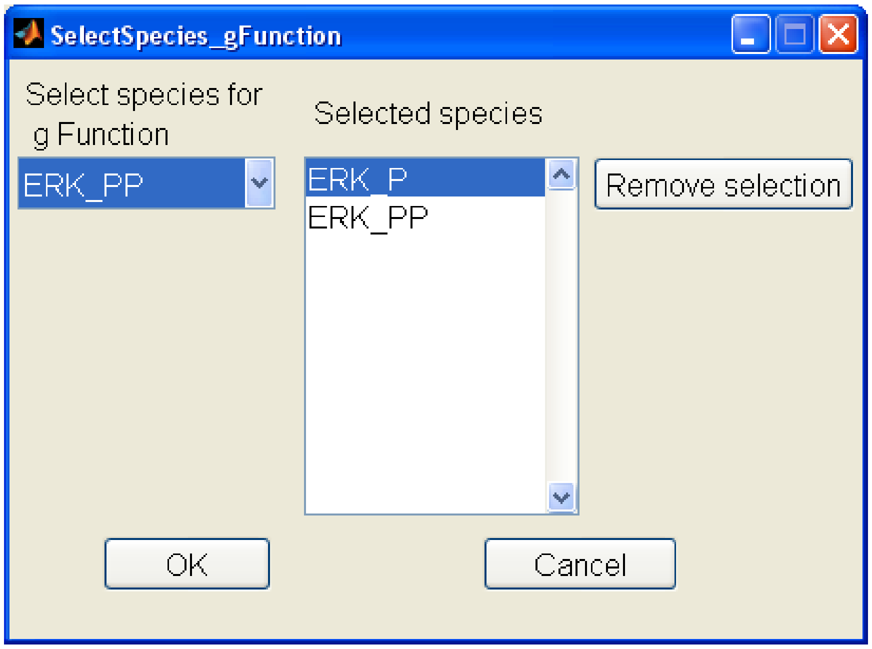



Done

Sensitivity analysis has been completed

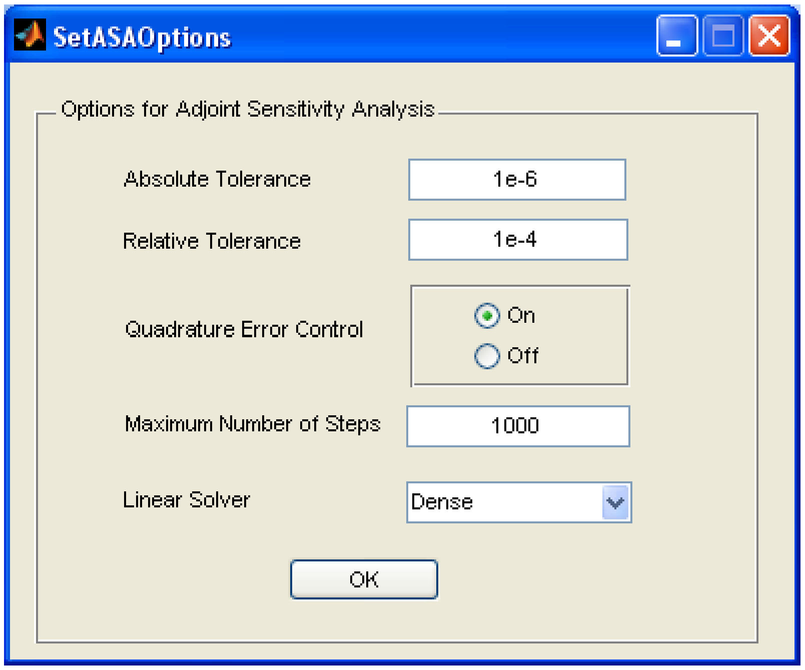
OK

__Adjoint Sensitivity Calculations__ computes scaled adjoint sensitivity coefficients (*Wj*). When the "Run" button is clicked, the user is prompted to define the *g* function in the interface below.



In this case, *g* function can be defined as the concentration of single species, or the concentration of multiple species. For example, as shown in the figure above, user can define *g* function as the sum of the concentrations of "ERK-P" and "ERK-PP".

Once the *g* function is specified by the user and "OK" button is pressed, "SetASAOptions" (<u>SetUp</u> <u>A</u>djoint <u>S</u>ensitivity <u>A</u>nalysis <u>Options</u>) interface appears as below.

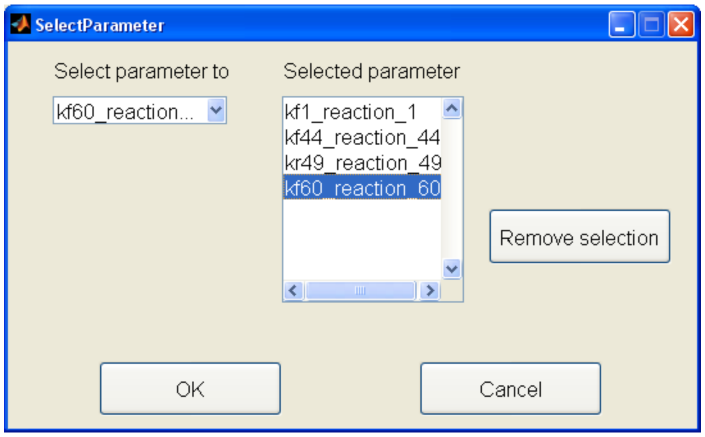


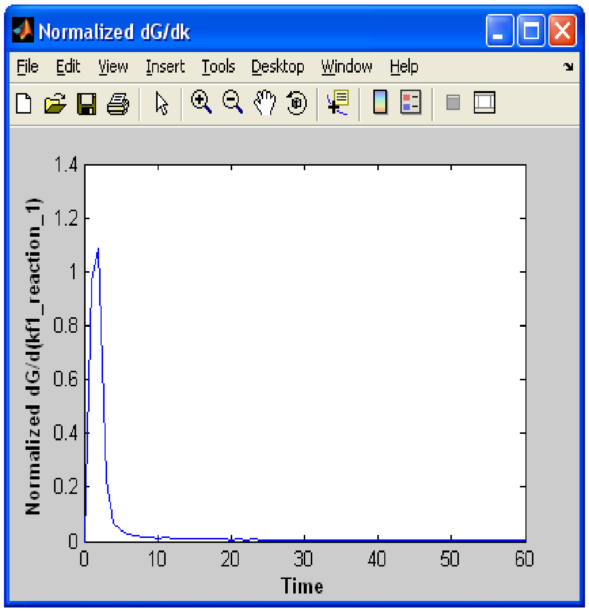
Absolute Tolerance 1e-6
Relative Tolerance 1e-4
Quadrature Error Control [✓ On] Off
Maximum Number of Steps 1000
Linear Solver Dense

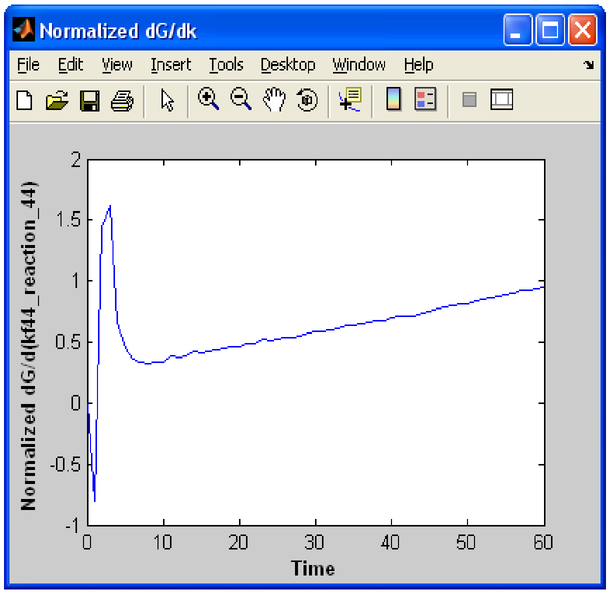
At current stage, five options are provided in IBRENA to perform adjoint sensitivity computation. These options are “Absolute Tolerance”, “Relative Tolerance”, “Quadrature Error Control”, “Maximum Number of Steps” and “Linear Solver”. The common inputs for these options are “1e-6”, “1e-3”, “On”, “1000” and “Dense”. The appropriate setups for these options are problem-dependent. Users can vary these inputs according to the integration results. Please see the SUNDIALS manual (http://www.llnl.gov/CASC/sundials/download/download.html) for details on these option settings. Other option settings are also available in SUNDIALS and these are not currently implemented in IBRENA for the sake of simplicity. These options can however be varied by experienced users by using script M-files. An example file for the EGF network (TestEGFAdjointSA.m) is provided with this software and it is stored in the folder “Example/TestAdjointSA”.

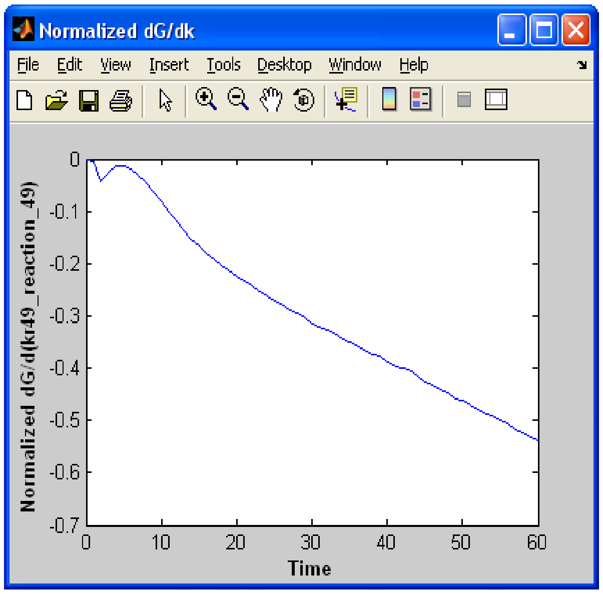
After the user sets up the options and clicks “OK”, adjoint sensitivity calculation starts. Similar to before, the user will be prompted to input the file name and store the data once the computation is finished. In addition, four m files which be generated and stored in the MATLAB “current directory”. These contain the: (i) g function, (ii-iv) right hand side (R.H.S) equation of kinetic ODE equations (Eq.2b), adjoint sensitivity equation (Eq.16), and quadrature equations (Eq.18). The user can use these function m-files to calculate adjoint sensitivity coefficients in the MATLAB environment independent of our interface. As mentioned above, an example files for adjoint sensitivity analysis of the EGF network is provided with this software.

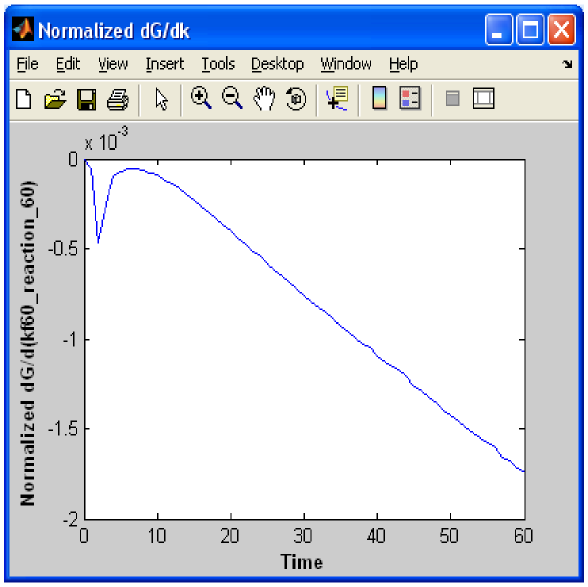
Plotting of adjoint sensitivity coefficients can be achieved in IBRENA by choosing the option of “Plot from Global Variables (Adjoint Calculations)” followed by clicking of “Plot”. Similar to before, the user can select the parameter k_j, or multiple parameters as shown below. Time courses of normalized dG/dk appear upon pressing “OK”. Samples plots are shown below and these are similar to the results of long calculations of the EGF network discussed above.




Normalized dG/dk
File Edit View Insert Tools Desktop Window Help
Normalized dG/d(kf1_reaction_1)
0
0.2
0.4
0.6
0.8
1
1.2
1.4
0
10
20
30
40
50
60
Time


Normalized dG/dk
File Edit View Insert Tools Desktop Window Help
Normalized dG/d(kf44_reaction_44)
-1
-0.5
0
0.5
1
1.5
2
0
10
20
30
40
50
60
Time


Normalized dG/dk
File Edit View Insert Tools Desktop Window Help
Normalized dG/d(kr49_reaction_49)
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
0
10
20
30
40
50
60
Time


Normalized dG/dk
File Edit View Insert Tools Desktop Window Help
x 10^-3
Normalized dG/d(kf60_reaction_60)
0
-0.5
-1
-1.5
-2
0
10
20
30
40
50
60
Time

5 Principal Component Analysis (PCA)

Both dynamic simulation (Chapter 3) and sensitivity analysis (Chapter 4) typically result in large output matrices. For example, the number of sensitivity coefficients measured depends on the size of the reaction network, and choice of system variables and system parameters. The sensitivity coefficients and results from dynamic simulation also vary with both time and stimulus dosage. Manual analysis of W_{ij} and C_i data can thus be complex and tedious. Reduction in the size of such output is desirable particularly if such analysis provides new mechanistic insight. With this goal two analysis strategies are described in Chapter 5 (PCA) and Chapter 6 (SVD). While either analysis strategy can be applied to dynamic simulation and sensitivity analysis data, as explained below, PCA allows the results of sensitivity analysis to be directly related to the response of a biochemical reaction network to perturbation. SVD analysis, on the other hand, is better suited for analysis of dynamic simulation results and array-based experimental data. Results from SVD analysis are same as PCA when the covariance matrix is analyzed.

5.1 Theory

5.1.1 PCA of sensitivity coefficient data matrix

As discussed above, PCA can be applied to further analyze sensitivity coefficient data. Such analysis allows us to study the effect of variation in reaction rate constants on biological reaction network response. This is done by quantifying the response function of the system, $Q(\mathbf{k})$ (Eq. 20):

$$Q(\mathbf{k}) = \sum_{h=1}^{q}\sum_{g=1}^{p}\left[\frac{C_{out}\left(t_h, Stim\_Dos_g, \mathbf{k}\right) - C_{out}\left(t_h, Stim\_Dos_g, \mathbf{k^0}\right)}{C_{out}\left(t_h, Stim\_Dos_g, \mathbf{k^0}\right)}\right]^2 \qquad (20)$$

In this expression, the effect of changing the reaction rate constants from $\boldsymbol{k^0}$ to $\boldsymbol{k}$ on the concentration of the species that represents system output (C_{out}) is measured. t_h is the hth time point, q is the total number of time points, and $Stim\_Dos_g$ denotes one of the p stimulus dosage considered. $\boldsymbol{k^0}$ and $\boldsymbol{k}$ are $2n$-dimensional vectors containing both the n forward and n reverse rate constants (n is the total number of reversible reactions), prior to and following perturbation respectively. This response function thus incorporates information on changes in species concentrations over a range of times and stimulus dose. If $\boldsymbol{\Delta\kappa}$ is a $2n$-dimensional vector whose elements are the logarithmic perturbation in the rate constants, either $\ln(k^f_j/k^{f\,0}_j)$ or $\ln(k^r_j/k^{r\,0}_j)$ (j=1..n), then the above response function can be related to the sensitivity coefficients using (Vajda, Valvko et al. 1985):

$$Q(\boldsymbol{k}) \approx Q(\boldsymbol{\Delta\kappa}) = (\boldsymbol{\Delta\kappa})^{\boldsymbol{T}}\boldsymbol{S^T S}(\boldsymbol{\Delta\kappa}) \qquad (21)$$

Here, the elements of the scaled sensitivity coefficient matrix $\boldsymbol{S}$ (and its transpose $\boldsymbol{S^T}$) are W_{ij}

derived from the sensitivity analysis in Chapter 4. The exact form of the matrix depends on the definition of the response function. For example, in the EGF-induced signaling reaction network (Liu, Swihart et al. 2005), if we consider the response function at q time points, three stimulus EGF dosages (p=3) and with respect to the system output, $C_{ERK\text{-}PP}$ ($C_{out} = C_{59}$), then $\boldsymbol{S}$ has dimensions of $(3q)\times(2n)$. The first q rows of this matrix have the elements of $W_{59,j}$ with respect to both the forward and reverse rate constants, at various times and at a fixed EGF dosage (say the lowest dose). The next q rows have the same data at the intermediate EGF concentration and the final rows have this information at the highest EGF dosage. Regardless of the dimensions of $\boldsymbol{S}$, the product of $\boldsymbol{S}^T\boldsymbol{S}$ always has dimensions of $(2n)\times(2n)$. Overall, by appropriate design of $\boldsymbol{S}$ we can quantitatively weight the effects time, stimulus dosage and choice of system variable on the reaction network response.

The matrix $\boldsymbol{S}^T\boldsymbol{S}$ can be diagonalized using its eigenvalues and eigenvectors as shown in Eq.22, where the individual eigenvalues (λ_l) form the diagonal elements of the diagonal matrix $\boldsymbol{\Lambda}$, and $\boldsymbol{U}$ denotes the matrix whose columns are the normalized eigenvectors. u_{jl} represents an element of $\boldsymbol{U}$.

$$\boldsymbol{S}^T\boldsymbol{S}=\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \tag{22}$$

The eigenvectors represent the principle axes of the response function, in terms of the list of reactions. $\boldsymbol{\Delta\Psi} = \boldsymbol{U}^T\cdot\boldsymbol{\Delta\kappa}$ is the linear transformation of the perturbation vector, $\boldsymbol{\Delta\kappa}$, to the principle axes. With this definition, the response function is given by $Q(\boldsymbol{\Delta\kappa}) = \boldsymbol{\Delta\Psi}^T\boldsymbol{\Lambda}\boldsymbol{\Delta\Psi} = \sum_{i=1}^{2n}\lambda_i(\Delta\Psi_i)^2$. The response function is thus most sensitive to changes in rate constants along the principal axis corresponding to the largest eigenvalue and is least sensitive to changes along the principal axis corresponding to the smallest eigenvalue. In such analysis, the eigenvalue provides an absolute measure of the significance of some part of the biological system that is composed of strongly coupled reactions. The relative importance of eigenvectors can be measured based on the corresponding eigenvalues according to:

$$\text{Percent of eigenvalue} = 100 \times \lambda_i / \sum_{i=1}^{2n} \lambda_i \tag{23}$$

Each eigenvector is a linear combination of reactions, and the relative magnitude of each element in eigenvector column measures the relative importance of each reaction for the corresponding eigenvalue. As part of the current modeling effort, we use the PCA parameter e_j (Eq. 24) as a measure of the importance of the jth reaction:

$$\text{PCA parameter, } e_j=\sum_{l=1}^{2n}\lambda_l u_{jl} / \sum_{l=1}^{2n}\lambda_l \tag{24}$$

Taken together, eigenvalues and eigenvectors of $\boldsymbol{S^T S}$ evaluated using PCA provide a measure of the significance of individual reactions.

5.1.2 PCA analysis of genomic/proteomic data matrix

cDNA- or protein- arrays allow simultaneous measurement of the expression of large numbers (from tens to thousands) of mRNA or protein in multiple samples. This can generate large genomic and proteomic data sets that can vary as a function of treatment conditions and time. For the analysis of such multi-dimensional data, with the goal of capturing the essence of the experiments, PCA has been applied to reduce the dimensionality of the data sets to a small number of uncorrelated variables (called Principle Components or PCs).

In a sample study, using PCA to study the TNF-induced signaling protein data matrix (29 measurement × 26 samples), Janes *et al.* found that the first two principle components capture most of the information of the signaling network. While the 1st PC (Principal Component or eigen vector) captured the early, pro-survival signals, the 2nd PC accounted for the late, pro-death signals (Janes, Kelly et al. 2004). In such analysis the following equation which is analogous to Eq.22 is used:

$$\boldsymbol{XX^T} = \boldsymbol{U\Lambda U^T} \qquad (25)$$

Here, $\boldsymbol{X}$ ($m \times p$) matrix stores protein concentration data, with the p columns of $\boldsymbol{X}$ corresponding to observations at specific time points and the m rows containing species concentration. Data preprocessing steps may be performed prior to PCA based on the rationale outlined in the next section. Following this, eigenvector-eigenvalue analysis is performed according to Eq. 25, and this yields the m×m $\boldsymbol{U}$ matrix with orthonormal eigenvectors ($\boldsymbol{UU^T} = \boldsymbol{I}$, identity matrix). Each eigenvector contain the coefficients (also called "loading") for each original element so that the corresponding principal component can be expressed as the linear combination of original elements. $\boldsymbol{\Lambda}$ (m×m) is the diagonal matrix with rank-ordered eigenvalues located along the main diagonal from highest to lowest. Each eigenvalue captures a fraction of the variance of the biochemical network and thus highlights the relative importance of the corresponding PC. Thus, the first PC explains the largest percentage of the total variance of the system; the second explains the second largest percentage, and so on. In the case of signal transduction, the variance of each variable may refer to the degree of the fluctuation of the signaling element after stimulation of the biochemical system. The more the variance for the signaling element, the larger the response it intrigues. Therefore, the principal components with large variance have greater effect on the signal. Such information, which is quantified by the parameter "Percent of eigenvalue" (Eq. 23), can be used to infer important components of a signal network, their relation to other components in the system and such analysis can also aid model reduction. In addition, the PCA parameter can be calculated using Eq. 24 in order to quantify the relative importance of particular protein/gene species.

Similar to the above discussion for genomic/proteomic experiment analysis, PCA analysis can also be performed on dynamic simulation results with the goal of quantifying "percent of eigenvalue" and "PCA parameter".

5.1.3 Preprocessing step

Prior to PCA (and SVD), several data transformation steps including mean centering and normalization (scaling) may be performed on the raw data matrix (Alter, Brown et al. 2000; Holter, Mitra et al. 2000; Janes, Kelly et al. 2004; Pittelkow and Wilson 2005). Mean-centering involves the subtraction of column/row mean from each element in the column/row according to Eq.26A, 26B. Normalization similarly involves the division step shown in Eq.26C, 26D. In these equations, x_{ij} refers to a matrix element in i^{th} row and j^{th} column. m is the number of rows and n refers to column number.

Row centering: $$x_{ij} = x_{ij} - \frac{1}{n}\sum_{j=1}^{n} x_{ij} \tag{26A}$$

Column centering: $$x_{ij} = x_{ij} - \frac{1}{m}\sum_{i=1}^{m} x_{ij} \tag{26B}$$

Row normalization: $$x_{ij} = x_{ij} / \sqrt{\sum_{j=1}^{n} {x_{ij}}^2} \tag{26C}$$

Column normalization: $$x_{ij} = x_{ij} / \sqrt{\sum_{i=1}^{m} {x_{ij}}^2} \tag{26D}$$

The specific data preprocessing step performed by the user depends on the structure of the matrix, physical meaning of the source data being analyzed, and the questions being addressed. If the data are protein concentration time course data, mean-centering is required prior to PCA in order to obtain the covariance matrix $\mathbf{XX}^{\mathrm{T}}$. If the data set is represented by the reaction sensitivity coefficient matrix (***S***), data preprocessing is not necessary since eigenvector analysis of the original ***S*** can display the response function of the biological system upon perturbation. Normalization steps are required if each row/column has different measurement units or if the user does not want to take the magnitude of individual measurements into consideration during analysis, i.e., if the relative change of the system properties has more valuable information than the absolute value itself. Also, normalization can correct system biases when source data are experimental data sets.

Based on the above discussion, our recommended rules for data preprocessing steps are as follows: i) for *insilico* protein temporal concentration data use row mean-centering and row normalization steps; ii) for *insilico* protein sensitivity coefficient data preprocessing steps are not required; iii) for experimental protein concentration data with same measurement unit and in the same magnitude, use row centering only; iv) for experimental protein concentration data with different measurement units, use row centering along with normalization; v) for cDNA microarray data, use both row and column centering along with normalization as illustrated by others (Holter, Mitra et al. 2000; Pittelkow and Wilson 2005).

5.2 Implementation

5.2.1 PCA analysis of scaled sensitivity coefficient data matrix

We illustrate the application of PCA to analyze sensitivity analysis results using the EGF-induced signal transduction network example discussed earlier (Liu, Swihart et al. 2005). For this case, the sensitivity coefficient matrix ***S*** was constructed by performing simulations at 3 different EGF doses prior to PCA analysis. The following steps were followed:

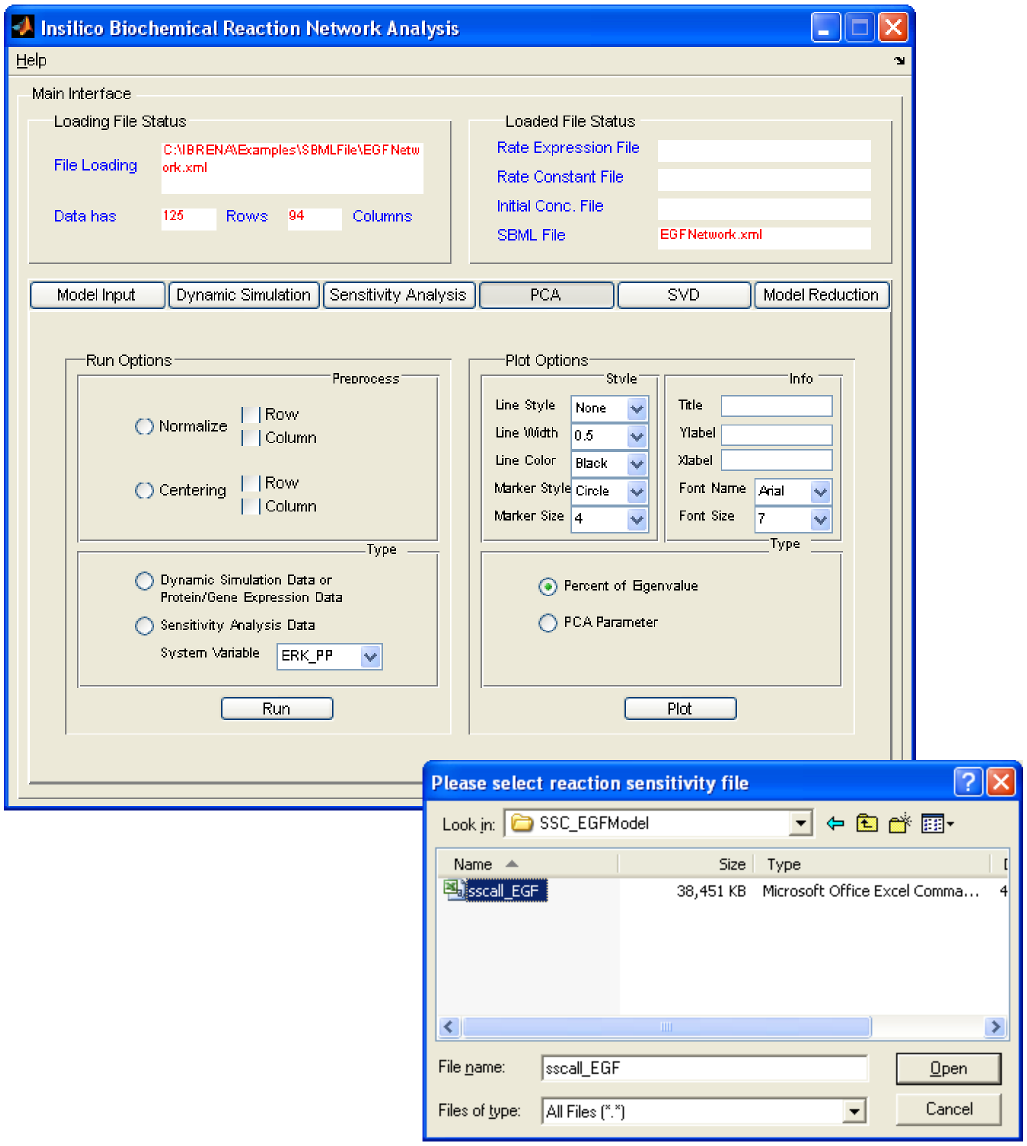
i) Following loading of the EGF reaction network files, the sensitivity matrix was formulated at EGF dose of 50 ng/ml using the "long calculation" method discussed in the previous chapter. Simulation time was set to 60 min. and time interval was 1 min. Sensitivity analysis results were stored in "ssc50.csv" via the pop-up window feature as shown before.

ii) Similar to above, we also obtained sensitivity matrices at EGF dose of 0.5 (results stored in "sscpoint5.csv") and 0.125 ng/ml ("sscpoint125.csv"). Each of these three matrices has 23502 rows (=94 forward and 94 reverse reactions*125 system parameters (Species) + 1 row containing the time points+1 row containing the comment) and 61 columns (one for each time point). Within Wij data matrix, the first 94 rows correspond to W_{ij} with respect to 1st forward reaction. The next 94 rows correspond to W_{ij} with respect to 2nd forward reaction and so on. While the first 94*125 rows corresponds to W_{ij} with respect to forward reaction, the next 94*125 rows correspond to the coefficients with respect to the reverse reactions.

Once these three sensitivity coefficient matrices were obtained, these were merged into a single large matrix. In the process of merger, while the first two rows containing the comment and the details of the simulation time points was preserved in the first matrix, this row was deleted in subsequent matrices. The entire merged matrix was named "sccall_EGF.csv", and it is provided with the software. In this file, the first 62 columns correspond to sensitivity coefficient data for EGF dose=50ng/ml. This is followed by

61 columns for stimulus=0.5ng/ml and 61 for stimulus=0.125ng/ml.

iii) Following compilation of ***S*** ("sccall_EGF.csv"), PCA was implemented by following the steps illustrated below. None of the options in the "Preprocess" panel were selected for the sample calculations since preprocessing is not required in this case. The system output was quantified in terms of the effect of reaction rate perturbation on the 59th species (C_{out} is ERK-PP in this model). Upon pressing "Run", the user is prompted to load the reaction sensitivity file which contains the matrix ***S***. For the purpose of this simulation "sccall_EGF.csv" was selected (see below). In the final step, prior to PCA analysis, the user is prompted to save the results in a comma delimited text file. The first column of this output file contains the eigenvalues (diagonal values of Λ), and this is followed by the eigenvector matrix. Comment statements in the output file clarify the format of the output file. The user is notified when calculations are completed.



Loading File Status
File Loading: C:\IBRENA\Examples\SBMLFile\EGFNetwork.xml
Data has 125 Rows 94 Columns

Loaded File Status
Rate Expression File
Rate Constant File
Initial Conc. File
SBML File: EGFNetwork.xml

Run Options
Preprocess
Normalize: Row, Column
Centering: Row, Column
Type
Dynamic Simulation Data or Protein/Gene Expression Data
Sensitivity Analysis Data
System Variable: ERK_PP

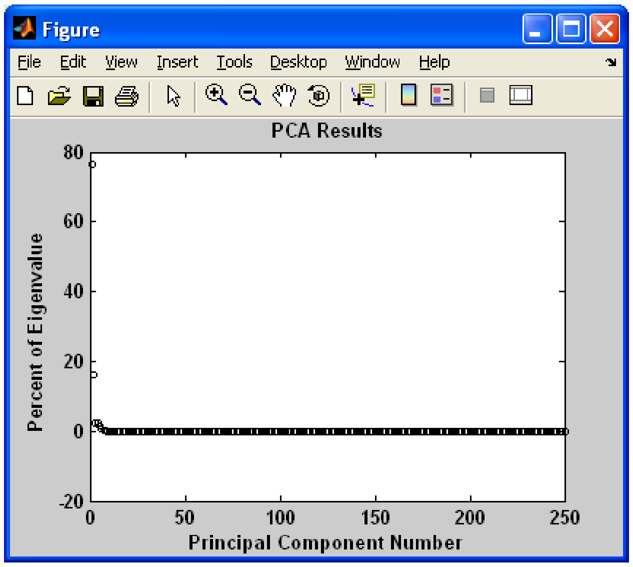
Plot Options
Style
Line Style: None
Line Width: 0.5
Line Color: Black
Marker Style: Circle
Marker Size: 4
Info
Title:
Ylabel:
Xlabel:
Font Name: Arial
Font Size: 7
Type
[✓ Percent of Eigenvalue]
PCA Parameter

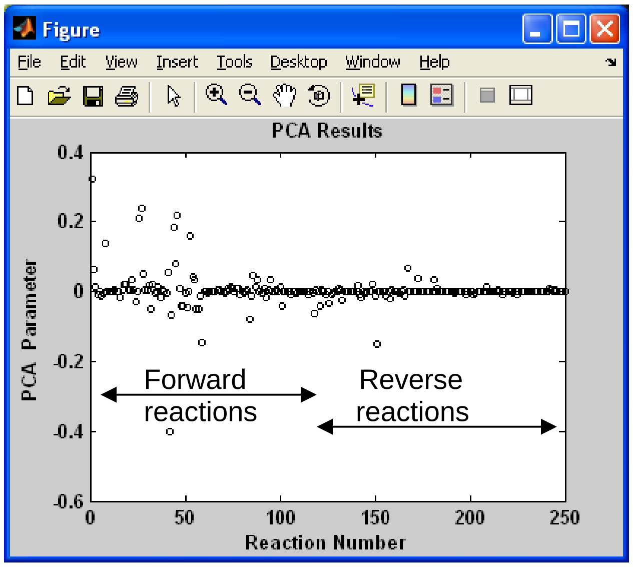
Look in: SSC_EGFModel

Name	Size	Type
sscall_EGF	38,451 KB	Microsoft Office Excel Comma...

File name: sscall_EGF
Files of type: All Files (*.*)

iv) Upon completion of the computations, the user can use the Right Panel in the “PCA” Tab to plot both the “PCA parameter” and “percent of eigenvalue”. Sample plots for these two functions are illustrated below and these replicate previously published results from our group (Liu, Swihart et al. 2005). In the case of the eigenvalue plot, the first principal component captures 75% of the variance. In the second plot (PCA parameter plot), PCA parameters are plotted for all 125 reversible reactions. In this case, the PCA parameter corresponding to the 125 forward reactions is plotted first (Reaction number 1-125) and this is followed by the 125 reverse reactions (Reaction number 126-250).





5.2.2 PCA of other data matrices

While the focus of the above example is on implementation of PCA to analyze the sensitivity coefficient matrix, it is apparent that similar analysis can be performed to interpret dynamic simulation computations and protein/gene expression data using the same module. In this regard, dynamic simulation result files generated in Chapter-3 can be directly read into IBRENA. It is also possible to format gene and protein expression data matrices following the guidelines below for analysis using IBRENA. The only difference between sensitivity and dynamic simulation data analysis is that, during PCA analysis the user must choose the “Dynamic Simulation Data or Protein/Gene Expression Data” button in the left panel instead of the “Sensitivity Analysis Data” button in the later case, prior to pressing “Run”. The user is then prompted to simply input the data file, and this is followed by PCA analysis and storage of results in local file.

PCA analysis of sensitivity coefficient matrix is more complex than dynamic simulation data analysis. In this regard, while sensitivity analysis extracts particular rows of the sensitivity coefficient data file based on the specific model specified in "Model Input" and the choice of "system variable", analysis of dynamic simulation data is more straightforward since the entire file is analyzed without discriminating between the contents of the various rows. Dynamic simulation/protein/gene expression data analysis thus does not require a "Model Input", nor does it require the selection of system variables.

With regard to the format of the input file for both PCA analysis (this chapter) and SVD analysis (next chapter), there are two types of input files used with minor difference:

a) Sensitivity analysis data matrix: Here, the first column and the first row of the input file is either left blank or contains a comment statement. The data matrix populate the second row/column and beyond. Such files can be created in Excel and saved in .csv format.

b) Dynamic Simulation data or protein/Gene expression data matrix: This file is identical to the sensitivity analysis data file, only the data populate all columns including the first one. Thus, a comment statement only occupies the first row of the input file.

The user can open the example .csv files provided with this software (see "Examples" folder) for further guidance.

6 Singular Value Decomposition (SVD)

6.1 Theory

As discussed at the beginning of the previous chapter, SVD is well suited for analysis of dynamic simulation and gene-expression/protein-array experimental results. SVD theorem declares that the matrix $\boldsymbol{X}$ (m×p dimension with m≥p) with rank r can be decomposed as:

$$\boldsymbol{X}=\boldsymbol{U\Sigma V^T} \tag{27}$$

Where $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ have dimensions m×p, p×p and p×p respectively. The columns of $\boldsymbol{U}$ are called left singular vectors and the columns of $\boldsymbol{V}$ (i.e. the rows of $\boldsymbol{V^T}$) are called right singular vectors. Both $\boldsymbol{U}$ and $\boldsymbol{V}$ are orthonormal with $\boldsymbol{UU^T=I}$, $\boldsymbol{VV^T=I}$ ($\boldsymbol{I}$ is the identity matrix). $\boldsymbol{\Sigma}$ is the diagonal matrix with non-negative singular values arranged in decreasing order ($\sigma_1>\sigma_2>\ldots$).

In the context of genomic microarray data analysis, m is the number of genes and p is the number of arrays. Thus, the left singular vectors are termed eigenassays, while the right singular vector represents the eigengenes. These eigengenes and eigenarrays extracted from SVD analysis of genome-wide temporal expression data may represent independent processes and cellular states (Alter, Brown et al. 2000).

As part of the SVD analysis, the following parameters are plotted by the software:

i) ***Fraction of variance***: The square of each singular value is proportional to the variance explained. Thus, the relative variance is plotted in order to quantify the importance of corresponding eigengenes/ eigenassays:

$$F_k=\sigma_k^2/\sum_{l=1}^{r}\sigma_l^2 \tag{28}$$

This plot of fraction of variance is sometimes also termed "fraction of eigen-expression" in literature (Alter, Brown et al. 2000).

ii) ***Right singular vector***: The right and left singular vectors are called eigengenes and eigenarrays respectively in the context of gene expression data. These parameters describe fundamental and independent patterns underlying gene/protein expression (Holter, Mitra et al. 2000), or *in silico* simulations (Wall, Andreas et al. 2003). A linear combination of these underlying patterns weighted by the singular values closely approximates the experimental data. As currently implemented, IBRENA plots the first three right singular vectors, by default. Additional plots can be generated by the user using the output data file, if necessary.

iii) ***Projection plot***: Visualization of the projection of data into SVD subspace can aid classification or grouping of data. One method for achieving this is by forming a projection scatter plot. This involves computation of the coefficients matrix $\boldsymbol{C}$ (n×p):

$$\boldsymbol{C}=\boldsymbol{X}\boldsymbol{V} \tag{29}$$

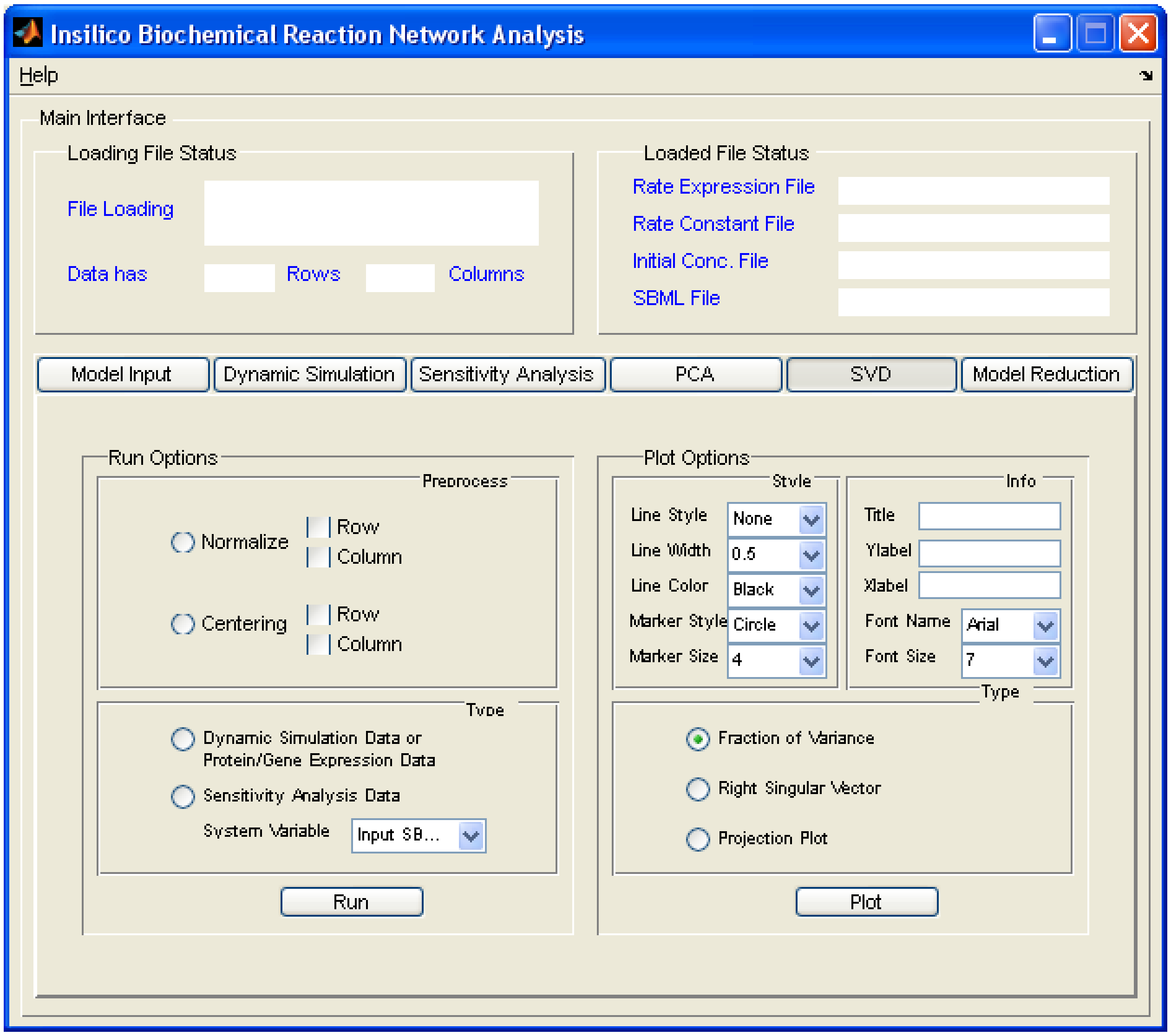
Projection plots can then be constructed by plotting the first two modes of $\boldsymbol{C}$ which correspond to the largest singular values (i.e., by plotting the first column of $\boldsymbol{C}$ along the X-axis and the second column along the Y-axis). If $\boldsymbol{X}$ is preprocessed to have zero mean and unit norm, the elements of $\boldsymbol{C}$ vary from -1 to 1, and the resulting projection plot is the same as the correlation plot.

6.2 Implementation

For the purpose of validation of the SVD method, we use gene array data from a previous publication (Alter, Brown et al. 2000). This paper analyzes elutriation data for yeast *Saccharomyces cerevisiae* involving 5981 genes across 14 time points (from 0 to 390 min). SVD analysis in IBRENA is conducted in a manner similar to PCA analysis described in the previous chapter.

i) Following the procedure outline previously (pg. 10103, (Alter, Brown et al. 2000)), various filtering and normalizing steps were performed with the initial data set to obtain the modified file: “norm_elutriation_yeast.csv”. This file is provided with the package. Please note that the first line of this input file is a comment statement. Experiment data appear after this first line.

ii) IBRENA software was used for SVD analysis on this processed file. The procedure is illustrated in the figures below. Upon selecting the “protein/gene dynamic data” option under the SVD tab and pressing “Run”, an input window appears that requests the user to load the file with the experimental dataset. The output storage site is then specified by the user after SVD computations. The raw $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{\Sigma}$ values are stored in separate output files. The user is notified when computations are complete.


Insilico Biochemical Reaction Network Analysis

Main Interface

Loading File Status
File Loading
Data has Rows Columns

Loaded File Status
Rate Expression File
Rate Constant File
Initial Conc. File
SBML File

Run Options

Preprocess
Normalize: Row, Column
Centering: Row, Column

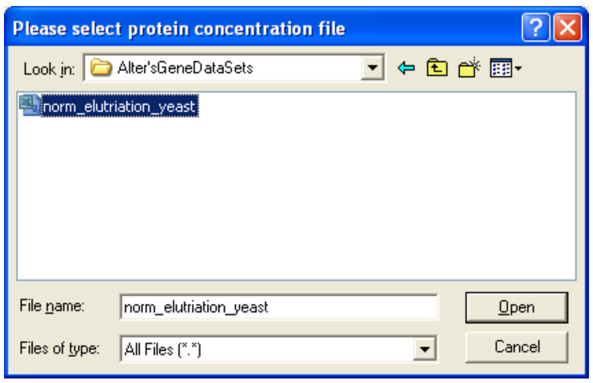
Type
Dynamic Simulation Data or Protein/Gene Expression Data
Sensitivity Analysis Data
System Variable: Input SB...

Plot Options

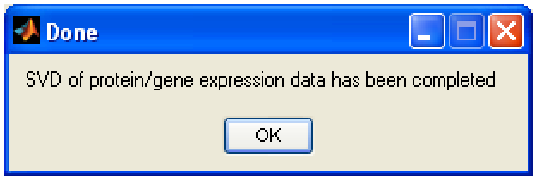
Style
Line Style: None
Line Width: 0.5
Line Color: Black
Marker Style: Circle
Marker Size: 4

Info
Title:
Ylabel:
Xlabel:
Font Name: Arial
Font Size: 7

Type
[✓ Fraction of Variance]
Right Singular Vector
Projection Plot

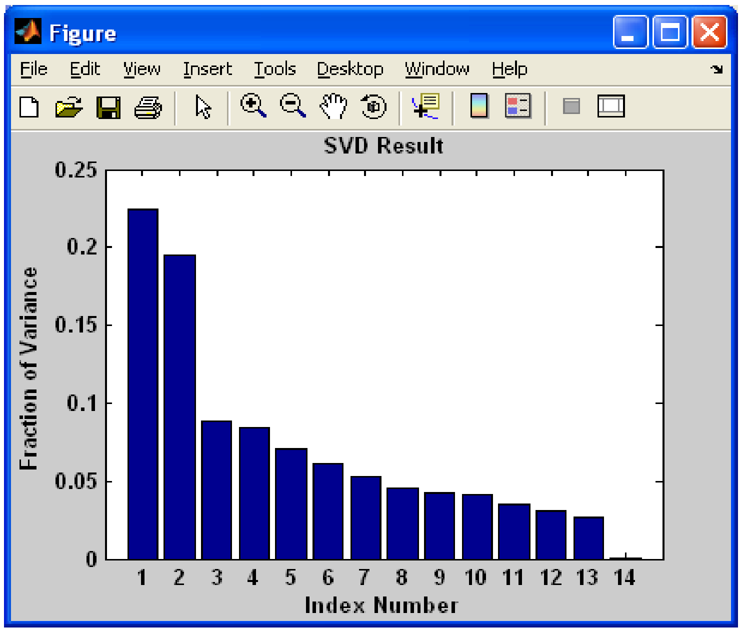

Please select protein concentration file

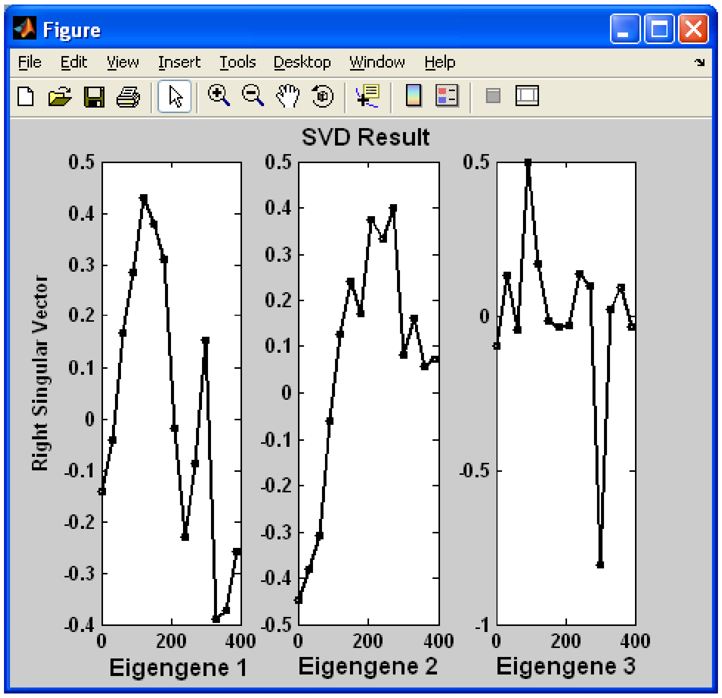
Look in: Alter'sGeneDataSets
norm_elutriation_yeast
File name: norm_elutriation_yeast
Files of type: All Files (*.*)

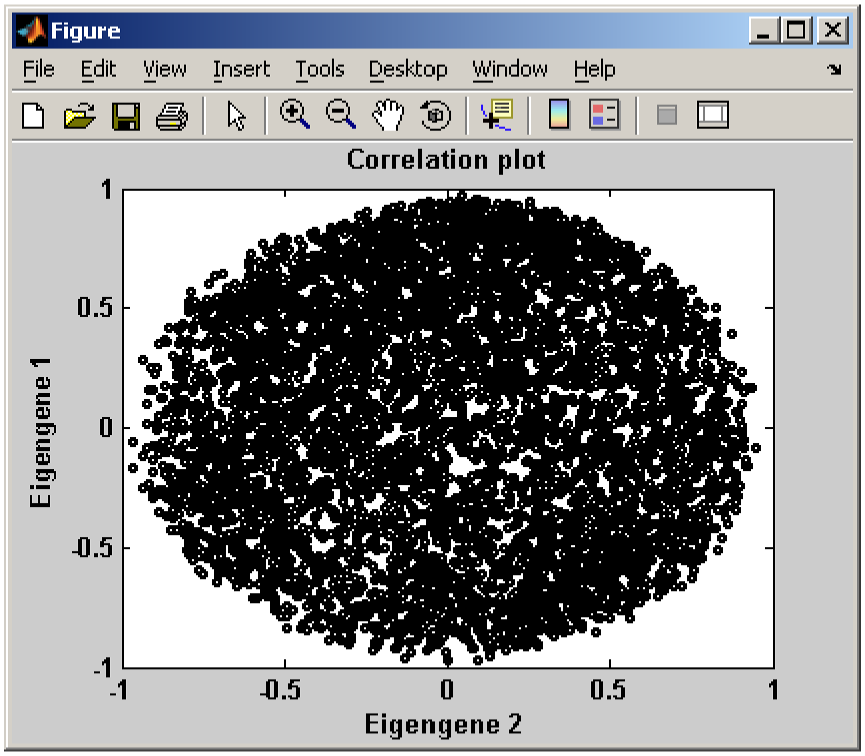

Done

SVD of protein/gene expression data has been completed

iii) Figures can then be generated for: 1) fraction of variance; 2) right singular vector and 3) projection plot using the left-side panel of the IBRENA software. These figures are illustrated below. In the first panel, fraction of variance is plotted for all 14 eigengenes. Expression patterns of the three highest-ranked eigengenes (i.e row vector of V^T) is plotted in the right singular vector plot. Finally, a projection plot (correlation plot in this case) is shown where the first two columns of ***C*** are plotted. These plots are also shown in the publication by Alter et. al (2000) (see Fig. 1b for eigen value plot, Fig. 1c for eigenvector plot, and Fig. 2b for correlation plot).







Similar to the previous chapter, SVD analysis of sensitivity coefficient data can also be performed by choosing this option in the left-hand panel during the run step of the program. Please see section 5.2.2 for details on differences in the input file format for analysis of sensitivity coefficient data matrix versus dynamic simulation/gene/protein expression data matrix.

7 Model Reduction

7.1 Theory and Algorithm

It is often desirable to reduce complex mathematical models by deleting reactions that do not contribute substantially to system output. These reactions with minimal contribution have two features in common: i) They exhibit low sensitivity coefficients (low W_{ij}), i.e. perturbation of these reactions does not result in marked changes in system output. ii) Their reaction flux of is low, i.e. they are not very active at any time during the course of the simulation. Based on the above rationale, here, we combined flux analysis (described below) and sensitivity analysis (Chapter 4) to delete redundant reactions with the goal of reducing the complexity of the mathematical model.

The algorithm implemented is based on our previous publication with modifications that are outlined below (Liu, Swihart et al. 2005). These modifications are incorporated in order to automate the process of model reduction. Model reduction using the algorithm described below is controlled by three parameters, primarily by the reduction factor λ, and also by two other parameters δ (critical flux coefficient) and ε (error tolerance). In IBRENA, these three parameters are specified by the user at the start of the simulation. In each run-time, δ and ε are unchanged while λ is automatically changed in a step-wise fashion if the relative error is larger than ε. The sequential computational steps include:

i) The mean reaction flux $\overline{F}_j (= \left| \int_{t1}^{t2} v_j dt / (t_2 - t_1) \right|)$ is calculated for each of the reactions over the time interval specified by the user.

ii) The maximum absolute value of the scaled sensitivity coefficient is calculated for all the reactions in this same time interval. This maximum value is denoted $|W_{ij}|_{\max}$ and this is based on the long calculations of chapter 4.

iii) A parameter $|W_{crit}|$ is chosen for the entire network based on a fitted 'reduction factor', λ according to the equation $\left|W_{crit}\right| = \lambda \cdot \left|W_{ij}\right|_{\max}$.

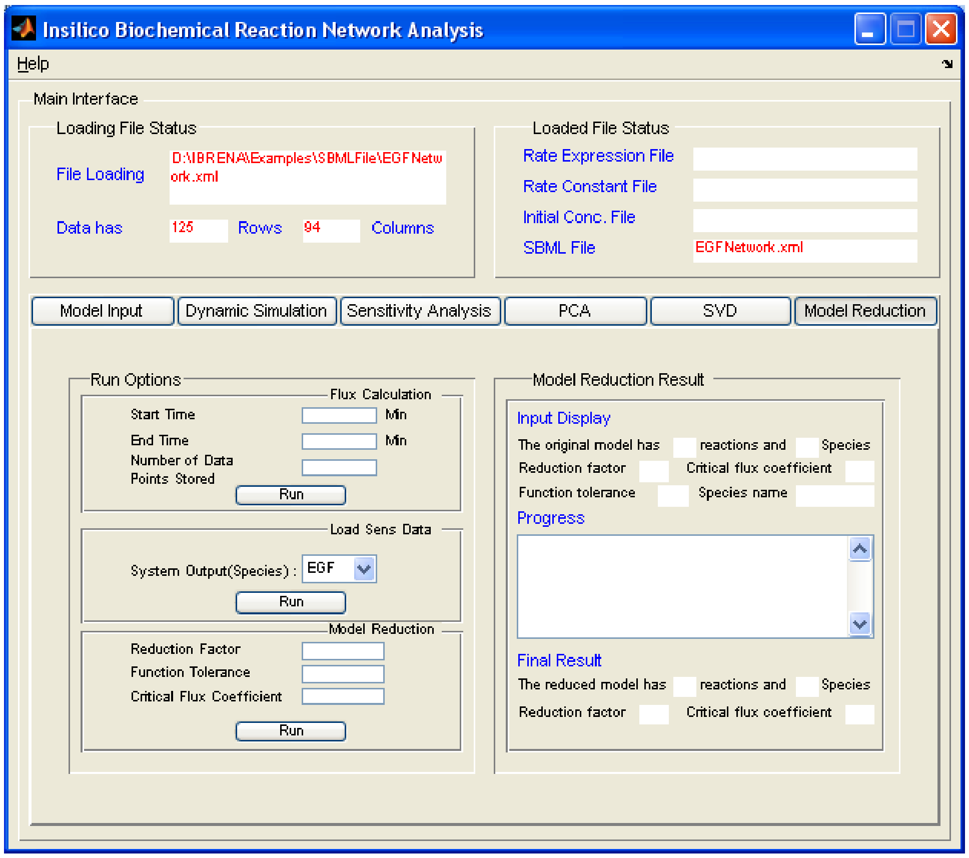
iv) All reactions with absolute value of scaled sensitivity coefficient greater than $|W_{crit}|$ are classified to be 'essential reactions' and the remaining was termed 'non-essential'. Among the essential reactions, the reaction with the lowest flux is calculated.

v) Some of the non-essential reactions that have low flux are eliminated during model reduction based on a parameter called critical flux coefficient δ (for example, δ=0.9). By definition these deleted reactions have mean reaction flux<F_{crit} with $F_{crit} = \delta(smallest\ \overline{F}_j \text{ among essential reactions})$. Thus, the deleted reactions have low reaction flux.

vi) After selected non-essential reactions are deleted, the relative difference in the system output (as measured by C_j) between the original model (Ori) and the reduced model (Red) is estimated using the following expression ($= \frac{1}{m}\sum_{i=1}^{m}\left|(1-C_j^{Red}/C_j^{Ori})_{at\ time\ i}\right|$) where *i* specifies the time point and *m* the number of data points. If this relative error is larger than the user-specified error tolerance ε, the reduction factor is dropped by 0.02 and the algorithm steps back to step iii). This procedure is repeated until the relative error is just lower than the error tolerance.

7.2 Implementation

Model reduction is currently only possible using the full-version of IBRENA, and not the stand-alone application. The user interface for model reduction is shown below.



File Loading D:\IBRENA\Examples\SBMLFile\EGFNetwork.xml

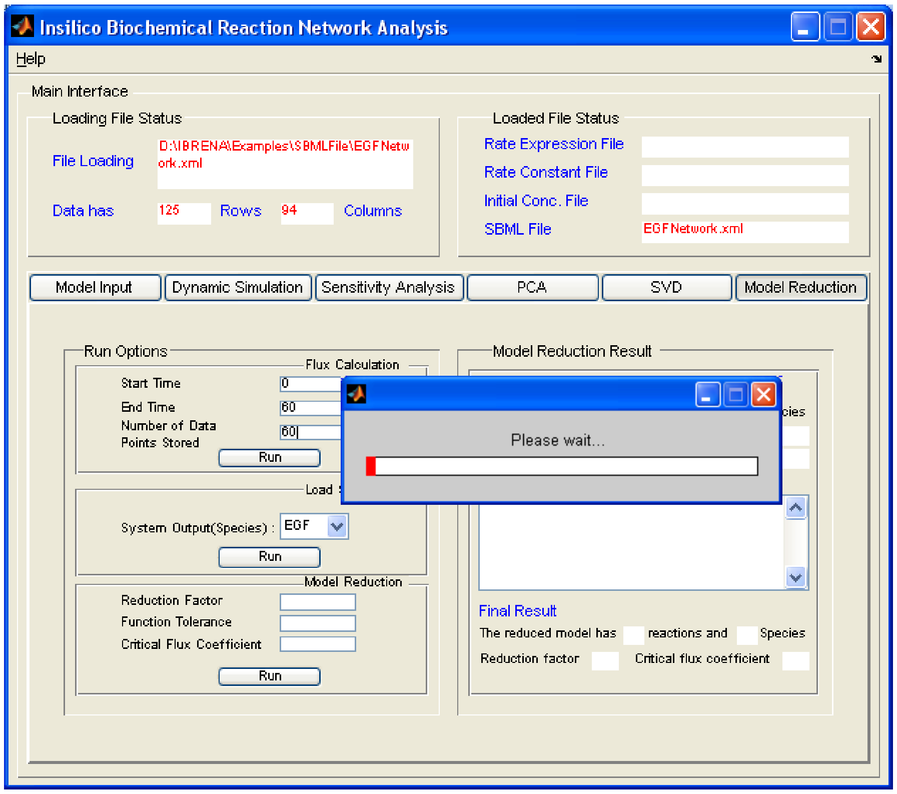
Data has 125 Rows 94 Columns

SBML File EGFNetwork.xml

System Output(Species) : EGF

Based on the algorithm described above, flux and sensitivity data sets have to be simulated and read into the program memory prior to initiation of model reduction. Thus three steps as necessary:

i) Flux Calculation: Average flux for individual reactions ($\overline{F}_j$) is calculated after the user inputs "start time", "end time" and "number of data points stored" in the upper section of "Run options" interface. As described previously a wait bar shows progress with the computations and the user is notified upon completion of computations. Selected windows are shown below for the sake of illustration.


Insilico Biochemical Reaction Network Analysis

Help

Main Interface

Loading File Status

File Loading: D:\IBRENA\Examples\SBMLFile\EGFNetwork.xml

Data has 125 Rows 94 Columns

Loaded File Status

Rate Expression File

Rate Constant File

Initial Conc. File

SBML File: EGFNetwork.xml

Model Input | Dynamic Simulation | Sensitivity Analysis | PCA | SVD | Model Reduction

Run Options

Flux Calculation

Start Time: 0

End Time: 60

Number of Data Points Stored: 60

Run

Load

System Output(Species) : EGF

Run

Model Reduction

Reduction Factor

Function Tolerance

Critical Flux Coefficient

Run

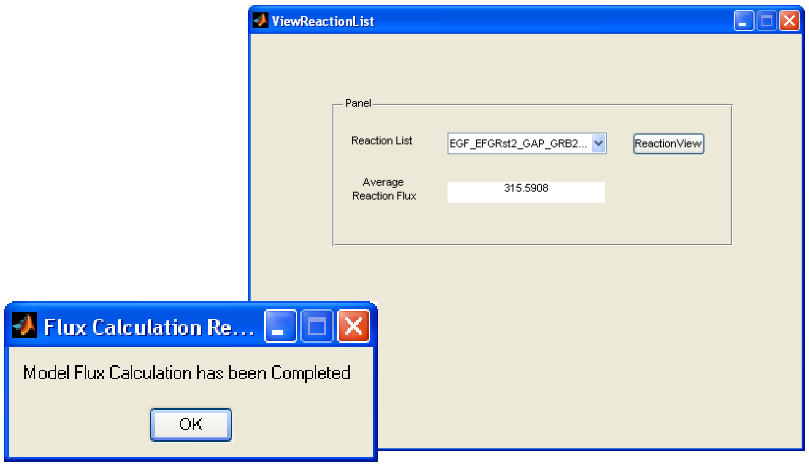
Model Reduction Result

Please wait...

Final Result

The reduced model has reactions and Species

Reduction factor Critical flux coefficient


ViewReactionList

Panel

Reaction List: EGF_EFGRst2_GAP_GRB2...

ReactionView

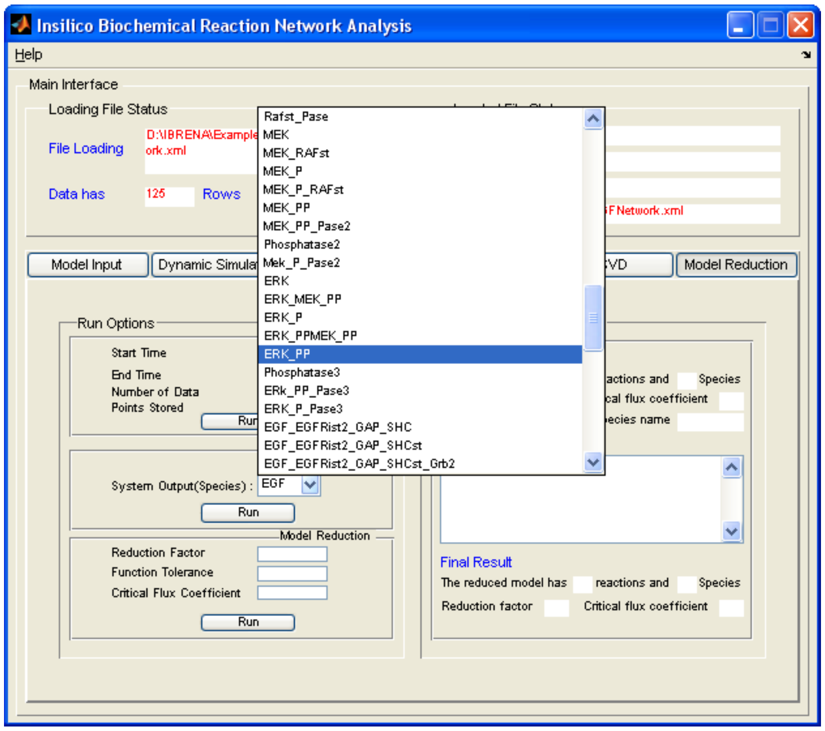
Average Reaction Flux: 315.5908

Flux Calculation Re...

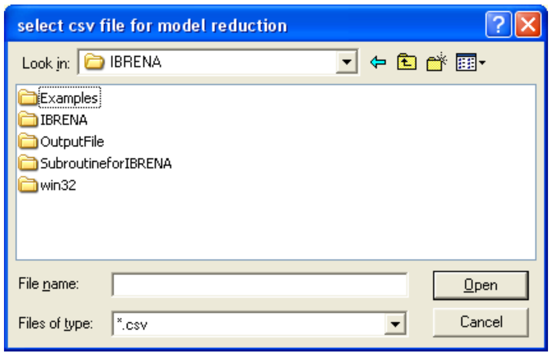
Model Flux Calculation has been Completed

OK

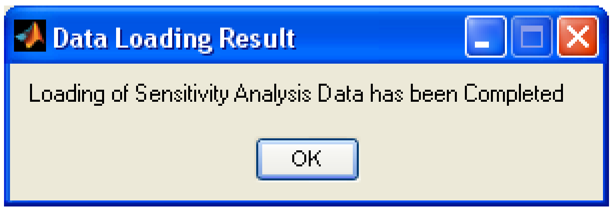
ii) Sensitivity Analysis: Load Sensitivity Analysis data from previous “Long Calculation” step as described in “Sensitivity Analysis” tab. The user should first choose the desired species from the popup menu in the interface as system output. Then a popup window will notify the user to choose the Sensitivity Analysis Data File. The user is notified when this step is completed.



File Loading: D:\IBRENA\Example ork.xml
Data has 125 Rows
F Network.xml
System Output(Species) : EGF

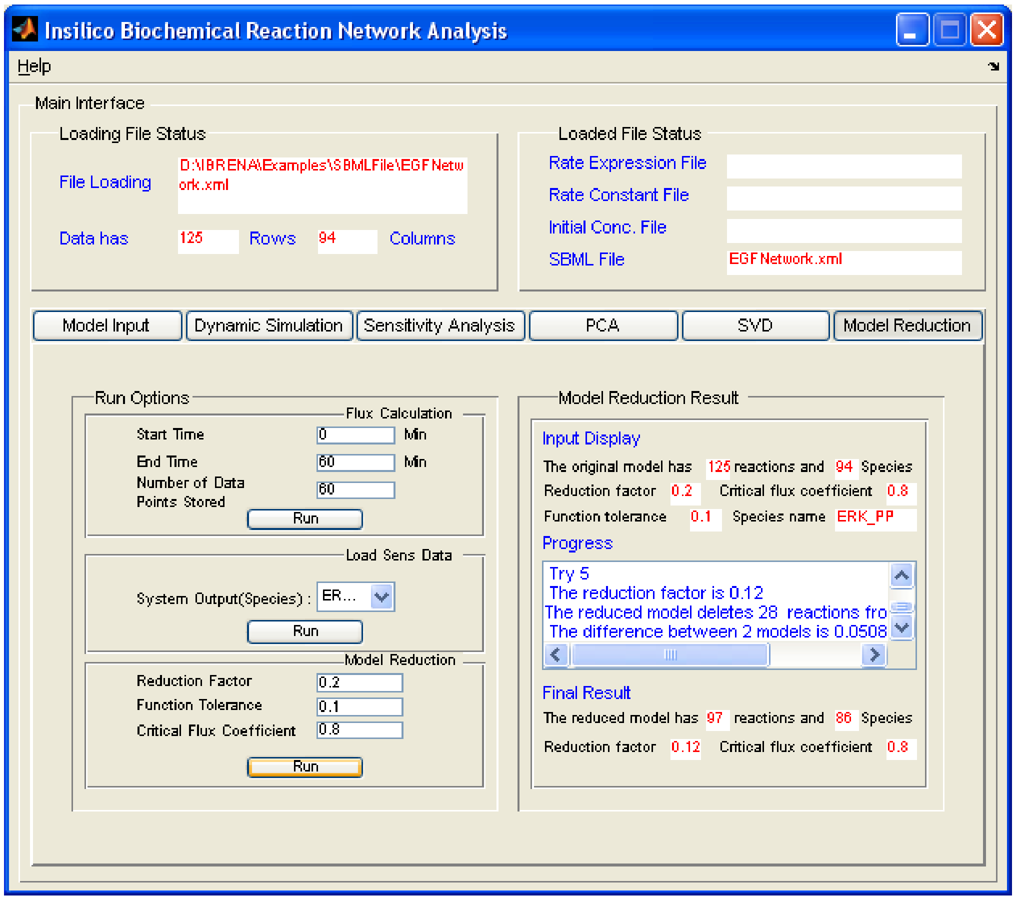


Files of type: *.csv



Loading of Sensitivity Analysis Data has been Completed

iii) Model Reduction: Once the user inputs λ, δ, ε and then presses "Run", the program will initiate model reduction. The progress report can be tracked using the self-explanatory interface shown on the right side. Output are stored/analyzed in three formats: 1) Reduced SBML model can be saved as SBML File in the hard drive. This can be visualized using an array of plotting/visualization software available at sbml.org; 2) The time course of specified species concentration in both the original and reduced model is plotted in order to compare these two models. 3) Finally, an interface with reaction lists in the reduced model is displayed for further analysis. These three outputs, for the analysis of the EGF network, are shown below.



File Loading: D:\IBRENA\Examples\SBMLFile\EGFNetwork.xml

Data has 125 Rows 94 Columns

Rate Expression File:

Rate Constant File:

Initial Conc. File:

SBML File: EGFNetwork.xml

Start Time: 0 Min

End Time: 60 Min

Number of Data Points Stored: 60

System Output(Species) : ER...

Reduction Factor: 0.2

Function Tolerance: 0.1

Critical Flux Coefficient: 0.8

The original model has 125 reactions and 94 Species

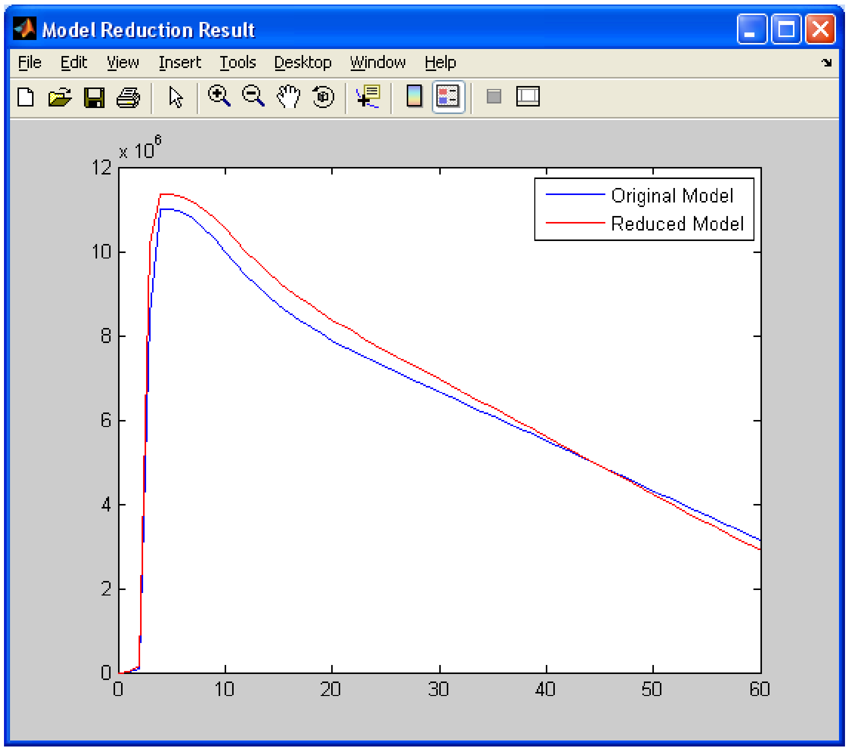
Reduction factor 0.2 Critical flux coefficient 0.8

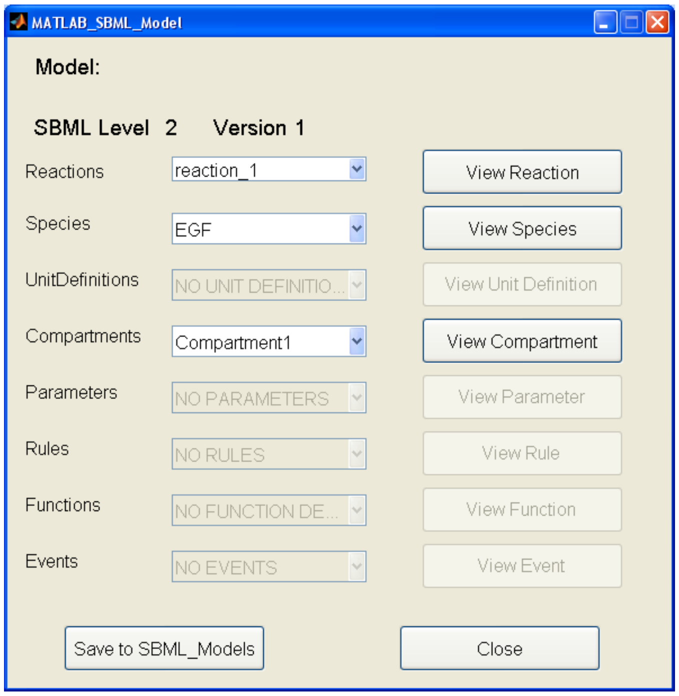
Function tolerance 0.1 Species name ERK_PP

```
Try 5
The reduction factor is 0.12
The reduced model deletes 28  reactions fro
The difference between 2 models is 0.0508
```

The reduced model has 97 reactions and 86 Species

Reduction factor 0.12 Critical flux coefficient 0.8

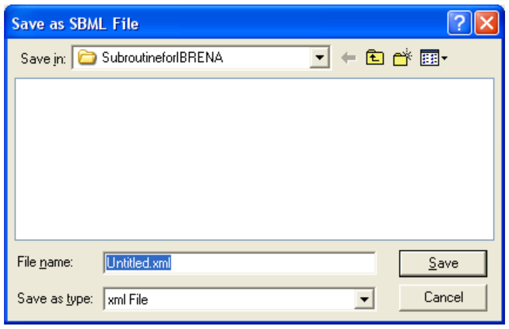




Model:

SBML Level 2 Version 1

Reactions	reaction_1	View Reaction
Species	EGF	View Species
UnitDefinitions	NO UNIT DEFINITIO...	View Unit Definition
Compartments	Compartment1	View Compartment
Parameters	NO PARAMETERS	View Parameter
Rules	NO RULES	View Rule
Functions	NO FUNCTION DE...	View Function
Events	NO EVENTS	View Event



Save in: SubroutineforBRENA

File name: Untitled.xml

Save as type: xml File

8 References

Alter, O., P. O. Brown, et al. (2000). "Singular value decomposition for genome-wide expression data processing and modeling." Proc Natl Acad Sci U S A **97**(18): 10101-6.

Cao, Y., S. T. Li, et al. (2003). "Adjoint sensitivity analysis or differential-algebraic equations: The adjoint DAE system and its numerical solution." Siam Journal on Scientific Computing **24**(3): 1076-1089.

Ederer, M. and E. D. Gilles (2007). "Thermodynamically feasible kinetic models of reaction networks." Biophys J **92**(6): 1846-57.

Hindmarsh, A. C., P. N. Brown, et al. (2005). "SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers." Acm Transactions on Mathematical Software **31**(3): 363-396.

Holter, N. S., M. Mitra, et al. (2000). "Fundamental patterns underlying gene expression profiles: simplicity from complexity." Proc Natl Acad Sci U S A **97**(15): 8409-14.

Janes, K. A., J. R. Kelly, et al. (2004). "Cue-signal-response analysis of TNF-induced apoptosis by partial least squares regression of dynamic multivariate data." Journal of Computational Biology **11**(4): 544-61.

Keating, S. M., B. J. Bornstein, et al. (2006). "SBMLToolbox: an SBML toolbox for MATLAB users." Bioinformatics **22**(10): 1275-7.

Kholodenko, B. N., O. V. Demin, et al. (1999). "Quantification of short term signaling by the epidermal growth factor receptor." J Biol Chem **274**(42): 30169-81.

Liu, G., M. T. Swihart, et al. (2005). "Sensitivity, principal component and flux analysis applied to signal transduction: the case of epidermal growth factor mediated signaling." Bioinformatics **21**(7): 1194-202.

Machne, R., A. Finney, et al. (2006). "The SBML ODE Solver Library: a native API for symbolic and fast numerical analysis of reaction networks." Bioinformatics **22**(11): 1406-7.

Pittelkow, Y. and S. R. Wilson (2005). "Use of principal component analysis and the GE-biplot for the graphical exploration of gene expression data." Biometrics **61**(2): 630-632.

Schoeberl, B., C. Eichler-Jonsson, et al. (2002). "Computational modeling of the dynamics of the MAP kinase cascade activated by surface and internalized EGF receptors." Nat. Biotechnol. **20**(4): 370-5.

Vajda, S., P. Valvko, et al. (1985). "Principal Component Analysis of Kinetic Models." International Journal of Chemical Kinetics **17**: 55-81.

Wall, M. E., R. Andreas, et al. (2003). Singular value decomposition and principal component analysis. A Practical Approach to Microarray Data Analysis. D.P. Berrar, W. Dubitzky and M. Granzow. Norwell MA**:** 91-109.

Yang, J., W. J. Bruno, et al. (2006). "On imposing detailed balance in complex reaction mechanisms." Biophys J **91**(3): 1136-41.

Appendix I

A Reaction equations and kinetic parameters

Reaction number	Reaction equation	Kinetic parameter	
		Forward Rate Constant	Reverse Rate Constant
v1	[EGFR]+[EGF] ↔ [EGF-EGFR]	2.99E-06	2.30E-01
v2	[EGF-EGFR]+[EGF-EGFR] ↔ [(EGF-EGFR)2]	1.00E-03	6.00E+00
v3	[(EGF-EGFR)2] ↔ [(EGF-EGFR*)2]	6.00E+01	6.00E-01
v4	[(EGF-EGFR*)2-GAP-Grb2]+[Prot] ↔ [(EGF-EGFR*)2-GAP-Grb2-Prot]	1.04E-05	9.96E-02
v5	[(EGF-EGFR*)2-GAP-Grb2-Prot] → [(EGF-EGFRi*)2-GAP-Grb2]+[Proti]	1.06E+00	0.00E+00
v6	[EGFR] ↔ [EGFRi]	3.00E-02	3.00E-01
v7	[(EGF-EGFR*)2] → [(EGF-EGFRi*)2]	3.00E-02	3.00E-01
v8	[(EGF-EGFR*)2]+[GAP] ↔ [(EGF-EGFR*)2-GAP]	1.00E-04	1.20E+01
v9	[(EGF-EGFR*)2-GAP] → [(EGF-EGFRi*)2-GAP]	3.00E-02	3.00E-01
v10	[EGFRi]+[EGFi] ↔ [EGF-EGFRi]	3.26E+00	6.60E-01
v11	[EGF-EGFRi]+[EGF-EGFRi] ↔ [(EGF-EGFRi)2]	1.00E-03	6.00E+00
v12	[(EGF-EGFR)2i] ↔ [(EGF-EGFRi*)2]	6.00E+01	6.00E-01
v13	→ [EGFR]	1.30E+02	0.00E+00
v14	[(EGF-EGFRi*)2]+ [GAP] ↔ [(EGF-EGFRi*)2-GAP]	1.00E-04	1.20E+01
v15	[Proti] → [Prot]	6.00E+05	0.00E+00
v16, 63	[(EGF-EGFR*)2-GAP]+[Grb2] ↔ [(EGF-EGFR*)2-GAP-Grb2]	1.00E-03	1.65E+01
v17, 64	[(EGF-EGFR*)2-GAP-Grb2]+[Sos] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos]	1.00E-03	3.60E+00
v18, 65	[(EGF-EGFR*)2-GAP-Grb2-Sos]+[Ras-GDP] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP]	1.50E-03	7.80E+01
v19, 66	[(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos]+[Ras-GTP]	3.00E+01	1.00E-05
v20, 67	[Ras-GTP*]+[(EGF-EGFR*)2-GAP-Grb2-Sos] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP]	2.10E-04	2.40E+01
v21, 68	[(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP] ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos]+[Ras-GDP]	1.38E+00	2.20E-05
v22, 69	[(EGF-EGFR*)2-GAP]+[Shc] ↔ [(EGF-EGFR*)2-GAP-Shc]	2.10E-03	6.00E+00
v23, 70	[(EGF-EGFR*)2-GAP-Shc] ↔ [(EGF-EGFR*)2-GAP-Shc*]	3.60E+02	3.60E+00
v24, 71	[(EGF-EGFR*)2-GAP-Shc*]+[Grb2] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2]	1.00E-03	3.30E+01
v25, 72	[(EGF-EGFR*)2-GAP-Shc*-Grb2]+[Sos] ↔ [(EGF-EGFR*)2-GAP-Shc-Grb2-Sos]	1.00E-03	1.28E+00
v26, 73	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos]+[Ras-GDP] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP]	1.50E-03	7.80E+01
v27, 74	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos] + [Ras-GTP]	3.00E+01	1.00E-05
v28, 75	[Raf]+[Ras-GTP] ↔ [Raf-Ras-GTP]	1.00E-04	3.18E-01
v29, 76	[Raf-Ras-GTP] ↔ [Raf*]+[Ras-GTP*]	6.00E+01	7.00E-05
v30, 77	[Ras-GTP*]+[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP]	2.10E-04	2.40E+01
v31, 78	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos]+[Ras-GDP]	1.38E+00	2.20E-05
v32, 79	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos] ↔ [(EGF-EGFR*)2-GAP]+[Shc-Grb2-Sos]	6.00E+00	2.40E-05
v33	[Shc*-Grb2-Sos] ↔ [Grb2-Sos]+[Shc*]	1.20E+01	2.10E-03
v34, 80	[(EGF-EGFR*)2-GAP-Grb2-Sos] ↔ [(EGF-EGFR*)2-GAP]+[Grb2-Sos]	1.80E+00	4.50E-04
v35	[Grb2-Sos] ↔ [Grb2] +[Sos]	9.00E-02	4.50E-04
v36	[Shc*] ↔ [Shc]	3.00E-01	0.00E+00
v37, 81	[(EGF-EGFR*)2-GAP-Shc*] ↔ [(EGF-EGFR*)2-GAP]+[Shc*]	1.80E+01	9.00E-05
v38	[Shc*]+[Grb2] ↔ [Shc*-Grb2]	1.00E-03	3.30E+01
v39, 82	[(EGF-EGFR*)2-GAP-Shc*-Grb2] ↔ [(EGF-EGFR*)2-GAP]+[Shc*-Grb2]	1.80E+01	9.00E-05
v40	[Shc*-Grb2]+[Sos] ↔ [Shc*-Grb2-Sos]	3.00E-03	3.84E+00
v41, 83	[(EGF-EGFR*)2-GAP-Shc*] + [Grb2-Sos] ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos]	3.00E-03	2.57E+00
v42, 84	[Raf*]+[Phosphatase1] ↔ [Raf*-Phosphatase1]	7.10E-03	1.20E+01
v43, 85	[Raf*-Phosphatase1] → [Raf]+[Phosphatase1]	6.00E+01	0.00E+00
v44, 86	[MEK] + [Raf*] ↔ [MEK-Raf*]	1.17E-03	1.10E+00
v45, 87	[MEK-Raf*] → [MEK-P] +[Raf*]	2.10E+02	0.00E+00
v46, 88	[MEK-P]+[Raf*] ↔ [MEK-P-Raf*]	1.17E-03	1.10E+00
v47, 89	[MEK-P-Raf*] → [MEK-PP] + [Raf*]	1.74E+02	0.00E+00
v48, 90	[MEK-PP]+[Phosphatase2] ↔ [MEK-PP-Phosphatase2]	1.43E-03	4.80E+01
v49, 91	[MEK-PP-Phosphatase2] → [MEK-P] + [Phosphatase2]	3.48E+00	0.00E+00
v50, 92	[MEK-P]+[Phosphatase2] ↔ [MEK-P-Phosphatase2]	2.70E-05	3.00E+01
v51, 93	[MEK-P-Phosphatase2] → [MEK]+[Phosphatase2]	3.48E+00	0.00E+00
v52, 94	[ERK]+[MEK-PP] ↔ [ERK-MEK-PP]	5.34E-03	1.98E+00
v53, 95	[ERK-MEKK-PP] → [ERK-P]+[MEK-PP]	9.60E+02	0.00E+00
v54, 96	[ERK-P]+[MEK-PP] ↔ [ERK-P-MEK-PP]	5.34E-03	1.98E+00
v55, 97	[ERK-P-MEK-PP] → [ERK-PP]+[MEK-PP]	3.42E+02	0.00E+00
v56, 98	[ERK-PP]+[Phosphatase3] ↔ [ERK-PP-Phosphatase3]	1.41E-03	3.60E+01
v57, 99	[ERK-PP-Phosphatase3] → [ERK-P]+[Phosphatase3]	1.48E+01	0.00E+00
v58,100	[ERK-P] + [Phosphatase3] ↔ [ERK-P-Phosphatase3]	5.00E-04	3.00E+01
v59,101	[ERK-P-Phosphatase3] → [ERK]+[Phosphatase3]	1.48E+01	0.00E+00
v60	[EGFRi] → [EGFRideg]	4.00E-02	0.00E+00
v61	[EGFi]→ [EGFideg]	1.00E-02	0.00E+00
v62	[(EGF-EGFRi*)2] ↔ [(EGF-EGFRi*)2deg]	4.00E-02	0.00E+00
v102	[(EGF-EGFR*)2-GAP] ↔ [(EGF-EGFRi*)2-GAP]	3.00E-02	3.00E-01
v103	[(EGF-EGFR*)2-GAP-Shc] ↔ [(EGF-EGFRi*)2-GAP-Shc]	3.00E-02	3.00E-01
v104	[(EGF-EGFR*)2-GAP-Shc*] ↔ [(EGF-EGFRi*)2-GAP-Shc*]	3.00E-02	3.00E-01
v105	[(EGF-EGFR*)2-GAP-Grb2-Sos] ↔ [(EGF-EGFRi*)2-GAP-Grb2-Sos]	3.00E-02	3.00E-01
v106	[(EGF-EGFR*)2-GAP-Grb2-Sos]+Prot ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos-Prot]	1.04E-05	9.96E-02
v107	[(EGF-EGFR*)2-GAP-Grb2-Sos-Prot] ↔ Proti+[(EGF-EGFRi*)2-GAP-Grb2-Sos]	1.06E+00	0.00E+00
v108	[(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP] ↔ [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GDP]	3.00E-02	3.00E-01
v109	[(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP]+Prot ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP-Prot]	1.00E-05	9.96E-02
v110	[(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GDP-Prot] ↔ Proti+[(EGF-EGFRi*)2-GAP-Grb2-Sos]	1.06E+00	0.00E+00
v111	[(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP] ↔ [(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GTP]	3.00E-02	3.00E-01
v112	[(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP]+Prot ↔ [(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP-Prot]	1.04E-05	9.96E-02
v113	[(EGF-EGFR*)2-GAP-Grb2-Sos-Ras-GTP-Prot] ↔ Proti+[(EGF-EGFRi*)2-GAP-Grb2-Sos-Ras-GTP]	1.06E+00	0.00E+00
v114	[(EGF-EGFR*)2-GAP-Shc*-Grb2] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2]	3.00E-02	3.00E-01
v115	[(EGF-EGFR*)2-GAP-Shc*-Grb2]+Prot ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Prot]	1.04E-05	9.96E-02
v116	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Prot] ↔ Proti+[(EGF-EGFRi*)2-GAP-Shc*-Grb2]	1.06E+00	0.00E+00
v117	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos]	3.00E-02	3.00E-01
v118	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos]+Prot ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Prot]	1.04E-05	9.96E-02
v119	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Prot] ↔ Proti+[(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos]	1.06E+00	0.00E+00
v120	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GDP]	3.00E-02	3.00E-01
v121	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP]+Prot ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP-Prot]	1.04E-05	9.96E-02
v122	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GDP-Prot] ↔ Proti+[(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GDP]	1.06E+00	0.00E+00
v123	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP] ↔ [(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GTP]	3.00E-02	3.00E-01
v124	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP]+Prot ↔ [(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP-Prot]	1.04E-05	9.96E-02
v125	[(EGF-EGFR*)2-GAP-Shc*-Grb2-Sos-Ras-GTP-Prot] ↔ Proti+[(EGF-EGFRi*)2-GAP-Shc*-Grb2-Sos-Ras-GTP]	1.06E+00	0.00E+00

Note: [#] Denotes rate constants that are different from Schoeberl et al.

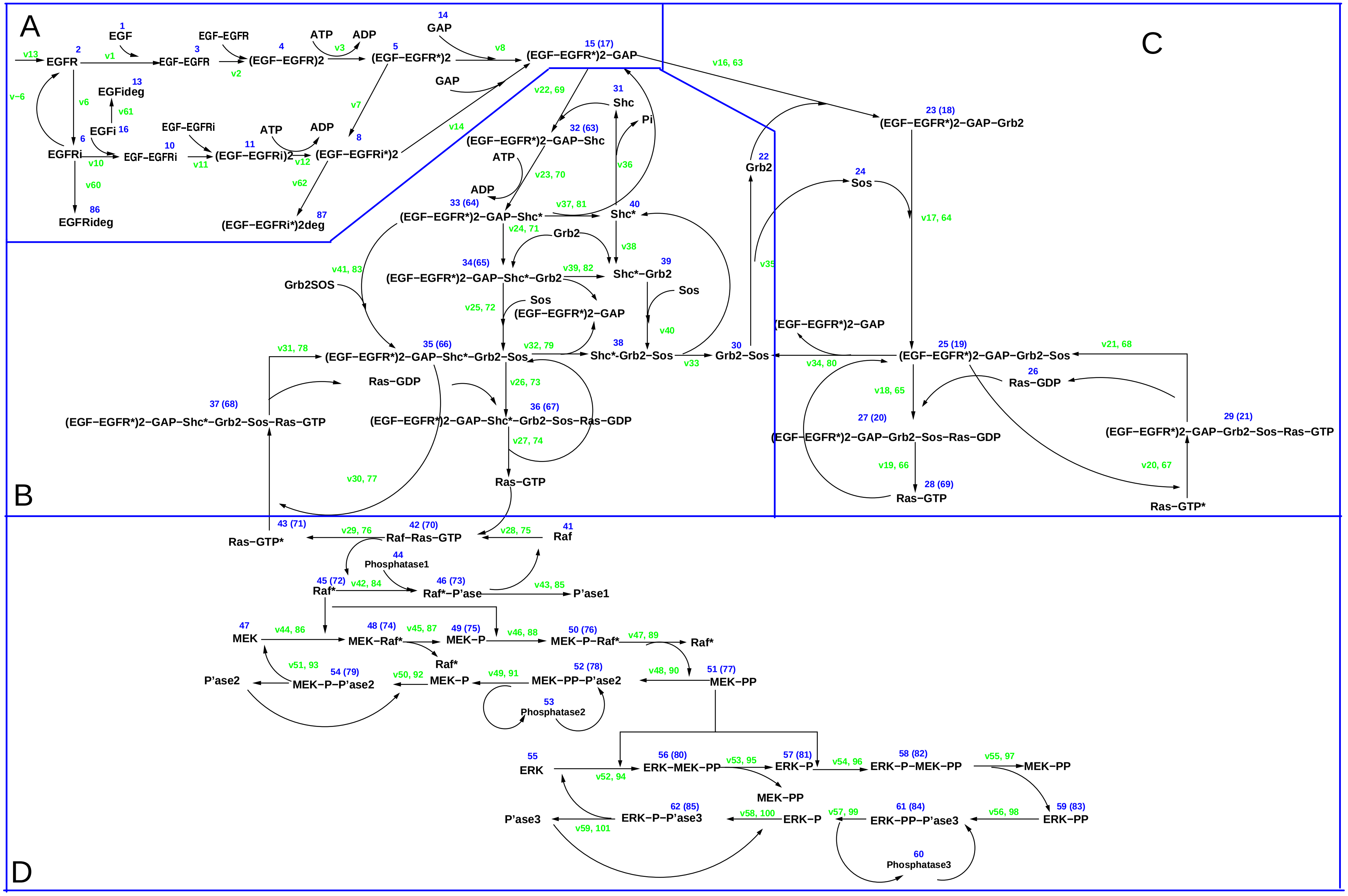
first order rate constants in 1/min and second order rate constants in 1/(molecule/cell)/min

B Initial protein concentrations

Protein	Number of molecules per cell (in Schoeberl et.al[†])	Number of molecules per cell (in the paper)
Receptors Total	5.00×10^{4}	same as Schoeberl et.al[†]
GAP	1.20×10^{4}	same as Schoeberl et.al[†]
Shc	1.01×10^{4}	1.60×10^{5}
Grb2	5.10×10^{4}	9.00×10^{4}
Sos	6.63×10^{4}	3.6×10^{4}
Ras-GDP Total	1.14×10^{7}	7.2×10^{4}
Raf-kinase	4.00×10^{4}	same as Schoeberl et.al[†]
Mek Total	2.20×10^{4}	same as Schoeberl et.al[†]
Erk Total	2.10×10^{4}	same as Schoeberl et.al[†]
Phosphatase 1	4.00×10^{4}	same as Schoeberl et.al[†]
Phosphatase 2	4.00×10^{4}	same as Schoeberl et.al[†]
Phosphatase 3	1.00×10^{4}	same as Schoeberl et.al[†]
Coated-Pit Protein	8.10×10^{4}	same as Schoeberl et.al[†]

[†] Schoeberl, B., Eichler-Jonsson, C., Gilles, E. D. and Muller, G. (2002) Nat. Biotechnol., 20(4), 370-5.

EGF reaction network schematic. Please see Schoeberl et al. (2002) or Liu et al. (2005) for details of individual components and reactions.



A
B
C
D
1 EGF
2 EGFR
3 EGF–EGFR
4 (EGF–EGFR)2
5 (EGF–EGFR*)2
6 EGFRi
8 (EGF–EGFRi*)2
10 EGF–EGFRi
11 (EGF–EGFRi)2
13 EGFideg
14 GAP
15 (17) (EGF–EGFR*)2–GAP
16 EGFi
22 Grb2
23 (18) (EGF–EGFR*)2–GAP–Grb2
24 Sos
25 (19) (EGF–EGFR*)2–GAP–Grb2–Sos
26 Ras–GDP
27 (20) (EGF–EGFR*)2–GAP–Grb2–Sos–Ras–GDP
28 (69) Ras–GTP
29 (21) (EGF–EGFR*)2–GAP–Grb2–Sos–Ras–GTP
30 Grb2–Sos
31 Shc
32 (63) (EGF–EGFR*)2–GAP–Shc
33 (64) (EGF–EGFR*)2–GAP–Shc*
34(65) (EGF–EGFR*)2–GAP–Shc*–Grb2
35 (66) (EGF–EGFR*)2–GAP–Shc*–Grb2–Sos
36 (67) (EGF–EGFR*)2–GAP–Shc*–Grb2–Sos–Ras–GDP
37 (68) (EGF–EGFR*)2–GAP–Shc*–Grb2–Sos–Ras–GTP
38 Shc*-Grb2–Sos
39 Shc*–Grb2
40 Shc*
41 Raf
42 (70) Raf–Ras–GTP
43 (71) Ras–GTP*
44 Phosphatase1
45 (72) Raf*
46 (73) Raf*–P'ase
47 MEK
48 (74) MEK–Raf*
49 (75) MEK–P
50 (76) MEK–P–Raf*
51 (77) MEK–PP
52 (78) MEK–PP–P'ase2
53 Phosphatase2
54 (79) MEK–P–P'ase2
55 ERK
56 (80) ERK–MEK–PP
57 (81) ERK–P
58 (82) ERK–P–MEK–PP
59 (83) ERK–PP
60 Phosphatase3
61 (84) ERK–PP–P'ase3
62 (85) ERK–P–P'ase3
86 EGFRideg
87 (EGF–EGFRi*)2deg
EGF–EGFR
EGF–EGFRi
ATP
ADP
GAP
Pi
Grb2
Sos
Grb2SOS
(EGF–EGFR*)2–GAP
Ras–GDP
Ras–GTP
Ras–GTP*
Raf*
P'ase1
P'ase2
P'ase3
MEK–P
MEK–PP
ERK–P
v1
v2
v3
v–6
v6
v7
v8
v10
v11
v12
v13
v14
v16, 63
v17, 64
v18, 65
v19, 66
v20, 67
v21, 68
v22, 69
v23, 70
v24, 71
v25, 72
v26, 73
v27, 74
v28, 75
v29, 76
v30, 77
v31, 78
v32, 79
v33
v34, 80
v35
v36
v37, 81
v38
v39, 82
v40
v41, 83
v42, 84
v43, 85
v44, 86
v45, 87
v46, 88
v47, 89
v48, 90
v49, 91
v50, 92
v51, 93
v52, 94
v53, 95
v54, 96
v55, 97
v56, 98
v57, 99
v58, 100
v59, 101
v60
v61
v62