Software Development for Simulating and Engineering Gene Circuits

by

Kanishk Asthana

Department of Biomedical Engineering
Duke University

Date: _____

Approved:

_____
Lingchong You, Supervisor

_____
Alexander Hartemink

_____
Ashutosh Chilkoti

Thesis submitted in partial fulfillment of
the requirements for the degree of Master of Science in the Department of
Biomedical Engineering in the Graduate School
of Duke University

2014

ABSTRACT

Software Development for Simulating and Engineering Gene Circuits

by

Kanishk Asthana

Department of Biomedical Engineering
Duke University

Date:_____

Approved:

_____
Lingchong You, Supervisor

_____
Alexander Hartemink

_____
Ashutosh Chilkoti

An abstract of a thesis submitted in partial
fulfillment of the requirements for the degree
of Master of Science in the Department of
Biomedical Engineering in the Graduate School of
Duke University

2014

Abstract

Mathematical modeling has become an increasingly important aspect of biological research. Computer simulations help to improve our understanding of complex systems by testing the validity of proposed mechanisms and generating experimentally testable hypotheses. However, significant overhead is generated by the creation, debugging, and perturbation of these computational models and their parameters, especially for researchers who are unfamiliar with programming or numerical methods. Dynetica 2.0 is a user-friendly dynamic network simulator designed to expedite this process. Models are created and visualized in an easy-to-use graphical interface, which displays all of the species and reactions involved in a graph layout. System inputs and outputs, indicators, and intermediate expressions may be incorporated into the model via the versatile "expression variable" entity. Models can also be modular, allowing for the quick construction of complex systems from simpler components. Dynetica 2.0 supports a number of deterministic and stochastic algorithms for performing time-course simulations. Additionally, Dynetica 2.0 provides built-in tools for performing sensitivity or dose response analysis for a number of different metrics. Its parameter searching tools can optimize specific objectives of the time course or dose response of the system. Systems can be translated from Dynetica 2.0 into MATLAB code or the SBML format for further analysis or publication. Finally, since it is written in Java, Dynetica 2.0 is platform independent, allowing for easy sharing and collaboration between researchers.

Dedication

I dedicate this thesis to my parents, Prabhakar and Vineeta.

Contents

List of Tables

List of Figures

Acknowledgements

I would like to thank:

Dr. Lingchong You, for his valuable feedback and constant support.

Derek Eidum, for his valuable feedback and constructive criticism.

Samir Unni, for his feedback and support.

Dr. Igor Shats for proving valuable feedback with the early test versions.

Anna Lee for her feedback regarding bugs and interface improvements.

My parents for providing constant emotional and financial support.

1. Introduction

1.1 A Brief Overview of Computational Modeling

Over the past several decades, mathematical modeling has become an important tool in biological research. Due to the rapid expansion of biological knowledge, kinetic modeling has become a realistic goal, particularly for experimentally well-characterized systems.

Kinetic models have been essential for the progress of the fields of systems and synthetic biology where scientists have successfully applied these models to explain many biological phenomena. For example, scientists recently constructed a whole cell kinetic model of the bacterium *Mycoplasma genitalium* which was used to predict phenotypic features of the bacterium (Karr et al., 2012); kinetic modelling and synthetic circuits were recently used to successfully examine the role of stochastic pulses in regulating the stress response in bacteria (Locke et al., 2011), quantitatively understand quorum sensing in bacteria (Pai et al., 2012) and create genetic oscillators (Danino et al., 2010).

A kinetic model essentially represents a mathematical integration of existing data and mechanisms on a particular system, and may be useful in a number of ways. A kinetic model can be used to test the consistency in the experimental data or mechanisms (Ferrezuelo et al., 2012), provide mechanistic explanations for counter-intuitive observations (Tan et al., 2012), to facilitate the formulation of experimentally testable hypotheses (Yao et al., 2011) or to provide insight into emergent properties, such as

robustness(Wong et al., 2012, Wang et al., 2010, Pai et al., 2012), which may be otherwise difficult to grasp intuitively.

Kinetic models need to be simulated using a kind of algorithm: the kind of algorithms can be used varies. Because this process is computationally intensive simulations of the model using various algorithms is usually done using a modeling software.

1.2Need for GUI Software

As seen in Table 1, majority of the latest annotated kinetic models submitted to the Biomodels Database (Li, Donizelli et al. 2010) use either COPASI, MATLAB or both softwares for modelling or simulation needs. Although these softwares are extremely powerful for model creation and simulation, they have a high learning threshold. This high learning threshold makes it hard for beginning modelers who have little or no programming experience to start modelling synthetic gene regulatory networks. To alleviate this problem and facilitate modelling many GUI softwares have been created such as OpenAlea (Pradal, Dufour-Kowalski et al. 2008), CellDesigner (Funahashi, Morohashi et al. 2003), VCell (Moraru et al., 2008) and E-Cell (Takahashi et al., 2003). However these modeling softwares have several drawbacks, for example: OpenAlea does not cater to general modelling needs and is specialised for solving plant modelling problems; CellDesigner allows GUI modelling of Biochemical networks, however it requires an external mathematical package (COPASI) for simulating networks or fitting parameters; Both E-Cell and VCell allow simulation, GUI modelling and provide very

high functionality, however they have a high learning threshold making them hard to intuitively use by beginners. Moreover, there are proprietary tools available such as Monolix(Team, 2012), which allow both modelling and simulation capabilities in a single user friendly package. However since these tools are proprietary they are not openly accessible for facilitating teaching and research. Moreover, many of the other softwares are generally available only as plugins and require some main package or software for either simulating or modelling a network.

Therefore, there is a need for an open, user friendly GUI based software with a low learning threshold for both simulating and modelling: synthetic biochemical and gene regulatory networks. To cater to this need we have developed the software Dynetica 2.0.

To provide added functionality while maintaining a simple GUI interface the following additions were contributed:

1. Modular System implementation, improvement of custom GUI, and import, export and merger facilities for modules was implemented by me.
2. SBML input and export was implemented by Samir Unni.
3. MATLAB export, Parameter Search, and Sensitivity Analysis functionality was implemented by Derek Eidum.
4. Rest of the improvements like inclusion of Expression Variables, and improving figure windows was implemented by Dr. Lingchong You.

This manuscript will cover all the features implemented for Dynetica 2.0. Detailed explanation of Modular System implementation and other improvements will be

in a separate section to highlight the contributions made by me, and will remain the main part of the thesis.

To further allow greater adoption and usage of Dynetica 2.0 by the synthetic biology community. The source code for Dynetica 2.0 has been released at https://github.com/youlab/dynetica under a GPL Version 2 license.

Table 1: Description of Modelling Softwares for Latest Biomodel Database Entries. Source: Biomodels Database (Li et al., 2010)

Publication	Modelling Software	GUI Based Modelling Software	Simulation Software	Type of Models
(Kallenberger et al., 2014)	PottersWheel (Maiwald and Timmer, 2008)	Yes	MATLAB (Mathworks Inc.)	ODE Model
(Muraro et al., 2014)	OpenAlea(Pradal et al., 2008)	Yes	Systems Biology ToolBox for MATLAB(Schmidt and Jirstrand, 2006)	ODE Model
(Ribba et al., 2012)	Monolix (Team, 2012)	Yes	Monolix (Team, 2012)	ODE Model
(Smallbone and Corfe, 2014)	COPASI (Hoops et al., 2006)	No	COPASI (Hoops et al., 2006)	ODE Model
(Kerkhoven et al., 2013)	PySCeS (Olivier et al., 2005)	No	COPASI (Hoops et al., 2006)	ODE Model
(Benson et al., 2014)	Unknown	NA	Unknown	ODE Model
(Barrack et al., 2014)	Unknown	NA	Unknown	ODE Model
(Gardner et al., 2000)	XPP (http://www.math.pitt.edu/~bard/xpp/xpp.html)	No	MATLAB(Mathworks Inc.)	ODE Model
(van Eunen	JWS Online Cellular	Yes	Mathematica(Wolfra	ODE

et al., 2013)	Systems Modeling (jjj.bio.vu.nl)		m Research)	Model
(Proctor et al., 2013b)	COPASI(Hoops et al., 2006)	No	COPASI(Hoops et al., 2006)	ODE Model
(Messiha et al., 2014)	COPASI(Hoops et al., 2006)	No	COPASI(Hoops et al., 2006)	ODE Model
(Begitt et al., 2014)	Unknown	NA	Unknown	ODE Model
(Vizan et al., 2013)	XPP (http://www.math.pitt.edu/~bard/xpp/xpp.html)	No	COPASI (Hoops et al., 2006)	ODE Model
(Mitchell and Mendes, 2013)	CellDesigner (Funahashi et al., 2003)	Yes	COPASI (Hoops et al., 2006)	ODE Model
(Stanford et al., 2013)	Systems Biology Table (http://www.sbtab.net/)	No	MATLAB(Mathworks Inc) and COPASI(Hoops et al., 2006)	ODE Model
(Sen et al., 2013)	Unknown	NA	Unknown	ODE Model
(Roblitz et al., 2013)	BioPARKIN (Dierkes et al., 2012)	No	MATLAB (Mathworks Inc)	ODE Model
(Schittler et al., 2010)	MATCONT (Dhooge et al., 2003)	No	MATLAB (Mathworks Inc)	ODE Model
(Pathak et al., 2013)	SBMLsqueezer(CellDesigner Plugin) (Drager et al., 2008)	Yes	COPASI(Hoops et al., 2006) and CellDesigner (Funahashi et al., 2003)	ODE Model
(Demin et al., 2013)	DBSolve Optimum (Gizzatkulov et al., 2010)	Yes	DBSolve Optimum(Gizzatkulov et al., 2010)	ODE Model
(Sharp et al., 2013)	COPASI (Hoops et al., 2006)	No	COPASI (Hoops et al., 2006)	ODE Model
(Proctor et al., 2013a)	COPASI (Hoops et al., 2006)	No	COPASI and R	ODE Model

1.3An Overview of Dynetica

Models in Dynetica 2.0 are represented by the substances and reactions that make up the system, as well as the parameters and expressions which govern them. Models are drawn in a graph layout and can be compartmentalized into individual modules. These modules can be copied, saved, and imported independently, thus facilitating the construction of large systems from basic subcomponents.

Simulations may be run using either of the following algorithms: Runge-Kutta (Fourth Order, Fixed Time Step), Runge-Kutta-Fehlberg (Fourth Order, Variable Time Step), Doob-Gillespie algorithms (both First Reaction and Direct Method implementations), and the Euler-Maruyama method for stochastic differential equations.

Additionally, Dynetica 2.0 provides a number of useful tools for analyzing the behavior of models. Sensitivity analysis allows the user to examine how the time courses or characteristic metrics of a system change with perturbations in the system's parameters or initial conditions. Repeated stochastic simulations allow the user visualize the distribution of a system's substance values across many trials. Parameter searching perturbs the system's parameter values in order to match the system's output to some user-specified behaviors.

Finally, Dynetica 2.0 supports the import and export of models in SBML format, and provides tools for exporting models to either deterministic or stochastic MATLAB scripts, allowing the user to perform any analyses not covered by Dynetica.

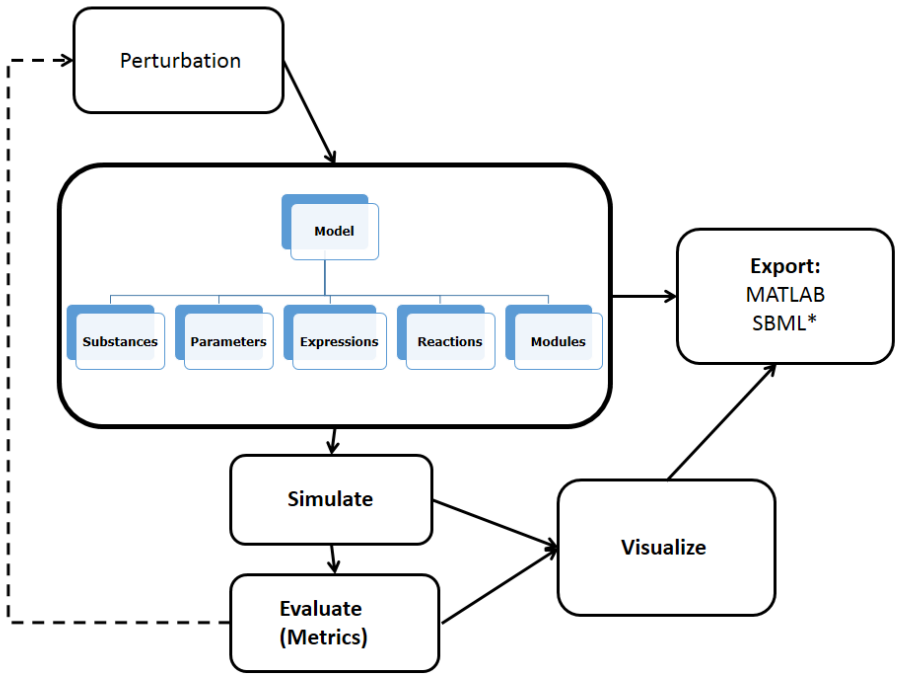

Figure 1. The design paradigm of Dynetica 2.0. The user specifies the module components: substances, reactions, parameters, and expression variables.

The module representation is independent of algorithm used to simulate it. Additionally, Dynetica 2.0 provides analysis tools which measure the behavior of the system in the form of metrics, which are calculated independent of how the simulation is performed. Depending on the type of analysis performed, these metrics may lead to or be measured as the result of some perturbation in the system parameters or initial conditions. Additionally, the model representation may be imported from or exported to SBML format, and can generate simple MATLAB simulation scripts.

The areas of improvement in Dynetica 2.0 can be summarized in four segments: interface enhancements, sensitivity analysis support, genetic parameter searching, and import/export functions.

2. Overview of New Features Implemented in Dynetica

2.1 Interface

The three primary interface enhancements in Dynetica 2.0 are expression variable support, modular system construction, and import/export support.

2.1.1 Expression Variables

An expression variable is an entity in Dynetica 2.0 that can can be represented by any mathematical or boolean function of substance concentrations, parameters, and time. Expression variables can be used in reaction kinetics equations just like substance concentrations or parameter values. Because of this flexibility, they can be used as time-dependent system inputs, intermediate expressions, and output variables or indicators.

2.1.2 Modules

When building large and/or complex systems in Dynetica, it often becomes difficult to keep track of the various components and how they interact. One way to mitigate this problem is by taking advantage of the modular nature of many of these large networks. Although there may be many components, they do not interact with all other components. Rather they tend to form small subnetworks, which then interact with one another via a handful of interconnects.

Dynetica 2.0 allows users to take advantage of this phenomenon by creating such subnetworks, called “modules.” Fig. 2, displaying a model for programmed altruistic death, exemplifies how Dynetica modules can be harnessed to enhance conceptual simplicity. The substances comprising the cheater and cooperator modules, on the bottom

right and bottom left corners of the network graph, respectively, have each been isolated into their own module.

2.1.3 Import/Export support

A major improvement in Dynetica 2.0 over the previous version is its ability to import and export models in SBML(Systems Biology Markup language). The earlier version of Dynetica only allowed models to be stored in the native DYN format used by Dynetica. This made it hard for models created in Dynetica to be further extended using more advanced modelling softwares, as they would have to be recreated in the new modelling platform. This new feature allows models created in Dynetica to be imported by other software packages that recognize the SBML format. Another useful export feature in the newest version is the ability to generate MATLAB code for a model created in Dynetica.

In addition, Dynetica 2.0 allows the user to: merge existing models with each other or add one model to another as a module. These features give the user incredible power when creating new models because it allows existing models to be reused or extended. Moreover, these new features allow the user to quickly create and test synthetic circuits by combining circuit elements from a library. Another advantage of using this new feature the initial MATLAB or SBML code for a model can be generated much more quickly and robustly, as the user won't have to rewrite the entire code when the model is modified or extended. Moreover, parameter values and constants are also preserved during merger.

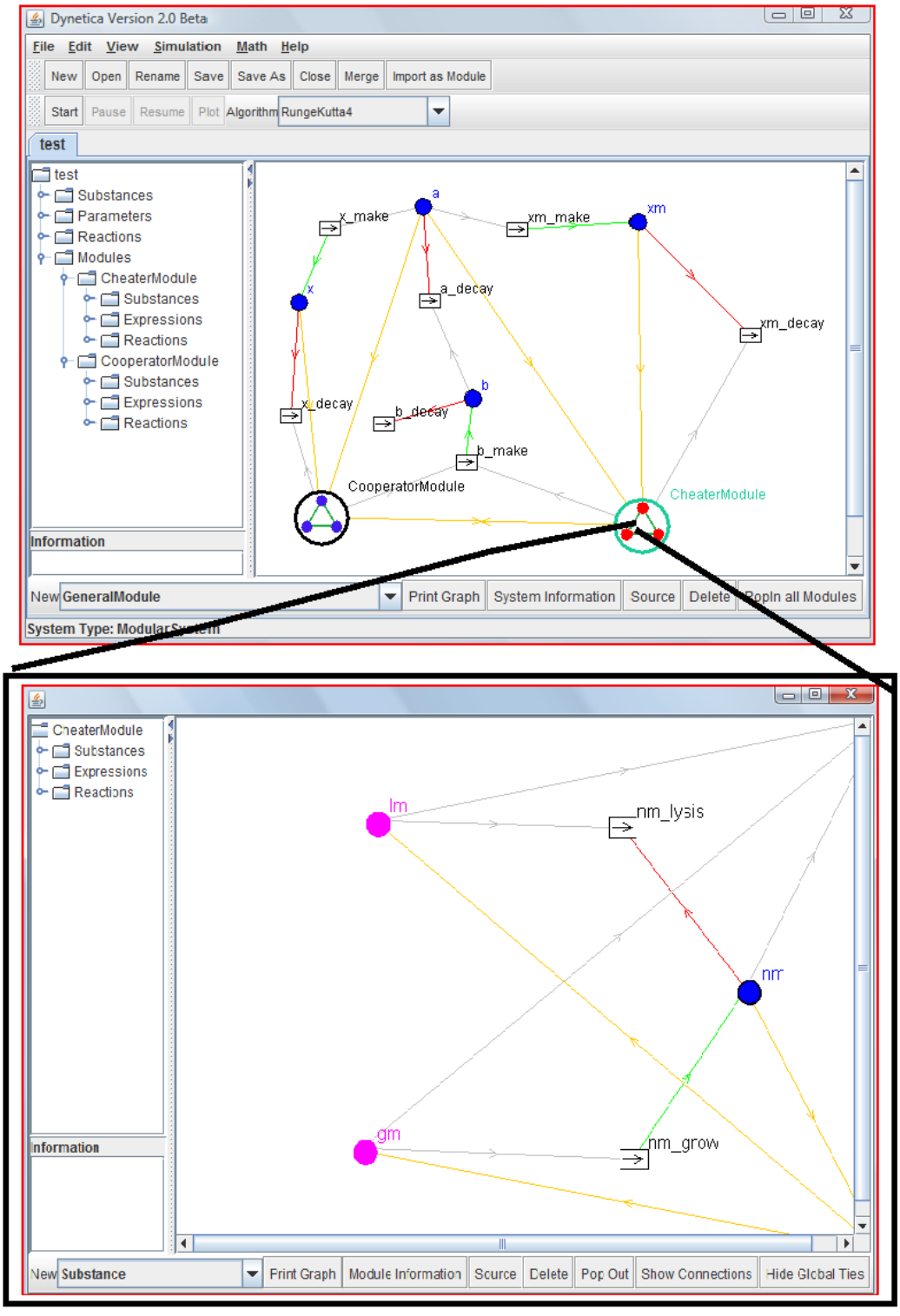
Dynetica Version 2.0 Beta
File Edit View Simulation Math Help
New Open Rename Save Save As Close Merge Import as Module
Start Pause Resume Plot Algorithm RungeKutta4
test
test
Substances
Parameters
Reactions
Modules
CheaterModule
Substances
Expressions
Reactions
CooperatorModule
Substances
Expressions
Reactions
Information
a
x_make
xm_make
xm
x
a_decay
xm_decay
b
x_decay
b_decay
b_make
CooperatorModule
CheaterModule
New GeneralModule
Print Graph
System Information
Source
Delete
Popln all Modules
System Type: Modular System
CheaterModule
Substances
Expressions
Reactions
Information
lm
nm_lysis
nm
gm
nm_grow
New Substance
Print Graph
Module Information
Source
Delete
Pop Out
Show Connections
Hide Global Ties

Figure 2. The module feature of Dynetica's UI. The upper window represents the entire cheater-cooperator system, while the bottom window shows just the cheater module.

2.2 Sensitivity Analysis Support

2.2.1 Repeated Stochastic Simulations

Analyze transient effects under stochastic conditions and generate (what could be a signature) distribution of

values.

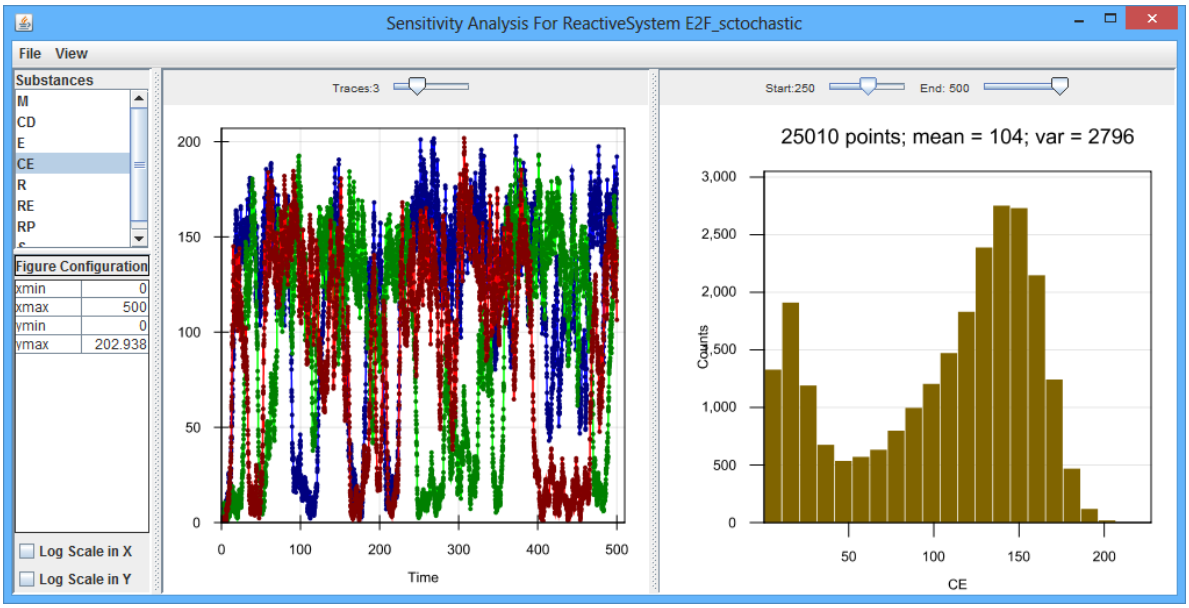

Figure 3. A stochastic simulation of the RbE2F system. The histogram on the right shows the distribution of [CE] values after 500 ms.

2.2.2 Metrics

Metrics are the key units used to analyze the behavior of a system in Dynetica 2.0. A metric is a value which describes some characteristic of a single time course. These are used in both sensitivity analysis and parameter searching.

- The metric implemented in Dynetica 2.0 include:
- Final, min, max value
- Range
- Maximum rate
- Time to reach a user-specified fraction of steady state value
- Area under curve
- Peak frequency in the power spectrum
- Correlation coefficient when compared to another substance or expression variable in the same system

2.2.3 Sensitivity Analysis

Dynetica 2.0 provides the user the ability to explore how the behavior of the system varies with changes in either parameter values or initial substance concentrations through its sensitivity analysis tool. The user selects a parameter or substance from the model to act as the independent variable, and defines any number of metrics to evaluate. The simulation is run multiple times over a range of values for the independent variable, and the value of each metric is plotted as a function of the parameter or initial concentration value. Alternatively, the time courses for each trial can be plotted on the

same axes in order to visualize the change in behavior as the independent variable is perturbed.

2.2.3.1 Deterministic

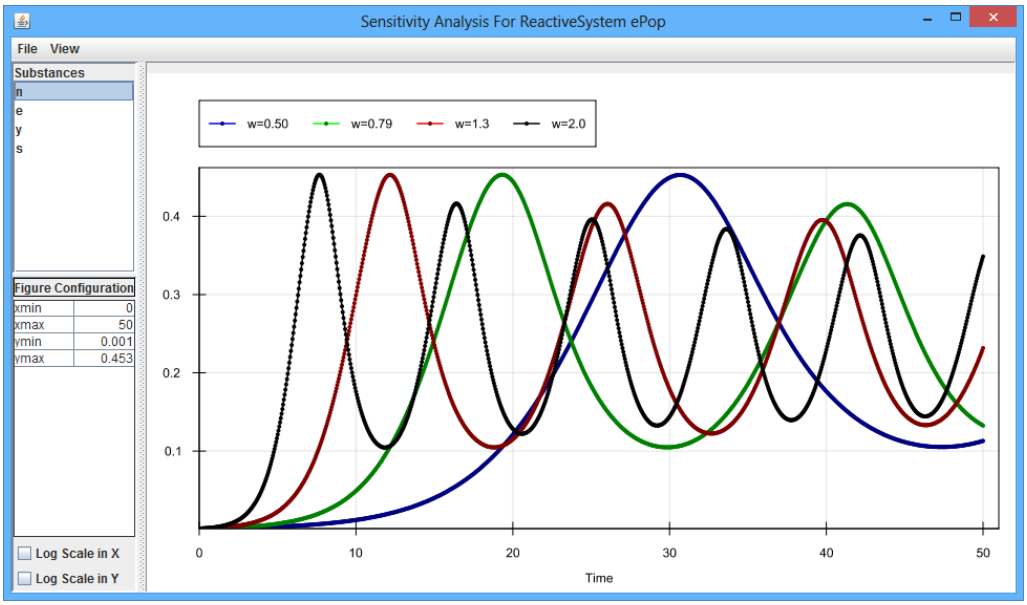

Fig. 4 shows an example of a sensitivity analysis conducted using deterministic simulations. The parameter 'w' is varied from 0.5 to 2.0, resulting in a change in the length of the transient response and the oscillation frequency at steady state for the substance 'n'. Note the use of color to facilitate the differentiation of the different time courses.

Frequency is not the only metric that Dynetica's sensitivity analysis tools support. The other difference between the timecourses in fig. 4 - the time to steady state - can also be measured, using the "time to steady state" metric.

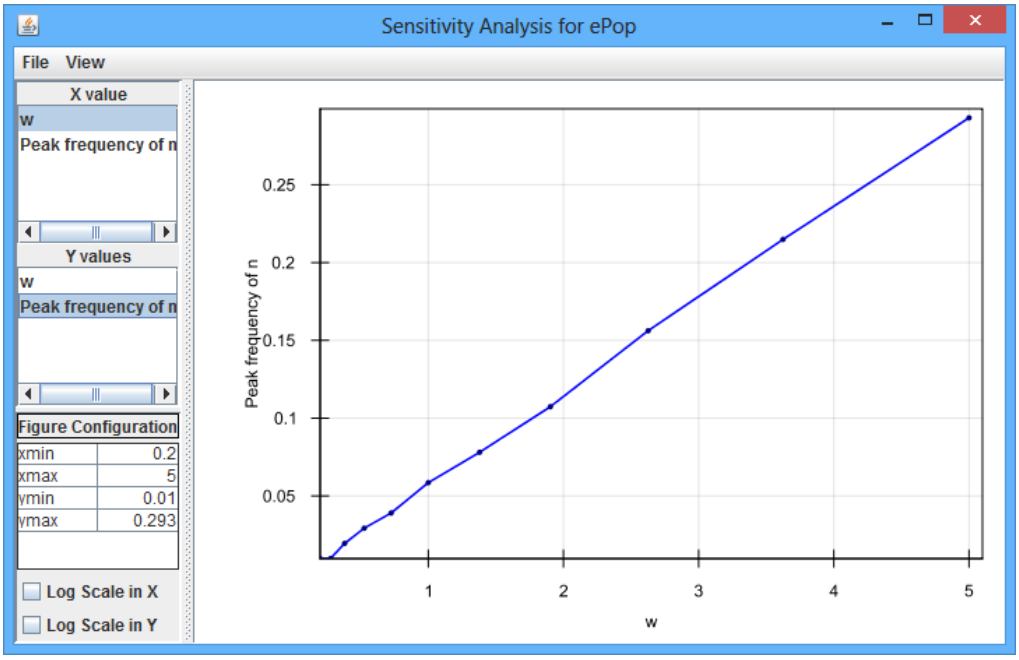

Fig. 5 contains an example of a more sophisticated sensitivity analysis. Rather than simply showing the timecourses for the various substances comprising the system under simulation, the frequencies of oscillation are shown. This is particularly useful for a system such as that in fig. 4, where one of the few differences between the timecourses is their oscillation frequency.

2.2.3.2 Stochastic

The above tools may be used with stochastic algorithms as well. However, this may prove less meaningful because the change in system behavior may be due to random noise rather than the deviations in the independent variable. To examine how stochastic systems respond to changes in parameters or initial concentrations, Dynetica 2.0 provides a tool for performing repeated stochastic simulations for each value of the independent

variable. Both the timecourses and the distribution of values can be viewed for each point, as shown in the figure below.

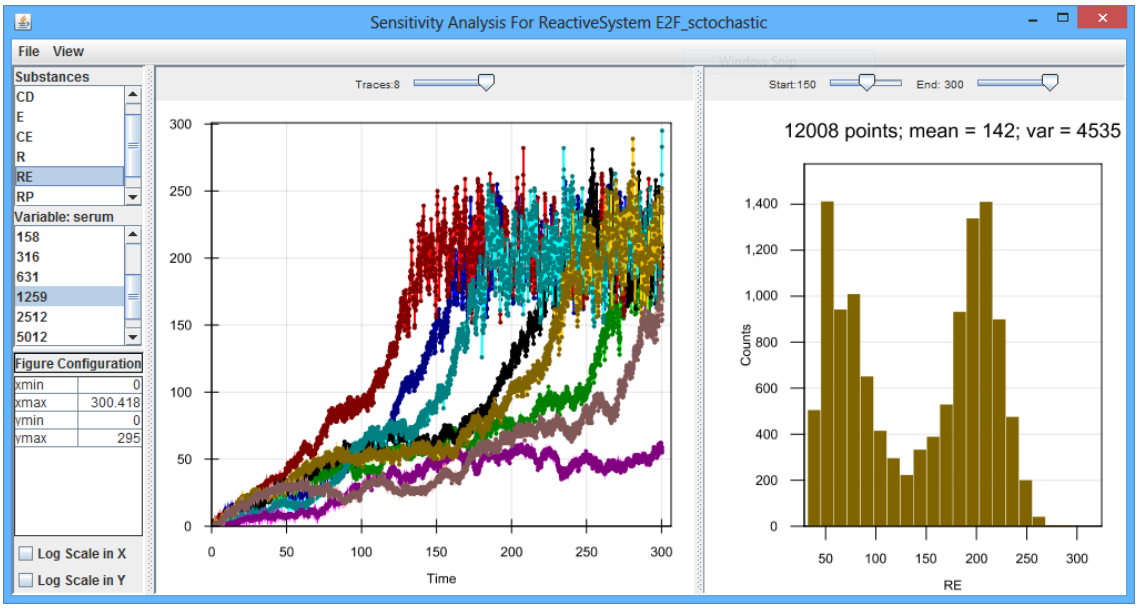

Figure 6. The above figure demonstrates the results of a stochastic sensitivity analysis on an RbE2F system. The options panel on the far left allows the user to select a substance or expression variable to plot, which value of the independent variable to look at, and the figure window options. The center panel shows the resulting timecourses. The user can select how many are displayed. The right panel shows the distribution of values across the repeated simulations, as well as the mean and variance. The slider bars at the top allow the user to specify the time window of interest.

2.3 Parameter search

Dynetica 2.0 provides tools for searching the parameter space of a system in order to optimize user defined objectives. This is useful for when the user knows the desired behavior of a system and wants to find a set of parameters for which the system closely matches this behavior. The search may be performed over all or any subset of the system's parameters.

The user defines the desired behavior of the system by specifying a number of target objectives, which the algorithm will attempt to match. The user selects a metric and specifies a goal for that metric's value, which can be to minimize, maximize, or target its value to a specific number. Target objectives can also be given different weights, with higher weighted objectives having a greater influence on the search than lower ones.

The search is done using a genetic algorithm. The population is made up of individual parameter vectors. In order to calculate the evolutionary fitness of each individual parameter vector, its parameters are applied to the system and the simulation is run. The vector then receives a fitness score between 0 and 100 based on how well the simulation matches the user-defined metrics. At each generation, high scoring parameter vectors survive and reproduce, while low scoring vectors die off, maintaining a constant population size. The implementation details for this algorithm can be found in the supplementary materials.

To run a parameter search, the user must specify which parameters will be perturbed. All others will remain fixed at their user defined values. The user also must specify the target objectives for the algorithm to optimize, each of which consists of a metric, a goal for that metric, and a relative weight. The user must also specify the simulation duration, population size, and maximum number of generations the algorithm will run. The user can also define a stopping threshold, which is a fraction specifying what score is needed to cause the search to complete. For example, the default value of 0.05 indicates that a fitness score of 95 or higher (out of 100) is sufficient to stop the search. Setting this value to 0 means the search will not stop unless the target objectives are matched exactly or the maximum number of generations is reached.

The user may also choose how the initial population of parameter vectors is created. By default, each parameter vector is created by random normal perturbations around the user's defined value. This allows the algorithm to use the user-given values as an initial guess. However, the user may also chose to create the initial population by selecting uniformly random numbers between the parameter's specified minimum and maximum values. This may produce better results by exploring new areas of the parameter space. Note, however, that if this option is selected, it is important to specify realistic constraints for the parameter's minimum and maximum values.

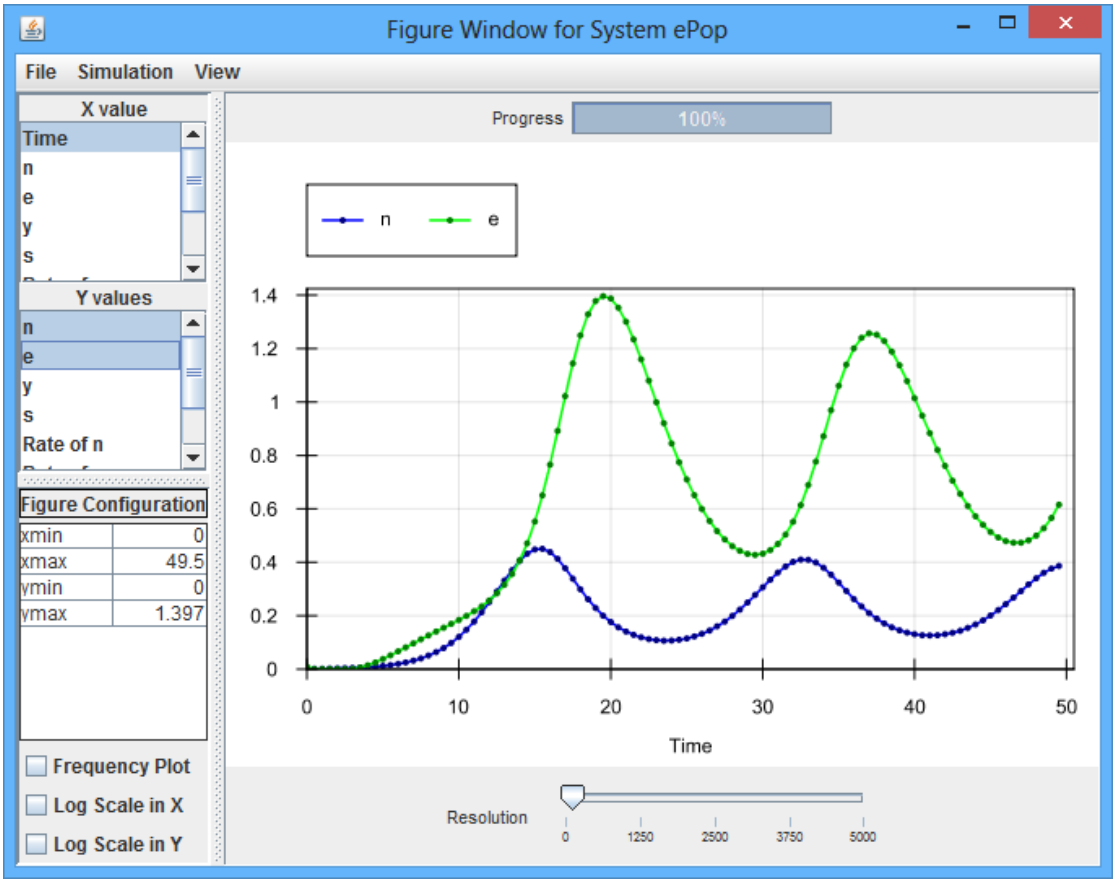
Figure Window for System ePop
File
Simulation
View
X value
Time
n
e
y
s
Y values
n
e
y
s
Rate of n
Figure Configuration
xmin 0
xmax 49.5
ymin 0
ymax 1.397
Frequency Plot
Log Scale in X
Log Scale in Y
Progress
100%
n
e
1.4
1.2
1
0.8
0.6
0.4
0.2
0
0
10
20
30
40
50
Time
Resolution
0
1250
2500
3750
5000

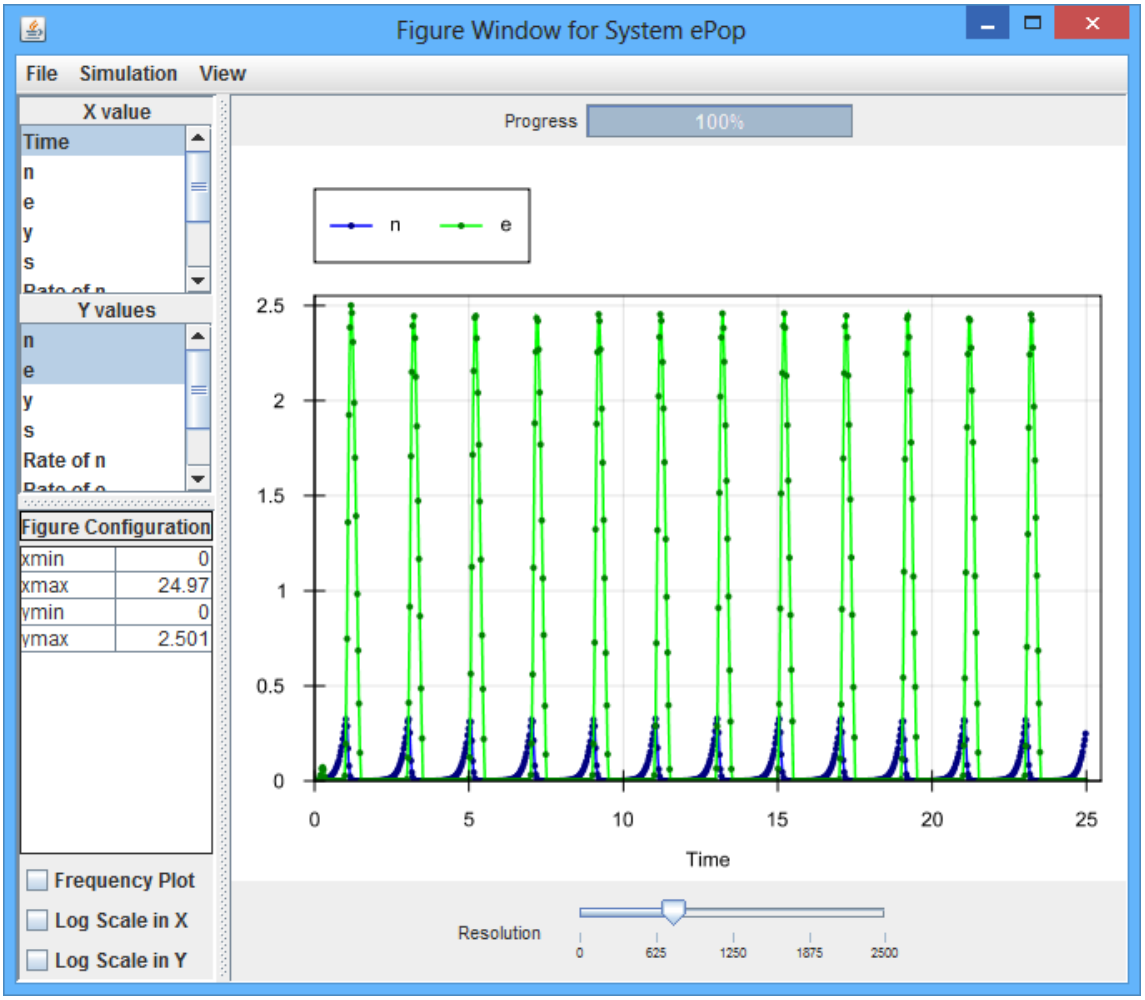

Figure 7. Shows the resulting time course plots before (above) and after(below) running a parameter search.

Dynetica 2.0 also features a tool for searching the parameter space of a system when the desired behavior is not a feature of a single simulation, but of the dose response or sensitivity analysis curves. In addition to the inputs above, the user must specify a parameter or initial substance concentration to use as the independent variable, as well as a range of values to scan over. The same genetic search algorithm is used; however, rather than using a single metric value, the fitness score is calculated as a function of the shape of curves when the specified metrics are plotted as against the independent

variable.

For each metric selected, there are five target functions which can be used. The range of the sensitivity plot can be minimized or maximized, resulting in the metric being minimally or maximally responsive to perturbations in the independent variable, respectively. The user may also select to maximize or minimize the difference between the first two local extrema, or the first local extrema and the steady state value. Lastly, the user may choose for the response curve to be maximally linear, in which case a linear regression will be case the score will be based on the coefficient of determination.

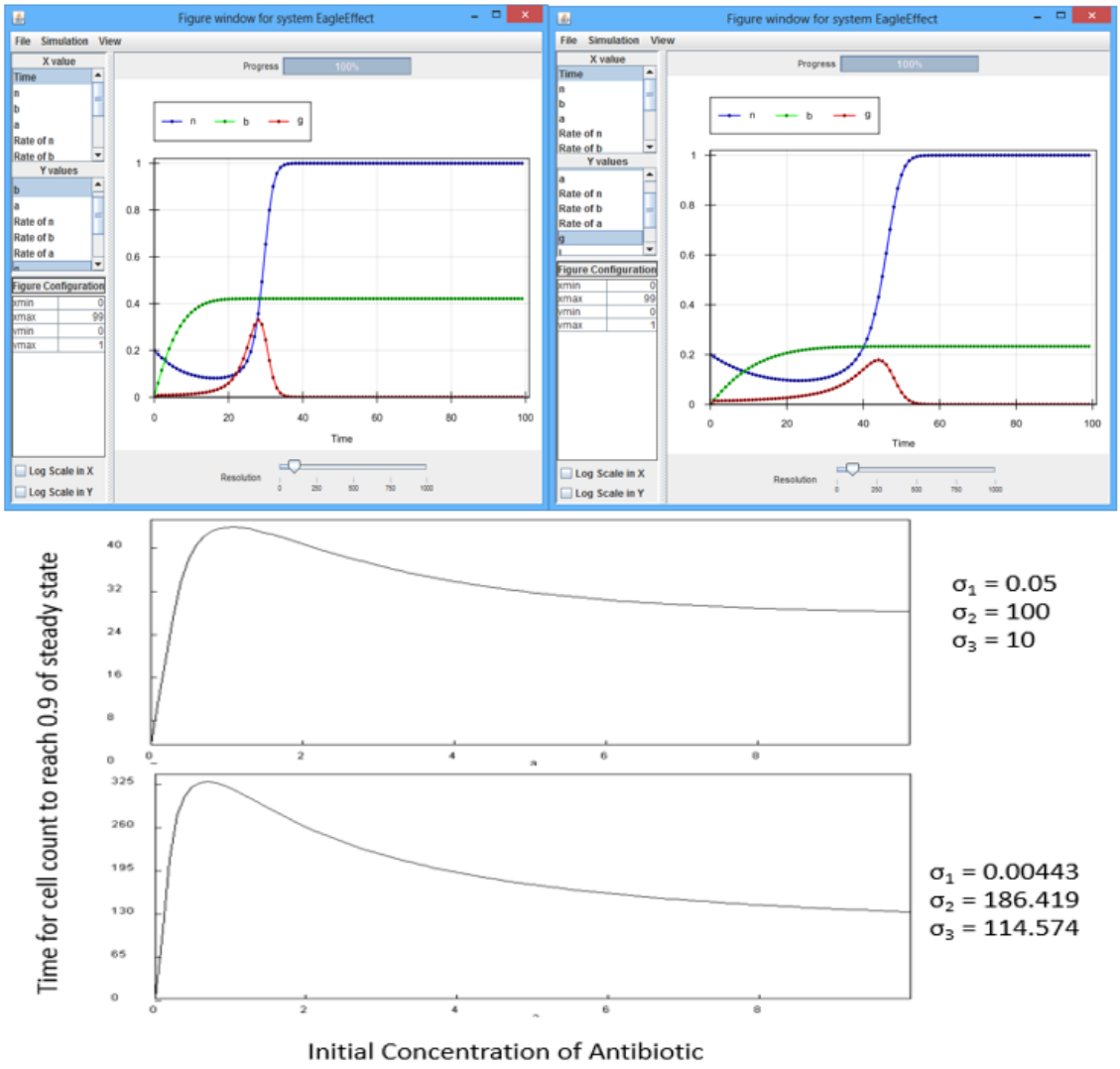

Figure 8. Shows the resulting time course plots before and after running a parameter search.

3. Modular Systems in Dynetica

3.1 Introduction

Modular Systems are an extension of Dynetica 2.0 that allow the user to create and maintain models that are modular in organization. This ability add and remove modules gives the user incredible power when creating models, as it allows previous models to be reused and extended. Moreover, many gene circuits are modular in organization and construction. Using Dynetica 2.0 the user can quickly create models which can be used to study the properties of a new gene circuit constructed using well known smaller circuits as modules. Moreover, by using this new feature the initial MATLAB or SBML code for a model can be generated much more quickly and robustly, as the user won't have to rewrite the entire code when the model is modified or extended.

3.2 Design Paradigm

Like the previous version of Dynetica, a model can have Reactions, Substances, Parameters and Expressions (RSPEs) at the highest level of organization. In this version in addition to RSPEs, the highest level of organization (henceforth referred to as the *Super system*) also contains Modules. Like the Super system each module can also have its own RSPEs. These RSPEs remain globally accessible, i.e. individual RSPEs are visible to all the other RSPEs in the system, independent of whether they are present in the super system or in a module; for example a reaction can have substances inside a module as reactants and substances outside the module as products: the system behaves

as if the modules don't exist. Moreover, the user can build models of arbitrary complexity by importing or exporting modules to and from Dynetica 2.0 (at the moment the ability to import modules into a system is restricted to the system type "Modular System"). In addition, an existing model can be extended by merging its contents with an empty Modular System.

To aid in the creation and maintenance of modules, Dynetica 2.0 has introduced many new and intuitive features.

3.3 Creating and Editing Modules:

Modules can be created from the main Dynetica 2.0 window by clicking the new Entity dropdown menu at the bottom of the screen and clicking on "GeneralModule". This should open a dialog box asking the user to name the new Module.

Once created the module interface looks similar to the interface of the main system. New entities can be created within the module by using the entity creation dropdown menu at the bottom of the window. Moreover, the structure of the module can be visualized using the module tree on the left. In addition to providing visual representation of the module structure, the module tree can be used to access each element in the module.

RSPEs can also be easily added to the Super system from a module and added to a module from the super system or from another module, using intuitive gestures.

The drag and drop gesture common in windows file systems can be used to drag and drop RSPEs into a module using the main system window. Moreover, by clicking

and dragging an RSPE within a module to the bottom of the module window transfers the RSPE to the main system. Drag and drop gesture can also be used to transfer entities from Poped-out modules to other modules that have not been Poped-out (explained below).

3.4 Module Visualization:

A module can be visualized easily by double clicking the module in the main system. Moreover, by right clicking a module and then clicking on Pop Out, the contents of a module are immediately displayed in the main system window. These contents can be hidden again by clicking on "Pop In all Modules" at the bottom of the main window. Moreover, modules can be Poped In and Out from their respective windows.

The module window also provides a few intuitive features to help the user understand and visualize the model structure. These features include the ability to hide and display connections to the super system and the ability to visualize the immediate connection partners for RSPE's in the module.

In addition Dynetica 2.0 displays the kinetics or value of an RSPE when you mouse over it.

3.5 Importing Exporting and Merging modules:

To create a module that the user wants to store as an independent system a new system type of "Reactive System" can be created from the File menu. This system can then be manipulated in a manner similar to a "Modular System" except no new modules can be created in this system.

To import the above system or any older systems as a module, the saved .dyn file for that system can simply be either merged with or imported as a module when “Merge” or “Import as Module” buttons are clicked at the top of the main window. Therefore by merging or importing a system a module, models of arbitrary complexity can be easily made.

3.6 Manipulating the System Tree:

The main system tree displays the complete model and its structure. It contains the nodes Parameters, Reactions, Substances, Expressions and Modules. Expanding each parent node allows the user to view and access the elements inside that node.

By clicking the Modules node in the system tree and expanding a particular module node shows the RSPEs within that module. Moreover right clicking each node shows several options, including the ability to Pop In and Pop Out modules. The structure of this tree changes dynamically as RSPE’s are moved from and to modules.

Conclusion

Dynetica 2.0 has added support for much more sophisticated analyses of stochastic simulations. For a variety of metrics, not only can they be repeated with a fixed parameter set to generate outcome distributions, they can also be varied to accomplish sensitivity analyses. For small systems, stochastic analyses can provide realistic insight into laboratory phenomena. For those systems that are too large for stochastic algorithms to be practical, sensitivity analyses can also be conducted using deterministic algorithms. The aforementioned metrics can also be used for the opposite of a sensitivity analysis. A genetic search algorithm is used to determine a set of parameter values that would produce the desired value of a specified metric. Significant interface and experience improvements have been made to Dynetica, in the form of the expression variable/module features, as well as import/export functionality. These enhancements ease the process of constructing and evaluating models in Dynetica 2.0, as well as the collaboration process when working with researchers who utilize other pieces of dynamic network simulation software.

Some of the limitations of Dynetica 2.0 are its inability to model spatial gradients and multiple compartments such as those found in Eukaryotic cells and organ systems. Despite these limitations we believe that Dynetica 2.0 serves its purpose by providing a robust and easy to use modelling and simulation environment for non-programmers. In addition many basic modelling tasks do not require the added complexity of multiple comparments, computationally intensive simulations, and spatial modelling.

Appendix A: Important Web links

Dynetica Main Page:

http://www.genome.duke.edu/labs/YouLab/software/dynetica/index.php

Dynetica Github Page:

https://github.com/youlab/dynetica

Dynetica Youtube tutorials:

https://www.youtube.com/channel/UCfn8saYPtPKkeIw786bHpmA

References

BARRACK, D. S., THUL, R. & OWEN, M. R. 2014. Modelling the coupling between intracellular calcium release and the cell cycle during cortical brain development. *J Theor Biol,* 347**,** 17-32.

BEGITT, A., DROESCHER, M., MEYER, T., SCHMID, C. D., BAKER, M., ANTUNES, F., KNOBELOCH, K.-P., OWEN, M. R., NAUMANN, R., DECKER, T. & VINKEMEIER, U. 2014. STAT1-cooperative DNA binding distinguishes type 1 from type 2 interferon signaling. *Nat Immunol,* 15**,** 168-176.

BENSON, N., METELKIN, E., DEMIN, O., LI, G. L., NICHOLS, D. & VAN DER GRAAF, P. H. 2014. A systems pharmacology perspective on the clinical development of Fatty Acid amide hydrolase inhibitors for pain. *CPT Pharmacometrics Syst Pharmacol,* 3**,** e91.

DANINO, T., MONDRAGON-PALOMINO, O., TSIMRING, L. & HASTY, J. 2010. A synchronized quorum of genetic clocks. *Nature,* 463**,** 326-330.

DEMIN, O., KARELINA, T., SVETLICHNIY, D., METELKIN, E., SPESHILOV, G., DEMIN, O., FAIRMAN, D., VAN DER GRAAF, P. H. & AGORAM, B. M. 2013. Systems Pharmacology Models Can Be Used to Understand Complex Pharmacokinetic-Pharmacodynamic Behavior: An Example Using 5-Lipoxygenase Inhibitors. *CPT: pharmacomet. syst. pharmacol.,* 2**,** e74.

DHOOGE, A., GOVAERTS, W. & KUZNETSOV, Y. A. 2003. MATCONT: A MATLAB package for numerical bifurcation analysis of ODEs. *ACM Trans. Math. Softw.,* 29**,** 141-164.

DIERKES, T., WADE, M., NOWAK, U. & RÖBLITZ, S. 2012. Bioparkin-biology-related parameter identification in large kinetic networks.

DRAGER, A., HASSIS, N., SUPPER, J., SCHRODER, A. & ZELL, A. 2008. SBMLsqueezer: a CellDesigner plug-in to generate kinetic rate equations for biochemical networks. *BMC Syst Biol,* 2**,** 39.

FERREZUELO, F., COLOMINA, N., PALMISANO, A., GARI, E., GALLEGO, C., CSIKASZ-NAGY, A. & ALDEA, M. 2012. The critical size is set at a single-cell level by growth rate to attain homeostasis and adaptation. *Nat Commun,* 3**,** 1012.

FUNAHASHI, A., MOROHASHI, M., KITANO, H. & TANIMURA, N. 2003. CellDesigner: a process diagram editor for gene-regulatory and biochemical networks. *BIOSILICO,* 1**,** 159-162.

GARDNER, T. S., CANTOR, C. R. & COLLINS, J. J. 2000. Construction of a genetic toggle switch in Escherichia coli. *Nature,* 403**,** 339-42.

GIZZATKULOV, N. M., GORYANIN, I. I., METELKIN, E. A., MOGILEVSKAYA, E. A., PESKOV, K. V. & DEMIN, O. V. 2010. DBSolve Optimum: a software package for kinetic modeling which allows dynamic visualization of simulation results. *BMC systems biology,* 4**,** 109.

HOOPS, S., SAHLE, S., GAUGES, R., LEE, C., PAHLE, J., SIMUS, N., SINGHAL, M., XU, L., MENDES, P. & KUMMER, U. 2006. COPASI--a COmplex PAthway SImulator. *Bioinformatics,* 22**,** 3067-74.

HTTP://WWW.MATH.PITT.EDU/~BARD/XPP/XPP.HTML.

HTTP://WWW.SBTAB.NET/.

JJJ.BIO.VU.NL.

KALLENBERGER, S. M., BEAUDOUIN, J., CLAUS, J., FISCHER, C., SORGER, P. K., LEGEWIE, S. & EILS, R. 2014. Intra- and Interdimeric Caspase-8 Self-Cleavage Controls Strength and Timing of CD95-Induced Apoptosis. *Sci. Signal.,* 7**,** ra23-.

KARR, JONATHAN R., SANGHVI, JAYODITA C., MACKLIN, DEREK N., GUTSCHOW, MIRIAM V., JACOBS, JARED M., BOLIVAL JR, B., ASSAD-GARCIA, N., GLASS, JOHN I. & COVERT, MARKUS W. 2012. A Whole-Cell Computational Model Predicts Phenotype from Genotype. *Cell,* 150**,** 389-401.

KERKHOVEN, E. J., ACHCAR, F., ALIBU, V. P., BURCHMORE, R. J., GILBERT, I. H., TRYBILO, M., DRIESSEN, N. N., GILBERT, D., BREITLING, R., BAKKER, B. M. & BARRETT, M. P. 2013. Handling uncertainty in dynamic models: the pentose phosphate pathway in Trypanosoma brucei. *PLoS Comput Biol,* 9**,** e1003371.

LI, C., DONIZELLI, M., RODRIGUEZ, N., DHARURI, H., ENDLER, L., CHELLIAH, V., LI, L., HE, E., HENRY, A., STEFAN, M. I., SNOEP, J. L., HUCKA, M., LE NOVERE, N. & LAIBE, C. 2010. BioModels Database: An enhanced, curated and annotated resource for published quantitative kinetic models. *BMC Syst Biol,* 4**,** 92.

LOCKE, J. C. W., YOUNG, J. W., FONTES, M., JIMÉNEZ, M. J. H. & ELOWITZ, M. B. 2011. Stochastic Pulse Regulation in Bacterial Stress Response. *Science,* 334**,** 366-369.

MAIWALD, T. & TIMMER, J. 2008. Dynamical modeling and multi-experiment fitting with PottersWheel. *Bioinformatics,* 24**,** 2037-2043.

MESSIHA, H. L., KENT, E., MALYS, N., CARROLL, K. M., MENDES, P. & SMALLBONE, K. 2014. Enzyme characterisation and kinetic modelling of the pentose phosphate pathway in yeast. PeerJ PrePrints.

MITCHELL, S. & MENDES, P. 2013. A computational model of liver iron metabolism. *PLoS Comput Biol,* 9**,** e1003299.

MORARU, I. I., SCHAFF, J. C., SLEPCHENKO, B. M., BLINOV, M. L., MORGAN, F., LAKSHMINARAYANA, A., GAO, F., LI, Y. & LOEW, L. M. 2008. Virtual cell modelling and simulation software environment. *Systems Biology, IET,* 2**,** 352-362.

MURARO, D., MELLOR, N., POUND, M. P., HELP, H., LUCAS, M., CHOPARD, J., BYRNE, H. M., GODIN, C., HODGMAN, T. C., KING, J. R., PRIDMORE, T. P., HELARIUTTA, Y., BENNETT, M. J. & BISHOPP, A. 2014. Integration of hormonal signaling networks and mobile microRNAs is required for vascular patterning in Arabidopsis roots. *Proc Natl Acad Sci U S A,* 111**,** 857-62.

OLIVIER, B. G., ROHWER, J. M. & HOFMEYR, J. H. 2005. Modelling cellular systems with PySCeS. *Bioinformatics,* 21**,** 560-1.

PAI, A., TANOUCHI, Y. & YOU, L. 2012. Optimality and robustness in quorum sensing (QS)-mediated regulation of a costly public good enzyme. *Proc Natl Acad Sci U S A,* 109**,** 19810-5.

PATHAK, R. K., TAJ, G., PANDEY, D., ARORA, S. & KUMAR, A. 2013. Modeling of the MAPK machinery activation in response to various abiotic and biotic stresses in plants by a system biology approach. *Bioinformation,* 9**,** 443-9.

PRADAL, C., DUFOUR-KOWALSKI, S., BOUDON, F., FOURNIER, C. & GODIN, C. 2008. OpenAlea: a visual programming and component-based software platform for plant modelling. *Functional plant biology,* 35**,** 751-760.

PROCTOR, C. J., BOCHE, D., GRAY, D. A. & NICOLL, J. A. R. 2013a. Investigating Interventions in Alzheimer's Disease with Computer Simulation Models. *PLoS ONE,* 8**,** e73631.

PROCTOR, C. J., MACDONALD, C., MILNER, J. M., ROWAN, A. D. & CAWSTON, T. E. 2013b. A computer simulation approach for assessing therapeutic intervention points to prevent cytokine-induced cartilage breakdown. *Arthritis Rheum*.

RIBBA, B., KALOSHI, G., PEYRE, M., RICARD, D., CALVEZ, V., TOD, M., CAJAVEC-BERNARD, B., IDBAIH, A., PSIMARAS, D., DAINESE, L., PALLUD, J., CARTALAT-CAREL, S., DELATTRE, J. Y., HONNORAT, J., GRENIER, E. & DUCRAY, F. 2012. A tumor growth inhibition model for low-grade glioma treated with chemotherapy or radiotherapy. *Clin Cancer Res,* 18**,** 5071-80.

ROBLITZ, S., STOTZEL, C., DEUFLHARD, P., JONES, H. M., AZULAY, D. O., VAN DER GRAAF, P. H. & MARTIN, S. W. 2013. A mathematical model of the human menstrual cycle for the administration of GnRH analogues. *J Theor Biol,* 321**,** 8-27.

SCHITTLER, D., HASENAUER, J., ALLGOWER, F. & WALDHERR, S. 2010. Cell differentiation modeled via a coupled two-switch regulatory network. *Chaos,* 20**,** 045121.

SCHMIDT, H. & JIRSTRAND, M. 2006. Systems Biology Toolbox for MATLAB: a computational platform for research in systems biology. *Bioinformatics,* 22**,** 514-5.

SEN, P., VIAL, H. J. & RADULESCU, O. 2013. Kinetic modelling of phospholipid synthesis in Plasmodium knowlesi unravels crucial steps and relative importance of multiple pathways. *BMC Syst Biol,* 7**,** 123.

SHARP, G. C., MA, H., SAUNDERS, P. T. & NORMAN, J. E. 2013. A computational model of lipopolysaccharide-induced nuclear factor kappa B activation: a key signalling pathway in infection-induced preterm labour. *PloS one,* 8**,** e70180.

SMALLBONE, K. & CORFE, B. M. 2014. A mathematical model of the colon crypt capturing compositional dynamic interactions between cell types. *International Journal of Experimental Pathology,* 95**,** 1-7.

STANFORD, N. J., LUBITZ, T., SMALLBONE, K., KLIPP, E., MENDES, P. & LIEBERMEISTER, W. 2013. Systematic construction of kinetic models from genome-scale metabolic networks. *PLoS One,* 8**,** e79195.

TAKAHASHI, K., ISHIKAWA, N., SADAMOTO, Y., SASAMOTO, H., OHTA, S., SHIOZAWA, A., MIYOSHI, F., NAITO, Y., NAKAYAMA, Y. & TOMITA, M. 2003. E-Cell 2: Multi-platform E-Cell simulation system. *Bioinformatics,* 19**,** 1727-1729.

TAN, C., SMITH, R. P., SRIMANI, J. K., RICCIONE, K. A., PRASADA, S., KUEHN, M. & YOU, L. 2012. The inoculum effect and band-pass bacterial response to periodic antibiotic treatment. *Mol Syst Biol,* 8**,** 617.

TEAM, M. 2012. The Monolix software, Version 4.1. 2. Analysis of mixed effects models. LIXOFT and INRIA.

VAN EUNEN, K., SIMONS, S. M., GERDING, A., BLEEKER, A., DEN BESTEN, G., TOUW, C. M., HOUTEN, S. M., GROEN, B. K., KRAB, K., REIJNGOUD, D. J. & BAKKER, B. M. 2013. Biochemical competition makes fatty-acid beta-oxidation vulnerable to substrate overload. *PLoS Comput Biol,* 9**,** e1003186.

VIZAN, P., MILLER, D. S. J., GORI, I., DAS, D., SCHMIERER, B. & HILL, C. S. 2013. Controlling Long-Term Signaling: Receptor Dynamics Determine Attenuation and Refractory Behavior of the TGF-{beta} Pathway. *Sci. Signal.,* 6**,** ra106-.

WANG, J., LI, C. & WANG, E. 2010. Potential and flux landscapes quantify the stability and robustness of budding yeast cell cycle network. *Proceedings of the National Academy of Sciences,* 107**,** 8195-8200.

WONG, J. V., LI, B. & YOU, L. 2012. Tension and robustness in multitasking cellular networks. *PLoS Comput Biol,* 8**,** e1002491.

YAO, G., TAN, C., WEST, M., NEVINS, J. R. & YOU, L. 2011. Origin of bistability underlying mammalian cell cycle entry. *Molecular Systems Biology,* 7**,** n/a-n/a.