

AFRL-IF-RS-TR-2006-303
Final Technical Report
October 2006



USER EVALUATION OF BIOSPICE

UNIVERSITY OF TEXAS MEDICAL SCHOOL AT HOUSTON

Sponsored by
Defense Advanced Research Projects Agency
DARPA Order No. J058/U041

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

STINFO FINAL REPORT

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report was cleared for public release by the Air Force Research Laboratory Rome Research Site Public Affairs Office and is available to the general public, including foreign nationals. Copies may be obtained from the Defense Technical Information Center (DTIC) (<http://www.dtic.mil>).

AFRL-IF-RS-TR-2006-303 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE DIRECTOR:

/s/

CLARE D. THIEM
Work Unit Manager

/s/

JAMES A. COLLINS
Deputy Chief, Advanced Computing Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) OCT 06		2. REPORT TYPE Final		3. DATES COVERED (From - To) Jun 04 – Feb 06	
4. TITLE AND SUBTITLE USER EVALUATION OF BIOSPICE				5a. CONTRACT NUMBER FA8750-04-1-0242	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 61101E	
6. AUTHOR(S) Evyatar Av-Ron, John H. Byrne				5d. PROJECT NUMBER BIOC	
				5e. TASK NUMBER UT	
				5f. WORK UNIT NUMBER EX	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Department of Neurobiology and Anatomy The University of Texas Medical School at Houston Houston Texas 77030				8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Defense Advanced Research Projects Agency AFRL/IFTC 3701 North Fairfax Drive 525 Brooks Road Arlington Virginia 22203-1714 Rome New York 13441-4505				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSORING/MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2006-303	
12. DISTRIBUTION AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED. PA#06-701					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The BioComp project was established to provide the system biology community with a tool, a computing environment called Bio-SPICE, to carry out data collection, model development and computer simulation. Usability studies of the Bio-SPICE Dashboard and seven simulators/editors: BioSketchPad/Charon, BioSpreadSheet/ESS, Jarnac/JDesigner, JigCell, Simpathica, BioNets and Pathway Builder were performed. The usability of each simulator was examined and the interoperability of the various tools using System Biology Markup Language (SBML) as a language of exchange between the different applications. Each tool provides a model builder component and a simulator component. The model builder components varied from providing a scripting-like language, a tabular editor to a graphical user interface. The simulator components use either deterministic solver algorithms (e.g., CVODE, Backward Euler) or a stochastic solver using the Gillespie algorithm. Seven simulators/editors were examined with four models that produce oscillatory behavior or time course of a long duration. A tutorial manual and a use case document were written.					
15. SUBJECT TERMS Bio-SPICE, tutorial, simulator, SBML, interoperability, system biology					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UL	18. NUMBER OF PAGES 89	19a. NAME OF RESPONSIBLE PERSON Clare D. Thiem
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code)

Table of Contents

1. Introduction	1
1.1 The Bio-SPICE Project	1
1.1.1 The Bio-SPICE tool kit	2
1.1.2 System Biology Markup Language	2
1.2 Modeling	2
1.2.1 Stochastic	3
1.2.2 Deterministic	3
1.2.3 Use Case Models	3
1.3 Bio-SPICE Tools: Simulators and Editors	4
1.4 Tutorial and Use Case Documentation	5
2. Methods	6
2.1 Models	7
2.1.1 Circadian Rhythm I	7
2.1.2 Circadian Rhythm II	8
2.1.3 Cell Cycle Division	9
2.1.4 Glycolytis Oscillations	11
2.1.5 Memory Induction	11
2.2 Simulators and Editor	13
2.2.1 Jarnac and JDesigner	13
2.2.2 BioSketchPad and Charon	16
2.2.3 BioSpreadsheet and ESS	18
2.2.4 JigCell	20
2.2.5 Simpathica	22
2.2.6 BioNets	23
2.2.7 PathwayBuilder	24
3. Results	26
3.1 Simulators and Editors	26
3.1.1 Performance	26
3.1.2 Interoperability	27
3.1.3 Graphic Functionality	29
3.1.4 Usability and Documentation	30
3.2 Dashboard Integration	31
3.3 Tutorial and Use Case Documentation	34
3.3.1 Getting Started User Manual	34
3.3.2 Jarnac Use Case	35
4. Summary	35
5. References	36
Appendix A - Simulator Capabilities and Features	37
Appendix B - Interoperability	40
Appendix C - User Manual	42
Appendix D - SBW (Jarnac) Use Case	79

List of Figures

Figure 2.1. Circadian model (M1) ordinary differential equations.	7
Figure 2.2. Illustration of circadian model.	8
Figure 2.3. Circadian model (M2) ODEs scaled to 100 mass action equations.	9
Figure 2.4. Cell cycle division model (M3) ordinary differential equations.	10
Figure 2.5. Diagram of cell cycle division model.	10
Figure 2.6. Allosteric model ordinary differential equations.	11
Figure 2.7. Complexity of the memory model is shown by a partial list of the ODEs.	12
Figure 2.8. Illustration of the biochemical elements of the memory model.	13
Figure 2.9. Jarnac script editor displaying circadian model M1.	14
Figure 2.10. Jarnac display of simulation time course of circadian model.	15
Figure 2.11. JDesigner editor display of circadian model.	16
Figure 2.12. BioSketchPad editor display of cell division model M3.	17
Figure 2.13. Charon simulator display of initial time course of circadian model.	18
Figure 2.14. BioSpreadsheet editor display of circadian model.	19
Figure 2.15. BioSpreadsheet display of simulation time course of circadian model.	20
Figure 2.16. XPP display of simulation time course of memory model.	21
Figure 2.17. JigCell model builder display of memory model.	21
Figure 2.18. Simpathica editor display of circadian model.	22
Figure 2.19. Simpathica simulator time course display of circadian model.	23
Figure 2.20. BioNets editor and simulator display of circadian model.	24
Figure 2.21. Pathway Builder editor display of circadian model.	25
Figure 3.1. Performance of six solvers for models M1 to M5.	27
Figure 3.2. Interoperability of seven editors for models M1 to M5.	28
Figure 3.3. Graphic functionality of seven editors for models M1 to M5.	30
Figure 3.4. Usability and documentation of eight simulators/editors for models M1 to M5. ...	31
Figure 3.5. Dashboard with ESS simulation of circadian model displaying time course with 2DGrapher.	32
Figure 3.6. Dashboard with BioSpreadsheet editor displaying circadian model M1.	33
Figure 3.7. Dashboard with Jarnac simulation displaying allosteric model M4 time course and values.	34

1. Introduction

Systems biology is the study of the mechanisms underlying complex biological processes as integrated systems of many, diverse, interacting components. Systems biology involves (1) collection of large sets of experimental data (by high-throughput technologies and/or by mining the literature of reductionist molecular biology and biochemistry), (2) proposal of mathematical models that might account for at least some significant aspects of this data set, (3) accurate computer solution of the mathematical equations to obtain numerical predictions, and (4) assessment of the quality of the model by comparing numerical simulations with the experimental data. (<http://jigcell.biol.vt.edu/glossary.html>)

With the enormous growth of life science research it is clear that tools are needed to store and analyze the large amount of experimental data, to build, simulate and analyze mathematical models, and to visualize data and system dynamics. The Defense Advanced Research Projects Agency (DARPA) Bio-Computation (BioCOMP) Program was established to provide the system biology community with a tool, a computing environment called Bio-SPICE (Biological Simulation Program for Intra- and Inter-Cell Evaluation), to carry out data collection, model development and computer simulation.

To examine the current state of the Bio-SPICE system, we proposed to perform a usability study of several simulators/editors. Our goal was to examine the usability of specific editors/simulators and the interoperability of the various tools. A second goal was to develop a tutorial for first-time users of Bio-SPICE, and provide a Getting Started manual which will illustrate a use case on how to develop, implement and simulate a model with Bio-SPICE.

1.1 The Bio-SPICE Project

The Bio-SPICE project was started as part of the Bio-Computation program, funded by DARPA. The goal was to develop a computational framework that enables the construction of sophisticated models of intracellular processes that can be used to predict and control the behavior of living cells. In addition, Bio-SPICE is also being examined to generate new computational paradigms and engineering applications that utilize bio-molecules as an information processing, sensing, or structural components (<http://www.darpa.mil/ipto/programs/biocomp/index.htm>).

The Bio-SPICE web site <https://biospice.org/index.php> provides information and software downloads to the Bio-SPICE community. In order to download software from the web site, you must join and become a Bio-SPICE member. A second source for the Bio-SPICE project is on SourceForge.net, <http://sourceforge.net/projects/biospice>. The latest version of the Dashboard can be

downloaded from SourceForge.net without the need to register as a Bio-SPICE user as on the official Bio-SPICE web site. This reflects the transition from a DARPA funded program to a truly open source environment to fulfill the desire to see Bio-SPICE continue to mature and evolve long after the DARPA funding has ended.

1.1.1 The Bio-SPICE tool kit

The Bio-SPICE tool kit is comprised of the Dashboard and a range of tools.

- Dashboard: GUI application to create and run workflows.
- Data analysis tools: Tools to mine the data
- Database tools: The large amount of data that experiments produce needs to be stored and mined.
- Model analysis tools: A model can provide information through various means of analysis, e.g., bifurcation, parameter sensitivity.
- Model composition & visualization tools: In order to construct a model, tools are provided for model composition, as well as visualization.
- Simulator tools: Models need to be solved using various types of simulators, continuous ODE simulators or stochastic simulators.

1.1.2 System Biology Markup Language

The Bio-SPICE project chose the System Biology Markup Language (SBML) as a language of exchange between the different tools. SBML is a computer-readable format for representing **models of biochemical reaction networks**. For example, SBML is applicable to metabolic networks, cell-signaling pathways and regulatory networks. For further details see the web site <http://sbml.org/index.psp>.

1.2 Modeling

There are several methods of modeling system biology, in particular cellular processes and biochemical interactions. Quantitative modeling is usually carried out with ordinary differential equations (ODEs), which represent system variables that change as nonlinear functions of other variables and/or parameters. To describe mass action of biochemical reactions, equations may be written in stoichiometric form. There are two general types of solvers for quantitative models, stochastic and deterministic. When small populations of molecules are considered, stochastic modeling is more appropriate since molecular fluctuations may alter the dynamics. With large population sizes, deterministic modeling will be computationally faster as stochastic modeling of large populations is computationally time intensive and should produce results similar to an ODE model.

1.2.1 Stochastic

Stochastic solvers use the Gillespie algorithm or a variant of it. Basically, all possible reactions are examined and the reaction with the shortest time interval is “scheduled”. The executed reaction will affect the population of molecules and thereby other reactions. Therefore certain reaction times must be recalculated, and again, the shortest time interval reaction is scheduled. This algorithm was shown to describe the time evolution of a chemical system.

1.2.2 Deterministic

Deterministic solvers use a variety of algorithms to compute the value of the variables in a model. At each time step, all variables are calculated based on either only previous values (explicit methods) or incorporating estimates of the next value (implicit methods). Some of the more known algorithms are forward and backward Euler methods (explicit and implicit respectively), Runge Kutta, Gear, CVODE, and Crank-Nicolson. The algorithms differ in terms of ease of use (e.g., explicit methods requiring only initial conditions, implicit methods require estimations of variable values), speed of computation (how many function calls per time step) and accuracy especially important in cases of stiff systems.

1.2.3 Use Case Models

Five models were used to evaluate tools of the Bio-SPICE project:

- **Circadian rhythm**, mRNA transcription and protein phosphorylation (Smolen P et al, *J Neurosci*, 21:6644-6656, 2001)
14 ODEs; 20 mass action reactions.
- **Circadian rhythm II**, mRNA transcription and protein phosphorylation
95 ODEs; 100 mass action reactions.
- **Cell division cycle**, cdc2 and cyclin interactions (Tyson J, *PNAS*, 88:7328-7332, 1991)
6 ODEs; 14 reaction equations.
- **Allosteric model for glycolytic oscillations** (Goldbeter A & Lefever R, *Biophys J*, 12:1302-1315, 1972)
2 ODEs (minimal model)
- **Memory induction** (Pettigrew D et al, *J Comp Neurosci*, 18:163-181, 2005)
16 ODEs; 39 reaction equations.

Five models were used in our present study, a circadian rhythm model (M1), a second version (M2) of the circadian model which contained a 5-fold increase in the number of equations, a cell division cycle model (M3), an allosteric model for glycolytic oscillations (M4) and a memory induction model (M5).

The circadian rhythm model (M1) corresponds to the oscillations in the levels of core gene expression. Over a period of 24 hours the mRNA level oscillates.

The cell division cycle model (M3) describes the interaction of the proteins cyclin and cdc. The model for certain parameter regime showed limit cycle solutions.

The allosteric model for glycolytic oscillations (M4) is a reduced model, of two ODEs, and showed oscillations of substrate and product concentrations.

The memory induction model (M5) represents short-, intermediate-, and long-term phases of protein kinase A (PKA) activation. For a stimulus above threshold, the model describes the time course of PKA activation which correlates with the three phases of memory.

The above models were used to examine the Dashboard and seven applications of the Bio-SPICE project: BioSpreadSheet with ESS, BioSketchPad with Charon, JigCell, SBW Jarnac and JDesigner, BioNets, Simpathica and Pathway Builder. The goal was to simulate the models with each tool, graph the resulting time course, and convert the model to SBML, to compare usability and interoperability.

1.3 Bio-SPICE Tools: Simulators and Editors

To examine the current state of the Bio-SPICE system, we performed a usability study of the Bio-SPICE Dashboard and seven simulators/editors: BioSketchPad/Charon, BioSpreadSheet, Jarnac/JDesigner, JigCell, Simpathica, BioNetS and Pathway Builder. Our goal was to examine the usability of each simulator, both independently and within the Dashboard, and the interoperability of the various tools, using SBML as a language of exchange between the different applications. Most tools provide a model builder component and a simulator component. The model builder components varied from providing a scripting-like language, a tabular editor to a graphical user interface (GUI). Four simulator components use deterministic solver algorithms (e.g., LSODE, Runge Kutta, Backward Euler, etc) and two stochastic solvers use the Gillespie algorithm or an optimized variant. We examined the six simulators with four models that produce oscillatory behavior or time course of a long duration.

Due to the complex nonlinear equations of the memory induction model, only general solvers were tested. The simulators Jarnac and JigCell are general solvers that can resolve any type of nonlinear equation (excluding delayed equations). On the other hand, BioSketchPad, BioSpreadSheet and JDesigner are stoichiometric solvers that can only solve equations in the form of mass action equations. If a model contains a complex nonlinear equation, it might be very difficult to convert the

equation to a stoichiometric form and therefore a general solver may be more adaptable. A table presenting a comparison of the different simulators and their performance in solving the various models is provided in Appendix A.

We examined seven simulators/editors of the Bio-SPICE project both independently and as part of the Dashboard for capability, graphic functionality, usability and interoperability. The results are presented in Appendix A.

In order to test the model building component of each tool, which varies from using a script-like language (Jarnac), tabular editor (Simpathica, BioNetS, JigCell, BioSpreadSheet) and a GUI (BioSketchPad, JDesigner, Pathway Builder), we implemented four models (M1, M3, M4, M5; M2 is M1 scaled to 100 equations) in all six simulators (non-stoichiometric models M4 and M5 were only tested with general solvers, Jarnac and JigCell).

The solvers were used to simulate the models. We tested performance at various time steps and examined scalability. The results are presented in Appendix A. Briefly, Charon was found too slow for the models we tested. Of the three stochastic solvers, BioNetS was the fastest. Jarnac and JigCell are general solvers, which were simple to use, although Jarnac requires programming. The lack of full interoperability between the solvers makes it difficult to use the most appropriate tool for model implementation and simulation.

The Dashboard was used with two of the simulator analyzers, BioSpreadsheet/ESS and Jarnac. The execution time for the models using the Dashboard was faster for Jarnac, while slower with BioSpreadsheet/ESS. As of this publication, there is no component to provide input to a model, which is needed for model M5, and therefore this model could not be properly simulated with the Dashboard.

1.4 Tutorial and Use Case Documentation

The BioCOMP Program sought to provide the system biology community with a tool, a computing environment called Bio-SPICE, to carry out data collection, model development and computer simulation. In order to assist new users to Bio-SPICE, a tutorial manual (see Appendix C) has been written that provides a step-by-step guide to installing Bio-SPICE, developing a model and running a simulation. The tutorial provides screen shot images showing the user what their screen should look like as they follow the instructions. The manual is divided into five chapters:

- Introduction to Bio-SPICE - provides an introduction to systems biology and an overview of the Bio-SPICE toolkit.
- Getting Started - describes the step to download and install the Bio-SPICE system.
- Model Editor - describes how to enter model equations with one of the editors, BioSpreadsheet. A simple circadian rhythm model is presented as a use case.

- Using the Dashboard - describes how to construct a workflow and run a simulation using the model developed in the previous chapter.
- Bio-SPICE Tools - provides a short description of the tools that exist in the Bio-SPICE toolkit.

The goal of the tutorial manual is to provide first-time users with a step-by-step guide to using the Bio-SPICE toolkit. The first step is to download the Bio-SPICE software and registering with the Bio-SPICE community. Next, the user will need to download specific applications that run with Bio-SPICE. The tutorial describes how to implement a model using a Bio-SPICE editor. A model that describes circadian rhythms (M1) is presented and the user is shown how to implement the model. The tutorial then describes how to use the editor BioSpreadsheet and the stochastic solver ESS. With the model implemented, the user is presented with the Bio-SPICE Dashboard and learns how to install the analyzers they need in order to run a simulation of the model. They are shown how to develop a workflow, run a simulation and plot the time course of model variables.

2. Methods

One of the BioCOMP Program's objectives was to provide the system biology community with a tool, a computing environment called Bio-SPICE, to carry out data collection, model development and computer simulation. To examine the current state of the Bio-SPICE system, we performed a usability study of the Bio-SPICE Dashboard and seven simulators/editors: BioSketchPad/Charon, BioSpreadSheet, Jarnac/JDesigner, JigCell, Simpathica, BioNets and Pathway Builder. Our goal was to examine the usability of each simulator, both independently and within the Dashboard, and the interoperability of the various tools, using SBML as a language of exchange between the different applications.

Five models were used in our present study, a circadian rhythm model (M1), M1 scaled to 100 equations (M2), a cell division cycle model (M3), an allosteric model for glycolytic oscillations (M4) and a memory induction model (M5). The models were used to examine the Dashboard and applications of the Bio-SPICE project. We examined seven simulators/editors of the Bio-SPICE project both independently and as part of the Dashboard for capability, graphic functionality, usability and interoperability. In order to test the model building component of each tool, which varies from using a script-like language (Jarnac), tabular editor (Simpathica, BioNetS, JigCell, BioSpreadSheet) and a GUI (BioSketchPad, JDesigner, Pathway Builder), we implemented four models (M1, M3, M4, M5; M2 is M1 scaled to 100 equations) in all six simulators (non-stoichiometric models M4 and M5 were only tested with general solvers, Jarnac and JigCell). The solvers were used to simulate the models. We tested performance at various time steps and examined scalability.

The Dashboard was used with two of the simulator analyzers, BioSpreadsheet/ESS and Jarnac. The execution time for the models using the Dashboard was faster for

Jarnac, while slower with BioSpreadsheet/ESS. There is no component to provide input to a model, which is needed for model M5, and therefore this model could not be properly simulated with the Dashboard.

2.1 Models

Five models were used to evaluate tools of the Bio-SPICE project, in terms of usability and performance. The models consisted of ODEs, and three models were converted to stoichiometric form in order to evaluate tools that require such equations.

2.1.1 Circadian Rhythm I

The circadian rhythm model (Smolen et. al., 2001) corresponds to the oscillations in the levels of core gene expression. Our model uses a transcription factor (TF) which undergoes multiple phosphorylation steps. Over the space of a day, TF protein becomes fully phosphorylated and then degrades. This relieves TF repression so that another "burst" of TF transcription can occur.

$$\frac{d[\text{mRNA}]}{dt} = v_R * \frac{K_R}{K_R + [P_{12}]} - k_d [\text{mRNA}]$$

$$\frac{d[P_0]}{dt} = k_p [\text{mRNA}] - k_{ph} [P_0]$$

$$\frac{d[P_i]}{dt} = k_{ph} [P_{i-1}] - k_{ph} [P_i] \text{ for } i = 1 \dots 11$$

$$\frac{d[P_{12}]}{dt} = k_{ph} [P_{11}] - \frac{v_p [P_{12}]}{K_P + [P_{12}]}$$

Figure 2.1. Circadian model (M1) ordinary differential equations.

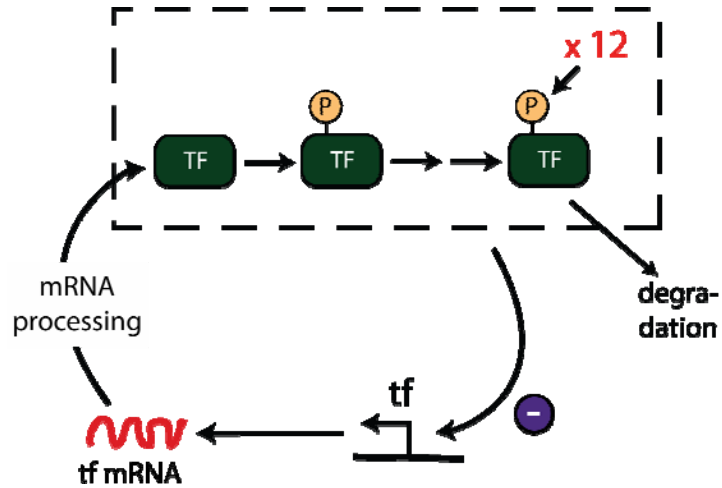


Figure 2.2. Illustration of circadian model.

2.1.2 Circadian Rhythm II

In order to evaluate scalability of the various simulators, a second version of the circadian model was developed. The original circadian model contains 14 ODEs which were converted to 20 mass action equations. The second version of the circadian rhythm model was increased to 100 mass action equations by simply adding more phosphorylation steps.

$$\frac{d[\text{mRNA}]}{dt} = v_R * \frac{K_R}{K_R + [P_{12}]} - k_d [\text{mRNA}]$$

$$\frac{d[P_0]}{dt} = k_p [\text{mRNA}] - k_{ph_0} [P_0]$$

$$\frac{d[P_i]}{dt} = k_{ph_{i-1}} [P_{i-1}] - k_{ph_i} [P_i] \text{ for } i=1 \dots 92$$

$$\frac{d[P_{12}]}{dt} = k_{ph_{11}} [P_{11}] - \frac{v_p [P_{12}]}{K_P + [P_{12}]}$$

Figure 2.3. Circadian model (M2) ODEs scaled to 100 mass action equations.

2.1.3 Cell Cycle Division

The cell division cycle model (Tyson, 1991) describes the interaction of the proteins cyclin and cdc2 and the activation of maturation promoting factor, a heterodimer of cyclin and cdc2. The model for certain parameter regime showed limit cycle solutions. The model contains six ODEs and was converted to 14 stoichiometric (mass action) equations.

2.1.4 Glycolytic Oscillations

The allosteric model (Goldbeter and Lefever, 1972) for glycolytic oscillations is a reduced model, of 2 ODEs, using a quasi-steady-state hypothesis and dimensionless variables; and showed oscillations of substrate and product concentrations. No attempt was made to try and convert this model to stoichiometric form.

$$\begin{aligned}\frac{d[\alpha]}{dt} &= v - \sigma\phi \\ \frac{d[\gamma]}{dt} &= q\sigma\phi - k_s \gamma \\ \phi &= \frac{\alpha e(1 + \alpha e)^{n-1} (1 + \gamma)^n}{L(1 + \alpha e')^n + (1 + \alpha e)^n (1 + \gamma)^n}\end{aligned}$$

Figure 2.6. Allosteric model ordinary differential equations.

2.1.5 Memory Induction

The memory model (Pettigrew et. al., 2005) contains 16 ODEs (partial list see Figure 2.7), several of which are highly nonlinear equations. The model represents short-, intermediate-, and long-term phases of protein kinase A (PKA) activation, as well as represents phosphorylation of the transcription factor CREB1 by PKA and consequent induction of the immediate-early gene *Aplysia* ubiquitin hydrolase (Apuh), which is essential for long-term synaptic facilitation (LTF). The original model contains delayed differential equations which were simplified to the version in Figure 2.7 without delay. This model was not converted to a stoichiometric form.

$$\begin{aligned}
\frac{d[\text{cAMP}]}{dt} &= f(5\text{-HT}) - ([\text{cAMP}] - [\text{cAMP}]_{\text{basal}}) \\
\frac{d[\text{C}]}{dt} &= V_{\text{syn}} + k_{\text{fpka}} [\text{RC}] [\text{cAMP}]^2 - k_{\text{bpka}} [\text{R}] [\text{C}] + \\
&\quad k_{\text{Ap-uch}} [\text{RC}] ([\text{Ap-uch}] - [\text{Ap-uch}]_{\text{basal}}) - k_{\text{dpka}} [\text{R}] \\
\frac{d[\text{pREG}]}{dt} &= V_{\text{rphos}} \text{ERK}_{\text{act}} \left(\frac{[\text{REG}] - [\text{pREG}]}{[\text{REG}] - [\text{pREG}] + K_{\text{rphos}}} \right) - \\
&\quad V_{\text{dreg}} \left(\frac{[\text{pREG}]}{[\text{REG}] + K_{\text{dreg}}} \right) + k_{\text{dsm}} [\text{pREG}] \\
\frac{d[\text{mRNA}_{\text{REG}}]}{dt} &= V_{\text{mREG}} - V_{\text{dmreg}} [5\text{-HT}] \frac{[\text{mRNA}_{\text{REG}}]}{[\text{mRNA}_{\text{REG}}] - K_{\text{dmreg}}} - k_{\text{dmreg}} [\text{mRNA}_{\text{REG}}] \\
\frac{d[\text{MAPKK}]}{dt} &= -k_{\text{fMAPKK}} [\text{Raf}^{\text{p}}] \frac{[\text{MAPKK}]}{[\text{MAPKK}] + K_{\text{MK}}} + k_{\text{bMAPKK}} \frac{[\text{MAPKK}^{\text{p}}]}{[\text{MAPKK}^{\text{p}}] + K_{\text{MK}}} \\
\frac{d[\text{ERK}]}{dt} &= -k_{\text{fERK}} [\text{MAPKK}^{\text{pp}}] \frac{[\text{ERK}]}{[\text{ERK}] + K_{\text{MK}}} + k_{\text{bERK}} \frac{[\text{ERK}^{\text{p}}]}{[\text{ERK}^{\text{p}}] + K_{\text{MK}}} \\
\frac{d[\text{P}_{\text{pka}}]}{dt} &= k_{\text{phos1}} \cdot \text{PKA}_{\text{act}} \cdot (1 - \text{P}_{\text{pka}}) - k_{\text{dephos1}} \cdot \text{PPhos} \cdot \text{P}_{\text{pka}} \\
\frac{d[\text{Ap-uch}]}{dt} &= k_{\text{ApSyn}} \left(\frac{\text{P}_{\text{pka}}^2}{\text{P}_{\text{pka}}^2 + K_{\text{pka}}^2} \right) \left(\frac{\text{P}_{\text{erk}}^2}{\text{P}_{\text{erk}}^2 + K_{\text{erk}}^2} \right) + k_{\text{ApSynBasal}} - [\text{Ap-uch}]
\end{aligned}$$

Figure 2.7. Complexity of the memory model is shown by a partial list of the ODEs.

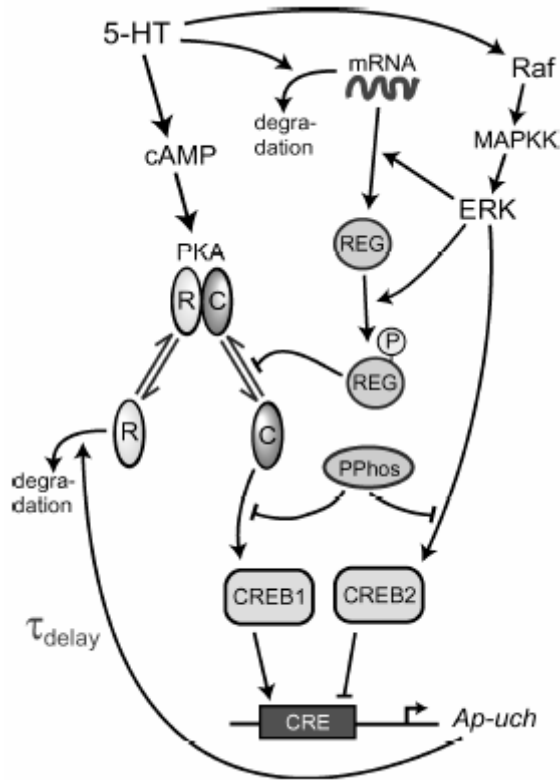


Figure 2.8. Illustration of the biochemical elements of the memory model.

2.2 Simulators and Editor

Seven simulators and editors were evaluated as part of the usability test of Bio-SPIKE tools.

2.2.1 Jarnac and JDesigner

Jarnac is a part of the SBW package (<http://sbw.kgi.edu>) which was a precursor of the Bio-SPIKE project. Jarnac provides a general solver that a script-like language to enter model equations. Users are not limited to stoichiometric equations with Jarnac. The graphic application JDesigner, on the other hand, does require stoichiometric equations, so that not every model that runs in Jarnac can work in JDesigner. Jarnac provides an output viewer to display the time course of the variables of the model.

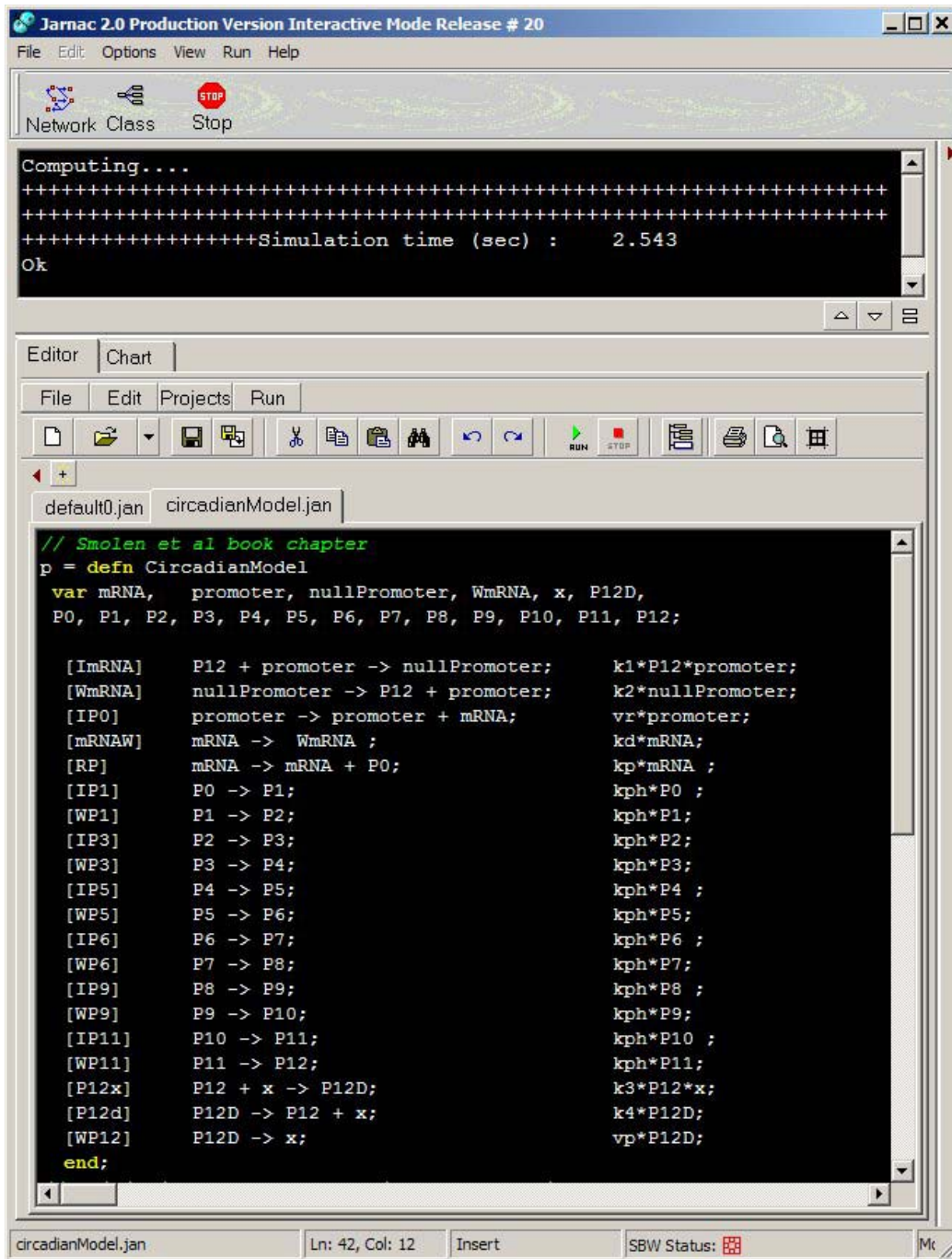


Figure 2.9. Jarnac script editor displaying circadian model M1.

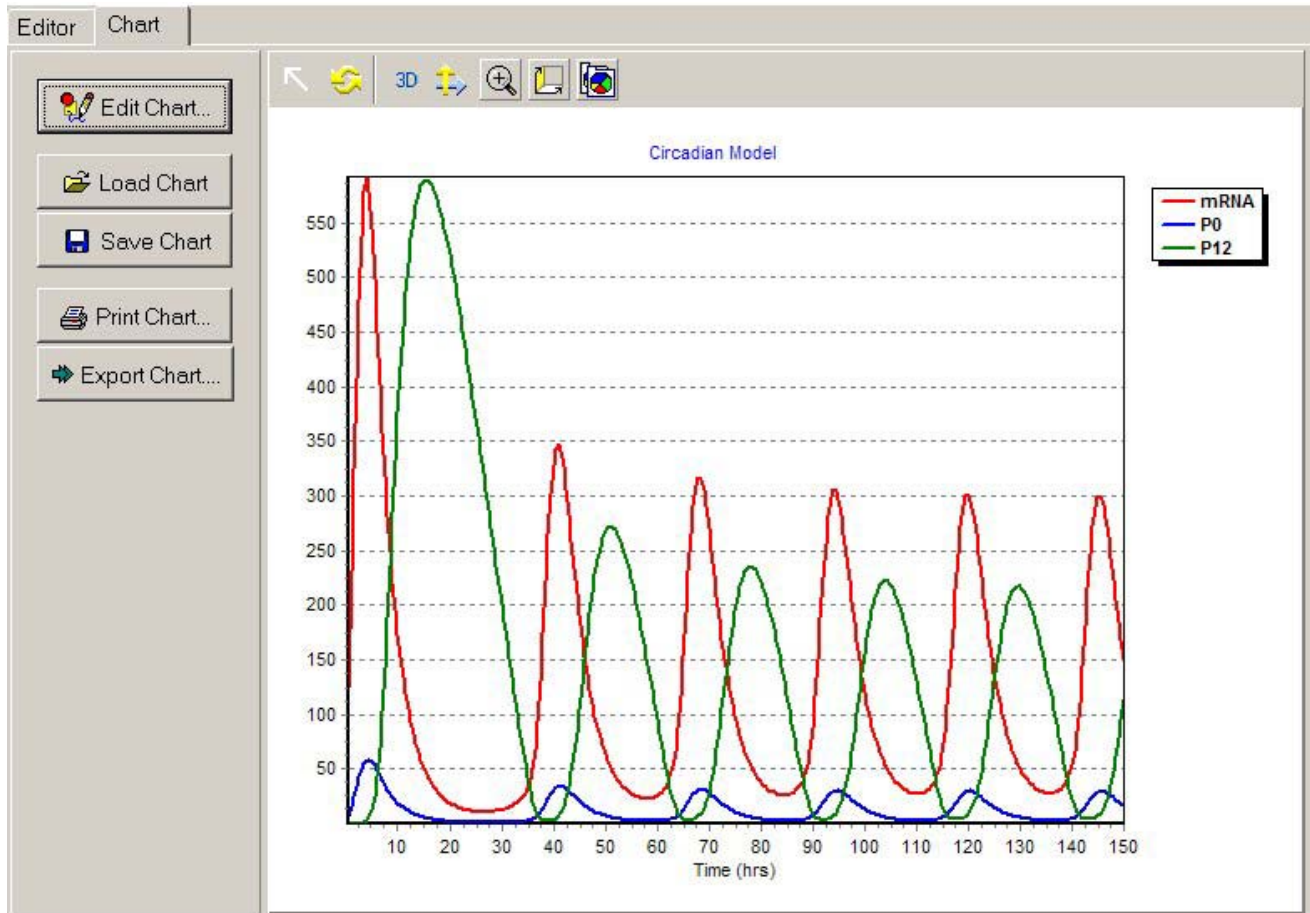


Figure 2.10. Jarnac display of simulation time course of circadian model.

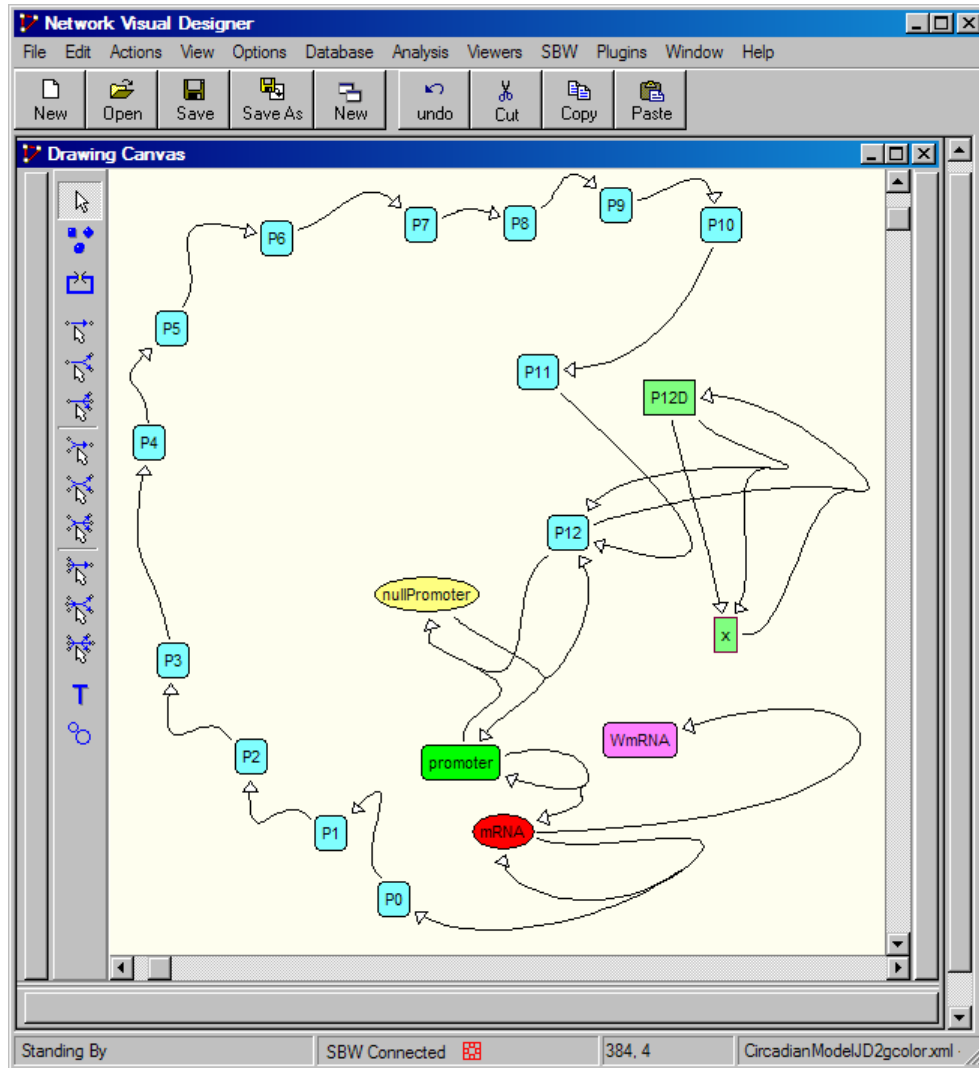


Figure 2.11. JDesigner editor display of circadian model.

2.2.2 BioSketchPad and Charon

The editor BioSketchPad is used with the simulator Charon. BioSketchPad only accepts models in stoichiometric form. Once the model is implemented in BioSketchPad, the simulator Charon is invoked and the model is exported into Charon in the required format.

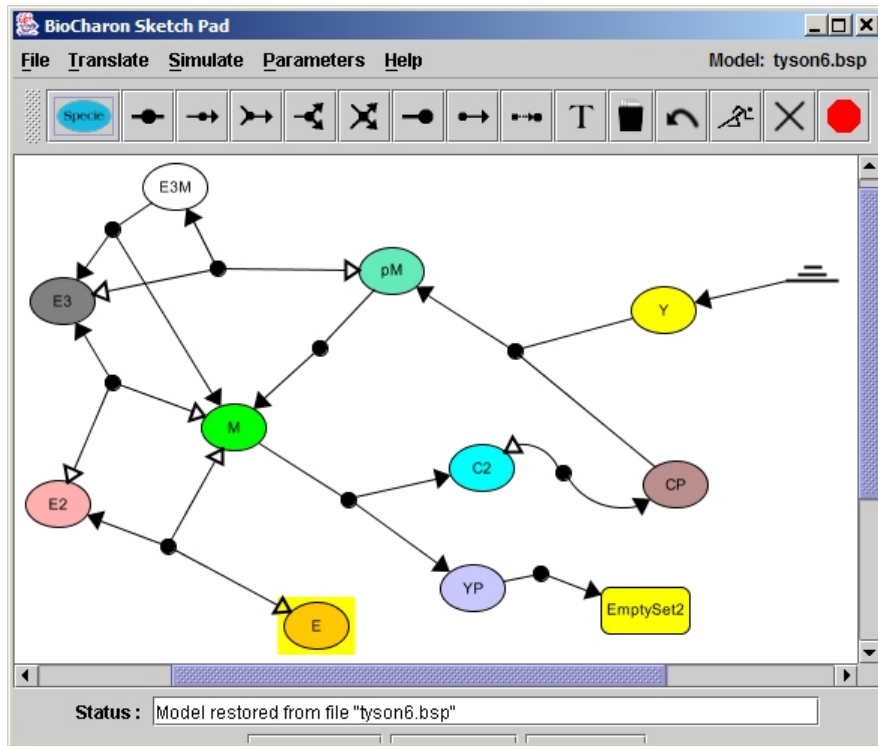


Figure 2.12. BioSketchPad editor display of cell division model M3.

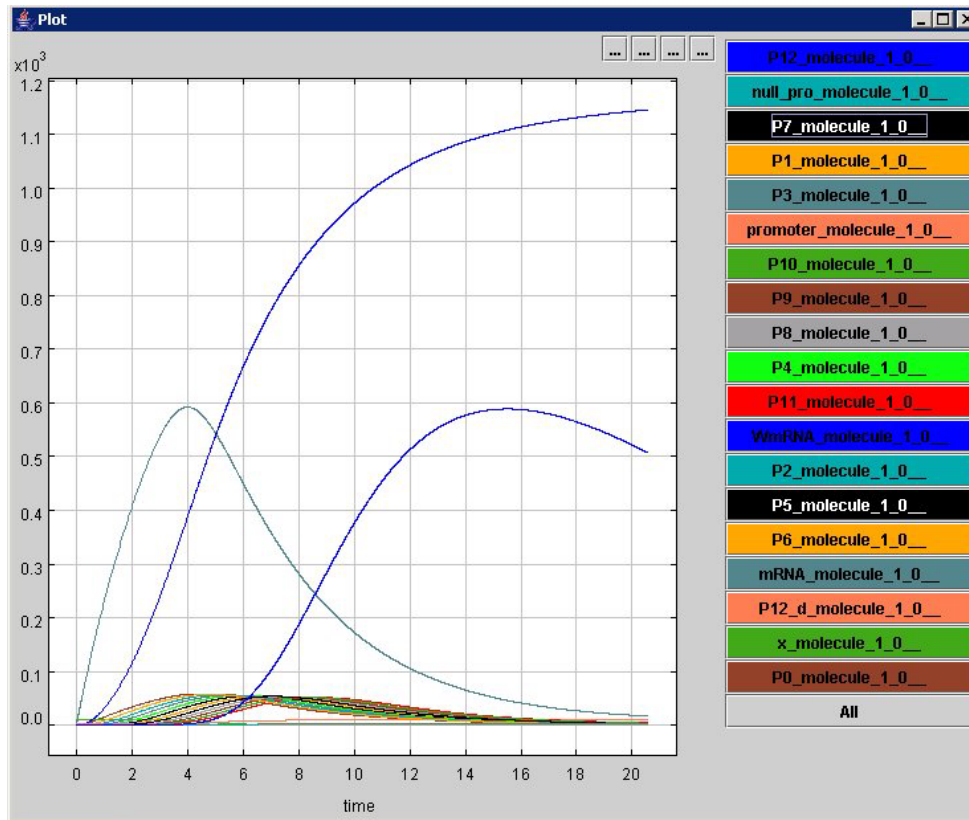


Figure 2.13. Charon simulator display of initial time course of circadian model.

2.2.3 BioSpreadsheet and ESS

The editor BioSpreadsheet (<http://biocomp.ece.utk.edu>) is used with the Exact Stochastic Simulator (ESS). BioSpreadsheet accepts only models in stoichiometric form. The solver ESS is a stochastic solver.

BioSpreadsheet v2.0 - [C:\UTenn\circadianModel]

File View Model Data Help

Reaction	Rate	Comment
P12 + promoter -> nullPromoter	18.2	
nullPromoter -> P12 + promoter	65	
promoter -> mRNA + promoter	130	
mRNA -> UmRNA	0.26	
mRNA -> mRNA + P0	0.26	
P0 -> P1	2.6	
P1 -> P2	2.6	
P2 -> P3	2.6	
P3 -> P4	2.6	
P4 -> P5	2.6	
P5 -> P6	2.6	
P6 -> P7	2.6	
P7 -> P8	2.6	
P8 -> P9	2.6	
P9 -> P10	2.6	
P10 -> P11	2.6	
P11 -> P12	2.6	
P12 + x -> P12D	3.9	
P12D -> P12 + x	39	
P12D -> x	3.9	

Add Reaction Remove Reaction

Information Species Reactions

Figure 2.14. BioSpreadsheet editor display of circadian model.

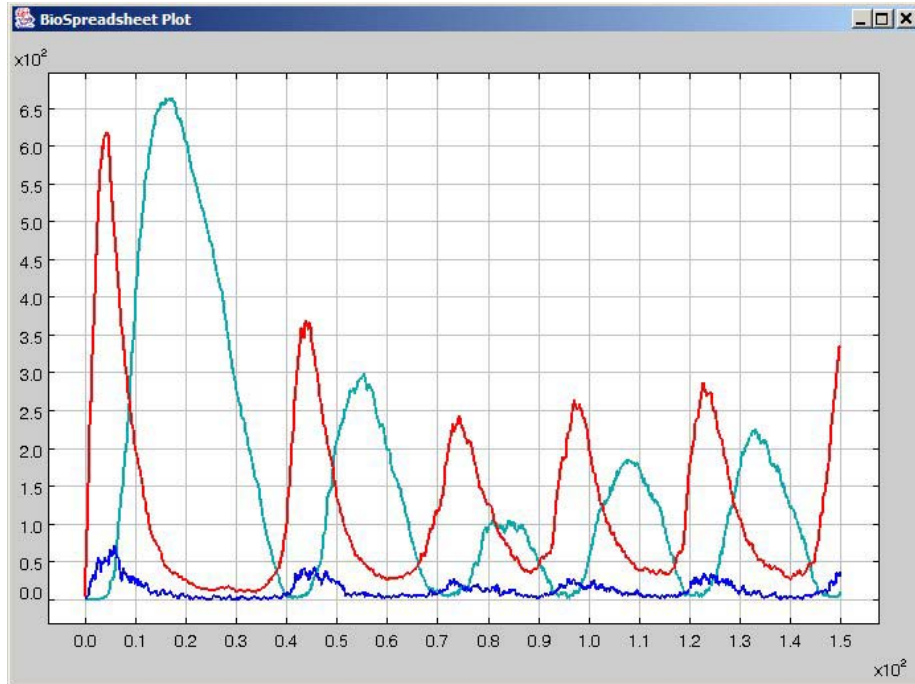


Figure 2.15. BioSpreadsheet display of simulation time course of circadian model.

2.2.4 JigCell

The application JigCell (<http://jigcell.biol.vt.edu>) is comprised of several components, a model builder as well as a run manager. The model builder provides users with a spreadsheet-like tabular GUI to implement a model. The user can define the type of equations that comprise the model, once a model is completely defined in the spreadsheet, the run manager is invoked in order to solve the equations using the XPP solver and display the time course.

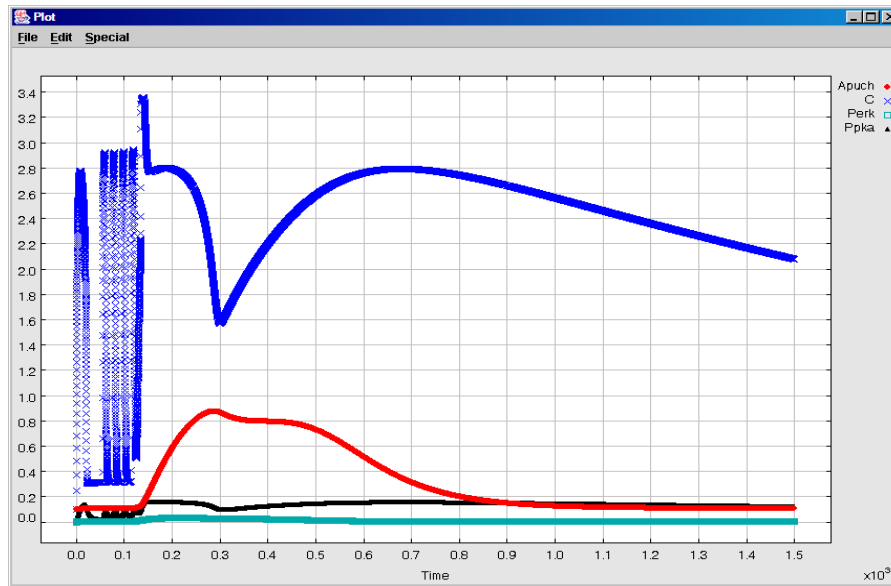


Figure 2.16. XPP display of simulation time course of memory model.

#	Reaction	Name	Type	Equation	Para
8	-> C	Csource	MassAction0		kf=0.0020
9	-> R	Rsource	MassAction0		kf=0.0020
10	C -> Cwaste	Cwaste	MassAction1	$(4.8E-4)*C$	kf=4.8E-4
11	R -> Rwaste	Rwaste	MassAction1	$(4.8E-4)*R$	kf=4.8E-4
12	RC -> RCwaste	RCwaste	MassAction1	$(4.8E-4)*RC$	kf=4.8E-4
13	C + R -> RC	RCproduct	MassAction2	$(12.0/(1.0+RegP/6.4E-4))*C*R$	kf=12.0/(1.0+RegP/6.4E-4)
14	RC -> R + C	RCrev	MassAction1	$105.0*cAMP*cAMP*RC$	kf=105.0*cAMP*cAMP
15	RC -> C	RCtoC	MassAction1	$(0.0070*(Apuch-0.1))*RC$	kf=0.0070*(Apuch-0.1)
16	R -> Rwaste	RtoC	MassAction1	$(0.0070*(Apuch-0.1))*R$	kf=0.0070*(Apuch-0.1)
17	-> Raf	Rafsource	MassAction0	$(0.0010*(0.5-Raf))$	kf=0.0010*(0.5-Raf)
18	Raf -> Rafwaste	Rafwaste	MassAction1	$0.00345*rstim*Raf$	kf=0.00345*rstim
19	ErkPP -> ErkPPwaste	ErkPPwaste	Michaelis-Menten	$0.12*ErkPP*1.0/(0.08+ErkPP)$	M1=1.0, J1=0.08, k1=0.12
20	-> Mapkk	Mapkksource	MassAction0	$(0.12*(0.5-Mapkk-MapkPP)/(0.5-Mapkk-MapkPP+0.08))$	kf=0.12*(0.5-Mapkk-MapkPP)/(0.5-Mapkk-MapkPP+0.08)
21	Mapkk -> Mapkkwaste	Mapkkwaste	Michaelis-Menten	$0.7*Mapkk*(0.5-Raf)/(0.08+Mapkk)$	M1=0.5-Raf, J1=0.08, k1=0.7
22	-> MapkkPP	MapkkPPsource	MassAction0	$(0.7*(0.5-Raf)*(0.5-Mapkk-MapkPP)/(0.5-Mapkk-MapkPP+0.08))$	kf=0.7*(0.5-Raf)*(0.5-Mapkk-MapkPP)/(0.5-Mapkk-MapkPP+0.08)
23	MapkkPP -> MapkkPPwaste	MapkkPPwaste	Michaelis-Menten	$0.12*MapkkPP*1.0/(0.08+MapkkPP)$	M1=1.0, J1=0.08, k1=0.12
24	-> Erk	Erksource	MassAction0	$(0.12*(0.5-Erk-ErkPP)/(0.5-Erk-ErkPP+0.08))$	kf=0.12*(0.5-Erk-ErkPP)/(0.5-Erk-ErkPP+0.08)
25	Erk -> Erkwaste	Erkwaste	Michaelis-Menten	$0.44*Erk*MapkkPP/(0.08+Erk)$	M1=MapkkPP, J1=0.08, k1=0.44
26	-> ErkPP	ErkPPsource	MassAction0	$(0.44*MapkkPP*(0.5-Erk-ErkPP)/(0.5-Erk-ErkPP+0.08))$	kf=0.44*MapkkPP*(0.5-Erk-ErkPP)/(0.5-Erk-ErkPP+0.08)
27	-> mRnaReg	mRnaRegsource	MassAction0	$(2.0E-5)$	kf=2.0E-5
28	mRnaReg -> mRnaRegwaste	mRnaRegwaste	MassAction1	$(3.0E-5)*mRnaReg$	kf=3.0E-5
29	mRnaReg -> mRnaRegdeg	mRnaRegdegrade	Michaelis-Menten	$0.00225*mRnaReg*rstim/(0.01+mRnaReg)$	M1=rstim, J1=0.01, k1=0.00225
30	-> Reg	Regsource	MassAction0	$(4.0*mRnaReg*(ErkPP+0.015))$	kf=4.0*mRnaReg*(ErkPP+0.015)
31	Reg -> Regwaste	Regwaste	MassAction1	$0.02*Reg$	kf=0.02
32	Reg -> Regdeg	Regdegrade	Michaelis-Menten	$0.16*Reg*1.0/(0.0015+Reg)$	M1=1.0, J1=0.0015, k1=0.16
33	-> RegP	RegPsource	MassAction0	$((ErkPP+0.015)*(Reg-RegP)/(Reg-RegP+1.5))$	kf=(ErkPP+0.015)*(Reg-RegP)/(Reg-RegP+1.5)
34	RegP -> RegPwaste	RegPwaste	MassAction1	$0.02*RegP$	kf=0.02
35	RegP -> RegPdeg	RegPdegrade	MassAction1	$(0.16/(Reg+0.0015))*RegP$	kf=0.16/(Reg+0.0015)
36	-> cAMP	cAMPsource	MassAction0	$(3.6*rstim/(rstim+14.0))$	kf=3.6*rstim/(rstim+14.0)
37	cAMP -> cAMPwaste	cAMPwaste	MassAction1	$(1.0-0.06/cAMP)*cAMP$	kf=1.0-0.06/cAMP
38	-> Ppka	Ppkasource	MassAction0	$(0.01*C*(1.0-Ppka))$	kf=0.01*C*(1.0-Ppka)
39	Ppka -> Ppkawaste	Ppkawaste	MassAction1	$0.15*Ppka$	kf=0.15
40	-> Perk	Perksource	MassAction0	$(0.0050*(ErkPP+0.015)*(1.0-Perk))$	kf=0.0050*(ErkPP+0.015)*(1.0-Perk)
41	Perk -> Perkwaste	Perkwaste	MassAction1	$0.05*Perk$	kf=0.05
42	-> Apuch	Apuchsource	MassAction0	$(Ppka*Ppka/(Ppka*Ppka+0.0225))*Perk*Perk/(Perk*Perk+4.0E-4)*0.0289$	kf=Ppka*Ppka/(Ppka*Ppka+0.0225)*Perk*Perk/(Perk*Perk+4.0E-4)*0.0289
43	Apuch -> Apuchwaste	Apuchwaste	MassAction1	$0.01*Apuch$	kf=0.01
44	-> Apuch	Apuchsource1	MassAction0	0.0010	kf=0.0010
45	-> rtime	rt	MassAction0	1.0	kf=1.0
46	-> rstim	rt	MassAction0	0.0	kf=0.0
47	stim		Species	value	value=0.0

Figure 2.17. JigCell model builder display of memory model.

2.2.5 Simpathica

Simpathica (<http://bioinformatics.nyu.edu/Projects/Simpathica>) is a deterministic simulator that supports stoichiometric equations. Reaction editing is handled through a drop-down menu to select reaction types and text fields to enter values and reacting species. When adding species and reactions, a chart of the system is produced in real time. Integration and plotting is handled through Octave and a trace analysis tool is also provided.

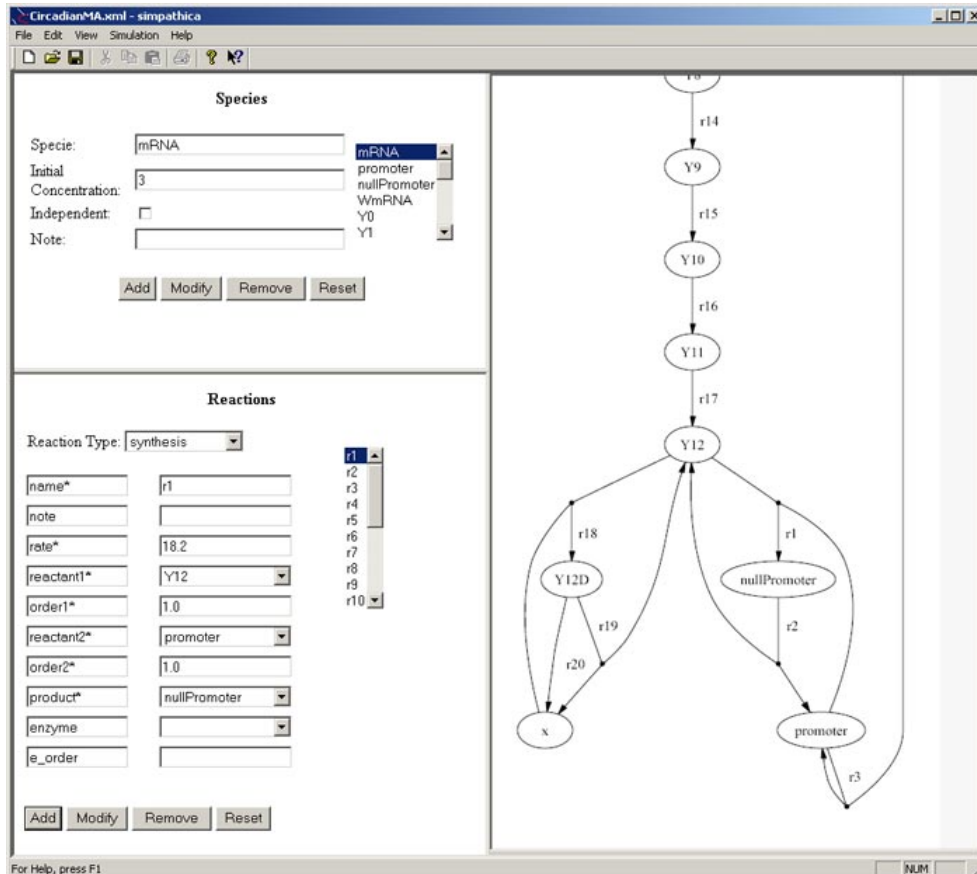


Figure 2.18. Simpathica editor display of circadian model.

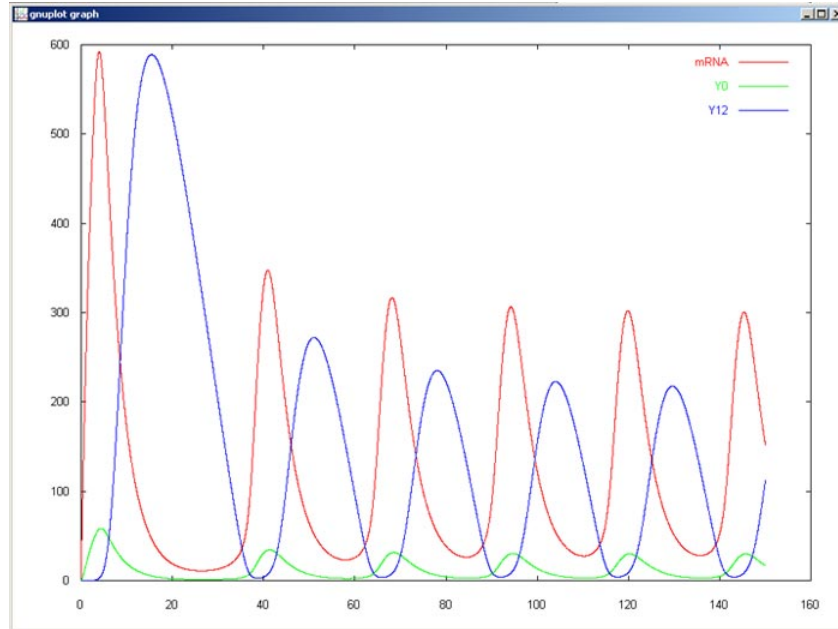


Figure 2.19. Simpathica simulator time course display of circadian model.

2.2.6 BioNets

BioNetS (<http://x.amath.unc.edu/BioNetS>) is a stochastic simulator which handles equations in stoichiometric form and can integrate in discrete or continuous modes. BioNetS features line by line species and reaction editing and a custom graphing component. Output from the simulation can be read into Matlab and the graph output can be saved as an image. BioNetS uses its own XML formatted files however, and does not have the ability to import or export SBML.

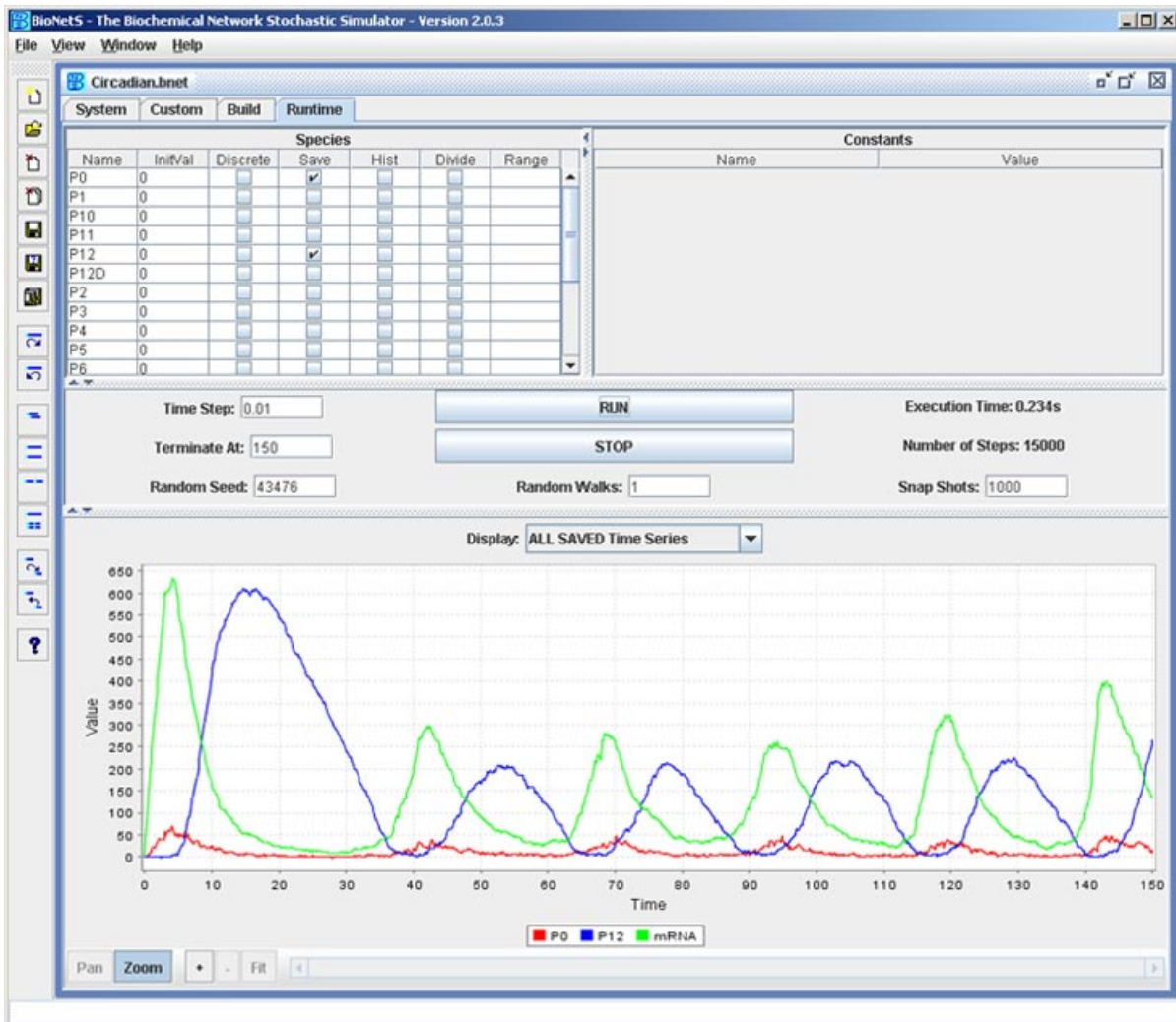


Figure 2.20. BioNets editor and simulator display of circadian model.

2.2.7 PathwayBuilder

Pathway Builder (<http://biospice.lbl.gov/PathwayBuilder>) is a visual model editing program. Pathway Builder allows the user to represent species and reactions as geometric objects of any size and color to distinguish between different pathways. Species and reactions can be grouped together and can be minimized to limit on screen clutter. A pathway can also include groups of subpathways and they can be visually distinct from one another.

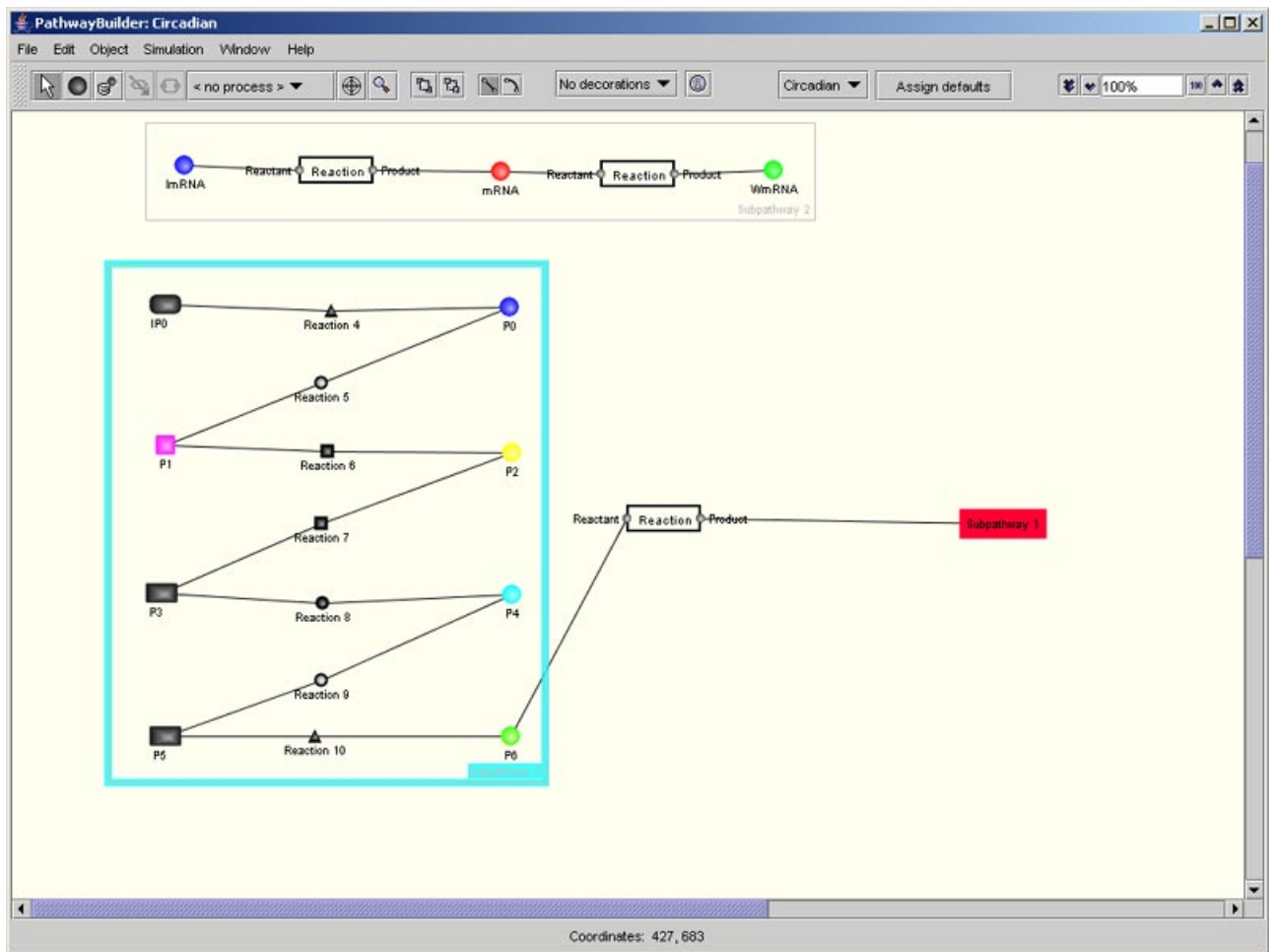


Figure 2.21. Pathway Builder editor display of circadian model.

3. Results

The Bio-SPICE system was designed to provide users with tools to carry out data collection, model development and computer simulation. To examine the current state of the Bio-SPICE system, we performed a usability study of the Bio-SPICE Dashboard and seven simulators and editors. Our goal was to examine the usability of each simulator, both independently and within the Dashboard, and the interoperability of the various tools, using SBML as a language of exchange between the different applications.

Five models were used in our present study, a circadian rhythm model (M1), M1 scaled to 100 equations (M2), a cell division cycle model (M3), an allosteric model for glycolytic oscillations (M4) and a memory induction model (M5).

We examined seven simulators/editors for capability, graphic functionality, usability and interoperability. In order to test the model building component of each tool, which varies from using a script-like language (Jarnac), tabular editor (Simpathica, BioNetS, JigCell, BioSpreadSheet) and a GUI (BioSketchPad, JDesigner, Pathway Builder), we implemented four models (M1, M3, M4, M5; M2 is M1 scaled to 100 equations) in all six simulators (non-stoichiometric models M4 and M5 were only tested with general solvers, Jarnac and JigCell). The solvers were used to simulate the models. We tested performance at various time steps and examined scalability.

The Dashboard was used with two of the simulator analyzers, BioSpreadsheet/ESS and Jarnac. The execution time for the models using the Dashboard was faster for Jarnac, while slower with BioSpreadsheet/ESS. There is no component to provide input to a model, which is needed for model M5, and therefore this model could not be properly simulated with the Dashboard.

3.1 Simulators and Editors

Five models M1 to M5 were used to evaluate the performance of the six simulators. Only models M1, M2 and M3 were converted to stoichiometric form, and were used for the evaluation of the simulators that require equations in stoichiometric form, BioSketchPad/Charon, BioSpreadsheet/ESS, Simpathica and BioNets.

3.1.1 Performance

Figure 3.1 shows the execution times of the various simulators to run the different models. The simulator BioNets was the fastest, for all three stoichiometric models. Only JigCell and Jarnac, as general solvers, were able to execute all five models. Charon was the slowest simulator, and its execution time extended beyond the scale of the graph. It was found not suitable for the types of models we tested, as the

adaptive method would reduce the step size to a very small value and proceed very slowly in the simulation.

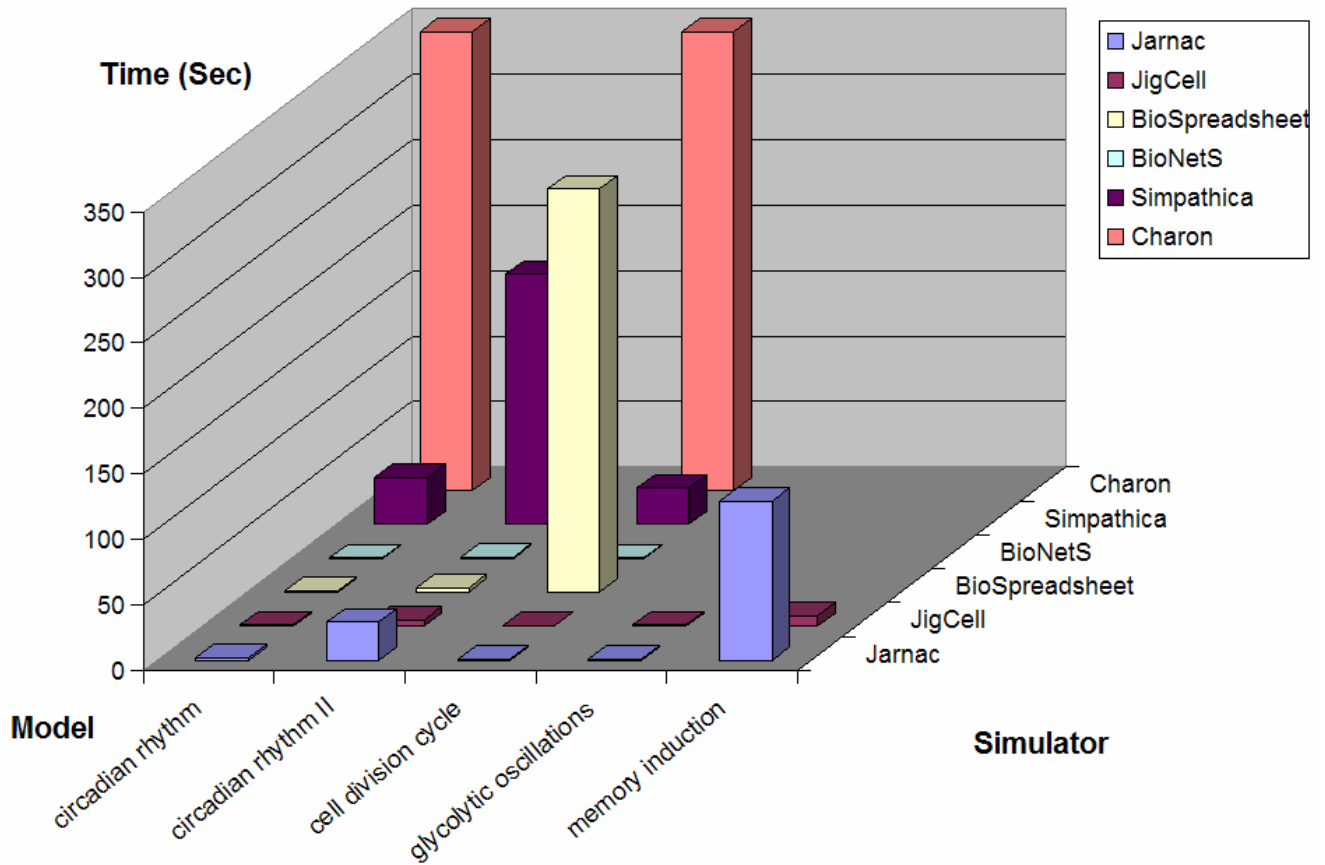


Figure 3.1. Performance of six solvers for models M1 to M5.

3.1.2 Interoperability

One of the goals of the Bio-SPIICE project was to provide a platform for the integration of diverse applications. Towards that end, the language SBML was chosen as a language of exchange between the various applications. Each application should provide an import/export function for SBML files. At the time of this writing, difficulties exist due to the fact that SBML is a rich language and various groups have implemented slightly different model representations.

Two methods were employed to test interoperability, using models M1 to M5 and using models from a public database. In the first, we tested the interoperability of the seven simulators/editors by using each application as a source for a SBML file and scoring the other applications on their ability to import that file.

The simulators/editors were measured on how well they imported:

- model equations
- kinetic rate constants and their values
- initial values of variables

The simulators/editors had variable amount of success. Some simulators were able to list the kinetic rate parameters but did not provide the initial condition values for the parameters (0.85 score). Others were only able to display the model equations (0.5 score). And some did very poorly in the attempt to import (0.05 score).

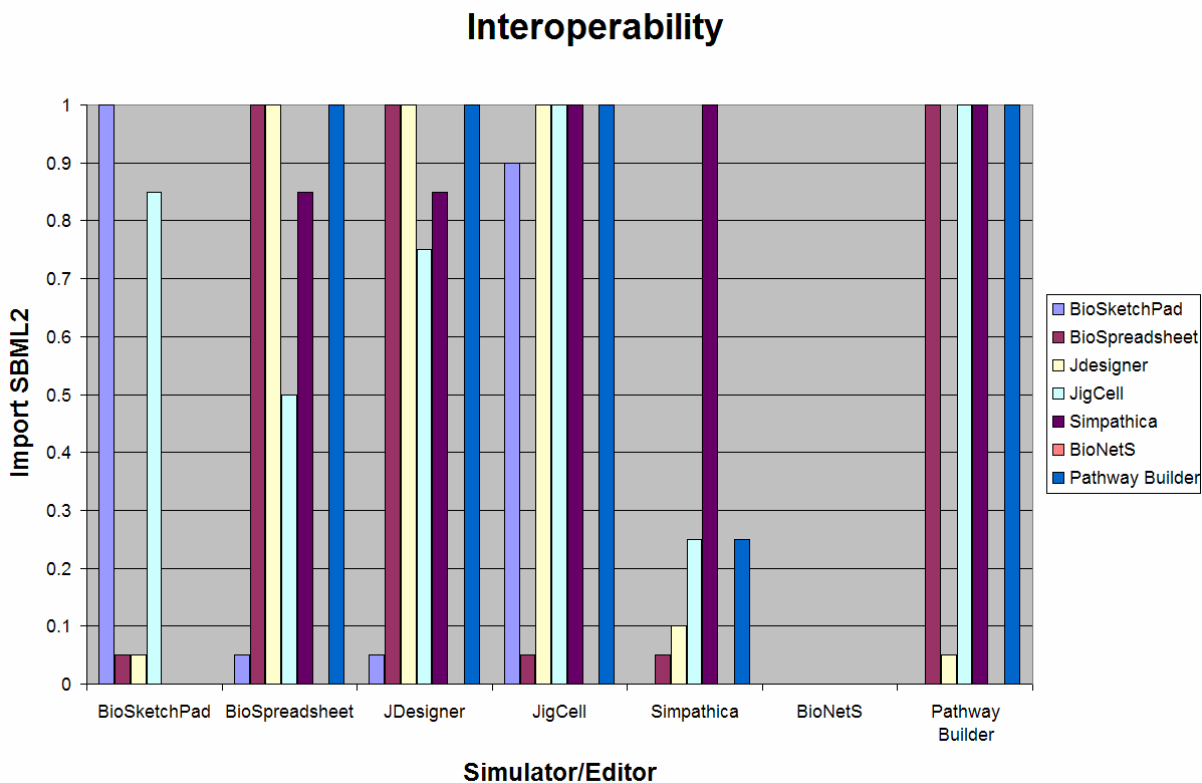


Figure 3.2. Interoperability of seven simulators/editors for models M1 to M5.

The second method for evaluating interoperability was carried out using models published in the BioModels Database (<http://www.ebi.ac.uk/biomodels/>). The 45 models in this database are in SBML form, curated by the authors of this database. The models were tested with four simulators/editors: JigCell, JDesigner, BioSpreadsheet and Simpathica. A table with the scores is provided in Appendix B.

3.1.3 Graphic Functionality

An important aspect of any editor is the capabilities provided to the user to implement a model as well as model maintenance and update. In our usability study, we examined various capabilities of the different editors, e.g. whether the editor provides cut and past functionality, and for single or multiple lines, can users group nodes together for move/copy. Since biochemical models will become more and more complex as our understanding of processes grows, the graphic capabilities of editors will play an important role in the choice of editors. We also qualified the graphic support that a simulator provides for displaying and printing the time course of a simulation.

Figure 3.3 displays the scores of the various editors with five being a high score and one a low score. Editors that provide single line copy/paste functionality received a score of three. Both Jarnac and JigCell provide the ability to cut and paste multiple lines. The graphic editors provide only cut and paste for single nodes or single connections. Four of the simulators provide the capability to print the time course of the simulation.

Graphic Functionality

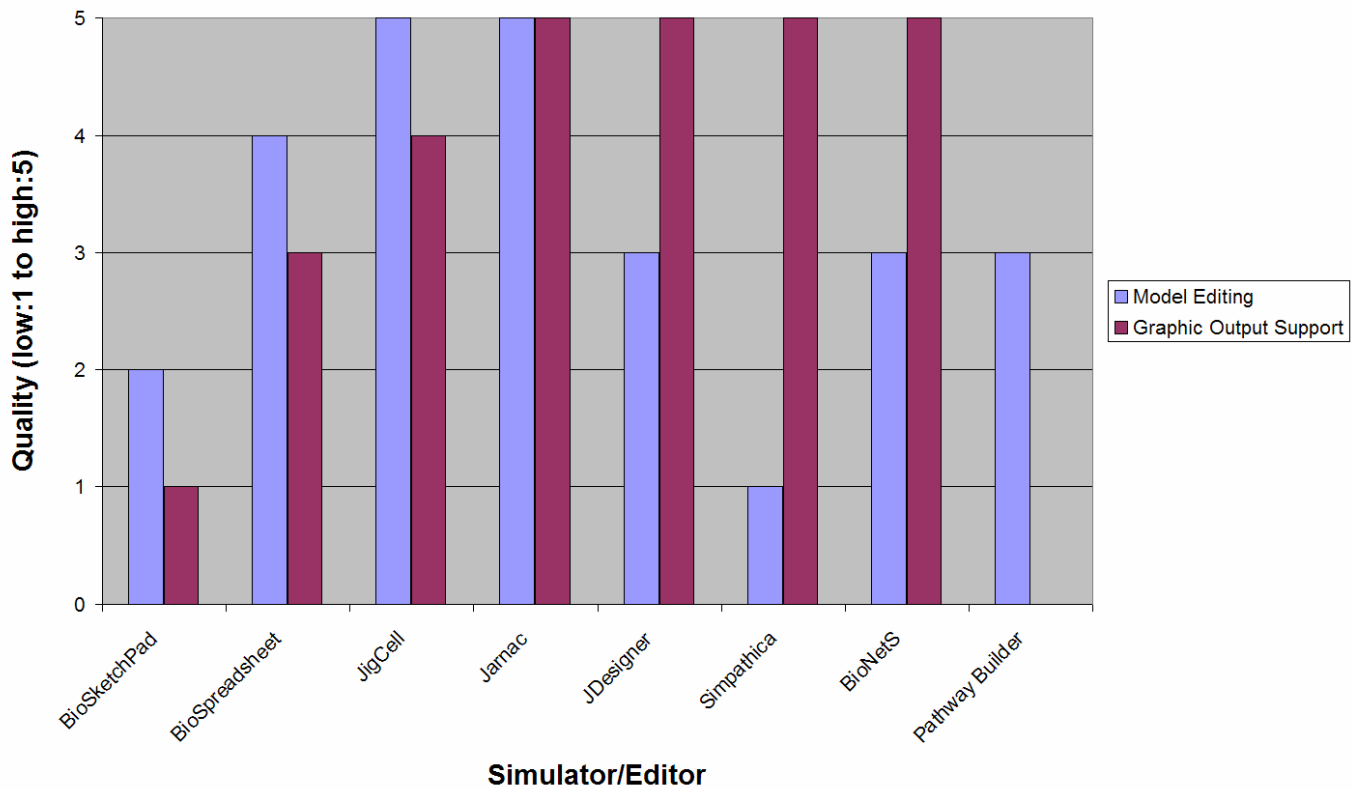


Figure 3.3. Graphic functionality of seven simulators/editors for models M1 to M5.

3.1.4 Usability and Documentation

The usability of an editor is determined by the ease in which it can be learned and used. Graphic editors tend to be simpler to learn and use. This can be seen from Figure 3.4 where all five graphic editors had top score for usability. Jarnac is a powerful tool, but requires programming, and hence the relatively low score for usability. Not all editors/simulators are provided with good documentation. All the latest versions of the editors/simulators provide reasonable documentation and several provide extensive documentation which merited a high score.

Usability and Documentation

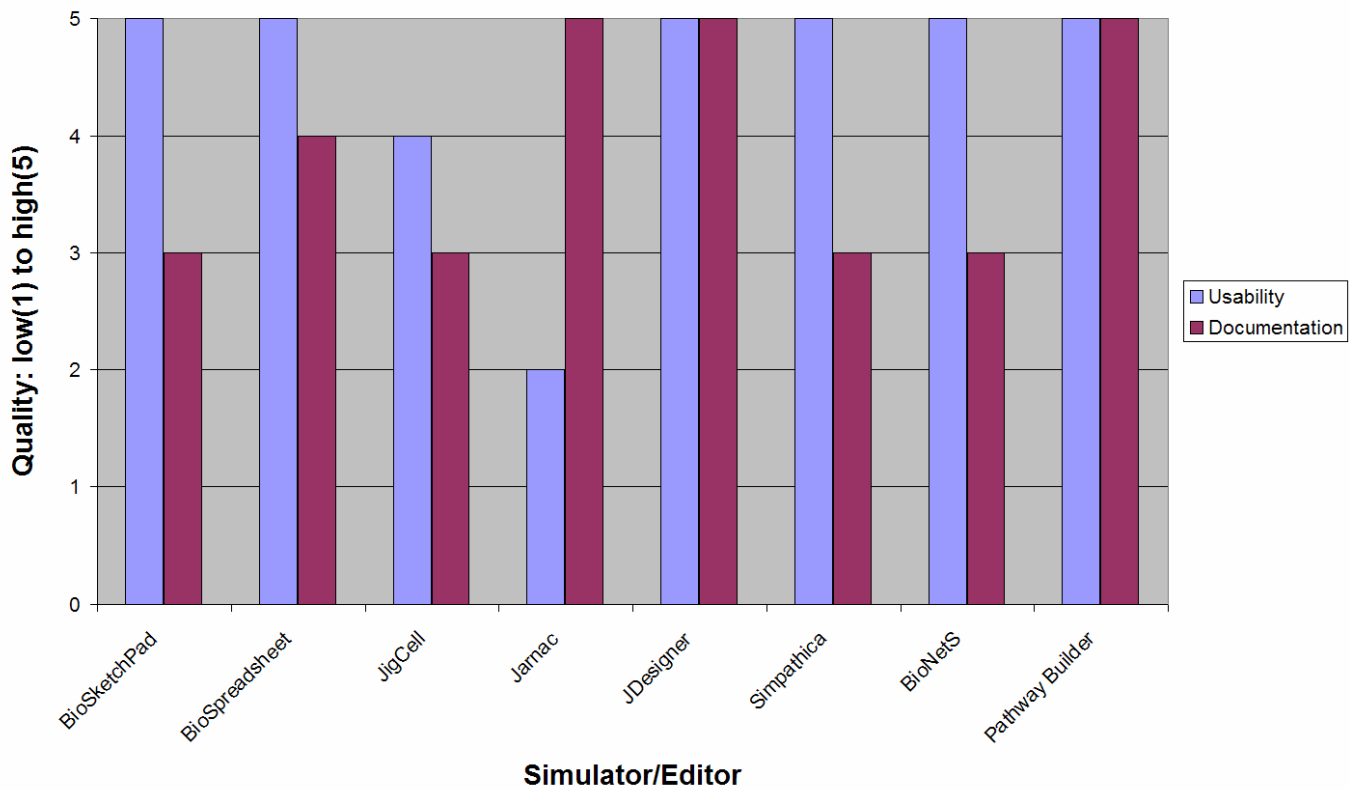


Figure 3.4. Usability and documentation of eight simulators/editors for models M1 to M5.

3.2 Dashboard Integration

The goal of the Bio-SPICE project was to develop an environment that allows different tools to interoperate in a workflow. A workflow may contain various tools in order to implement, analyze and simulate a model. The language SBML was chosen as a language of exchange between the various tools. Each simulator/editor tool, as part of a work flow, would be required to read a SBML input and produce a SBML output. The Dashboard was developed as the environment where a workflow can be implemented and the various tools incorporated. Towards the goal of evaluating the Bio-SPICE project in terms of usability we have examined the interoperability of various editors/simulators as part of a workflow in the Dashboard.

Figure 3.5 displays the Dashboard with the left pane containing the tools that are installed in the Dashboard and on the right pane, a workflow. The workflow contains three analyzers, an input document, the circadian model, the ESS solver and a display analyzer. Running the workflow causes a simulation to run and the time

course of certain variables to be displayed. At present, most solvers can not execute the SBML code exported by other tools other than their corresponding editor.

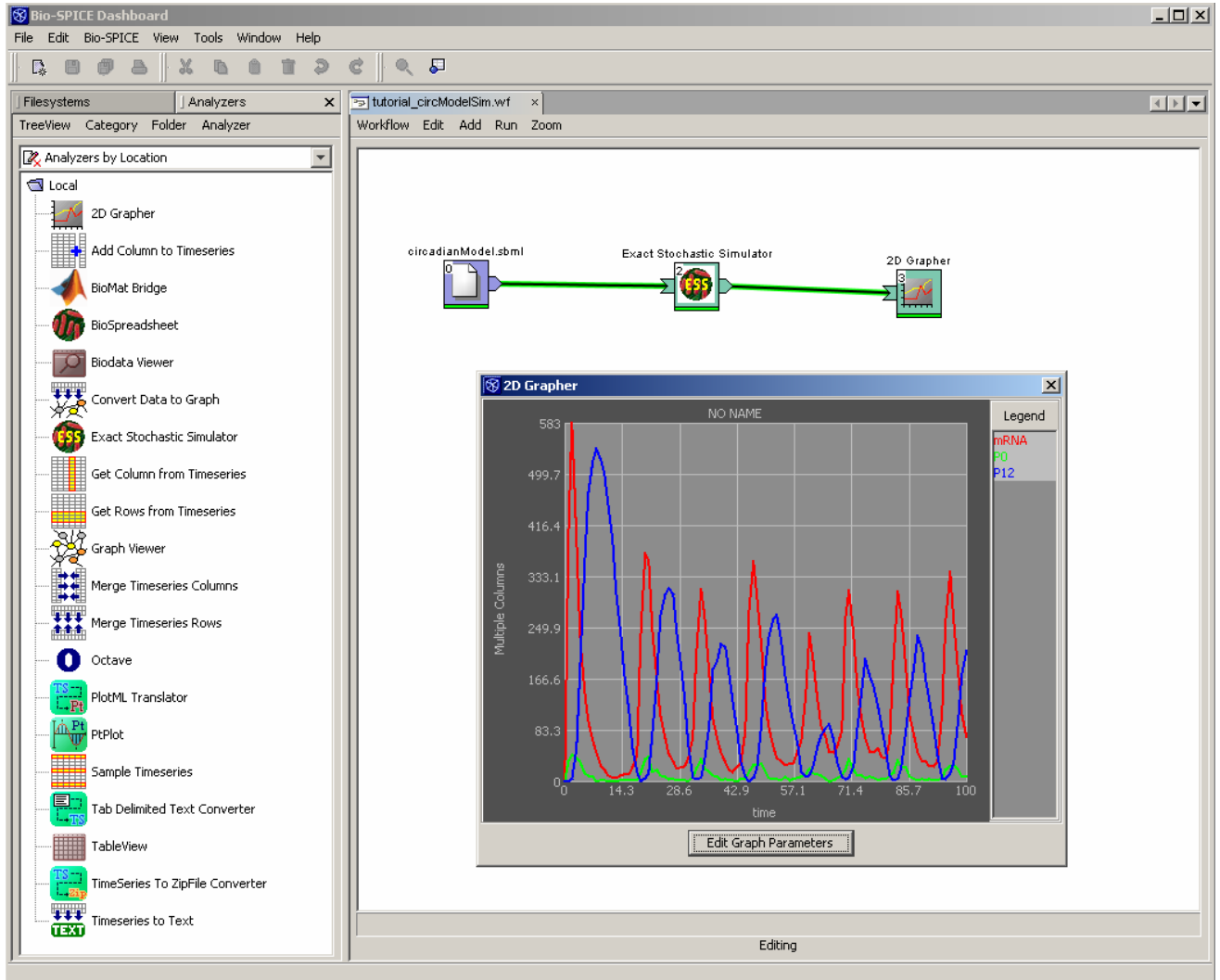


Figure 3.5. Dashboard with ESS simulation of circadian model displaying time course with 2DGrapher.

Figure 3.6 displays the Dashboard with the BioSpreadsheet editor analyzer. Running the workflow opens the editor with the document, which in this case is a circadian model.

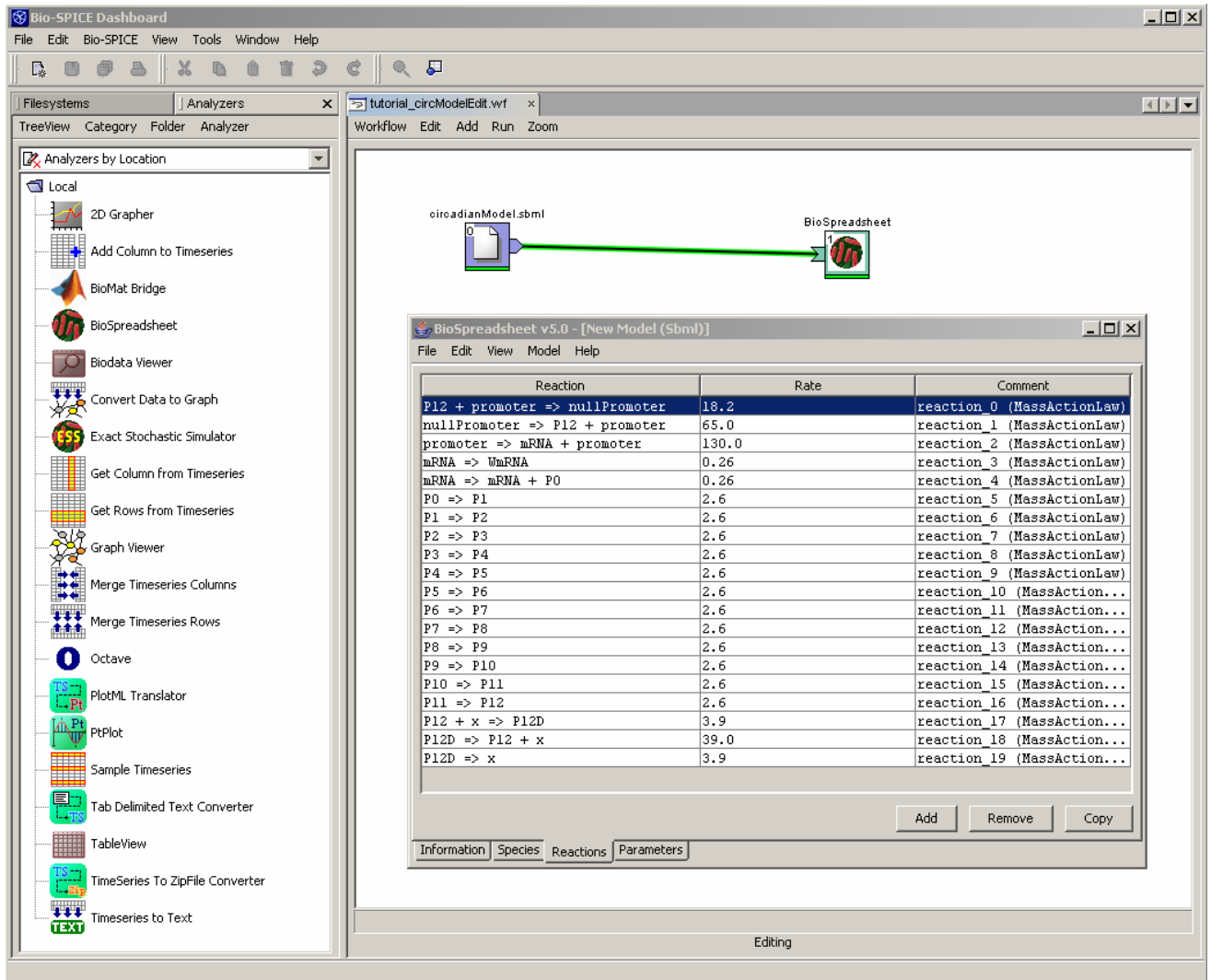


Figure 3.6. Dashboard with BioSpreadsheet editor displaying circadian model M1.

The Jarnac analyzer can also be used with the Dashboard, as seen in Figure 3.7. In the workflow, the allosteric model M4 is used as the source document. The Jarnac analyzer is connected to two output analyzers, a tabular output and a graphic output.

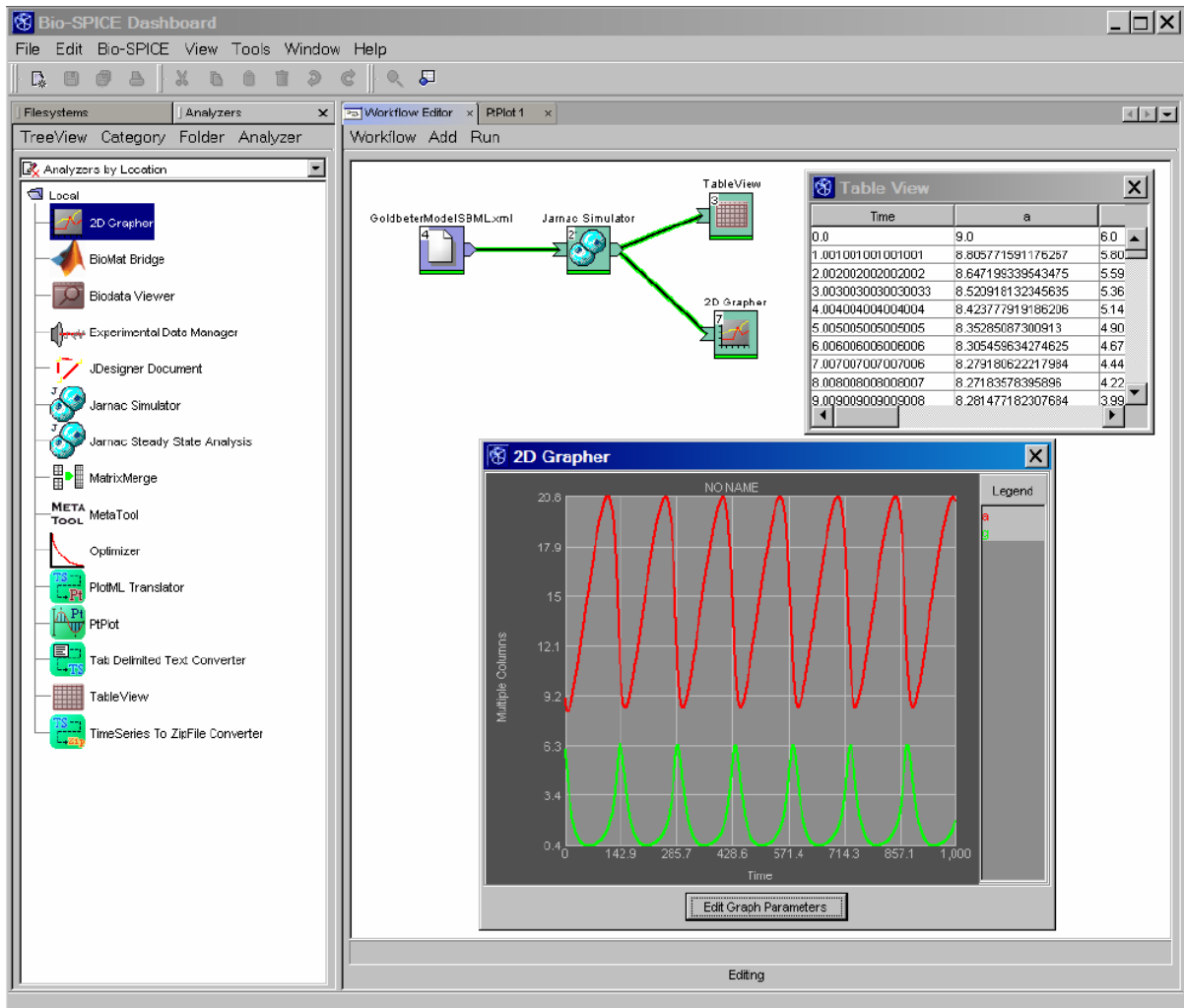


Figure 3.7. Dashboard with Jarnac simulation displaying allosteric model M4 time course and values.

3.3 Tutorial and Use Case Documentation

Two documents have been written in order to assist new users of Bio-SPICE, a Getting Started manual and a Jarnac use case document.

3.3.1 Getting Started User Manual

In order to assist new users of Bio-SPICE, a tutorial was written as a Getting Started manual. The Tutorial manual is divided into five chapters:

- *Introduction to Bio-SPICE* - provides an introduction to systems biology and an overview of the Bio-SPICE toolkit.

- *Getting Started* - describes the step to download and install the Bio-SPICE system.
- *Model Editor* - describes how to enter model equations with one of the editors, BioSpreadsheet. A simple circadian rhythm model is presented as a use case.
- *Using the Dashboard* - describes how to construct a workflow, run a simulation and display time course.
- *Bio-SPICE Tools* - provides a short description of the tools that exist in the Bio-SPICE toolkit.

The tutorial is presented in Appendix C.

3.3.2 Jarnac Use Case

The Jarnac use case presents to users how to implement a model using Jarnac and how to run a simulation using Jarnac and the Dashboard. The use case is divided into three sections:

- Download SBW and analyzer
- Modeling with Jarnac
- Dashboard workflow simulation

The use case document is presented in Appendix D.

4. Summary

The goals of this project were two-fold. The first goal was to assist developers of Bio-SPICE tools towards transparent integrations of their tools in the Dashboard. At each stage of development we reported on the status of usability and interoperability. The second goal was to help users by providing performance evaluations of the various simulation tools, illustrating the capabilities of the different tools. As well, a Getting Started manual was also written to assist new users of Bio-SPICE to develop models and use Bio-SPICE tools.

Five models were used in the usability study: a circadian rhythm model (M1), M1 scaled to 100 equations (M2), a cell division cycle model (M3), an allosteric model for glycolytic oscillations (M4) and a memory induction model (M5). We examined

seven simulators/editors for capability, graphic functionality, usability and interoperability. In order to test the model building component of each tool, which varies from using a script-like language (Jarnac), tabular editor (Simpatica, BioNetS, JigCell, BioSpreadSheet) and a GUI (BioSketchPad, JDesigner, Pathway Builder), we implemented five models (M1, M2 is M1 scaled to 100 equations, M3, M4, M5) in all six simulators (non-stoichiometric models M4 and M5 were only tested with general solvers, Jarnac and JigCell). The solvers were used to simulate the models. We tested performance at various time steps and examined scalability. The Dashboard was used with two of the simulator analyzers, BioSpreadsheet/ESS and Jarnac.

Summary of the evaluations can be found in Appendix A. The simulator BioNets was the fastest, for all three stoichiometric models. Only JigCell and Jarnac, as general solvers, were able to execute all five models. The simulator JigCell performed best in terms of interoperability. Graphic functionality varied but most simulators provide multiple copy/paste capability. The documentation of each tool was also evaluated, and had improved towards the end of the project cycle.

Based on our evaluations, users can examine our results before investing time and effort with a tool to decide what may be best suited for their needs.

5. References

Goldbeter A, Lefever R (1972) Dissipative structures for an allosteric model. Application to glycolytic oscillations. *Biophys J*, 12: 1302-1315.

Pettigrew D, Smolen P, Baxter DA, Byrne JH (2005) Dynamic Properties of Regulatory Motifs Associated with Induction of Three Temporal Domains of Memory in Aplysia . *J Comp Neurosci* 18: 163-181.

Smolen P, Baxter DA, Byrne JH (2004) Mathematical Modeling and Analysis of Intracellular Signaling Pathways. In *From Molecules to Networks-An introduction to cellular and molecular neuroscience* (JH Byrne and JL Roberts Eds.), pp. 391-430. Elsevier Academic Press, Amsterdam.

Tyson JJ (1991) Modeling the cell division cycle: cdc2 and cyclin interactions. *Proc Natl Acad Sci USA*, 88: 7328-7332.

Appendix A – Simulator Capabilities and Features

Simulator Capability					
Simulator	Type	Model Type	Performance (steps 1.0 / 0.1 / 0.01)	Accuracy (%) (steps 1.0 / 0.1 / 0.01)	Scalability
BioSketchPad / Charon	Deterministic	Stoichiometric	M1: > 1hr M2: NA M3: 3 hrs M4: NA M5: NA	M1: Fixed step of 0.001 was too large M3: Must use a step size smaller than 0.005.	NA
BioSpread-Sheet	Stochastic	Stoichiometric	M1: 0.9 sec M2: 3.7 sec M3: 308 sec M4: NA M5: NA	M1: 600 Molecules M2: Same as M1 M3: 1200 Molecules M4: NA M5: NA	For M3, examining 10, 100, and 1000 molecules, only 1000 molecules provided a reasonable solution.
Jarnac / JDesigner	Deterministic	Ja: General, JD: Stoichiometric	M1: 1.4 / 5.3 / 50.7 s M2: 28.9 / 196.2 / 1742 s M3: 0.9 / 2.9 / 19.5 s M4: 0.9 / 4.4 / 39.3 s M5: 75 / 107 / 376 s	M1: 0.004/ 0.001/ <10 ⁻⁴ M2: 0.00003/ <10 ⁻⁵ / <10 ⁻⁵ M3: 0.005/ 0.0025/ <10 ⁻⁴ M4: <0.02 / <0.02 / <0.02 M5: <0.0003 / 0.01? / 0.0003	Did not scale linearly with number of equations, M1 contained 20 equations and M2 contained 100 equations. M5: 0.1 showed larger error than 1.0.
JigCell	Deterministic	General	M1: 1.1 / 3.2 / 13.8 s M2: 4.3 / 35.1 / 132 s M3: 0.8 / 2.3 / 14.2 s M4: 0.9 / 3.7 / 31.6 s M5: 8 / 53 / NA s	M1: 0.3 / 0.01 / <10 ⁻⁴ M2: 0.0001/ <10 ⁻⁵ / <10 ⁻⁵ M3: 0.005 / 0.004 / <10 ⁻⁴ M4: <0.02 / <0.02 / <0.02 M5: <0.0003 / <0.0003 / NA	Scaled linearly with number of equations. M5: could not simulate at 0.01
Simpathica	Deterministic	Stoichiometric	M1: 36 s M2: 191 s M3: 28 s M4: NA M5: NA	Adaptive Error (SLODE Solver)	Scales Linearly
BioNetS	Stochastic	Stoichiometric	M1: 0.219 s M2: 1.026 s M3: 0.203 s M4: NA M5: NA	NA	M2 executed 5 times slower than M1. Only measurable at a small step size.

Simulator / Tool Usability and Interoperability

Simulator	Version	Import SBML	Documen-tation	Dashboard Enabled	Wish List	Usability	Issues / Comments
BioSketch-Pad / Charon	BSP: 2.2 C: 1.0	BSS: 0% JD: 85% JC: 0%	Average	No Analyzer Provided	Group node / edges for copy / paste	Easy / Intuitive	Charon requires a very small integration step which causes the execution to be extremely slow.
BioSpread-Sheet/ESS	4.0	BSP: JD: 100% JC: 50% Sim: 85% Path: 100%	Average	Runs BSS models with ESS analyzer	Move model editor rows up or down	Easy / Intuitive	
Jarnac / JDesigner	Ja: 2.14 JD: 1.952	BSP: 0% BSS: 100% JC: 75% Sim: 85% Path: 100%	Good	Runs JD model with Ja analyzer runs BSS model but incorrect results	JD: Group node/edges for copy/paste	Ja: Requires Programming JD: Easy / Intuitive	
JigCell	6.0	BSP: 90% BSS: 0% JD: 100% Sim: 100% Path: 100%	Average	JigCell analyzer not providing output	Simpler runManager	Intuitive editor	
Simpathica	1.3	BSS: 0% JD: 10% JC: 25% Path: 25%	Average	No	Simpler Model Editing Ability	Easy / Intuitive	Does not retain species name in SBML output, instead saves ID name.
BioNetS	2.0.3	NA	Average	Has Separate Dashboard Module	SBML Importing and Exporting	Easy / Intuitive	
Pathway Builder	0.9.56	BSS: 100% JD: 0% JC: 100% Sim: 100%	Good	No Analyzer Provided	Laid Out Model When Importing SBML	Easy / Intuitive	Does not layout model when importing SBML

Simulator / Tool Graphic Functionality

Simulator	Model Display	Print Model	Print/Save Output Data	Graphic Plot Support	Access to Output Data	Model Editing
BioSketchPad / Charon	GUI	No	No	None	None	Single Node Copy / Paste
BioSpread-Sheet	Text	No	uses Dashboard	uses Dashboard	Uses Dashboard	Multiple Row Copy / Paste, but inserts only as last row entry
Jarnac / JDesigner	Ja:Text JD:GUI	Ja: Yes, JD: Yes	Ja: No JD: Yes	Good	Ja: Viewing Only JD: Full Access	Ja: Full Featured Editor JD: Single Node Copy / Paste
JigCell	Text	No	No	Fair	Viewing only	Multiple Row Copy / Paste
Simpathica	Text / GUI	Yes	Yes	Gnu Plot	Full Access	Individual Entry Updating, No Copy / Paste
BioNetS	Text	No	From Matlab	Good	From Matlab	Single Cell Copy / Paste
Pathway Builder	GUI	Yes	NA	NA	NA	Single Node

Appendix B - Interoperability

BioModels Database

In an effort to promote the SBML format as a standard for biological modeling, a database of published models was assembled by the BioModels Database (<http://www.ebi.ac.uk/biomodels/>). The BioModels Database allows biologists to submit, find information on and retrieve a variety of models. For this model compilation to be useful, simulators must have the ability to read and execute the SBML formatted models. 45 models in the database were tested with the Bio-SPICE simulators BioSpreadSheet, JigCell, JDesigner, and Simpathica. Between the different simulators and models, there were varying degrees of success of readability and execution. For each simulator, models are rated on the following scale: (1) model cannot be read by the simulator, (2) simulator fails to read large portion of the model, (3) some missing parameters, (4) model loads but doesn't integrate properly, and (5) the model loads and integrates.

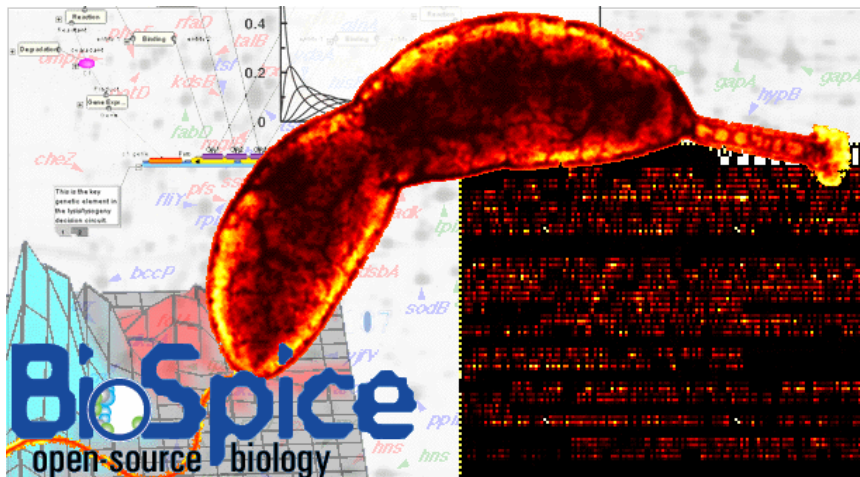
BioModel #	JigCell	JDesigner	BioSpreadSheet v4	Simpathica
01	4	1	1	1
02	5	2	1	1
03	5	1	3	1
04	5	1	3	1
05	3	5	3	1
06	3	5	3	1
07	2	1	1	1
08	3	5	3	1
09	3	5	3	1
10	5	1	3	1
11	5	5	3	1
12	4	5	3	1
13	4	1	3	1
14	4	5	3	1
15	3	5	1	1
16	3	5	1	1
17	5	3	3	1
18	3	1	1	1
20	1	1	1	1
21	5	3	1	1
22	3	1	1	1
23	4	3	1	1
24	1	1	1	1
25	1	1	1	1
26	3	1	1	1
27	4	1	1	1
28	4	2	1	1

29	4	1	1	1
30	4	2	1	1
31	5	1	1	1
32	3	1	1	1
33	4	2	1	1
34	1	1	1	1
35	4	5	1	1
36	5	5	1	1
37	4	5	1	1
38	4	5	1	1
39	5	1	1	1
40	1	5	3	1
41	3	5	1	1
42	5	2	1	1
43	5	3	1	1
44	5	3	1	1
45	5	3	1	1

Appendix C – User Manual

Getting Started with Bio-SPICE: A Tutorial for New Users

Version 1.0



Evyatar Av-Ron and John H. Byrne
Department of Neurobiology and Anatomy
The University of Texas Medical School at Houston
Houston, TX 77030

<http://biospice.org>

Chapter C-1 Introduction to Bio-SPICE	45
Introduction to systems biology	45
Modeling biochemical and gene networks	45
Boolean methods	45
Differential equation-based models.....	45
Stochastic.....	44
Deterministic	44
The Bio-SPICE project.....	46
The Bio-SPICE web site and SourceForge.net.....	47
The Bio-SPICE tool kit.....	45
Outline.....	48
Chapter C-2 Getting Started.....	49
The Dashboard.....	49
System Requirements.....	49
Bio-SPICE on SourceForge.net.....	49
Installing the Dashboard from SourceForge.....	50
Chapter C-3 Model Editor.....	53
Choosing an editor.....	53
Downloading an editor.....	53
Using an editor	56
Implementation of a model	56
Entering Model in BioSpreadsheet.....	59
Chapter C-4 Using the Dashboard	62
Dashboard description	62
Dashboard workflow	62
Running the Dashboard.....	63
Installing UTK-ORNL tools	64
Opening a source file.....	65
Building a workflow	66
Running a simulation	69
Updating a file.....	71
Installing Additional Tools in the Dashboard	72
Importing an Analyzer	74
Chapter C-5 Using Bio-SPICE Tools.....	75
Appendix C-1: Bio-SPICE Components List.....	76
Appendix C-2: Bio-SPICE Usability Testing	76

Disclaimer

The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government.

Acknowledgements

This material is based on research sponsored by DARPA under agreement number FA8750-04-1-0242. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

Chapter C-1 Introduction to Bio-SPICE

With the enormous growth of life science research it is clear that tools are needed to store and analyze the large amount of experimental data, to build, simulate and analyze mathematical models, and to visualize data and system dynamics. This chapter presents the software project Bio-SPICE (Biological Simulation Program for Intra- and Inter-Cell Evaluation) and a short description of system biology.

Introduction to systems biology

Systems biology is the study of the mechanisms underlying complex biological processes as integrated systems of many, diverse, interacting components. Systems biology involves (1) collection of large sets of experimental data (by high-throughput technologies and/or by mining the literature of reductionist molecular biology and biochemistry), (2) proposal of mathematical models that might account for at least some significant aspects of this data set, (3) accurate computer solution of the mathematical equations to obtain numerical predictions, and (4) assessment of the quality of the model by comparing numerical simulations with the experimental data. (<http://jigcell.biol.vt.edu/glossary.html>)

Modeling biochemical and gene networks

There are several methods of modeling system biology, in particular cellular processes and biochemical interactions. Two very different approaches are Boolean networks and ordinary differential equations (ODEs). For qualitative modeling of large systems, Boolean networks may be more appropriate than an ODE-based model where details of reaction rates are required.

Boolean methods

Qualitative modeling of large systems can most effectively be carried out using a Boolean method. There are several variants of Boolean methods, e.g., PetriNets and Cellular Automata. A Boolean network contains nodes that can have the value 1 (on) or 0 (off). For a specific node, the change from one state to the next is a function of the nodes that are connected to the specified node.

Differential equation-based models

Quantitative modeling is usually carried out with ordinary differential equations, which represent system variables that change as nonlinear functions of other variables and/or parameters. To describe mass action of biochemical reactions, equations may be written in stoichiometric form. There are two general types of solvers for quantitative models, stochastic and deterministic. When small populations of molecules are considered, stochastic modeling is more appropriate since molecular fluctuations may alter the

dynamics. With large population sizes, deterministic modeling will be computationally faster as stochastic modeling of large populations is computationally time intensive and should produce results similar to an ODE model.

Stochastic

Stochastic solvers use the Gillespie algorithm or a variant of it. Basically, all possible reactions are examined and the reaction with the shortest time interval is “scheduled”. The executed reaction will affect the population of molecules and thereby other reactions. Therefore certain reaction times must be recalculated, and again, the shortest time interval reaction is scheduled. This algorithm was shown to describe the time evolution of a chemical system.

Deterministic

Deterministic solvers use a variety of algorithms to compute the value of the variables in a model. At each time step, all variables are calculated based on either only previous values (explicit methods) or incorporating estimates of the next value (implicit methods). Some of the more known algorithms are forward and backward Euler methods (explicit and implicit respectively), Runge Kutta, Gear, CVODE and Crank-Nicolson. The algorithms differ in terms of ease of use (e.g., explicit methods requiring only initial conditions, implicit methods require estimations of variable values), speed of computation (how many function calls per time step) and accuracy especially important in cases of stiff systems.

The Bio-SPICE project

The Bio-SPICE project was started as part of the Bio-ComputationP, funded by the Defense Advanced Research Projects Agency (DARPA). The goal was to develop a computational framework that enables the construction of sophisticated models of intracellular processes that can be used to predict and control the behavior of living cells. In addition, Bio-SPICE is also being examined to generate new computational paradigms and engineering applications that utilize biomolecules as an information processing, sensing, or structural components (<http://www.darpa.mil/ipto/programs/biocomp/index.htm>). In order to understand cellular behavior, we need to understand how the underlying genetic code is executed and to characterize the dynamics of cellular events (see editorial at <http://www.liebertonline.com/toc/omi/7/3>).

The Bio-SPICE project chose the System Biology Markup Language (SBML) as a language of exchange between the different tools. SBML is a computer-readable format for representing **models of biochemical reaction networks**. For example, SBML is applicable to metabolic networks, cell-signaling pathways and regulatory networks. For further details see the web site <http://sbml.org/index.psp>.

The Bio-SPICE web site and SourceForge.net

The Bio-SPICE web site <https://biospice.org/index.php> provides information and software downloads to the Bio-SPICE community.



Figure C-1.1 Bio-SPICE web site.

In order to download software from the web site, you must join and become a Bio-SPICE member. Chapter 2, Getting Started describes how to go about joining.

A second source now exists for the Bio-SPICE project is on SourceForge.net, <http://sourceforge.net/projects/biospice>. The latest version of the Dashboard can be downloaded from SourceForge.net without the need to register as a Bio-SPICE user as on the official Bio-SPICE web site. This reflects the transition from a DARPA funded program to a truly open source environment to fulfill the desire to see Bio-SPICE continue to mature and evolve long after the DARPA funding has ended.

The Bio-SPICE tool kit

The Bio-SPICE tool kit is comprised of the Dashboard and a range of tools.

- Dashboard: GUI application to create and run workflows.
- Data analysis tools: Tools to mine the data
- Database tools: The large amount of data that experiments produce needs to be stored and mined.
- Model analysis tools: A model can provide information through various means of analysis, e.g., bifurcation, parameter sensitivity.
- Model composition & visualization tools: In order to construct a model, tools are provided for model composition, as well as visualization.
- Simulator tools: Models need to be solved using various types of simulators, continuous ODE simulators or stochastic simulators.

Outline

The manual is organized as follows:

- Introduction – Chapter C-1 provides a general overview of System Biology and the Bio-SPICE software.
- Getting Started – Chapter C-2 describes how to download the Bio-SPICE package and launch the application.
- Model Editor – Chapter C-3 presents a simple use case and describes how to edit a model.
- Using the Dashboard – Chapter C-4 shows how to use the Dashboard to simulate the model described in Chapter C-3.
- Using Bio-SPICE Tools – Chapter C-5 provides a brief summary of all the Bio-SPICE tools available.

Chapter C-2 Getting Started

This chapter presents the Dashboard application, and shows how to download and install the Dashboard.

The Dashboard

The Bio-SPICE Dashboard is an environment for integrating varied data and tools useful to biologists. The main categories of tools this environment was designed for are model building, model analysis, experimental data analysis, and visualization. However, conceivably any tool that analyzes, transforms, produces, or helps to interpret data could be integrated into the Dashboard. The Dashboard functions in a manner loosely analogous to UNIX shells (especially with respect to UNIX pipe facilities).

The Dashboard is based on the NetBeans application platform, a Java-based tool kit. Tools may be written in any language, however, as the core Dashboard libraries provide support for accessing non-Java tools that are either network enabled, command-line tools, or OAA agents.

Currently, facilities for Java and OAA tools are available, as well as support for TCP/IP, web, and command-line tools via an XML wrapping API.

Users or tool developers interested in integrating their tools into the Dashboard environment can find additional guidance in the Developer's Manual.

System Requirements

Java SDK version 1.4.5. Download from <http://java.sun.com>.

Linux users: It may be necessary to have super-user permission (i.e., root) before you install the software. Check with your local system administrator.

Bio-SPICE on SourceForge.net

The Bio-SPICE software may be downloaded from the SourceForge.net site at <http://sourceforge.net/projects/biospice/>.

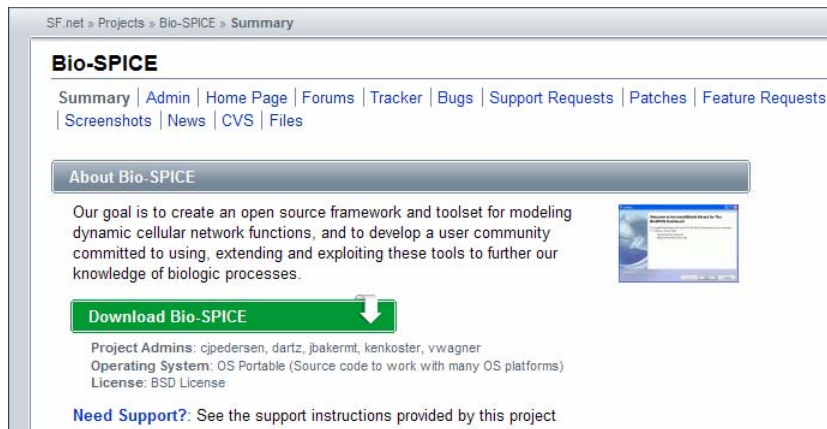


Figure C-2.1 SourceForge Dashboard page.

Click on the green button `Download Bio-SPICE`, see Figure C-2.1, and you will be directed to the download page. There are three versions of the software for the operating systems: Windows (32bit), Linux (32bit) and Mac (PPC), shown in Figure C-2.2.

File Releases


Below is a list of all files released by this project. Before downloading, you may want to read Release Notes. The current release for each package is shown.

Package	Release (date)	Filename	Size (bytes)	Downloads	Architecture	Type
Dashboard						
Latest	Dashboard 7.0 [Notes] (2005-11-07 16:52)					
		Dashboard-mac-7.0.zip	13402477	6	PPC	.zip
		Dashboard-SetupLinux-7.0.bin	79920321	6	i386	Other Binary Package
		Dashboard-SetupWindows-7.0.exe	70822043	31	i386	.exe (32-bit Windows)
	Dashboard 6.0 [Notes] (2005-02-15 16:53)					
Totals:	2	6	325126102	43		

Figure C- 2.2 SourceForge Dashboard download page.

Installing the Dashboard from SourceForge

There are three versions of the dashboard, for Windows, Linux and Macintosh operating systems. Select the version you need. Then save either the Windows file `Dashboard-SetupWindows-7.0.exe` or the Linux file `Dashboard-SetupLinux-7.0.bin` or the Macintosh file `Dashboard-mac-7.0.zip` on your computer.

The installer will have an icon which looks like this,  for MS Windows, double-click on the icon to launch the installer, see Figure C-2.3.

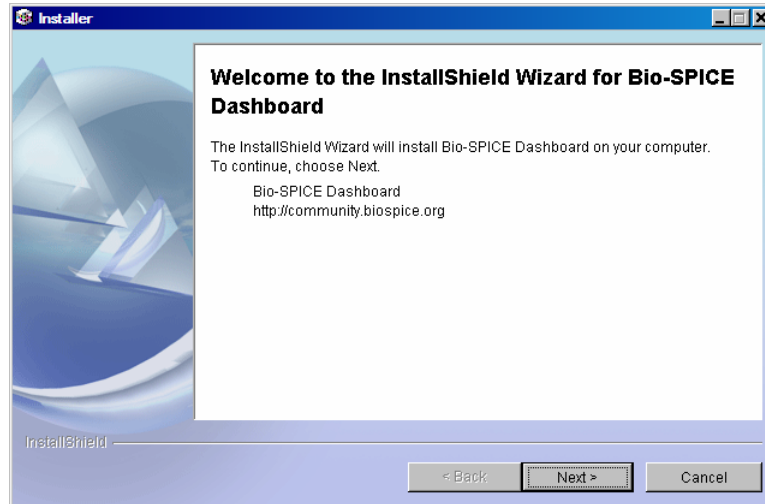
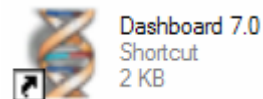


Figure C-2.3 Dashboard InstallShield Wizard.

Note: In order for the Dashboard to function you must have a Java VM installed, of a version 1.4 or later. If you do not have Java installed or, have an older version, visit the Java site <http://java.sun.com> in order to download the appropriate Java VM.

The installer will provide a default folder/directory for installations, but you may change the destination folder/directory. Click the button `Browse` to open a dialog box displaying the file structure for your system. Traverse the structure to the folder in which you want to install Bio-SPICE. Select the folder name and click the button `open` and then `next`. The installer will then display the installation files. Leave the two check marks for installation of both the Dashboard and OAA for a complete install. Click the button `next` twice to start the installation. After installation click the buttons `next` and then `finish`.

The installation will place four folders, `_jvm`, `_uninstall`, `Dashboard`, and `OAA` in the designated folder, and place three shortcuts, to the Dashboard, to uninstall the Dashboard, and to the OAA Facilitator, on the Desktop.



To launch the Dashboard, double-click on the Dashboard shortcut. In the next chapter we will show how to use the Dashboard to construct a biochemical model.

The Bio-SPICE documentation, shown in Figure C-2.4, has been integrated into the Dashboard. Click on the menu item `Help>Help Contents` in order to open the documentation window.

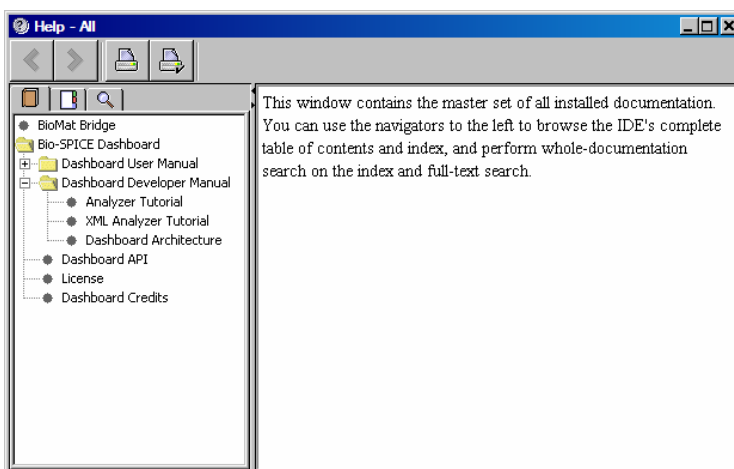


Figure C-2.4 Access the Dashboard manual from Bio-SPICE web site.

Chapter C-3 Model Editor

This chapter presents how to implement a model with Bio-SPICE. We will use the model editor *BioSpreadsheet* to construct a biochemical model.

Choosing an editor

There are several model editors available in Bio-SPICE. Each simulator provides an accompanying editor, which is best suited for use with the specific simulator. On the other hand, each editor provides an export function that allows you to save the model in SBML format. SBML is the language of exchange for models and each simulator provides you with an import function to read SBML models.

Note: At present there is not full interoperability between all editors and simulators. The import function of one tool may not read correctly the entire SBML model from another tool. See Appendix B for a table of interoperability of Bio-SPICE tools.

Simulator contributions to earlier versions of Bio-SPICE all included editors for model construction, due to the fact that no standard data format was established. As the need for interoperability expanded, SBML was chosen as a language of exchange for the various tools. All editors provide an SBML import/export function. One of the original contributors to Bio-SPICE was a team from the University of Tennessee at Knoxville (UTK) and Oak Ridge National Lab (ORNL) which provided the editor BioSpreadsheet with the accompanying stochastic solver ESS.

Downloading an editor

With the move of Bio-SPICE to SourceForge.net, it is best to download Bio-SPICE tools from the web site of the contributing group, if such a site exists. A few of the contributing teams have been found to do a better job making recent tools releases and fixes available to others on their own web site than on the Bio-SPICE web site. Some tools though are only provided on the Bio-SPICE web site <https://biospice.org/>. Appendix A provides the list of tools from the various organizations and the web site most convenient to access them.

The tools from the University of Tennessee at Knoxville (UTK) and Oak Ridge National Lab (ORNL) can be found at: <http://biocomp.ece.utk.edu/>, see Figure C-3.1.

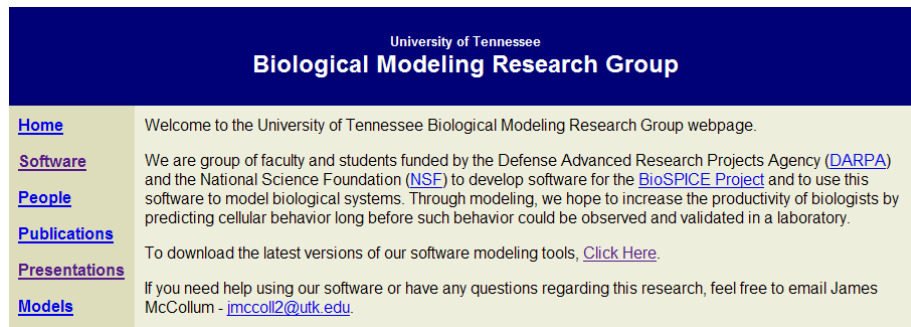


Figure C-3.1 University of Tennessee Biological Modeling Research Group webpage.

The editor BioSpreadsheet is a simple-to-use editor for developing models of mass action equations. It was developed in conjunction to the stochastic solver ESS (Exact Stochastic Simulator) as part of the software contribution of UTK/ORNL.

Click the `Software` link to open the download page <http://biocomp.ece.utk.edu/tools.html>. Click the link `Download Now` and save the file `utkornltools.zip` to disk. Unzip the file `utkornltools.zip` to install the folder `utkornltools`, which contains the following structure:

The unzipped `utkornltools` folder will look something like the image in Figure C-3.2.

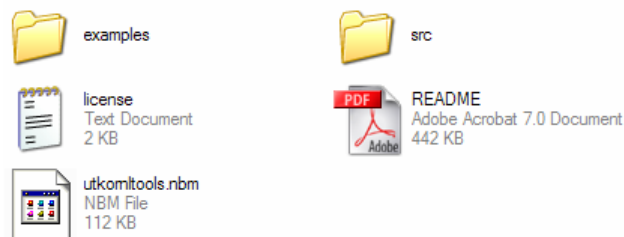


Figure C-3.2 Folder `utkornltools`.

The PDF file `README`, on page 3, describes how to install the UTK-ORNL tools in the BioSPICE Dashboard. Briefly, the file `utkornltools.nbm` must be installed by using `Install Manually Downloaded Modules` from the menu `Tools>Update Center`. Manual install is used when the tool file resides on your hard drive. Automatic install is used to download the tool file from the update center. Relaunch the Dashboard to have access to the UTK-ORNL tools. The Dashboard should look as in Figure C-3.3.

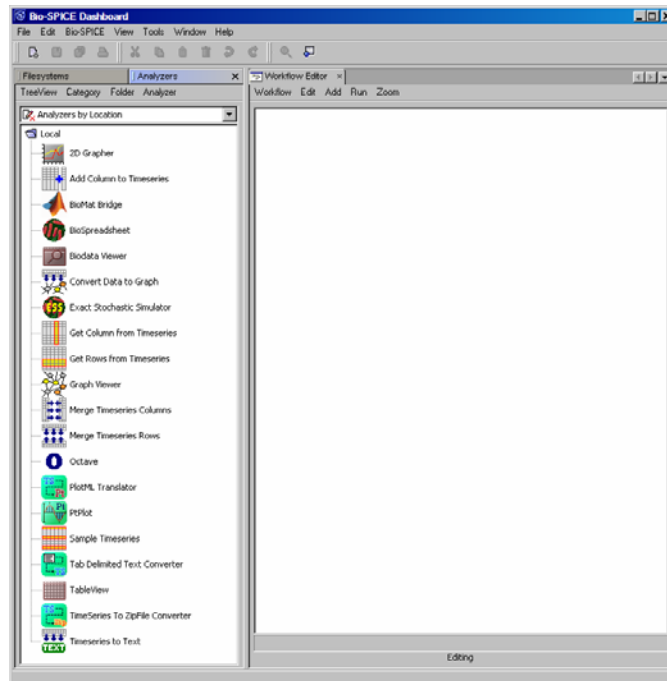


Figure C- 3.3 Dashboard with UTK-ORNL tools installed.

To launch the BioSpreadsheet editor, you must first place the BioSpreadsheet analyzer on a workflow. Double-click the BioSpreadsheet analyzer on the left pane of the Dashboard and drag the cursor, while holding the left mouse button, to the workflow area on the right pane of the Dashboard.

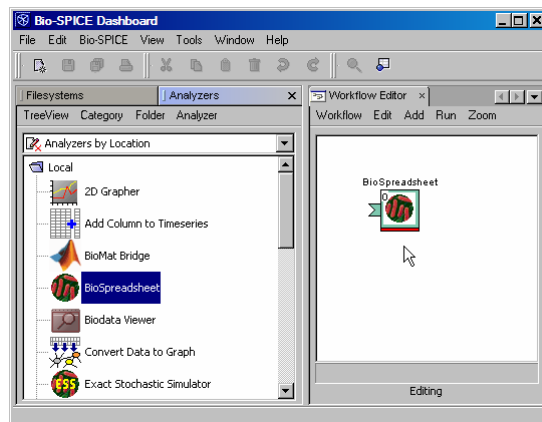


Figure C-3.4 BioSpreadsheet analyzer in workflow editor.

Click the menu `Run>Start` in order to start BioSpreadsheet which should open a window as in Figure C-3.5.

Using an editor

The BioSpreadsheet editor has four panes, each associated with a tab Information, Reactions, Species and Parameters, and its corresponding panel.

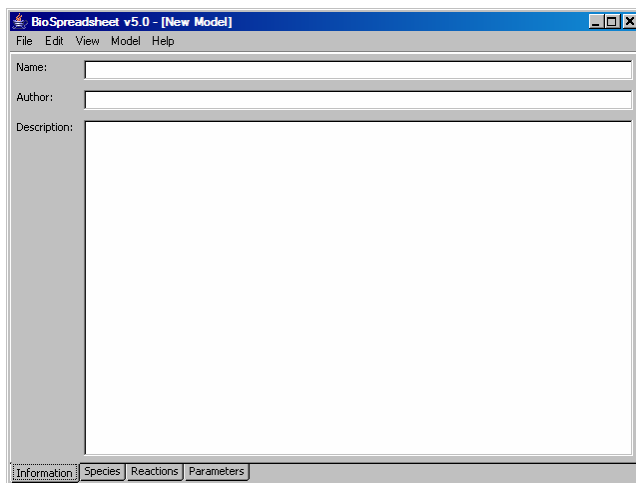


Figure C-3.5 BioSpreadsheet application window.

A model in BioSpreadsheet is composed of the model name, the species or variables, the list of reactions and possible parameters. By selecting the desired tag, the appropriate pane will be displayed.

Implementation of a model

In order to demonstrate the BioSpreadsheet editor, a circadian rhythm model will be implemented. This model corresponds to the oscillations in the levels of core gene expression due to negative feedback. The model uses a transcription factor (TF) which undergoes multiple phosphorylation steps. Over the space of a day, TF proteins becomes fully phosphorylated and relieve TF repression so that another "burst" of TF transcription can occur, see Figure C-3.6.

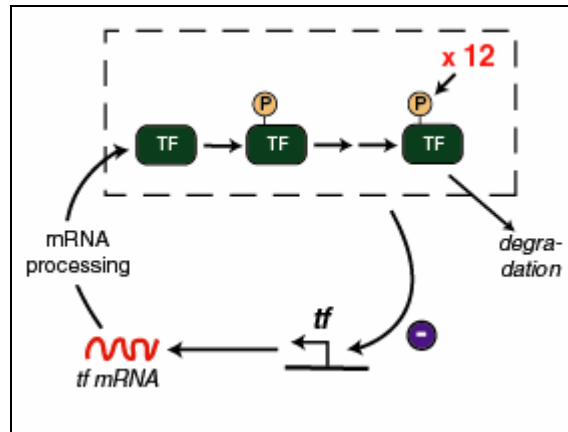


Figure C-3.6 Diagram of Circadian model.

Since the BioSpreadsheet editor only accepts models in the form of mass action equations, the ODE model of Figure C-3.7 must be converted to mass action form.

$$\frac{d[\text{mRNA}]}{dt} = v_R * \frac{K_R}{K_R + [\text{TF}_{12}]} - k_d[\text{mRNA}]$$

$$\frac{d[\text{TF}_0]}{dt} = k_p[\text{mRNA}] - k_{ph}[\text{TF}_0]$$

$$\frac{d[\text{TF}_i]}{dt} = k_{ph}[\text{TF}_{i-1}] - k_{ph}[\text{TF}_i] \text{ for } i = 1 \dots 11$$

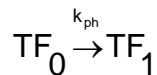
$$\frac{d[\text{TF}_{12}]}{dt} = k_{ph}[\text{TF}_{11}] - \frac{v_p[\text{TF}_{12}]}{K_P + [\text{TF}_{12}]}$$

Figure C-3.7 Circadian ODE model. Parameters: $v_r=7.0$, $K_r=0.0005$, $k_d=0.2$, $k_p=0.2$, $k_{ph}=2.0$, $v_p=3.0$, $K_p=0.0001$.

Converting a linear ODE is simple since every term represents a mass action reaction. For example, the equation above:

$$\frac{d[\text{TF}_0]}{dt} = k_p[\text{mRNA}] - k_{ph}[\text{TF}_0]$$

represents a linear production term and a linear degradation term. Two mass action equations are needed, one for each term. Of course for mass balance, the degradation term for P_0 is equivalent to the production term for P_1 .



The nonlinear terms, such as those using the Michaelis-Menten formalism are more difficult to convert into mass action form. Since the Michaelis-Menten formalism takes advantage of the quasi-steady-state approximation, this assumption which reduces the complexity of the model is not valid for a system of mass action equations, and must be expanded to its original form.

In order to unpack the equations, new variables are needed. There are two ODEs of the circadian model with Michaelis-Menten terms, the equations for mRNA and TF12. In the case of the ODE for mRNA, the Michaelis-Menten term contributes to the production of mRNA. While in the ODE for TF12, the Michaelis-Menten term is part of the removal of TF12. It is not our intent in this manual to deal with the subject of converting nonlinear ODEs to mass action equations. Figure C-3.9 provides the equations for the model of Figure C-3.7 in mass action format. Figure C-3.10 illustrates the response of the mass action equation model which is similar to the time course of the original model, shown in Figure C-3.8.

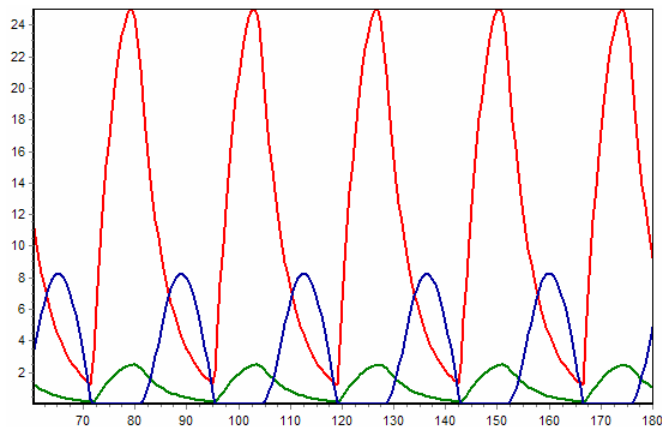


Figure C- 3.8 Circadian model oscillations of mRNA (red), P0 (green) and P12 (blue).

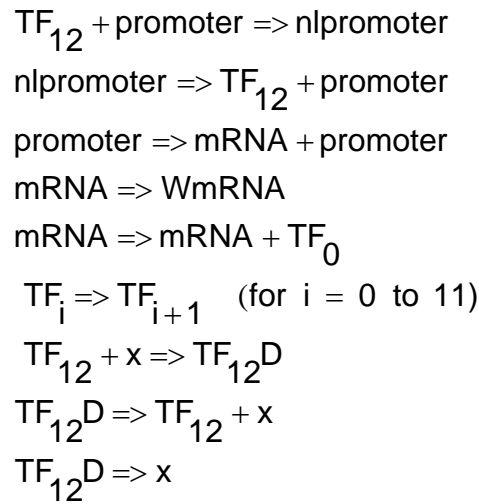


Figure C-3.9 Circadian model in mass action form. Rate values are: 18.2, 65.0, 130.0, 0.26, 0.26, 2.6 (i=0 to 11), 3.9, 39.0, 3.9.

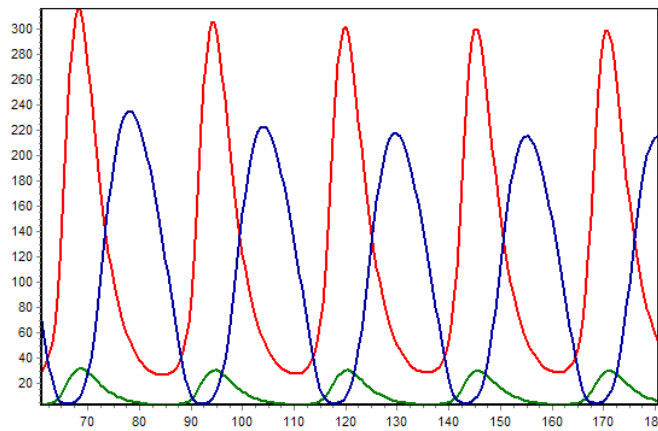


Figure C-3.10 Circadian oscillations of mass action model.

Entering Model in BioSpreadsheet

The BioSpreadsheet editor provides a Species panel for declaring the model species and a Reactions panel that is used for entering model equations.

Click the `Species` tab to open the Species panel. Click the button `Add`, to insert a blank line, as shown in Figure C-3.11.

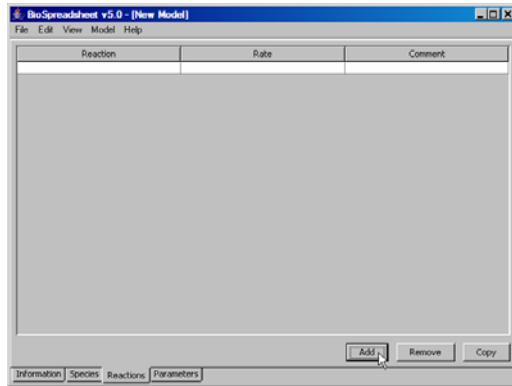


Figure C-3.11 BioSpreadsheet Reaction panel.

Each term of the mass action equations is a species that needs to be defined in the BioSpreadsheet Species panel. Only three species have initial values different from zero. The initial value of mRNA is 3, promoter is 2, and x is 10. Figure C-3.12 presents the complete species panel for the circadian model.

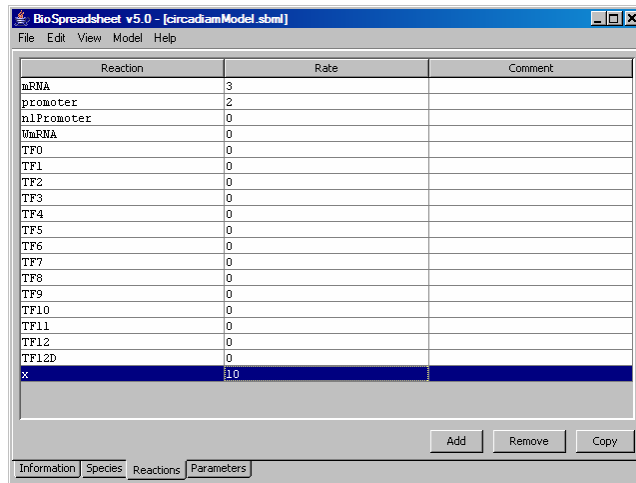


Figure C-3.12 Species panel of BioSpreadsheet editor.

The next step is to define the reactions of the model. Select the Reactions tab in the BioSpreadsheet editor. Use the button Add to insert a blank line in the panel. Select the column you want to write in. There are 20 mass action equations in the Circadian model. The final model should look like Figure C-3.12.

Reaction	Rate	Comment
TF12 + promoter => n1Promoter	18.2	
n1Promoter => TF12 + promoter	65.0	
promoter => mRNA + promoter	130.0	
mRNA => m1RNA	0.26	
mRNA => m1RNA + TF0	0.26	
TF0 => TF1	rateP	
TF1 => TF2	rateP	
TF2 => TF3	rateP	
TF3 => TF4	rateP	
TF4 => TF5	rateP	
TF5 => TF6	rateP	
TF6 => TF7	rateP	
TF7 => TF8	rateP	
TF8 => TF9	rateP	
TF9 => TF10	rateP	
TF10 => TF11	rateP	
TF11 => TF12	rateP	
TF12 + x => TF12D	3.9	
TF12D => TF12 + x	39.0	
TF12D => x	3.9	

Figure C-3.13 Mass action equations for the circadian model.

The last panel we need to modify is the panel Parameters. Since all the phosphorylation steps use the same rate constant, we can define the value as a parameter. Open the Parameter panel by selecting the Parameter tab. Click the button `Add` to insert a new blank line. Click the column with the mouse and enter the parameter `rateP`, click the Tab key and enter the value 2.6.

In order to use the model file within the Dashboard, it must be saved in SBML format. BioSpreadsheet provides an export function, in the File menu, in order to save the model in SBML format. Click `File>Export SBML` to open a Save dialog box. For our example, we have chosen the filename `circadianModel.sbml`.

To edit a SBML file, use the input SBML command to open the file in BioSpreadsheet. Click `File>Import SBML` to open a dialog box to select the desired SBML file.

Chapter C-4 Using the Dashboard

This chapter presents the Dashboard application, which is the environment for invoking Bio-SPICE tools. The tools provided by the Dashboard are referred to as analyzers, and any tool that is installed is represented by an icon in the analyzer pane. You will learn how to construct a workflow and run a simulation, as well as how to visualize the results.

Dashboard description

The Bio-SPICE Dashboard is an environment for integrating varied data and tools useful to biologists. The main categories of tools this environment was designed for are model building, model analysis, experimental data analysis, and visualization. However, conceivably any tool that analyzes, transforms, produces, or helps to interpret data could be integrated into the Dashboard. The Dashboard functions in a manner loosely analogous to UNIX shells (especially with respect to UNIX pipe facilities).

The Dashboard is based on the NetBeans application platform, a Java-based tool kit. Tools may be written in any language, however, as the core Dashboard libraries provide support for accessing non-Java tools that are network enabled, command-line tools, or OAA agents.

Currently, facilities for Java and OAA tools are available, as well as support for TCP/IP, web, and command-line tools via an XML wrapping API.

The Dashboard consists of two panes, a left pane for viewing analyzers and the file system, and a right pane for the workflow editor and output visualization. The Dashboard provides a library of tools that you can connect and configure for your needs. By default, the Dashboard contains several basic analyzers, e.g., a table viewer and a data plotter.

Dashboard workflow

The Dashboard provides an environment where you can connect data files and the various tools of Bio-SPICE. In order to connect the varied data and tools, the Dashboard provides a workflow editor. The workflow editor allows you to define source documents e.g., a SBML model file, and direct the document to a tool, e.g., an editor or a simulator. The tools can produce output data which you can visualize with a graphing tool.

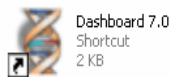
A workflow is an acyclic graph representing a high-level task that a user wishes to run. The individual parts of this task are all the nodes in a workflow, and consist of all the modules that will be run and the data they will be analyzing or producing. A workflow may contain source documents, destination documents, and analyzers. A source document represents data read from a file. Similarly, a destination document represents data being written to a file. Analyzers may have any number of inputs and/or outputs. For a workflow

to be valid, all required inputs and outputs from all nodes (documents and analyzers) must be satisfied. In addition, all source and destination documents must have a file associated with them. Analyzer inputs and outputs may be satisfied by connecting links to other documents and analyzers of matching type. In addition, some analyzer inputs (for example, text) may be satisfied by manually editing the input parameters.

In order to run the model we have developed with BioSpreadsheet, the stochastic solver ESS needs to be downloaded.

Running the Dashboard

Windows users should find shortcut links on their desktop to the Dashboard, the OAA facilitator, and the Dashboard Uninstaller. Double-click on the Dashboard shortcut icon



to launch the application.

To launch the application directly, locate the Dashboard's bin directory.

- On Windows, this is most likely: `C:\Program Files\Bio-SPICE\Dashboard[Version #]\Dashboard\bin`
- On Linux, this is most likely: `/opt/Bio-SPICE/Dashboard2.0.0/Dashboard/bin`

The executable to run the Dashboard is named "runide"

- Windows users: double-click on `runidew.exe`. There is also `runide.exe`, which starts a console window in addition to the Dashboard. Debugging and/or information messages are sometimes printed to this console window.
- Linux users: `run ./runide.sh`

By default the Dashboard opens the workflow editor on the right pane, as seen in Figure C-4.1. If you find the editor window has closed, you may open the workflow editor by clicking `Bio-SPICE>Open Workflow Editor`. You may open several workflow windows and use the workflow editor tab to select the desired workflow.

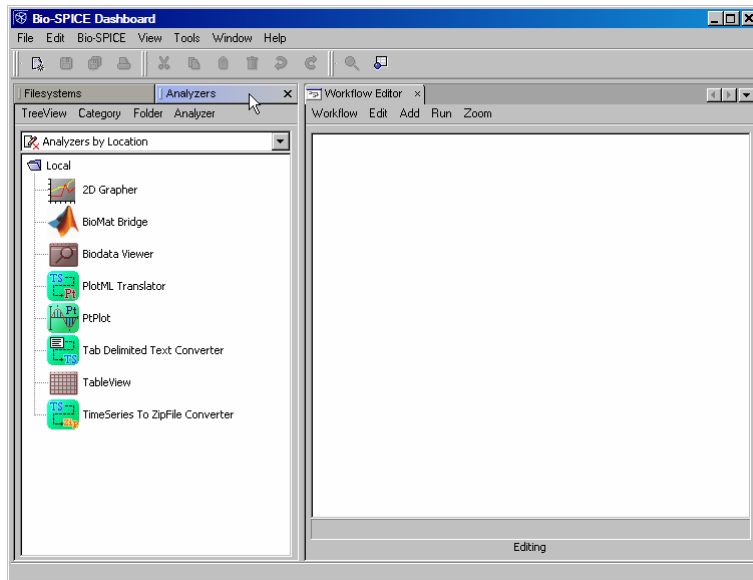


Figure C-4.1 Default Dashboard with basic set of Analyzers.

Installing UTK-ORNL tools

In the previous chapter we briefly described how to install the tools from UTK-ORNL in the Bio-SPICE Dashboard. The UTK-ORNL documentation describes the procedure on page 3 of the README document.

Briefly, the file `utkornltools.nbm` must be installed by using `Install Manually Downloaded Modules` from the menu `Tools>Update Center`. After relaunching the Dashboard, the UTK-ORNL tools will appear in the analyzer pane on the left.

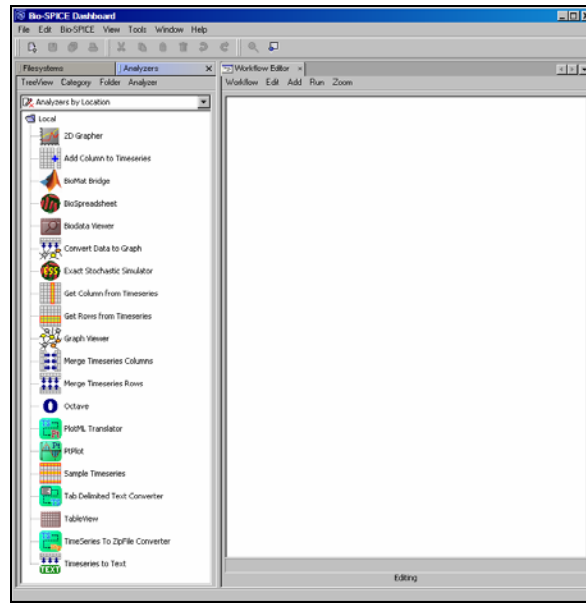


Figure C-4.2 Dashboard with UTK-ORNL tools installed.

Opening a source file

The first step in creating a simulation workflow is to provide input data for the simulation. This is called a source document. Click the menu `Add>Document>Source` as shown in Figure C-4.3.

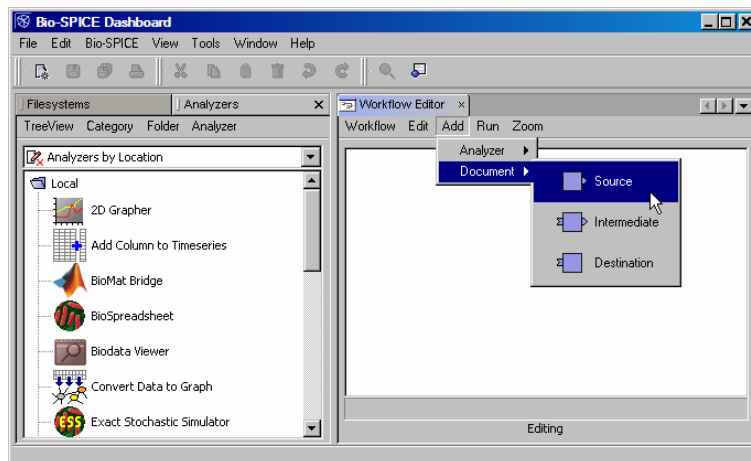


Figure C-4.3 Add a source document to the Workflow editor.

The source document icon will appear in the workflow editor pane (see Figure C-4.4). Click the document icon and right-click to open the pull-down menu. Click the item `edit`,

as shown in Figure C-4.4. A dialog box will open which allows you to locate the file to load as the source document.

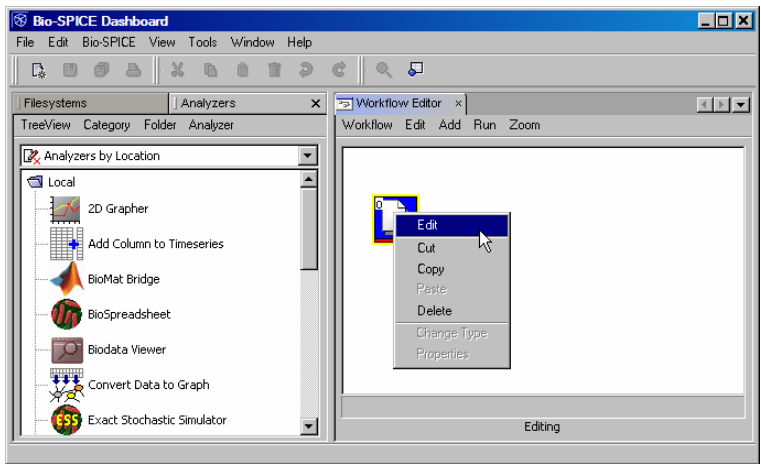


Figure C-4.4 Edit the source document to select the input file.

Select the source file `circadianModel.sbml` and the file name will appear above the source document icon, as shown in Figure C-4.5.

Building a workflow

Constructing a workflow entails connecting the building blocks from an initial source document to a final output document or plot. There are two ways to add a component box to the workflow. Using the button `Add`, you may incorporate a source document or an analyzer. As well, the analyzers presented in the component pane can be selected and dragged onto the workflow pane. In order to connect two components, select the right side of the leading component and extend the black line to the left side of the second component. The Dashboard will allow you to connect two components that are designed to be joined in a workflow, otherwise, attempting to connect two components that are not designed to be connected will fail.

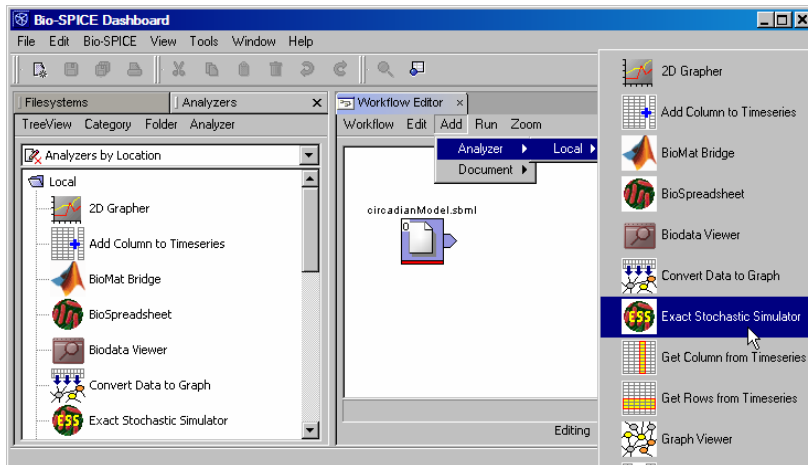


Figure C-4.5 Insert a BioSpreadsheet analyzer in the workflow.

The source document is attached to the analyzer by connecting the output socket (rightward protruding arrow head) to the input socket (leftward protruding arrow tail) of the analyzer. Right-click the arrow head and hold the button as you draw a line to the arrow tail, as seen in Figure C-4.6. Once the connection is established the line will remain when you release the button. Double-click the analyzer to open the analyzer parameter box to confirm the input source document format. When this is done, the red line on the bottom of the source document icon will turn to green, meaning the connection is established and verified, as seen in Figure C-4.7.

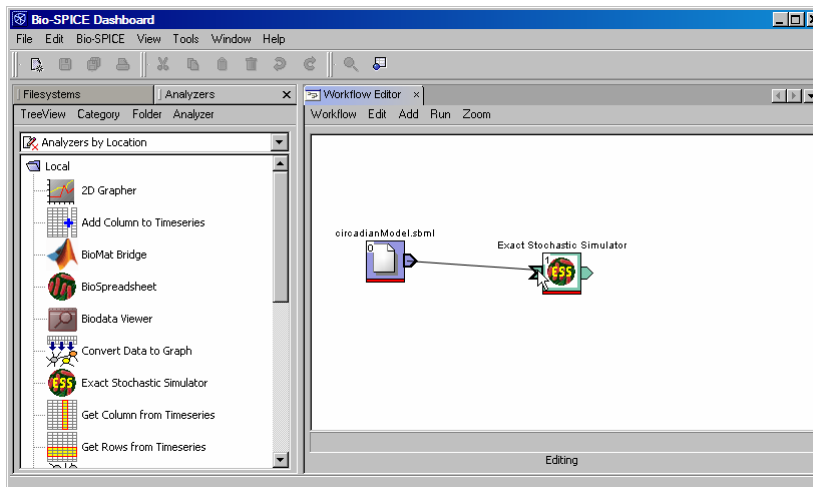


Figure C-4.6 Connect source document to analyzer in workflow.

The output from the ESS analyzer may be directed into the 2D Grapher analyzer which is provided with the Dashboard.

There are several ways of adding analyzers to the workflow. You may also select an analyzer from the analyzer pane. Click the analyzer 2D Grapher and right-click to open a menu, select Add To Workflow, as seen in Figure C-4.7.

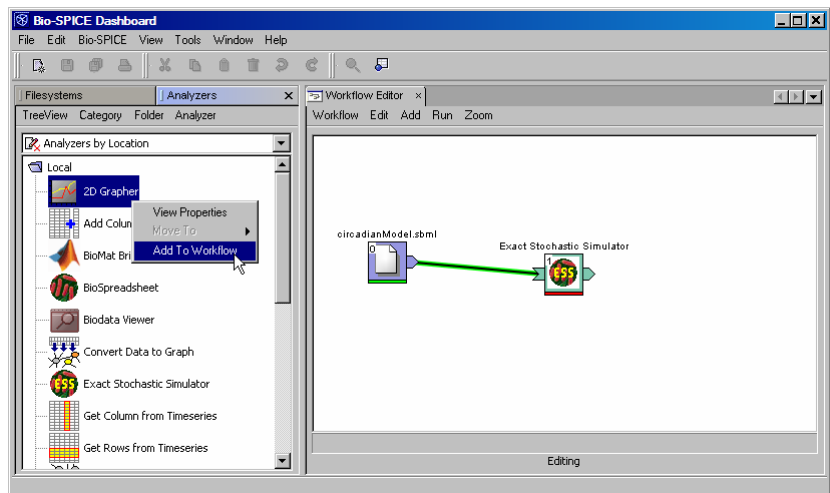


Figure C-4.7 Add ESS analyzer from analyzer pane.

Figure C-4.8 shows the complete workflow. Click the ESS analyzer to open a dialog window to set simulation parameters. Figure C-4.9 shows the three parameters that you must provide, the print interval, end time and seed number, the values 1, 100 and 1 were used respectively, within quotes (" ") due to the required string format.

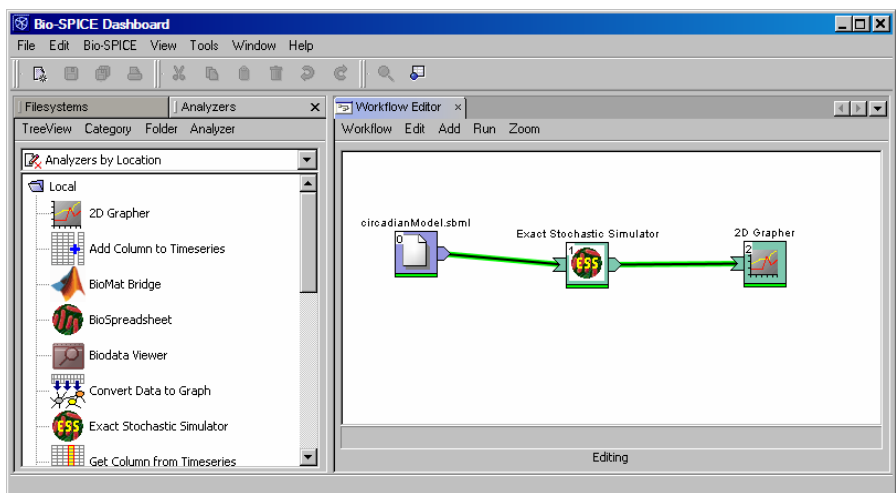


Figure C-4.8 Connect ESS analyzer to 2D Grapher analyzer in workflow.

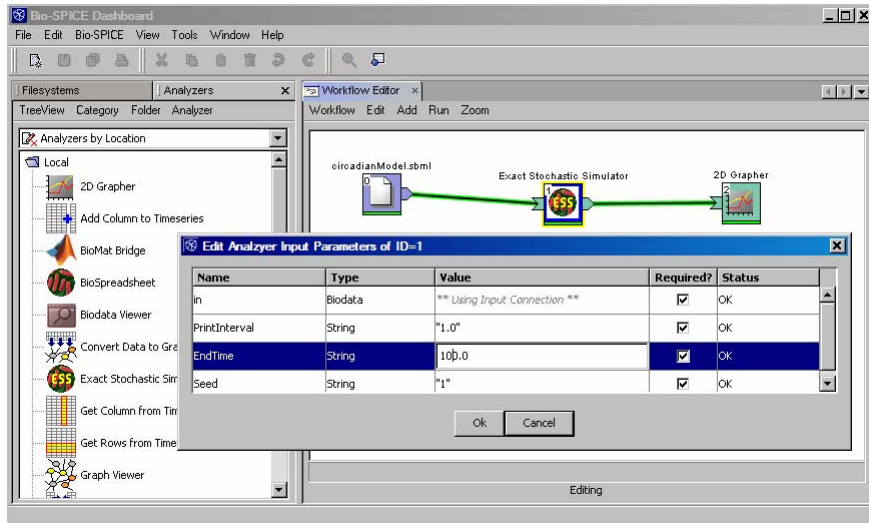


Figure C-4.9 ESS analyzer requires three parameters for simulation.

Save the workflow by clicking the menu `Workflow>Save` and typing the filename in the desired folder. For this tutorial, we have chosen `Tutorial_CircModelSim.wf` in folder `tutorialExamples`.

Running a simulation

With a complete workflow, it is possible to run a simulation and examine the time course of the model variables.

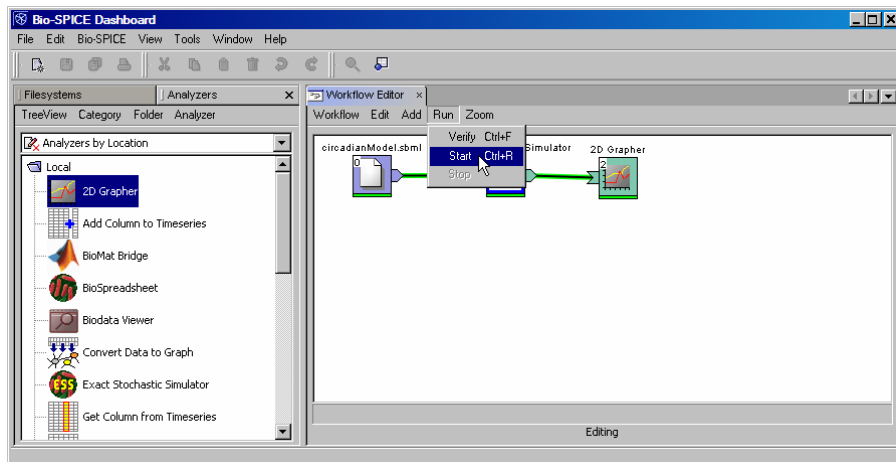


Figure C-4.10 Launching a simulation.

To launch a simulation click the menu `Run>Start`. The Dashboard displays the progression of the simulation by highlighting the analyzer of the workflow that is active with a green square.

When the simulation ends, 2D Grapher will display the time course of the model variables, as seen in Figure C-4.11.

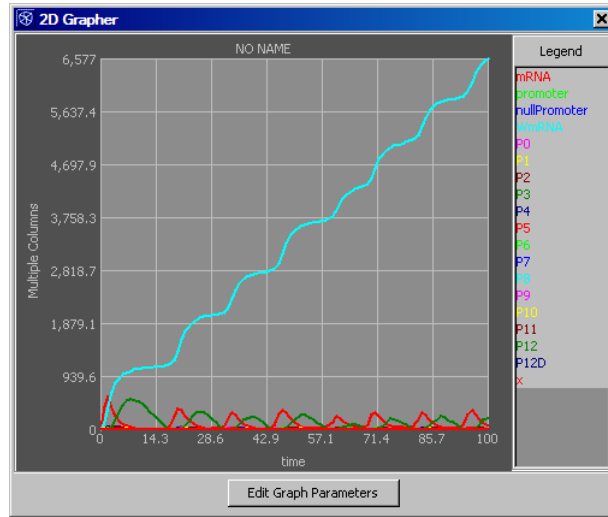


Figure C-4.11 Plot of circadian model time course.

The results displayed with 2D Grapher show the time course of all the variable of our model. Normally we want to limit the display to certain specified variables.

The 2D Grapher analyzer allows you to edit the variables to be displayed. To select the variable you wish to display click the button `Edit Graph Parameters`. A window will open, as seen in Figure C-4.12. Click the variables you wish to display, using the `Shift` key to select blocks of variables and the `Control (Ctrl)` key to select multiple individual variables.

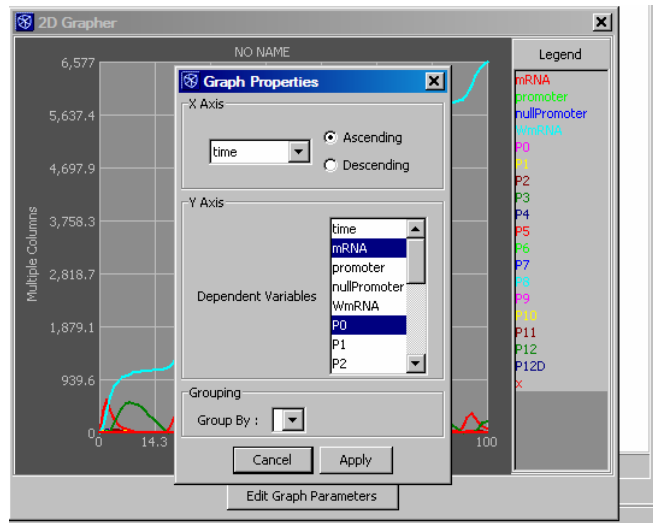


Figure C-4.12 Using the 2D Grapher property option you may select specific variable to display.

Click the button Apply in the Graph Properties dialog box in order to update the display, as seen in Figure C-4.13.

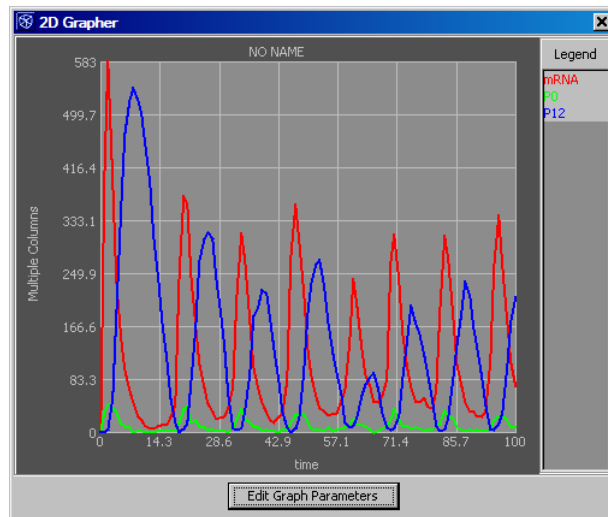


Figure C-4.13 Workflow with 2D Grapher analyzer.

Updating a file

So far we have seen how to run a simulation using the Dashboard. After running a simulation we would like to examine the behavior of the model using different parameter values. In order to modify the parameter values we need to create a workflow consisting

of a source document connected to the BioSpreadsheet analyzer. Click the edit menu of the source document, as shown in Figure C-4.4, to load the file you wish to edit. Figure C-4.14 presents such a workflow with the source document `circadianModel.sbml` that was described in Chapter C-3. Click the menu item `Run>Start` to execute the workflow and open the BioSpreadsheet editor with the desired file. Change the parameter value and save the file by clicking `File>Export SBML`. In our case, export the model using the same filename as before. Run the workflow with the updated parameters to view the change in system behavior. If you provide a new filename, the new filename will have to be provided as the source document, which you may change by right-clicking the icon and clicking the item `edit`.

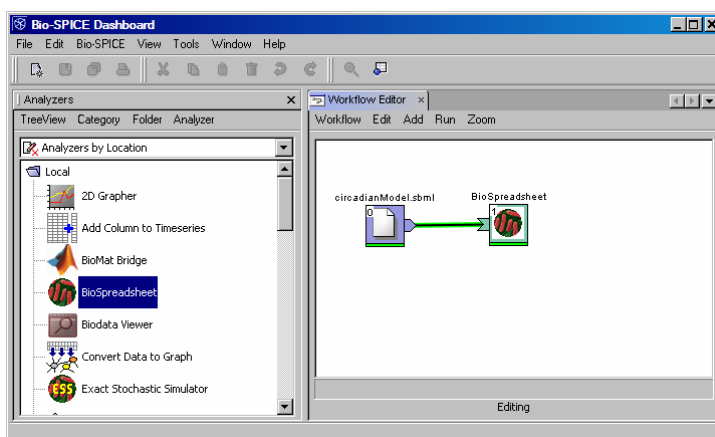


Figure C-4.14 A workflow to open BioSpreadsheet editor to update file.

Installing Additional Tools in the Dashboard

In the previous section, we saw how to construct a workflow and run a simulation.

There are several ways of installing tools in the Dashboard. The Dashboard provides a way to connect to an update center from which you may select and download tools.

Click the menu item `Tools>Update Center` to open the Update Center wizard, seen in Figure C-4.15. Click the button `Next` to view the possible updates available. The wizard will connect to the Bio-SPICE web site and download the list of available components.

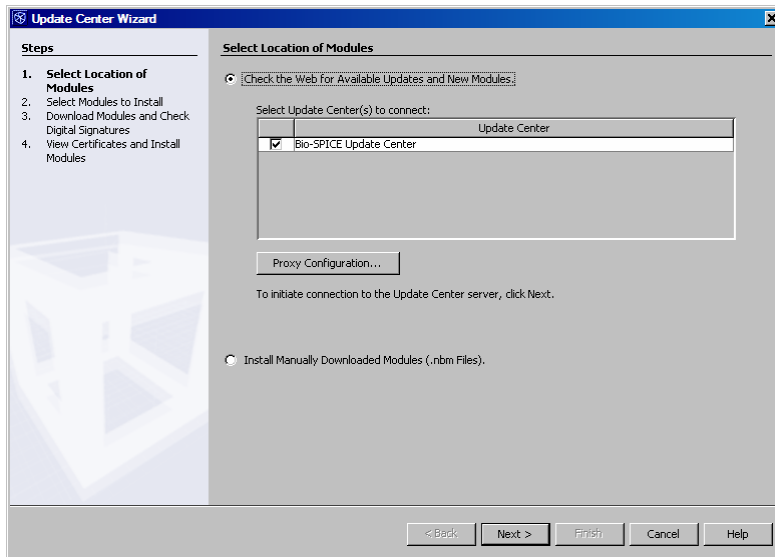


Figure C- 4.15 The Bio-SPICE update center wizard.

Scroll down the list of possible downloads and click the tool you need. Click the right arrow to move the module to the right pane titled include in install, as seen in Figure C-4.16.

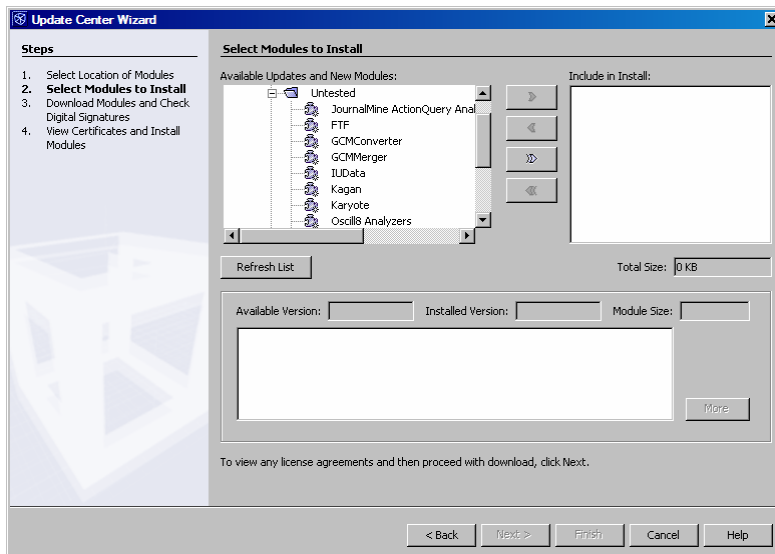


Figure C-4.16 Update center wizard with list of available downloads.

Click the button `Next` to download the program, you will be asked to accept the License Agreement. Click the button `Finish` in order to install the tool.

If the tool is properly installed, you will see the icon in the analyzer pane, after restarting the Dashboard.

Importing an Analyzer

Tool analyzers may be defined by an xml file which can be installed in the Dashboard by using the `import Analyzer` command. Click the pull-down menu item `Bio-SPICE>import Analyzer` to open a dialog box which displays the file hierarchy. Traverse the file structure in order to reach the folder/directory of interest. Click the file `toolName.xml` and click the button `open`. After importing the analyzer, the Dashboard needs to be restarted to show the analyzer icon.

Chapter C-5 Using Bio-SPICE Tools

The Bio-SPICE toolkit consists of the Dashboard application and numerous types of tools. Users are provided with model editor tools, simulators, database tools as well as visualization tools.

The Bio-SPICE web site, <https://biospice.org/index.php> provides the tools that may be incorporated into the Dashboard. The listing of the tools may be viewed in several ways, by alphabetical order, by order of organization that developed the tool, or by order of functional category.

The screenshot shows the Bio-SPICE website interface. At the top, there is a navigation bar with links for Home, Events Calendar, Site Map, Request Support, My Account, and Log Out. Below this is a search bar and a main menu with categories like ABOUT, RESEARCH, PRODUCTS, EVENTS, NEWS, DOCUMENTATION, WORK GROUPS, and MY ACCOUNT. A dropdown menu is open under 'Bio-SPICE Tools', showing options: 'Sort by Tool Name', 'by Organization' (selected), 'by Functional Category', and 'Tool Developer's Manual'. The main content area is titled 'Bio-SPICE Tools' and includes links for 'By Tool Name', 'By Organization', 'By Category', and 'Printer Friendly Version'. A paragraph states: 'The Bio-SPICE core application allows most tools to be downloaded via the Bio-SPICE Dashboard's Update Center.' Below this is a table listing tools by organization.

Organization	Tool Name	**Primary Functional Category	Summary
CalTech	SOS Tools View_detail	Model Analysis	SOS Tools solves feasibility or optimization sum of squares problem third-party Matlab toolbox. The solution is arrived at by performing sum of squares decomposition for multivariate polynomials, which is efficiently computed with semi-definite programming.
Columbia	Geneways View_detail	Data Analysis	Geneways allows access to the Geneways Database, which contains information from over fifty full text journals. Users can add additional information to an existing biochemical model.
Harvard	Biowarehouse2SBML View_detail	Data Analysis	Biowarehouse2SBML extracts stoichiometric and reversibility condition reactions in a pathway in the BioWarehouse, producing an SBML model suitable for metabolic flux analysis.
Harvard	Fluxor Computational Analyzer View_detail	Data Analysis	Fluxor Computational Analyzer performs flux predictions resulting from reactions, selected by the user, having their fluxes limited or completely disabled.
Harvard	Fluxor Spreadsheet View_detail	Model Composition & Visualization	Fluxor Spreadsheet provides a spreadsheet-like interface to allow users to specify nutrient conditions, external metabolites, and gene knockout. It displays the results of various kinds of flux predictions.
Indiana	JDesigner View_detail	Model Composition	JDesigner provides a visual biochemical network simulation tool that allows users to specify a biochemical network, derive the representative set of differential equations automatically, and simulate the network.

Figure C-5.1 Bio-SPICE tools listings by organization.

The tool page provides a description for each tool which can be accessed by clicking the link in red `View_detail` after each tool name.

Appendix C-1: Bio-SPICE Components List

Organization	Tools	Web Site
CalTech – California Institute of Technology	SOSTools	https://biospice.org
Columbia	GeneWays	http://geneways.genomecenter.columbia.edu/
Harvard	BioWarehouse2SBML, Fluxor	http://arep.med.harvard.edu/moma/
KGI – Keck Graduate Institute	JDesigner, Jarnac, MetaTool, Optimizer, SBWMatlab	http://sys-bio.org
LBL – Lawrence Berkeley Lab	HomologFinder, SensitivityAnalyzer, Pathway Builder	http://biospice.lbl.gov/PathwayBuilder/
NYU – New York University	NYUMAD, MYUSIM, Simpathica	http://bioinformatics.nyu.edu/Projects/Simpathica
MolSci – The molecular science institute	MONOD	http://monod.molsci.org/
SRI – Stanford Research Institute	BioWarehouse, BioMatBridge, SAL, Hybrid Automata, Symbolic Reachability Tool	http://bioinformatics.ai.sri.com/
TJU – Thomas Jefferson University	CloneUpdater, MetaCluster, PAINT	http://www.dbi.tju.edu/dbi/tools/paint/
Indiana University	CellX, Karyote Cell Analyzer, Karyote Genome Analyzer	https://systemsbiology.indiana.edu/cellx/
UCLA – University of California at Los Angeles	GeneScreen, IcdNA, MIAME Spice, NCA	http://www.ee.ucla.edu/~riccardo/
UCSB – University of California at Santa Barbara	BioSens	http://www.chemengr.ucsb.edu/~ceweb/faculty/doyle/biosens/
UNC – University of North Carolina	BioNets	http://x.amath.unc.edu:16080/BioNetS/
UPENN – University of Pennsylvania	Charon	http://www.cis.upenn.edu/mobies/charon
UTK-ORNL University of Tennessee and Oak Ridge National Labs	BioGrid, BioSmokey, BioSpreadsheet, ESS, OctaveBridge	http://biocomp.ece.utk.edu/
VaTech – Virginia Institute of Technology	BioPak, JigCell	http://jigcell.biol.vt.edu/
WRAIR – Walter Reed Army Institute of Research	GeneCite, Pathway Screen	

Appendix C-2: Bio-SPICE Usability Testing

Five models were used to test several Bio-SPICE tools, editors and simulators, in order to evaluate tool usability. The five models used were:

- **Circadian rhythm**, mRNA transcription and protein phosphorylation (Smolen P et al, *J Neurosci*, 21:6644-6656, 2001)
14 ODEs; 20 mass action reactions.
- **Circadian rhythm II**, mRNA transcription and protein phosphorylation (same as Circadian rhythm scaled to 95 ODEs; 100 mass action reactions.)
- **Cell division cycle**, cdc2 and cyclin interactions (Tyson J, *PNAS*, 88:7328-7332, 1991)
6 ODEs; 14 reaction equations.
- **Allosteric model for glycolytic oscillations** (Goldbeter A & Lefever R, *Biophys J*, 12:1302-1315, 1972)
2 ODEs (minimal model)
- **Memory induction** (Pettigrew D et al, *J Comp Neurosci*, 18:163-181, 2005)
16 ODEs; 39 reaction equations.

The simulators/editors were evaluated (when possible) with both a code specific implementation of the model using the editor tool and a SBML version which is supposed to be interoperable between simulators/editors. The table below indicates which models were tested with the various simulators/editors. These models can be found at the web site: <http://nba.uth.tmc.edu/darpa/> click menu MODEL CODE to find a copy of the table with links to the various models. PDF documents of posters presented at DARPA Bio-SPICE conferences can also be found in the Document page (click menu DOCUMENTS) as well as the Getting Started manual and Jarnac Use Case document.

	Circadian Rhythm I		Circadian Rhythm II		Cell Division Cycle		Allosteric Glycolytic Oscillations		Memory Induction	
	code specific	SBML	code specific	SBML	code specific	SBML	code specific	SBML	code specific	SBML
BioSpreadSheet/ ESS	CR	CR	CR100		CDC					
BioSketchPad/ Charon	CR				CDC					
Jarnac/ JDesigner	CR	CR	CR100		CDC	CDC	AGO		MI	
JigCell	CR	CR	CR100		CDC	CDC	AGO		MI	MI
Simpathica	CR	CR	CR100	CR100	CDC	CDC				
BioNets	CR		CR100		CDC					
PathwayBuilder	CR	CR			CDC	CDC				

Appendix D - SBW (Jarnac) Use Case

This use case describes how to use the System Biology Workbench (SBW) Jarnac simulator. This simulator is a general purpose simulator that can solve any ODE model. The editor is a text-based and provides a solver. You can export the model to SBML and use other solvers or run the Jarnac engine and view the time course with the Jarnac viewer, or use the Dashboard with the Grapher analyzer.

This use case is divided into the following sections:

- Download SBW and analyzer
- Modeling with Jarnac
- Dashboard workflow simulation and Interoperability

Download SBW and Analyzer


Jarnac is part of the SBW software package. You can download the SBW package from SourceForge.net, follow the link from <http://sbw.kgi.edu/>.

The analyzer needed to run Jarnac through the Dashboard can also be found through the sbw.kgi.edu site.

After downloading the SBW software install it on your computer.

Modeling with Jarnac

Once SBW is installed you should find a folder named Jarnac as well as a folder named JDesigner. You can launch the Jarnac application by double-clicking on the Jarnac

application icon  .

For this use case, we will implement an allosteric model for glycolytic oscillations by Goldbeter and Lefever, 1972, also published in Goldbeter 1990. This reduced model, of two ODEs, using a quasi-steady-state hypothesis and dimensionless variables; shows oscillations of substrate and product concentrations. The equations are presented in Figure D-1. It is clear that the nonlinear function for theta does not permit a simple conversion to mass action equations. For this reason the editor BioSpreadsheet can not be used. The Jarnac editor is appropriate as a general ODE editor. Figure D-2 shows the Jarnac editor with an implementation of the model of Figure D-1.

$$\frac{d[\alpha]}{dt} = v - \sigma\phi$$

$$\frac{d[\gamma]}{dt} = q\sigma\phi - k_s \gamma$$

$$\phi = \frac{\alpha e (1 + \alpha e)^{n-1} (1 + \gamma)^n}{L(1 + \alpha e')^n + (1 + \alpha e)^n (1 + \gamma)^n}$$

Figure D-1. Simplified model of glycolytic oscillations (parameter value used in Figure D-2: $n=2$, $v=0.2$, $s=1000$, $k_s=0.1$, $q=1$, $L=7500000$, $c=0.01$, $e=0.1$, $ep=d=0$).

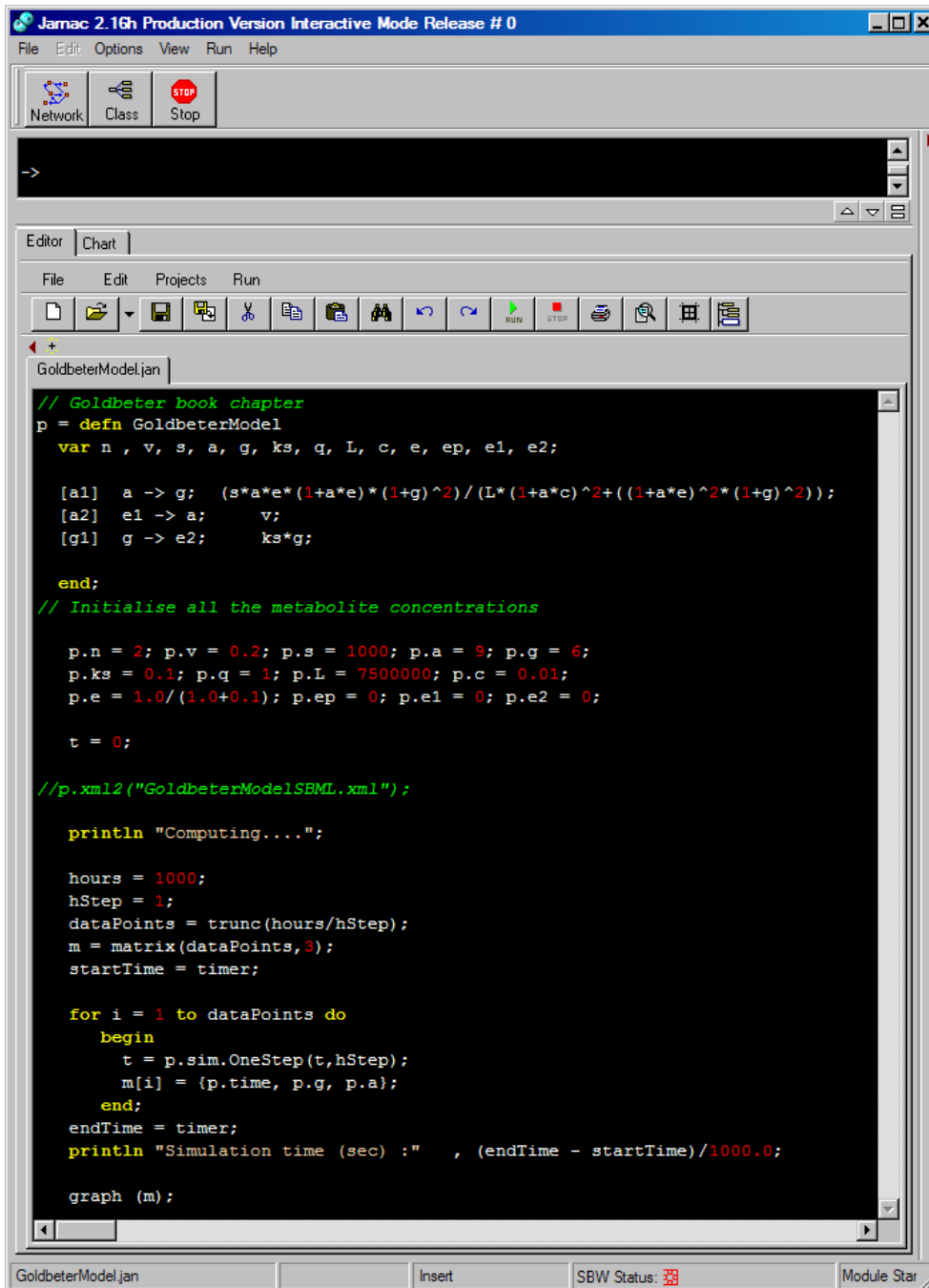


Figure D-2. Jarnac simulator with Goldbeter glycolytic oscillator model.

Jarnac application provides a full text editor where you type your model followed by the commands to execute the model. There are several ways to solve the equations of a model. Figure D-2 shows the use of a variable p for the definition of the model in order to be able to convert the model to SBML.

Jarnac provides commands to load, save or generate SBML code from a model. Using an object-oriented framework, Jarnac provides methods called on the model to carry out the desired operation. For example, in order to generate SBML level 2 code from a model, the command `xml2("filename.xml")` is called on the variable representing the model, such as `p` in Figure D-2.

Defining a model variable provides us with better control of the simulation step, so that we can single step the computation and test for conditions.

Once the model is entered using the editor, the user clicks the run icon in order to solve the equations.

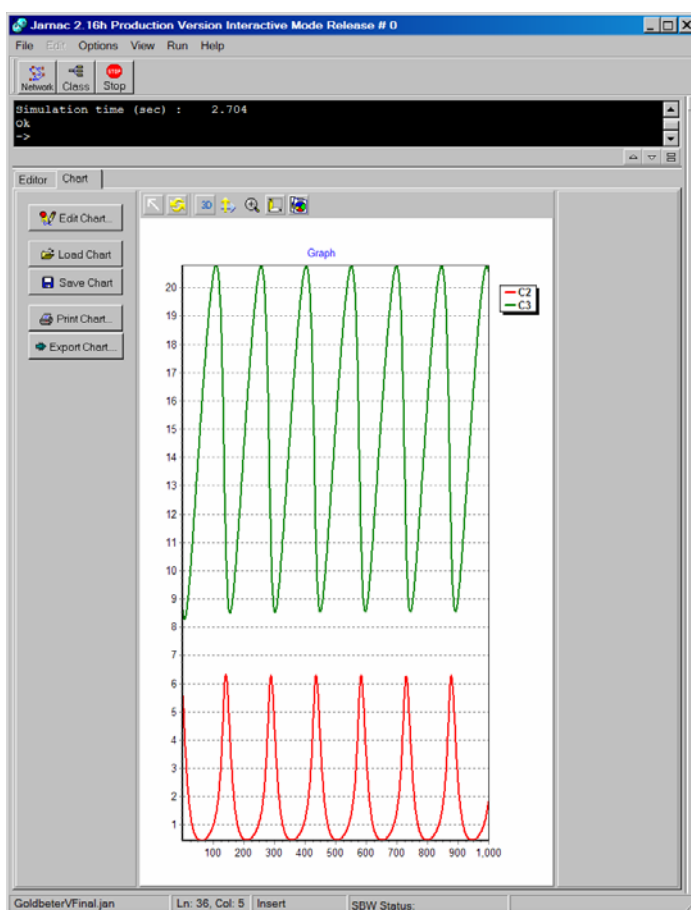


Figure D-3. Jarnac display of model variable time.

The model presented in Figure D-1 shows oscillations in substrate, presented in Figure D-3.

The next step is to build a Dashboard workflow in order to simulate the model using the Dashboard.

Dashboard workflow simulation

Before launching the Dashboard, you must install the SBW analyzer in order to have access to Jarnac and JDesigner from the Dashboard. Download the analyzer and double-click on the application in order to install it.

Once the analyzer is installed, launch the Dashboard. Figure D-4 illustrates the workflow and resulting time course plot and table data.

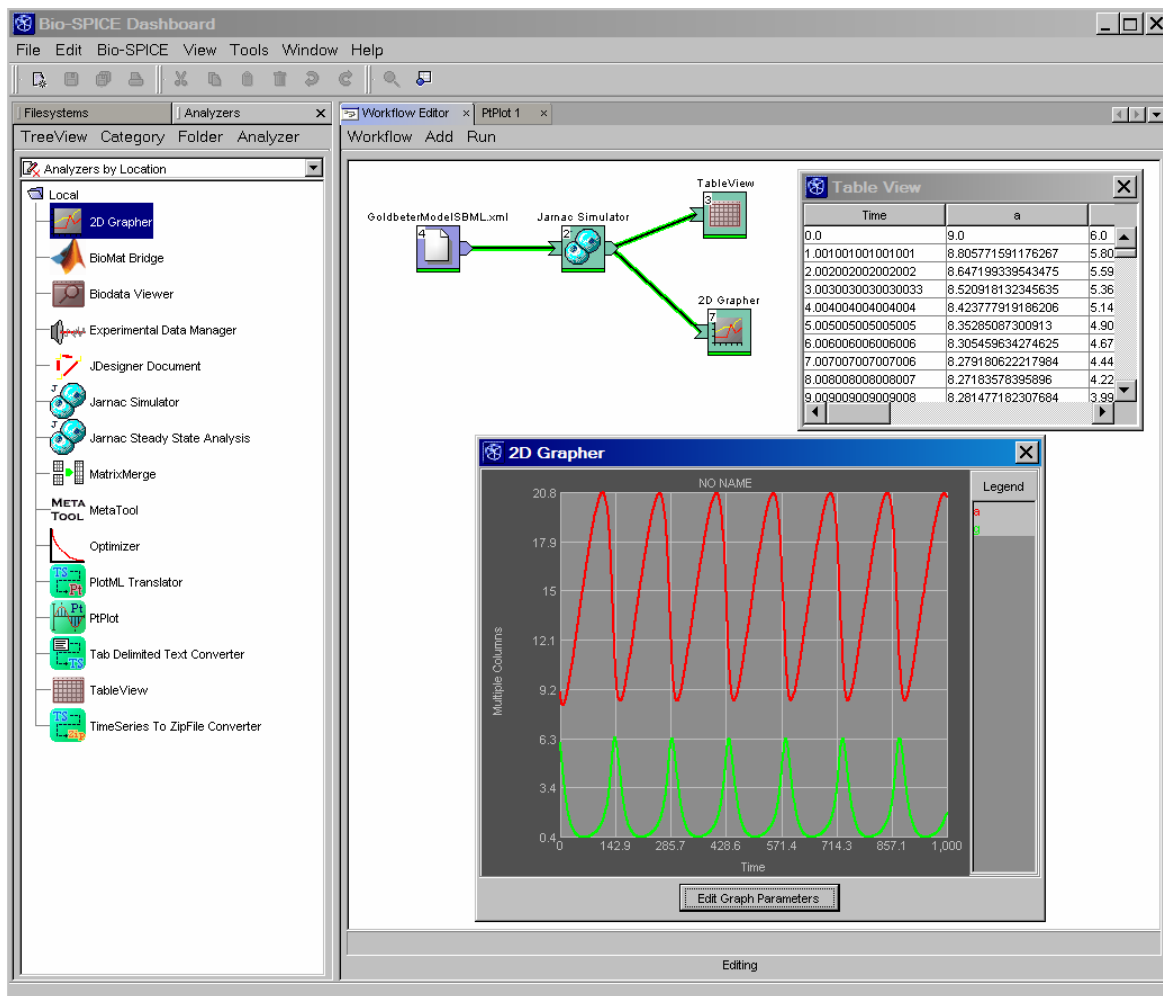


Figure D-4. Dashboard with SBW analyzer, glycolytic oscillator model, workflow and resulting table data and time course graph.

We can also test interoperability between different simulators with this workflow, using the circadian model we previously developed.

References

Goldbeter A, Lefever R (1972) Dissipative structures for an allosteric model. Application to glycolytic oscillations. *Biophys J*, 12: 1302-1315.

Goldbeter A (1990) *Rythmes et chaos dans les systèmes biochimiques et cellulaires*. Masson, Paris, 304 pp.