Article

GKIN: a tool for drawing genetic networks

Jonathan Arnold[1], Thiab R. Taha[2], Leonidas Deligiannidis[3]

[1]Genetics Department, The University of Georgia, Athens, GA 30602-0001, USA

[2]Computer Science Department, The University of Georgia, Athens, GA 30602-0001, USA

[3]Computer Science Department, Wentworth Institute of Technology, Boston, MA 02115, USA

E-mail: deligiannidisl@wit.edu

Received 20 October 2011; *Accepted 30 November 2011*; *Published online 1March 2012*

IAEES

Abstract

We present GKIN, a simulator and a comprehensive graphical interface where one can draw the model specification of reactions between hypothesized molecular participants in a gene regulatory and biochemical reaction network (or genetic network for short). The solver is written in C++ in a nearly platform independent manner to simulate large ensembles of models, which can run on PCs, Macintoshes, and UNIX machines, and its graphical user interface is written in Java which can run as a standalone or WebStart application. The drawing capability for rendering a network significantly enhances the ease of use of other reaction network simulators, such as KINSOLVER (Aleman-Meza et al., 2009) and enforces a correct semantic specification of the network. In a usability study with novice users, drawing the network with GKIN was preferred and faster in comparison with entry with a dialog-box guided interface in COPASI (Hoops, et al., 2006) with no difference in error rates between GKIN and COPASI in specifying the network. GKIN is freely available at http://faculty.cs.wit.edu/~ldeligia/PROJECTS/GKIN/.

Keywords GKIN; genetic network; drawing tool; Java; GUI.

1 Introduction

A living system can be viewed as a chemical reaction network (Beadle and Tatum, 1941). At the heart of a new systems biology approach to understanding living systems, is viewing living systems as chemical reaction networks (Arnold et al., 2004; Dong et al., 2008; Tang et al., 2011). New genomics techniques like RNA and protein profiling are providing ways to measure the state of these networks (Ideker et al., 2001; DeRisi et al., 1997). Simulating an ensemble of hypothesized biochemical and genetic regulatory networks and fitting these profiling data to networks allow us to predict what a cell is doing and to compare these predictions with observed states of a system over time (i.e., their observed kinetics) (Battogtokh et al., 2002; Brown and Sethna, 2003; Clarke et al., 2006; Yu et al., 2007; Violin et al., 2008; Toni et al., 2009; Jain et al., 2011) to evaluate the network models. One of the central problems of functional genomics then becomes the simulation and identification of biochemical and gene regulatory networks describing how a living system functions (Arnold et al., 2004; Martinez-Antonio, 2011).

Solving this problem begins by focusing on a particular process like the biological clock in Fig. 1 found in most eukaryotes and some prokaryotes (Yu et al., 2007), by identifying the regulatory control exercised by

genes and how the products of these genes participate in biochemical pathways relevant to the process in question (Ibrahim et al., 2011; Paris and Bazzoni, 2011; Tacutu et al., 2011). The resulting model of the biological clock as an example is then a network or "genetic network", in which genes, their products, metabolites, and reactions enter as nodes in the circuit (Kochut et al., 2003). The term genetic network then refers to a kinetics model in biochemistry and molecular biology, which includes efforts to account fully for the effects of concentrations of each species, the time evolution of biochemical events, and the accumulation of transient intermediates (Purich and Allison, 2000). Validating a genetic network or biological circuit depends upon our ability to simulate a particular reaction network and to predict how the network responds to various experimental perturbations. Perturbations may include gene knockouts, change in the environment like the regimen of light experienced, or the addition of a protein inhibitor like a drug. As the information about this circuit accumulates, the biological circuit is modified in the light of new experiments, and new predictions are made to refine the circuit (Wagner, 2001; Dong et al., 2008; Gardner et al., 2003; Xiong et al., 2004; Schulz et al., 2009).

Our goal here is to present a Graphical User Interface (GUI) for “drawing” an arbitrary reaction network satisfying multiplicative mass action kinetics (Bhalla and Iyengar, 1999). The drawing can be used as input to the simulator here and another simulator, KINSOLVER (Aleman-Meza et al., 2009). Exchange of models rendered in GKIN to other systems biology workbenches is done through the systems biology markup language (SBML) (Hucka et al., 2003). The “drawing” aspect is nontrivial but essential for modeling (Becker and Rojas, 2001), and this functionality of drag and drop to populate a network with species, reactions, and links on a canvas is only possessed by a few simulators (Funahashi et al., 2008; Mirschel et al., 2009; Sauro et al., 2003; Sorokin et al., 2006; Zhang, 2012) as reviewed in Funahashi et al. (2008). There are at least three alternatives to a network drawing approach (Vass et al., 2006), but given a choice most people prefer a drawing approach (see Table 1). There are strengths and limitations to drawing vs. scripting vs. wizarding vs. spread sheet approaches, and there is a nice complementarity between drawing and spread sheet approaches (Vass et al., 2006; Zhang, 2007).

2 Background

In this general model on which the simulator is based (Aleman-Meza et al., 2009), the cell is viewed as well-stirred, although it is possible to embed cell compartmentalization and scaffolding into a simulation with the existing simulator described in (Aleman-Meza et al., 2009). The simulator allows an arbitrary number of species into a reaction if higher-order kinetics need to be hypothesized. In principle, the number of reactions and participating species is only limited by the array dimensions set in the GKIN simulator. The model of a reaction network is deterministic, and each reaction has forward and backward reaction rate constants, denoted by kf and kb. Once the initial concentrations of all species are provided and the reaction rates specified, the time derivative of each species' concentration can be computed. The simulator solves recursively a system of ordinary differential equations (ODEs) for the trajectories of all species in the reaction network. As illustrated by some of the examples in (Aleman-Meza, et al., 2009), this task of solving these ODEs is not always straightforward, and there is a premium on speed in the computing of each model in a large model ensemble (Battogtokh et al., 2002; Brown and Sethna, 2003; Clarke et al., 2006; Yu et al., 2007; Violin et al., 2008; Toni et al., 2009). The resulting solutions to the ODEs are presented within the GKIN graphical interface.

Several packages exist for simulation of biochemical reaction networks. METAMODEL (Cornish-Bowden and Hofmeyr, 1991) can calculate steady-state fluxes of metabolic systems and provides a functionality where its results are stored in files and can be opened in external applications for further analysis. GEPASI (Mendes, 1993, 1997) simulates the kinetics of biochemical reaction systems and provides tools to fit models to data.

GKIN is not constrained by the size of the reaction network and has a menu-based interface, but unlike GEPASI, GKIN is open source. SCAMP **(**Sauro, 1993; Sauro et al., 1987) extends matrix algebra procedures for determining the flux control coefficients of enzymes to allow determination of the concentration control coefficients. In contrast to the metabolic control analysis tools (Fell, 1992; Sauro, 1993; Ko et al., 2009), another suite of tools focus on metabolic flux analysis for a network in steady state as in (Klamt et al., 2003). KINSIM (Barshop et al., 1983) (Wachsstock and Pollard, 1994) is an interactive simulation tool that provides methods to simulate kinetic process curves. It generates the differential equations of a given mechanism and applies a direct numerical integration. E-CELL (Tomita et al., 1999; Tomita et al., 2000; Ishii et al., 2004) is a generic object-oriented environment for simulating molecular processes in user-defined models and allows a stochastic component to the models. It provides a graphical interface that allows observation and interaction. Like E-CELL, GKIN can simulate large reaction networks using a variety of numerical integration methods; 5 numerical solution methods (as opposed to 2 methods in E-CELL or 1 in CellDesigner 3.5 (Funahashi et al., 2008) are built in with an option to add others so that challenging reaction networks can still be simulated. Input can take the form of multiple models as might arise in model fitting. There are four features that separate GKIN from most other simulators, its platform independence, flexibility in solution procedures, its capability to simulate efficiently large (10^4 -10^5) ensembles of models and its support of a dynamic user interface used to draw a kinetics model. Other modeling approaches for gene regulatory and biochemical networks are being pursued by others. These include Bayesian networks (Murphy et al., 1999) (Friedman et al., 2000), neural networks (Weaver et al., 1999), boolean networks (Huang, 1999; Shmulevich et al., 2002), and hybrid qualitative approximations to ODE modeling approaches (Ironi and Panzeri, 2009). The ultimate success of these competing approaches will be determined by their ability to predict successfully reaction network behavior. The advantage of the approach here is that it is well-rooted in chemistry and physics. To integrate the circuit design capability with other software, as used in (Snoep et al., 1999) relying on GEPASI (Mendes, 1993), MIST (Ehlde and Zacchi, 1995) and DBSolve (Goryanin et al., 1999), the GKIN tool inherits SBML exchange from KINSOLVER (Aleman-Meza et al., 2009).

Here we present a description of the underlying model on which the simulator is based. Then the implementation of the simulator is described. One example is used to illustrate the development of GKIN for the simulator and to present briefly a notation for reaction networks more extensively described elsewhere (Aleman-Meza et al., 2009). The results of a usability study of the drawing interface are presented.

3 Kinetics Model

A kinetics model is a specification of reactions between hypothesized molecular participants. These models that represent gene regulation and biochemical reactions, allow us to observe and study how cells behave over time. The standard multiplicative mass action kinetics leads to a specification of an underlying system of coupled ODEs that describes a particular reaction network (Bhalla and Iyengar, 1999). The generality of mass action kinetics has been discussed (Aleman-Meza et al., 2009). The specification of a reaction network model begins with the construction of a circuit-like diagram that captures the relationships of species, reactants and products, the reactions in which the species participate, and the relationships of the reactions. To illustrate a relatively simple kind of genetic network we will use a network for the biological clock of *Neurospora crassa* shown in Fig. 1.

Following the notation previously established (Aleman-Meza et al., 2009), molecular species are represented as rectangles. Species connect to other species via reactions, which are represented as circles. The lines connecting species and reactions are bidirectional. That means that reactants (on the left side of a reaction) participating in a reaction could also be products of the reaction, usually in smaller amounts, however. This is

determined by the forward and backward reaction rate constants (kf and kb). Incoming arrows into a reaction indicate the reactants participating in a reaction. Outgoing arrows from a reaction indicate the products of a given reaction. In Fig. 1 some reactants have bidirectional arrows (such as reactant *wc-1*^{1} in reaction S1), and that means they are catalysts, appearing on both right and left hand side of the reaction. A single arrow means the reactant is consumed and is not catalytic (such as *wc-1*^{r0} in reaction C1). A reaction can have arbitrary numbers of input and output species that can involve higher-order kinetics. An example is in reaction A with its cooperative kinetics: (A) n WCC + frq^{0} -> frq^{1}. A total of n molecules of WCC appear in the reaction on the left hand side (LHS), meaning higher-order kinetics. The simple genetic network describing the biological clock in *N. crassa* and shown in Fig. 1 consists of 26 reactions (*e.g.*, S1, L1, and D1 for transcription, translation, or decay) with 17 participating molecular species (*wc-2*^{1}, *wc-2*^{r1}, *WC-2*, *wc-1*^{1}, *wc-1*^{r0}, *wc-1*^{r1}, *WC-1*, *WCC*, *phot*, *frq*^{0}, *frq*^{1}, *frq*^{r1}, *FRQ*, *ccg*^{0}, *ccg*^{1}, *ccg*^{r1}, *and CCG*). Gene symbols in the network are denoted by lower case italics, and proteins, by all capital letters for *N. crassa*. Gene symbols superscripted 0, 1, r0, r1, indicate, respectively, a transcriptionally inactive (0) or active (1) gene or translational inactive (r0) or active (r1) messenger RNA (mRNA).

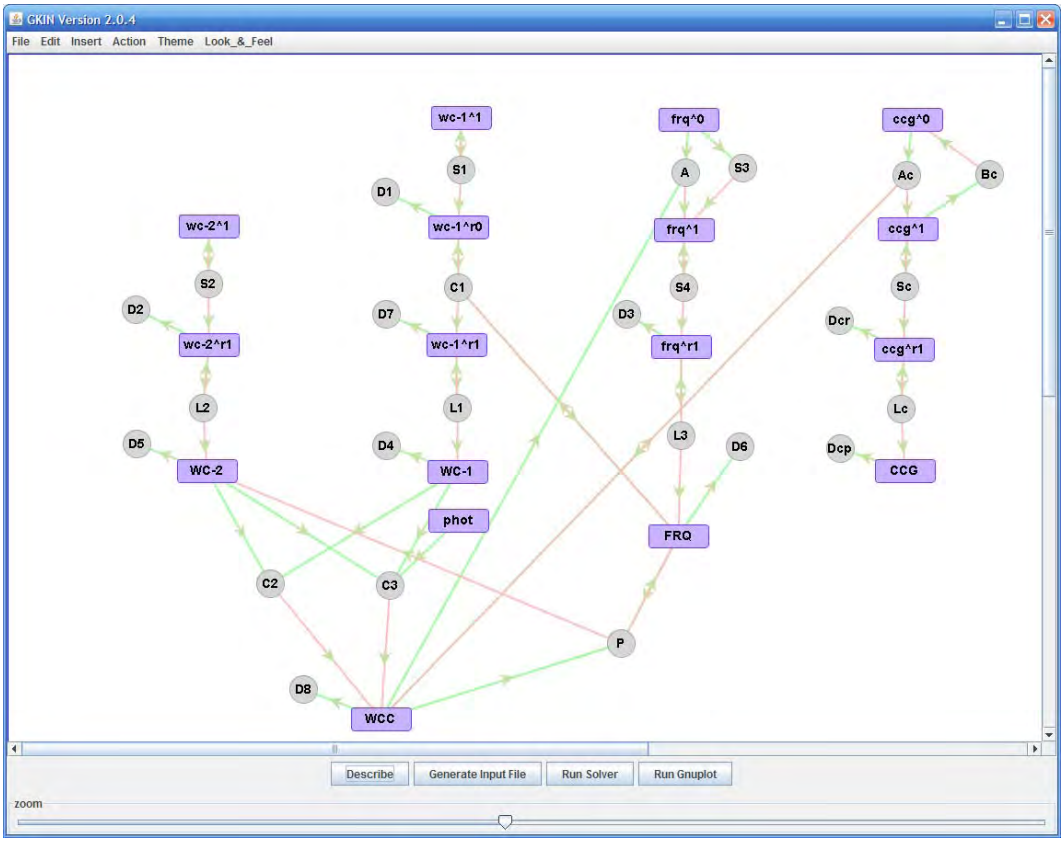

Fig. 1 Genetic network for the clock in *Neurospora crassa*. Molecular species in the network are represented by boxes. The *white-collar-1* (*wc-1*), *white-collar-2* (*wc-2*), *frequency* (*frq*), and *clock controlled gene* (*ccg*) gene symbols can be superscripted 0, 1, r0, r1, indicating, respectively, a transcriptionally inactive (0) or active (1) gene or a translationally inactive (r0) or active (r1) mRNA. Associated protein species are denoted by capitals. A phot denotes the photon species. Reactions in the network are represented by circles. Arrows entering circles identify reactions; arrows leaving circles identify products; and bidirectional arrows identify catalysts. The labels on each reaction, such as S4, also serve to denote the rate coefficients for each reaction. Reactions labeled with an S, L, or D denote transcription, translation, or degradation reactions, respectively. Reactions without products, such as D8, are decay reactions. Reactions, such as A and P, have cooperative kinetics: (A) nWCC + frq^{0} -> frq^{1} and (P) WCC + m FRQ -> WC-2 + mFRQ. The only reaction with a nonzero back reaction Abar is reaction A. The use of a tilde (^) denotes superscript in Fig. 1. The model is derived from Dong et al. (2008).

Table 1 Graphical interface in GKIN is preferred over a Wizard Interface of COPASI and leads to faster entry of model specification with similar error rates. Time units are in minutes (m).

1	What computer platform did you use?	GKIN Mac (16) Unix (1)	COPASI Mac (12); PC (5)
2	Were you able to install the software?	Yes (16)*	Yes (17)
3	How many minutes were taken in installing the software?	2.5 m average	2.7 m average
4	Were you able to specify the attached network by drawing or by the dialog boxes encountered? Print out the network as a diagram for GKIN and provide .cps file for COPASI.	Yes (17)	Yes (17)
5	How long did it take you to specify the network?	46 m average	65 m average
6	Were you able to generate a trajectory of the FRQ protein over time?	Yes (14); No (3)	Yes (15); No (2)
7	Were you able to save your specified network?	Yes (17)	Yes (17)
8	Which software did you prefer for specifying the model?	16	1
9	Was the network correctly specified?	No (16); Yes (1)	No (12); Yes (5)
10	Were there missing species?	No (17)	No (17)
11	Were there missing reactions?	No (13); Yes (4)	No (14); Yes (3)
12	Errors in reactions?	3.2 average	2.4 average
13	Rate constant errors?	1.6 average	2.4 average
14	Computational errors?$	No (16); Yes (1)	No (17)

*One user found the GKIN software already installed on her workstation. $A computational error is one in which the solver fails.

Such a model can be easily expressed as a series of chemical reactions or kinetics reactions. These (reversible) chemical reactions are of the form *S1 + S2 -> S3 + S4*, where *Si* is a participating molecular species (Purich and Allison, 2000). The symbol -> is used to indicate that the reaction is reversible with forward and backward reaction rate constants. Note that the backward reaction rate constants are often very small, and in some cases can be set to zero. In Fig. 1 only the A reaction has a nonzero back-reaction constant; all others are assumed zero. In order to write down the chemical reactions from Fig. 1, each reaction (depicted with circles) is written independently. Take the reaction represented by the circle P as an example. Arrows connecting species *FRQ* and *WCC* to reaction P denote that they are reactants. Other reaction network notations have been implemented as well (Funahashi et al., 2008). The outgoing arrows from reaction P towards *WC-2* and *FRQ* indicate that these are the products, i.e., appear on the right hand side (RHS). The

arrow also defines the forward reaction. This is expressed as a chemical reaction m*FRQ* + *WCC* ↔ m*FRQ* + *WC-2*. In that FRQ appears on the LHS and RHS, FRQ acts catalytically, and this is denoted by the double arrow. Details of deriving the corresponding ODEs and more elaborate examples can be found in (Aleman-Meza et al., 2009). The simulator receives as input, an ASCII file (or script) which is a representation of the chemical reactions, that might be redirected to other tools, such as SYCAMORE (Weidemann et al., 2008) or KINSOLVER (Aleman-Meza et al., 2009). The ASCII file is generated automatically from the drawing in GKIN. An SBML file can be accepted by KINSOLVER and an ASCII file, generated by KINSOLVER for exchange with GKIN. The participating species are assigned concentrations at an initial time t0, and the system is solved for concentrations at a final time t1. The parameters and initial conditions are all part of a fixed-format input file. The simulator takes the input file (which can describe an ensemble of models to be simulated) and solves the ODEs (for each model listed) and generates an output file.

While the corresponding system of ODEs is more concrete, the network in Fig. 1 is much more interpretable, fully and uniquely specifies the reaction network abiding by mass-action, and tells a story. As an example, from Fig. 1 the WCC protein activates the oscillator gene *frq* in reaction A. The active frq^1 gene is then transcribed into its cognate mRNA frq^{r1} in reaction S4, which in turn is translated into its cognate protein FRQ in reaction L3. The level of FRQ gives to the organism a readout of the time of day, being high at dusk and low at dawn. The FRQ protein, in turn, deactivates the WCC protein in the P reaction. The protein FRQ thereby closes a loop of dynamical frustration (i.e., negative feedback) wherein WCC turns on the oscillator gene whose product shuts down the activator WCC. This dynamical frustration explains in part how clock oscillations arise (Yu, et al., 2007). Yu et al. (Yu et al., 2007) give necessary and sufficient conditions for this network to display sustained oscillations and has a mathematical connection to the repressilator discussed in (Marchisio and Stelling, 2008).

4 User Interface

GKIN is a graphical user interface (GUI) to a rewritten KINSOLVER's (Aleman-Meza et al., 2009) simulator engine that allows one to design a circuit of a chemical kinetics model by drawing the circuit, modifying it, manipulating it, and maintaining it. GKIN will then generate input files to the GKIN solver, generate graphical reports using Gnuplot, and also list the corresponding set of ODEs, all of these from the same Java application. Both, Java and Gnuplot are freely available on the Internet, making GKIN open source. GKIN consists of two major components: a) the client, which is the main application or else the graphical interface, and b) the server which executes the GKIN ODE solver (Aleman-Meza et al., 2009). The client program (GKIN) is used to design the model, to generate the input file to the GKIN solver, to send (the generated input file) using TCP/IP to the server, to receive the results from the server, and to plot the results using Gnuplot.

The server machine runs a concurrent server process which accepts connections from GKIN clients and executes each request in a separate Java thread in the GKIN solver. When the GKIN solver engine finishes, the server process transmits the results back to the client. The client transmits the generated input file to the server, the server process accepts the connection, and the server starts a thread which executes on the GKIN solver. When the GKIN solver finishes, it sends the results back to the client. The server supports multiple clients connecting to the same server (in which case the server process starts a separate thread for each GKIN solver execution), or a single client could connect to multiple servers feeding them different input files. This multiple job capability is supported in other ways in other simulators (Aleman-Meza et al., 2009; Funahashi et al., 2008). Since the GKIN solver engine could take a considerable amount of time to finish a particular network simulation, we give the option to the user to cancel execution of the solver. This interrupt capability not only terminates the thread on the client side, but also the thread on the server side that is executing in the GKIN

solver engine as well. This is necessary, so that when the client cancels the client thread, the server does not waste CPU cycles executing the GKIN solver engine.

A snapshot of the client application, for the clock network is shown in Fig. 1. The user can create the molecular species, reactions, and the relationships between them (the lines) simply by using the mouse on the application's canvas; the user first selects the desired action from the "Insert" pull down menu shown in Fig. 1. In addition, via the Action->Properties pull down menu, the user can assign initial concentrations to species, set/unset the jfix value in the GKIN solver specifying ODE solution method (Aleman-Meza et al., 2009), and also set the forward and backward rate constants of a reaction (kf and kb). The user can drag and move around the species and the reactions to minimize edge crossings in the graph in Fig. 1; the lines are constrained so that a line is always attached to a species or a reaction. If the user deletes a species or a reaction, the attached lines are also deleted. An option in the Action->Properties pull down menu in Fig. 1 allows the rate constants to be displayed on the network diagram. The visual representation of the model can be saved and loaded at a later time to run experiments or can be modified. The main actions of the application are listed as buttons at the bottom of the window. The "zoom" slider enables one to zoom in and out of the visual representation of the model. The "Describe" button displays the reaction relationships and the ODEs of the species where the kf's and kb's are the forward and backward reaction rate constants respectively. Drawing a genetic network enforces a semantically correct specification of a kinetics model, a feature not present in KINSOLVER (Aleman-Meza et al., 2009).

The "Generate Input File" button opens a dialog window where the user specifies the details of the input file and then generates the input file to the GKIN solver, which is generated manually in the previous Java-Servlet interface to KINSOLVER (Aleman-Meza et al., 2009). The "Run Solver" button opens a dialog window, and the user provides four pieces of information: a) the host where the GKIN solver engine is installed, b) the port number of the server listening for incoming connections, c) the input file to be fed to the GKIN solver, and d) an output file (where the results from the GKIN solver simulation will be stored). After the GKIN solver engine finishes, the results are sent back to the GKIN client and saved locally onto disk. The user at this point can use the "Run Gnuplot" button in Fig. 1 to plot the graphs, for each species, without contacting the GKIN solver again. On the other hand, the Servlet based interface needs to contact the KINSOLVER engine for every graph (Aleman-Meza et al., 2009). If KINSOLVER takes one hour to generate the results, then plotting 10 graphs would require the user to run KINSOLVER 10 times and wait 10 hours. In contrast, the GKIN solver saves the results of each execution simultaneously, and Gnuplot can be executed for every species without contacting the GKIN solver again. Upon activating Gnuplot the user can specify global parameters to Gnuplot, such as the title, the axis labels, number ranges of axes, color, file names, and plotting style. After the user clicks the OK button, the user is presented with a dialog window where he/she selects which species plot to view. The result is a kinetics plot of the selected reactant or product generated by Gnuplot. When the user clicks on the "Done" button, the graph window closes and the previous window opens, where the user can select another species to plot. The user can also select the "Show Gnuplot Commands" to view the generated commands to Gnuplot for verification purposes or to copy the commands and generate manually the graph at a later time without going through GKIN.

5 Usability of Interface

At least four approaches have been utilized to create interfaces to specify genetic networks with: (1) drawing; (2) scripts; (3) dialog-box guiding or WIZARD; (4) spreadsheets (Vass et al., 2006). We assess the usability of a drawing approach (1) against a WIZARD approach used by COPASI (Hoops, et al., 2006). The choice of COPASI is made for three reasons. One, we have made comparisons with its predecessor, GEPASI, with

respect to its performance in computing solutions to ODEs (Aleman-Meza et al., 2009). Two, GEPASI and COPASI are widely used. Three, COPASI has an approach to the design of networks distinct from GKIN.

We wished to determine the accessibility of the software to a general but interested audience. As a consequence we selected junior and senior undergraduates and graduate students who were taking Bioinformatics. A total of 17 undergraduates or graduate students in Bioinformatics at UGA with no prior experience with either GKIN or COPASI were given Web addresses for GKIN and COPASI (selected Version 4.5 (Build 30)). The undergraduates were then asked to install GKIN and COPASI and answer questions 1-8 in Table 1. J. Arnold determined the answers to questions 9-14 based on the answers to questions 1-8 from the GKIN network diagram file and .cps file generated by each student. The first question assesses the broad accessibility of COPASI and GKIN. If the installation succeeded, the errors in specifying the genetic network in Fig. 1 and speed of specifying the network were assessed from output in GKIN or COPASI with questions 1-14. The information provided to each student included both a drawing of the network and a spreadsheet of reactions and their rates from Table 1 in Yu et al. (2007). Each student was left to choose which information to use to specify the network in GKIN and COPASI. Each student also was assessed for the rapidity of specifying the model as well. While it is well known that people prefer a drawing approach, we collected a response on the preference of a drawing approach with GKIN vs. a WIZARD approach with COPASI. This study cannot separate either approach from its implementation.

As expected the students almost universally preferred drawing the networks in GKIN as opposed to the Wizard interface of COPASI. As already noted, this preference could be due to the interface or the interface implementation in GKIN. We then asked if the time to enter the network was less for either interface using the results in Table 1. By a nonparametric Wilcoxon Signed Rank Test or Sign Test the P-value is between .008-.013 (Lehmann, 1975) with drawing the network being a faster entry system for specifying the network. It is possible that the two interfaces might have different error rates as summarized in Table 1. A Wilcoxon Signed Rank Test and the Sign test are not significant for the total of the errors catalogued in Table 1. For both GKIN and COPASI, the main source of errors in network specification was determining the stoichiometric coefficients (e.g., the Hill coefficients in Fig. 1) for the reactions. As a consequence we have added an example of how to do this in the GKIN online tutorial. We conclude that drawing the genetic network in GKIN is to be preferred over the Wizard Interface of COPASI and allows faster model specification with no difference in error rates relative to the Wizard interface of COPASI.

As a final note from the results in Table 1 it can be seen that students with no prior experience with either GKIN or COPASI could install and use both tools, thus indicating the general accessibility of the tools.

6 Conclusion

GKIN is a sophisticated open source, nearly platform independent, reaction network simulator using a previously-described transparent notation (Aleman-Meza et al., 2009) for designing arbitrary genetic networks satisfying mass action (by drag and drop) and which could easily be applied in a variety of other problem domains, including astronomy, pharmacology, and systems ecology. Besides its ease of use, the GKIN network designer enforces a correct semantic specification of a genetic network. While the drawing canvas for GKIN can specify an arbitrary reaction network satisfying mass action, it would be desirable to extend its capability by incorporating pre-existing modules as in PROMOT (Mirschel et al., 2009) or Marchisio and Stelling (2008). The models are also not stochastic (Adalsteinsson et al., 2004). The GKIN software is integrated with other tools with SBML (Hucka et al., 2003), with our existing simulator KINSOLVER (Aleman-Meza et al., 2009), and generates command lines to the simulator engine that might be integrated with other tools, such as SYCAMORE (Weidemann et al., 2008). The KINSOLVER engine is implemented in

a way to facilitate its incorporation in systems biology workflows and Web processes (Li et al., 2008; Zhang et al., 2009). GKIN has been essential in a flexible model program to understand carbon metabolism and the biological clock in Battogtokh et al. (2002) and Dong, et al. (2008).

Availability and Requirements

GKIN software with source code is freely available on the Web and does not require user registration for downloading at http://faculty.cs.wit.edu/~ldeligia/PROJECTS/GKIN/. The program requires java and gnuplot freely available on the Internet.

List of Abbreviations

ODE: ordinary differential equation; GUI, graphical user interface; CPU, central processing unit.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. QSB-0425762 and DBI-1062213, and we gratefully acknowledge NSF support. We thank BCMB 8211 and GENE 4220 students for evaluating the GKIN and COPASI software. We thank Megan Behringer for testing the software distribution Web site and completing the download of the software and making it operational without our assistance.

References

Adalsteinsson D, McMillen D, Elston TC. 2004. Biochemical Network Stochastic Simulator (BioNetS): software for stochastic modeling of biochemical networks. BMC Bioinformatics, 5: 1-24 (http://www.biomedcentral.com/1471-2105/5124)

Arnold J, Schuttler H-B, Logan D, et al. 2004. Metabolomics. In: Handbook of Industrial Mycology. 597-633, Marcel-Dekker, NY, USA

Aleman-Meza B, Yu YH, Schuttler HB, et al. 2009. KINSOLVER: A simulator for computing large ensembles of biochemical and gene regulatory networks. Computers and Mathematics with Applications, 57(3): 420-435

Barshop BA, Wrenn RF, Frieden C.1983. Analysis of numerical methods for computer simulation of kinetic processes: development of KINSIM-a flexible, portable system. Analytical Biochemistry, 130(1): 134-145

Battogtokh D, Asch DK, Case ME, et al. 2002. An ensemble method for identifying regulatory circuits with special reference to the qa gene cluster of *Neurospora crassa*. Proceedings of the National Academy of Sciences of the USA, 99(26): 16904-16909

Beadle GW, Tatum EL. 1941. Genetic Control of Biochemical Reactions in Neurospora. Proceedings of the National Academy of Sciences of the USA, 27(11): 499-506

Becker MY, Rojas I. 2001. A graph layout algorithm for drawing metabolic pathways. Bioinformatics, 17(5): 461-467

Bhalla US, Iyengar R. 1999. Emergent properties of networks of biological signaling pathways. Science, 283(5400): 381-387

Brown KS, Sethna JP. 2003. Statistical mechanical approaches to models with many poorly known parameters. Physics Review E, 68(2).

Clarke DC, Betterton MD, Liu X. 2006. Systems theory of Smad signalling. IEEE P Systems Biology, 153(6): 412-424

Cornish-Bowden A, Hofmeyr JH. 1991. MetaModel: a program for modelling and control analysis of

metabolic pathways on the IBM PC and compatibles. Comput Appl Biosci, 7(1): 89-93

DeRisi JL, Iyer VR, Brown PO. 1997. Exploring the metabolic and genetic control of gene expression on a genomic scale. Science, 278(5338): 680-686

Dong W, Tang X, Yu Y, et al. 2008. Systems biology of the clock in Neurospora crassa. PloS One, 3(8): e3105

Ehlde M, Zacchi G.1995. MIST: a user-friendly metabolic simulator. Comput Appl Biosci, 11(2): 201-207

Fell D. 1992. Metabolic control analysis: a survey of its theoretical and experimental development. Biochemistry Journal, 266: 313-330

Friedman N, Linial M, Nachman I, et al. 2000. Using Bayesian networks to analyze expression data. Journal of Computational Biology, 7(3-4): 601-620

Funahashi A, Matsuoka Y, Jouraku A, et al. 2008. CellDesigner 3.5: A versatile modeling tool for biochemical networks. IEEE P, 96(8): 1254-1265

Gardner TS, di Bernardo D, Lorenz D, et al. 2003. Collins JJ: Inferring genetic networks and identifying compound mode of action via expression profiling. Science, 301(5629): 102-105

Goryanin I, Hodgman TC, Selkov E. 1999. Mathematical simulation and analysis of cellular metabolism and regulation. Bioinformatics, 15(9): 749-758

Hoops S, Sahle S, Gauges R, et al. 2006. COPASI: a complex pathway simulator. Bioinformatics, 22(24): 3067-3074

Huang S. 1999. Gene expression profiling, genetic networks, and cellular states: an integrating conectp for tumorigenesis and drug discovery. Journal of Mol Med-Jmm, 77, 469-480

Hucka M, Finney A, Sauro HM, et al. 2003. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. Bioinformatics, 19(4): 524-531

Ibrahim SS, Eldeeb MAR, Rady MAH, et al. 2011. The role of protein interaction domains in the human cancer network. Network Biology, 1(1): 59-71

Ideker T, Thorsson V, Ranish JA, et al. 2001. Integrated genomic and proteomic analyses of a systematically perturbed metabolic network. Science, 292(5518): 929-934

Ishii N, Robert M, Nakayama Y, et al. 2004. Toward large-scale modeling of the microbial cell for computer simulation. Journal of Biotechnology, 113(1-3): 281-294

Ironi L, Panzeri L. 2009. A computational framework for qualitative simulation of nonlinear dynamical models of gene-regulatory networks. BMC Bioinformatics, 10(Suppl): 12: S14

Jain S, Bhooshan SV, Naik PK. 2011. Mathematical modeling deciphering balance between cell survival and cell death using insulin. Network Biology, 1(1): 46-58

Klamt S, Stelling J, Ginkel M, et al. 2003. FluxAnalyzer: exploring structure, pathways, and flux distributions in metabolic networks on interactive flux maps. Bioinformatics, 19(2): 261-269

Ko CL, Voit EO, Wang FS. 2009. Estimating parameters for generalized mass action models with connectivity information. BMC Bioinformatics, 10: 140

Kochut K, Arnold J, Sheth A, et al. 2003. IntelliGEN: A distributed workflow system for discovering protein-protein interactions. Distributed and Parallel Databases, 13(1): 43-72

Lehmann EL. 1975. Nonparametrics: statistical methods based on ranks. Holden-Day, San Francisico, CA, USA

Li P, Oinn T, Soiland S, et al. 2008. Automated manipulation of systems biology models using libSBML within Taverna workflows. Bioinformatics, 24(2): 287-289

Marchisio MA, Stelling J. 2008. Computational design of synthetic gene circuits with composable parts. Bioinformatics, 24(17): 1903-1910

Martinez-Antonio A. 2011. *Escherichia coli* transcriptional regulatory network. Network Biology, 1(1): 21-33

Mendes P. 1993. GEPASI: a software package for modelling the dynamics, steady states and control of biochemical and other systems. Comput Appl Biosci, 9(5): 563-571

Mendes P. 1997. Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. Trends in Biochemical Sciences, 22(9): 361-363

Mirschel S, Steinmetz K, Rempel M, et al. 2009. PROMOT: modular modeling for systems biology. Bioinformatics, 25(5): 687-689

Murphy K, Mian S. 1999. Modeling gene expression data using dynamic Bayesian networks. Technical Report, Berkeley, USA

Paris L, Bazzoni G. 2011.The polarity sub-network in the yeast network of protein-protein interactions. Network Biology, 1(3-4): 149-158

Purich DL, Allison RD. 2000. Handbook of biochemical kinetics. Academic Press, San Diego, CA, USA

Sauro HM. 1993. SCAMP: a general-purpose simulator and metabolic control analysis program. Comput Appl Biosci, 9(4): 441-450

Sauro HM, Hucka M, Finney A, et al. 2003. Next generation simulation tools: the Systems Biology Workbench and BioSPICE integration. Omics, 7(4): 355-372

Sauro HM, Small JR, Fell DA. 1987. Metabolic control and its analysis. Extensions to the theory and matrix method. European journal of biochemistry/FEBS, 165(1): 215-221

Schulz M, Bakker B, Klipp E. 2009. TIde: a software for the systematic scanning of drug targets in kinetic network models. BMC Bioinformatics, 10: 344

Shmulevich I, Dougherty ER, Kim S, et al. 2002. Probabilistic Boolean Networks: a rule-based uncertainty model for gene regulatory networks. Bioinformatics, 18(2): 261-274

Snoep J, Mendes P, Westerhoff H. 1999. Metabolic control analysis and kinetic modeling - towards a portable teaching module. The Biochemist, The Biochemical Society, 25-38

Sorokin A, Paliy K, Selkov A, et al. 2006. The pathway editor: a tool for managing complex biological networks. IBM Journal of Research and Development, 50: 561-574

Tacutu R, et al. 2011.Immunoregulatory network and cancer-associated genes: molecular links and relevance to aging. Network Biology, 1(2): 112-120

Tang XJ, Dong WB, Griffith J, et al. 2011. Systems Biology of the qa Gene Cluster in Neurospora crassa. PloS One, 6(6): e20671

Tomita M, Hashimoto K, Takahashi K, et al. 1999. E-CELL: software environment for whole-cell simulation. Bioinformatics, 15(1): 72-84

Tomita M, Hashimoto K, Takahashi K, et al. 2000. The E-CELL project: towards integrative simulation of cellular processes. New Generation Computing, 18: 1-12

Toni T, Welch D, Strelkowa N, et al. 2009. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. J R Soc Interface 6(31): 187-202

Vass M, Shaffer CA, Ramakrishnan N, et al. 2006. The JigCell model builder: a spreadsheet interface for creating biochemical reaction network models. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2: 155-164

Violin JD, DiPilato LM, Yildirim N, et al. 2008. beta(2)-adrenergic receptor signaling and desensitization elucidated by quantitative modeling of real time cAMP dynamics. Journal of Biological Chemistry,, 283(5): 2949-2961

Wachsstock DH, Pollard TD. 1994. Transient state kinetics tutorial using the kinetics simulation program, KINSIM. Biophysical Journal, 67(3): 1260-1273

Wagner A. 2001. How to reconstruct a large genetic network from n gene perturbations in fewer than n(2) easy

steps. Bioinformatics, 17(12): 1183-1197

Weaver D, Workman C, Stormo G. 1999. Modeling regulatory networks ith weight matrices. Pac Symp Biocomputing, 4: 112-123

Weidemann A, Richter S, Stein M, et al. 2008. SYCAMORE - a systems biology computational analysis and modeling research environment. Bioinformatics, 24(12): 1463-1464

Xiong MM, Li J, Fang XZ. 2004. Identification of genetic networks. Genetics, 166(2): 1037-1052

Yu Y, Dong W, Altimus C, et al. 2007. A genetic network for the clock of *Neurospora crassa*. Proceedings of the National Academy of Sciences of the USA, 104(8): 2809-2814

Zhang WJ. 2007. Computer inference of network of ecological interactions from sampling data. Environmental Monitoring and Assessment, 124: 253-261

Zhang WJ. 2012. A Java software for drawing graphs. Network Biology, 2(1): 38-44

Zhang Z, Cheung KH, Townsend JP. 2009. Bringing Web 2.0 to bioinformatics. Briefings in Bioinformatics, 10(1): 1-10