Software

Biochemical Network Stochastic Simulator (BioNetS): software for stochastic modeling of biochemical networks

David Adalsteinsson*[1], David McMillen[2] and Timothy C Elston[1]

Address: [1]Department of Mathematics, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599-3250, USA and [2]Department of Chemical and Physical Sciences, University of Toronto at Mississauga, Mississauga, ON L5L 1C6, Canada

Email: David Adalsteinsson* - david@amath.unc.edu; David McMillen - mcmillen@utm.utoronto.ca; Timothy C Elston - telston@amath.unc.edu

* Corresponding author

Abstract

Background: Intrinsic fluctuations due to the stochastic nature of biochemical reactions can have large effects on the response of biochemical networks. This is particularly true for pathways that involve transcriptional regulation, where generally there are two copies of each gene and the number of messenger RNA (mRNA) molecules can be small. Therefore, there is a need for computational tools for developing and investigating stochastic models of biochemical networks.

Results: We have developed the software package Biochemical Network Stochastic Simulator (BioNetS) for efficiently and accurately simulating stochastic models of biochemical networks. BioNetS has a graphical user interface that allows models to be entered in a straightforward manner, and allows the user to specify the type of random variable (discrete or continuous) for each chemical species in the network. The discrete variables are simulated using an efficient implementation of the Gillespie algorithm. For the continuous random variables, BioNetS constructs and numerically solves the appropriate chemical Langevin equations. The software package has been developed to scale efficiently with network size, thereby allowing large systems to be studied. BioNetS runs as a BioSpice agent and can be downloaded from http://www.biospice.org. BioNetS also can be run as a stand alone package. All the required files are accessible from http://x.amath.unc.edu/BioNetS.

Conclusions: We have developed BioNetS to be a reliable tool for studying the stochastic dynamics of large biochemical networks. Important features of BioNetS are its ability to handle hybrid models that consist of both continuous and discrete random variables and its ability to model cell growth and division. We have verified the accuracy and efficiency of the numerical methods by considering several test systems.

Background

Mathematical modeling of complex biological networks has a lengthy history [1-5]. In the past, the standard approach for modeling these systems has been to derive ordinary differential equations (ODEs) based on the law of mass action for the concentrations of the biochemical species involved in the network [6-16]. Experimental studies [17-19] have demonstrated, however, that sto-chastic effects can be significant in cellular reactions, particularly in the case of transcriptional regulation, where generally there are two copies of each gene and the number of messenger RNA (mRNA) molecules can be small. A number of recent experimental and modeling studies have addressed the role of fluctuations in gene expression [20-31]. Many modeling studies have employed the well-established Gillespie Monte Carlo

algorithm [32] or one of its more recent variants [33,34]. These algorithms offer an exact solution to the stochastic evolution of chemical systems, but they are computationally very expensive. A much more efficient approach is to approximate the species as continuous variables and formulate the problem in terms of stochastic differential equations (SDEs), often referred to as chemical Langevin equations [24,28,35]. This approximation works remarkably well for many cases, even when the number of particles involved is as small as ten, and the resulting simulations can run orders of magnitude more quickly than the discrete Monte Carlo approach. In other cases, when some or all of the particle numbers are very small, the system may need to be modeled using the discrete approach, or a hybrid method in which some species are treated discretely while others are evolved using the continuum approximation. With the increasing interest in formulating accurate models of large biochemical networks, there is a need for reliable software packages that correctly incorporate stochastic effects, yet are fast enough to simulate large interconnected sets of reacting species (as found, for example, in signaling cascades or genetic regulatory networks). We have developed the BIOchemical NETwork Stochastic Simulator, "BioNetS," to meet this need. BioNetS is capable of performing full discrete simulations using an efficient implementation of the Gillespie algorithm. It is also able to set up and solve the chemical Langevin equations, which are a good approximation to the discrete dynamics in the limit of large abundances. Finally, BioNetS can handle hybrid models in which chemical species that are present in low abundances are treated discretely, whereas those present at high abundances are handled continuously. Thus, the user can pick the simulation method that is best suited to their needs. All aspects of the software are highly optimized for efficiency.
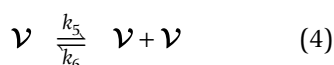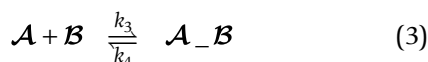
The remainder of this manuscript is arranged in the following way. In the Implementation section, the mathematical background for the Gillespie method, chemical Langevin equations and hybrid models is presented, along with a discussion of the numerical algorithms used in BioNetS. Under Results and Discussion we provide a brief introduction to BioNetS along with several examples. The examples serve two purposes: 1) to illustrate how to use the software and 2) to verify its efficiency and accuracy. More complete documentation can be found at http://x.amath.unc.edu/BioNetS, and in the documentation included with the package.

Implementation
We first develop the mathematical methodology on which BioNetS is built. Readers interested in using BioNetS without going into its underlying structure can proceed directly to the Results and discussion section.

Discrete reactions and the gillespie algorithm
BioNetS makes use of elementary reactions (zeroth, first and second order). The following examples illustrates each type of reaction:

$$\varnothing \; \underset{\delta}{\overset{\gamma}{\rightleftharpoons}} \; \mathcal{A} \qquad\qquad (1)$$

$$\mathcal{A} \; \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \; \mathcal{B} \qquad\qquad (2)$$

$$\mathcal{A} + \mathcal{B} \; \underset{k_4}{\overset{k_3}{\rightleftharpoons}} \; \mathcal{A}\_\mathcal{B} \qquad\qquad (3)$$

$$\mathcal{V} \; \underset{k_6}{\overset{k_5}{\rightleftharpoons}} \; \mathcal{V} + \mathcal{V} \qquad\qquad (4)$$

In the above reactions, the calligraphic letters denote a single molecule of a chemical species. The number of molecules of a particular species in the system at time t is denoted with uppercase letters (e.g., $A(t)$, $B(t)$, $A\_B(t)$, and $V(t)$). All the rate constants, γ, δ, and k_1-k_6, have units of per time. Eq. 1 represents a process in which a molecule A is produced when the reaction proceeds in the forward direction and is degraded in the reverse direction. In the forward direction the reaction is zeroth order and proceeds with an average rate of γ. In the backward direction, the reaction is first order, and the average rate of degradation is $\delta A(t)$. The forward reaction in Eq. 2 represents a process in which chemical species A is converted to species B. In this case A and B might represent two different conformations of the same molecule. In Eq. 2 both the forward and backward reactions are first order because the reaction rates are proportional to the respective concentrations. The forward reaction given in Eq. 3 is a second order reaction in which an A molecule and a B molecule come together to form the complex A_B. The average rate for the reaction is $k_1 A(t) B(t)$. The backward reaction is a first order reaction in which A_B dissociates at an average rate of $k_2 A\_B(t)$. In Eq. 4 the forward reaction produces a molecule V. The difference between this reaction and the forward reaction in Eq. 1 is that the average rate is $k_3 V(t)$. This leads to exponential growth of $V(t)$. This reaction is particularly useful if $V(t)$ is interpreted as the cell volume. In the backward reaction, two V molecules come together and degrade one of the V molecules. The average rate for this reaction is $k_4 V(t)(V(t) - 1)$. The $V(t) - 1$ term arises because two of $V(t)$ molecules must be chosen to react. This type of term also arises in reactions that produce homodimers. This reaction eventually stops the exponential growth of V. The net effect of these two reactions is to produce logistic growth. The total average reaction rate for the set of reactions given in Eqs. 1–4 is

$$\mu(t) = \gamma + \delta A(t) + k_1 A(t) + k_2 B(t) +$$
$$k_3 A(t)B(t) + k_4 A\_B(t) + k_5 V(t) + k_6 V(t)(V(t)-1)$$
$$= \sum_{i=1}^{4}(F_i + B_i) \qquad (5)$$

where F_i and B_i are the average forward and backward rates, respectively, for the *i*th reaction.

For the rest of this section, we assume that the volume of the cell is not changing and only consider Eqs. 1–3. In the Examples we consider a case in which the volume is changing. If $A(t)$, $B(t)$ and $A\_B(t)$ are present in large numbers, then the law of mass action can be applied to derive equations that govern the concentrations $[A] = A(t)/V$, $[B] = B(t)/V$ and $[AB] = A\_B(t)/V$, where V is the cell volume. These equations are

$$\frac{d[A]}{dt} = -(k_3'[B] + k_1 + \delta)[A] + k_4[AB] + k_2[B] + \gamma' \qquad (6)$$

$$\frac{d[B]}{dt} = -(k_3'[A] + k_2)[B] + k_1[A] + k_4[AB] \qquad (7)$$

$$\frac{d[AB]}{dt} = k_3'[A][B] - k_4[AB] \qquad (8)$$

The primed rate constants indicate that they have been appropriately scaled by the volume (i.e, $k_3' = k_3 V$ and $\gamma' = \gamma/V$), and, therefore, have units of either per time per concentration or concentration per time. Note that to convert to units of molar, we also have to appropriately scale the rate constants by Avagadro's number. Eqs. 6–8 represent a macroscopic description of the process, because they ignore fluctuations in the concentration that arise from the stochastic nature of chemical reactions.

In general, $A(t)$, $B(t)$ and $A\_B(t)$ are random variables that take on any nonnegative integer value. The Gillespie algorithm [32] can be used to generate sample paths of the process. This algorithm assumes that the random time ΔT_i, between the *i*th and $i + 1$ reaction, is exponentially distributed. For the simple example given by Eqs. 1–3, the mean waiting time between reactions, which characterizes the exponential distribution, is $\mu_{\Delta Ti} = \gamma + \delta A(t_i) + k_1 A(t_i) + k_2 B(t_i) + k_3 A(t_i)B(t_i) + k_4 A\_B(t_i)$, where t_i is the time at which the ith reaction occurred. Therefore, $t_i^{+1} = t_i + \Delta T_i$. Once the time at which the next reaction occurs has been determined, the following probabilities are used to determine which reaction occurred:

$$\Pr[\varnothing \xrightarrow{\gamma} \mathcal{A}] = \frac{\gamma}{\mu_{\Delta T_i}} \qquad (9)$$

$$\Pr[\mathcal{A} \xrightarrow{\delta} \varnothing] = \frac{\delta A(t_i)}{\mu_{\Delta T_i}} \qquad (10)$$

$$\Pr[\mathcal{A} \xrightarrow{k_1} \mathcal{B}] = \frac{k_1 A(t_i)}{\mu_{\Delta T_i}} \qquad (11)$$

$$\Pr[\mathcal{B} \xrightarrow{k_2} \mathcal{A}] = \frac{k_2 B(t_i)}{\mu_{\Delta T_i}} \qquad (12)$$

$$\Pr[\mathcal{A} + \mathcal{B} \xrightarrow{k_3} \mathcal{A}\_\mathcal{B}] = \frac{k_3 A(t_i)B(t_i)}{\mu_{\Delta T_i}} \qquad (13)$$

$$\Pr[\mathcal{A}\_\mathcal{B} \xrightarrow{k_4} \mathcal{A} + \mathcal{B}] = \frac{k_4 A\_B(t_i)}{\mu_{\Delta T_i}} \qquad (14)$$

Once the reaction has been determined, the chemical species are updated accordingly. As discussed in the Numerical methods section, BioNetS uses an efficient implementation of the Gillespie algorithm [33].

Another description of discrete stochastic processes is achieved through use of the master equation that governs how the probabilities of the various random variables in the process evolve in time. Let $p_{a, b, a\_b}(t) = \Pr [A(t) = a, B(t) = b, A\_B(t) = a\_b]$, then $P_{a,b,a\_b}(t)$ satisfies the master equation

$$\frac{dp_{a,b,a\_b}}{dt} = -[\gamma + (\delta + k_1)a + k_2 b + k_3 ab + k_4 a\_b]p_{a,b,a\_b} + \gamma p_{a-1,b,a\_b} + \delta(a+1)p_{a+1,b,a\_b}$$
$$+ k_1(a+1)p_{a+1,b-1,a\_b} + k_2(b+1)p_{a-1,b+1,a\_b}$$
$$+ k_3(a+1)(b+1)p_{a+1,b+1,a\_b-1} + k_4(a\_b+1)p_{a-1,b-1,a\_b+1} \qquad (15)$$

The master equation is the starting point for deriving various approximate schemes for describing the system [28]. In the next section, we discuss an approximate scheme that is valid in the limit of large, but finite molecule numbers. The simplest approximation scheme is achieved by considering the first moments of the process. We will use over bars to denote averaging. For example, $\bar{A}(t) = \sum_{a,b,a\_b} a p_{a,b,a\_b}(t)$. Eq. 15 can be used to derive equations that govern the time evolution of all the first moments. Because of the second order reaction in Eq. 3, the equations for the means are coupled to the second moments. In fact, the *n*th moment equations contain terms that involve the $n+1$ moments. Thus, there is no closure to the system. The simplest closure scheme is to assume that all moments factorize (e.g., $\overline{A^2} = \bar{A}^2$). This represents the macroscopic limit in which fluctuations are ignored. In this limit, we recover Eqs. 6–8 from the master equation.

The diffusion limit and the chemical langevin equations

The general form of the master equation for a system consisting of N chemical species and M reactions is

$$\frac{dp_{\mathbf{n}}}{dt} = \sum_{i=1}^{M}(F_i + B_i)p_{\mathbf{n}} + \sum_{i=1}^{M}F_i p_{\mathbf{n}-\delta_i} + \sum_{i=1}^{M}B_i p_{\mathbf{n}+\delta_i} \qquad (16)$$

where $\mathbf{n}$ is a N-dimensional vector of species numbers, F_i and B_i are the backward and forward rates for the ith reaction, and the vectors δ_i contain the stoichiometric constants for the ith reaction. For the simple model given by Eqs. 1–3, $N = 3$, $M = 3$, and $p_{\mathbf{n}}(t) = \Pr[A(t) = n_1, B(t) = n_2,$ and $A\_B(t) = n_3]$. The forward and backward rates are $F_1 = \gamma$, $B_1 = \delta n_1$, $F_2 = k_1 n_1$, $B_2 = k_2 n_2$, $F_3 = k_3 n_1 n_2$, and $B_3 = k_4 n_3$. The δ_i vectors are the rows of the stoichiometric matrix

$$\mathbf{\Delta} = \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & -1 & 1 \end{pmatrix} \qquad (17)$$

The (i,j) element in the above matrix represents the change in the jth chemical species when the ith reaction proceeds in the forward direction.

If the molecule numbers are large as compared to 1, then the master equation Eq. 16 can be approximated by the continuous process [28,35]

$$\frac{\partial \rho(\mathbf{n},t)}{\partial t} = -\sum_{j}^{N}\frac{\partial}{\partial n_j}A_j(\mathbf{n})\rho(\mathbf{n},t) + \frac{1}{2}\sum_{j,k}^{N}\frac{\partial^2}{\partial n_j \partial n_k}D_{j,k}(\mathbf{n})\rho(\mathbf{n},t) \qquad (18)$$

where

$$A_j(\mathbf{n}) = \sum_{i}^{M}\Delta_{j,i}(F_i - B_i) \qquad (19)$$

$$D_{j,k}(\mathbf{n}) = \sum_{i}^{M}\Delta_{i,j}\Delta_{i,k}(F_i + B_i) \qquad (20)$$

This result can be derived in several ways. One method is to note that Eq. 15 represents a second order finite differencing of Eq. 18, with a grid size of 1. Another method is to make use of the shift operator

$$f(n+k) = \exp\left(k\frac{\partial}{\partial n}\right)f(n) = \sum_{j=0}^{\infty}\frac{k^j}{j!}\frac{\partial^j}{\partial n^j}f(n) \qquad (21)$$

where $f(n)$ is an arbitrary smooth function and for our purposes k is an integer. If the shift operator is used in Eq. 15, the diffusion limit is achieved when the Taylor series expansion given in Eq. 21 is truncated at $j = 2$.

Sample paths consistent with Eq. 18 can be generated using the following set of SDEs

$$\frac{dN_j}{dt} = A_j(\mathbf{N}) + \sum_{k=1}^{M}\Delta_{k,j}\sqrt{F_k(\mathbf{N}) + B_k(\mathbf{N})}w_k(t) \qquad (22)$$

where the $w_k(t)$ are independent Gaussian white noise processes. These equations are often referred to as the chemical Langevin equations. For Eqs. 1 – 3, the explicit form of the SDEs are

$$\frac{dA}{dt} = -(k_1 + k_3 B + \delta)A + k_2 B + k_4 A\_B + \gamma' + \sqrt{\delta A + \gamma}w_1(t)$$
$$-\sqrt{k_1 A + k_2 B}w_2(t) - \sqrt{k_3 AB + k_4 A\_B}w_3(t) \qquad (23)$$

$$\frac{dB}{dt} = -(k_2 + k_3 A)B + k_1 A + k_4 A\_B + \sqrt{k_1 A + k_2 B}w_2(t)$$
$$-\sqrt{k_3 AB + k_4 A\_B}w_3(t) \qquad (24)$$

$$\frac{dA\_B}{dt} = -k_4 A\_B + k_3 AB + \sqrt{k_3 AB + k_4 A\_B}w_2(t) \qquad (25)$$

BioNetS generates numerical solutions to the SDEs given by Eq. 22 using either an explicit or semi-implicit Euler method. The form of these methods is

$$N_j(t+\Delta t) = N_j(t) + \Delta t A_j(\mathbf{N}(t+\epsilon\Delta t))$$
$$+\sqrt{\Delta t}\sum_{k=1}^{M}\Delta_{k,j}\sqrt{F_k(\mathbf{N}(t)) + B_k(\mathbf{N}(t))}Z_k(t) \qquad (26)$$

where $\epsilon = 0$ for the explicit method and $\epsilon = 1$ for the semi-implicit method and the $Z_k(t)$ are independent standard normal random variables. The advantage of using the chemical Langevin equations is that in the appropriate parameter regime, numerical solutions to the set of SDEs given by Eq. 22 can be generated much more efficiently than using the Gillespie algorithm. We expand upon this point in the Examples section. Higher order numerical algorithms for SDEs are available [36], but the noise structure of the chemical Langevin equations makes these schemes very cumbersome to implement. In the Examples, we verify that the Euler methods given by Eq. 26 are sufficient to produce reliable results. We note that the Δ matrix is generally sparse, and BioNetS takes advantage of this sparseness to optimize the efficiency of the two Euler methods (see Numerical Methods, below).

Hybrid schemes

It is often desirable to allow some of the chemical species to be treated as continuous random variables and some to be treated discretely. This is particularly true for the case of transcriptional regulation by transcription factors. In this situation there can be as few as one DNA/transcription factor binding site and mRNA abundances can be as small

as 10 or fewer. In contrast, protein abundances can be in the thousands. The technical difficulty with implementing hybrid schemes that include both discrete and continuous random variables is that the Gillespie method requires constant transition rates between reactions. This may not be the case, if some of the chemical species are evolving continuously in time. BioNetS overcomes this problem in one of two ways.

Let $N_d < N$ be the number of discrete chemical species and $M_d \leq M$ the number of reactions that produce a change in one of the N_d chemical species. The overall reaction rate at time t_j for the discrete set of chemical species is

$$\mu_{t_j} = \sum_{i=1}^{M_d} [F_i(t_j) + B_i(t_j)] \qquad (27)$$

If the time step Δt for the SDEs is small enough such that

$$p_t = \mu_{t_j} \Delta t < \varepsilon \ll 1 \qquad (28)$$

then p_t is approximately the probability of a transition in Δt. In the above equation ε is a user specified tolerance. The probability of two discrete transitions in Δt is proportional to $(\Delta t)^2$. Choosing $\varepsilon < 0.1$, which means the probability of two reactions in Δt is less than 0.01, generally produces good results. However, this should be verified on a case by case basis. At each time step, BioNetS checks to verify that Ineq. 28 is satisfied for the specified ε. If so, a uniform random number R is generated and compared against p_t. If $R < p_t$, then a transition occurred and the conditional probability R/p_t is used to determine which of the discrete transitions occurred. If $p_t > \varepsilon$, then the discrete reactions determine the fastest time scale in the system. In this case the Gillespie algorithm is used to update the discrete reactions, and the random time step Δt_j is used to update the SDEs.

A pseudo-code description of the above algorithm is displayed in Table 3:

Numerical methods
BioNetS generates code that is tailored to efficiently simulate biochemical reactions. The optimization techniques used by BioNetS allows the software to simulate large systems in reasonable times without requiring high-end computational hardware.

Techniques used to optimize the Gillespie method are:

• For the discrete variables, the program uses data structures that allow only the chemical species and reaction rates that are affected by the current reaction to be updated.

• A bisection search is used to determine which reaction occurred.

The code has both an explicit and a semi-implicit solver, for simulating the chemical Langevin equations. The user specifies at runtime which method to use. By default the semi-implicit solver will be used. The semi-implicit solver uses Newton's method to solve the implicit equations, and for that the program needs to compute the Jacobian and solve a linear system at each iteration. For updating the chemical Langevin equations and hybrid models optimization techniques include:
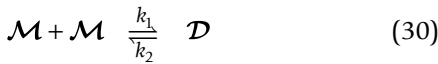
• The sparse nature of the stoichiometric matrix is used to efficiently store and per form matrix operations.

• After every reaction, only the species and reaction rates affected by that reaction are updated. This can be seen in the Rates.cpp file, where all the different cases have been worked out and written for optimal execution speed.

• The Jacobian is sparse, and the code takes full advantage of this fact. The program solves and factorizes the Jacobian using sparse methods. Before the code generation, BioNetS computes the entries in the Jacobian symbolically and finds a permutation that decreases the number of fill-ins during the LU factorization. As a result, no zero entries are saved, and the sparse structure is fully exploited. The sparse structure is then used in the LU solve. In the code, no pivots are visible, and no if-statements are left.

Results and discussion
In this section we present several examples which serve as illustrations of how to use BioNetS and test the accuracy and efficiency of the numerical methods. One particular concern is the accuracy of the Euler methods. While these methods are only of order $\sqrt{\Delta t}$, we show that when the approximations that lead to the chemical Langevin equations are valid, the difference between the numerical solutions of the SDEs and the exact discrete Gillespie method are negligible. Currently, the graphical user interface to BioNetS runs on the Macintosh OS X operating system, though the software will generate portable C/C++ code that can be compiled and run in any computing environment. The files needed to install and run BioNetS can be downloaded from http://x.amath.unc.edu/BioNetS. The following examples illustrate the way in which models are entered and run in BioNetS. More detailed documentation is available with the software package.

Dimerization
We begin with a simple system that consists of the following two reactions:

$$\varnothing \; \underset{\delta_m}{\overset{\gamma}{\rightleftharpoons}} \; \mathcal{M} \qquad (29)$$

$$\mathcal{M} + \mathcal{M} \; \underset{k_2}{\overset{k_1}{\rightleftharpoons}} \; \mathcal{D} \qquad (30)$$

$$\mathcal{D} \xrightarrow{\delta_d} \varnothing \qquad (31)$$

In this system, monomer molecules M are produced at an average rate γ and degraded at an average rate $\delta_m M(t)$. Two monomers can then bind to form a dimer molecule D. The average forward and backward rates for the this reaction are $k_1 M(t)(M(t) - 1)$ and $k_2 D(t)$, respectively. The dimers are degraded at a rate δ_d. We will treat two cases. In the first case the cell volume is assumed to be constant, and in the second case the cell is allowed to grow and divide. To model cell growth, the cell volume V_c is treated as a random variable $V_c = \alpha V$, where V is a non-negative discrete random variable and α represents a unit of volume. The random variable V is governed by the reaction

$$\mathcal{V} \xrightarrow{k_3} \mathcal{V} + \mathcal{V} \qquad (32)$$
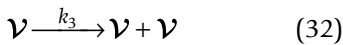
The above reaction causes V to grow exponentially fast with an average rate of k_3. Note that logistic growth is produced when the backward reaction in Eq. 32 is included.

Constant volume
We start by considering the simple case in which the volume of the cell remains constant. To use BioNetS follow these steps. Copy BioNetS onto your machine, and double click to launch. Help is included as part of the program, and accessed from the Help menu. The Help document will walk you through all the steps needed to enter reactions and run the simulator.

The user interface asks you to enter the reaction and corresponding rate constants in the top part of the script window as shown in Fig. 1a. In the bottom part of the script window, you can toggle between panels. The Species panel is shown in Fig. 1a and allows the user to specify how the simulator treats each chemical species, discrete or continuous. The Constants panel lists the order in which the rate constants are referenced. The Output panel allows the user to specify the ouput type. There are two ways to generate program output, either binary or ASCII. Binary output is based on MATLAB binary files, so it is possible to drive the program with MATLAB and use MATLAB's plotting routines to view the output. It is also possible to generate time series and histograms of the data from within BioNetS. Using ASCII files for I/O allows the simulator to be run through shell scripts. The Executable panel allows the user to generate either an executable file or source code. BioNetS generates portable C/C++ code

that can be compiled and run in any computing environment. BioNetS can directly compile the C/C++ code. However, this requires the Developer tools, included on all recent Apple machines and available directly from http://developer.apple.com for free. The compiled code can then be run from within BioNetS. The Comments panel is available for the user to enter descriptive comments about the model.

To run BioNetS as a BioSpice agent, you need to move the source directory onto a OAA-supported system. Once there, open up the MakeOAA file and specify the locations of your oaalib folder. Then just type "make -f MakeOAA" (without the quotes) to create the agent.

Figures 2A and 2B show plots of time series for the monomer number generated by BioNetS. The parameter values used to generate these figures are given in the caption. Figure 2A is the result obtained when M and D are treated as discrete variables. Figure 2B is the result from the chemical Langevin equations. The solid line shown in both panels is the result from the following equations for the first moments:

$$\frac{d\bar{M}}{dt} = -2k_1\bar{M}^2 - \delta_m\bar{M} + 2k_2\bar{D} + \gamma \qquad (33)$$

$$\frac{d\bar{D}}{dt} = -k_2\bar{D} + k_1\bar{M}^2 - \delta_d\bar{D} \qquad (34)$$

Figure 3A shows the probability density function (PDF) of the dimer concentration at various times for the discrete and continuous case. Notice that for all times, the agreement between the two different methods is very good. At the final time, $t = 200$ s, the system has reached steady state. These figures indicate that the chemical Langevin equations are accurately capturing the dynamics and steady-state behavior of the discrete system.

Cell growth and division
In this section we describe how cell growth and division can be modeled using BioNetS. We will assume that the cell is experiencing exponential growth up until the time it divides. As discussed above, the cell volume $V_c = \alpha V$ is treated as a random variable. In this model cell division occurs when V exceeds a threshold value V_{max}. Note that the choice of V_{max} influences the degree of variability observed in the cell division times: cells growing from $V = 1$ to $V_{max} = 2$ will have a large amount of variability in their division times, while those growing from $V = 100$ to $V_{max} = 200$ will have less variable times, and those ranging from $V = 1000$ to $V_{max} = 2000$ will be still less variable. Changing the range of V in this way requires rescaling the relationship of V to the cell volume by adjusting the value
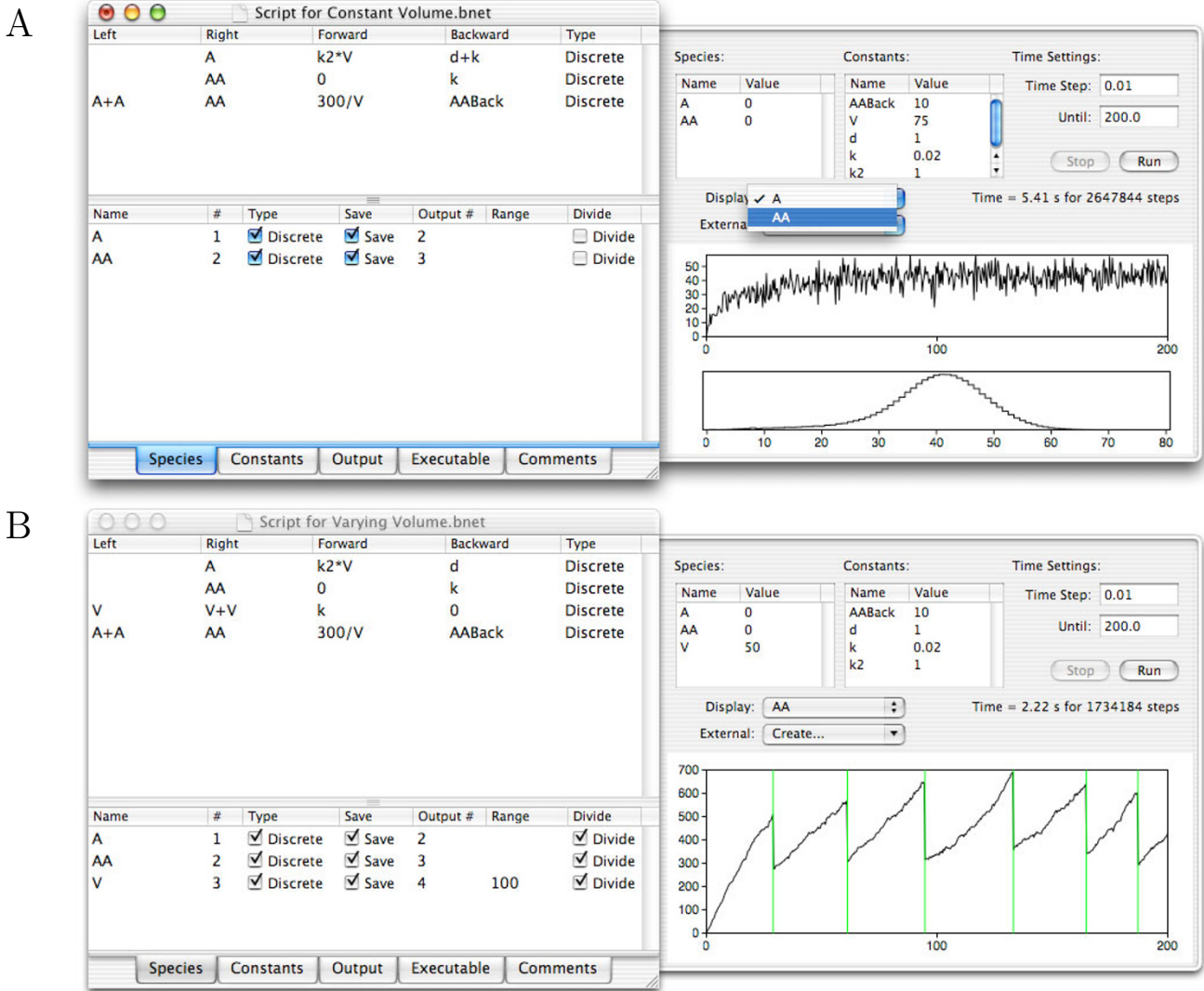
Figure 1
A) A screen shot of BioNetS that illustrates how the dimerization example with constant volume is entered. The tabs at the bottom of the screen allow the user to enter various options. B) A screen shot that illustrates how the dimerization example with cell growth and division is entered into BioNetS. In each screen shot, the area to the right of the reaction entry interface is a a slide-out testing panel that allows the user to enter rate constants and initial conditions, then view the results of a run from within BioNetS.

of α. When cell division occurs the volume is halved, and the proteins are randomly divided between the two cells using a binomial distribution. Only one of the daughter cells is tracked. Because second order reactions require two molecules to collide, the rate constants for these reactions should scale like $k_1 = k'_1/V_c$. We also assume that the production rate of monomers scales as $\gamma = \gamma'V_c$. This is a reasonable assumption, because as the cell grows the transcription and translation machinery increases. These assumptions produce the following rate equations for the concentrations

$$\frac{d[M]}{dt} = -2k'_1[M]^2 + 2k_2[D] + \gamma' - (\delta_m + k_3)[M] \qquad (35)$$

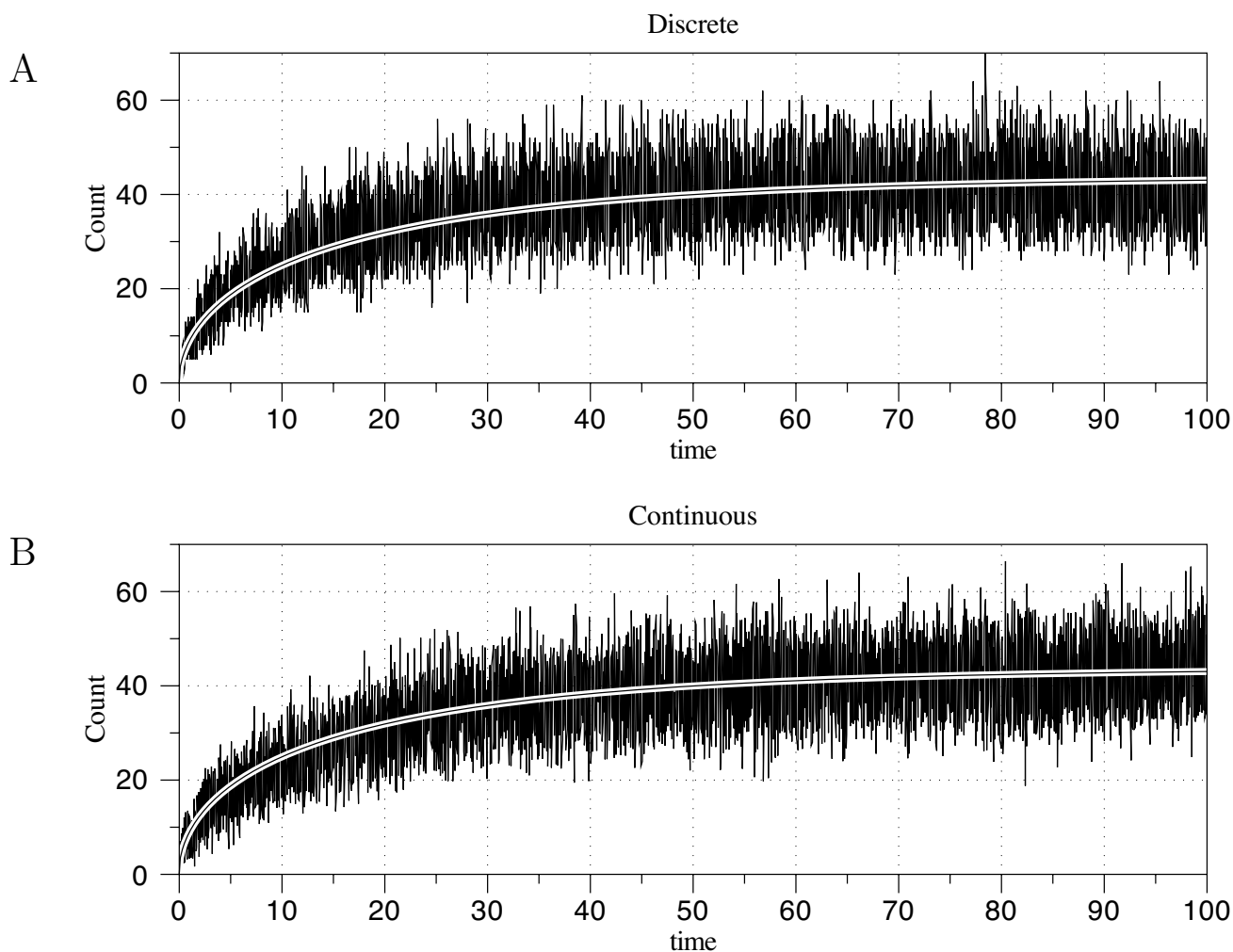$$\frac{d[D]}{dt} = k'_1[M]^2 - k_2[D] - (k_3 + \delta_d)[D] \qquad (36)$$

Figure 2
A) A single realization of $M(t)$ for the discrete process. B) A single realization of $M(t)$ produced by the chemical Langevin equations. The solid line in both panels is the result produced from Eqs. 33 and 34. The parameter values used to generate these figures were $\gamma = 50$, $\delta_m = 1.02$, $\delta_d = 0.02$, $k_1 = 0.01$, and $k_2 = 0.1$. The initial conditions used were $M(0) = 0$ and $D(0) = 0$ and a time step of 0.01 was used for the continuous case.

$$\frac{dV_c}{dt} = k_3 V_c \qquad (37)$$

The terms in Eqs. 35 and 36 that involve k_3 arise because of dilution due to cell growth. We use the same parameter values as in the constant volume case except $\delta_m = 1$ and $\delta_d = 0$. The cell growth rate is $k_3 = 0.02$ (assuming a scaling of 1 time unit to one minute, this yields an average cell division time of $\ln 2/k_3 \simeq 35$ minutes, typical for bacteria), the scale factor for the cell volume is $\alpha = 1$ (for simplicity), and $V_{max} = 100$. With these choices of parameter values, Eqs. 35 and 36 are identical with Eqs. 33 and 34, and we

expect the average behavior of this system to be similar to that of the constant volume case.

The screen shot shown in Fig. 1B illustrates how this model is entered into BioNetS. Figure 4A shows the time series for the volume and monomer number treating each variable discretely. The results for the continuous case are virtually identical. In Fig. 4B the concentration is plotted as a function of time. This figure should be compared with Fig. 2A. The solid line in Fig. 4B is the result from solving Eqs. 35–37. Figure 3B shows the PDFs for the dimer concentration at various times. Both the discrete and continuous results are shown. By comparing Fig. 3A with Fig. 3B,
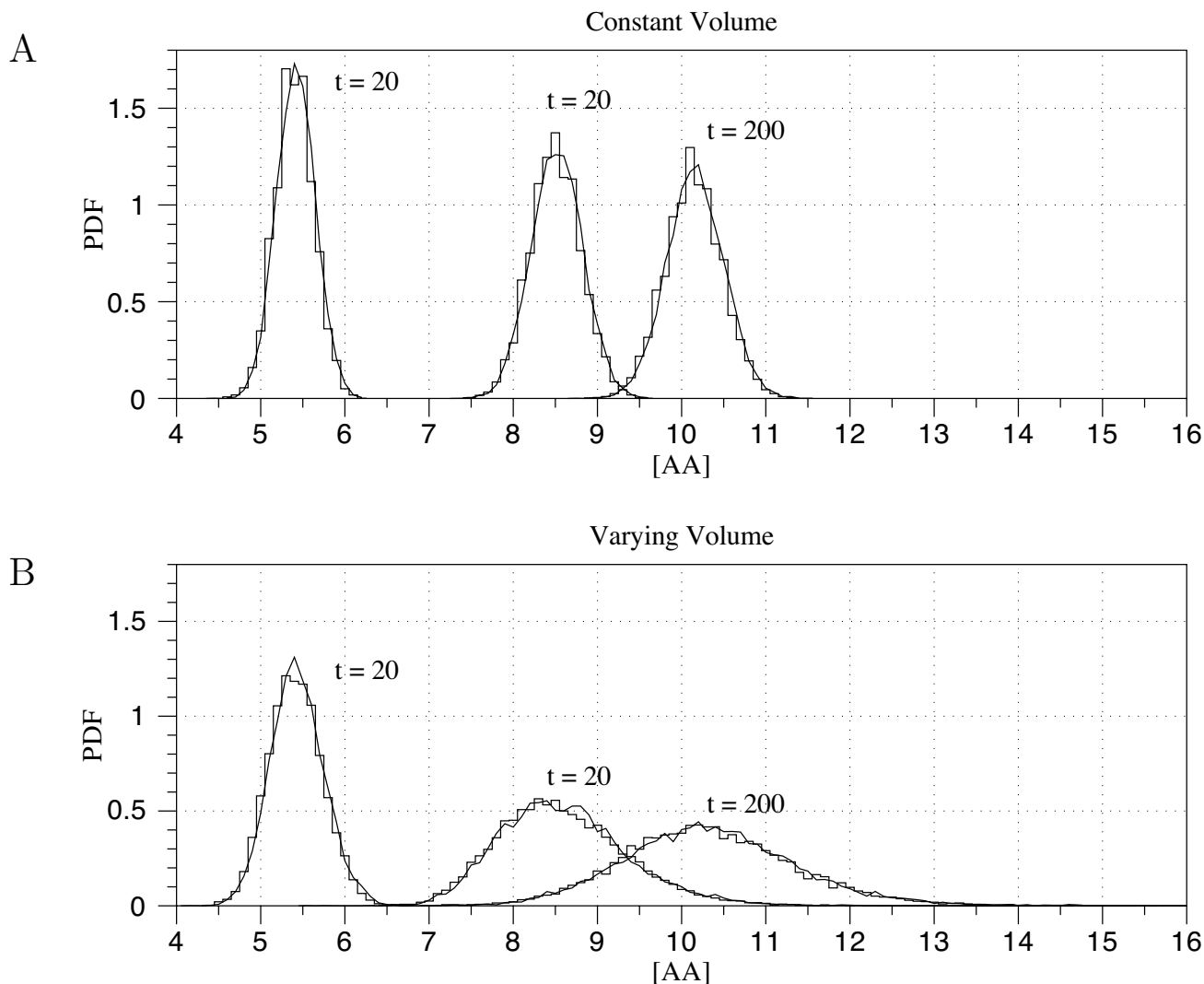
Figure 3
PDFs for the dimer concentration at various times. The staircase plots are the results for the discrete case and the continuous lines are the results for the continuous case. Panel A is for the case of constant volume and Panel B is the varying volume case. The parameter values and initial conditions are the same as in Figs. 2 and 4, for the constant volume case and varying volume case, respectively. Each PDF consists of 10, 000 realizations of the process.

we see not surprisingly that for this simple example the main effect of volume growth is to act as an additional noise source and increase the variability of the distributions.

A chemical oscillator

We next use BioNetS to simulate a two gene system that has been studied in the literature [37]. In this system, the protein A coded for by gene a acts as an activator for gene a and gene r, by binding to the promoter regions, P_a and

P_r, of the respective gene. This increases the rate of $mRNA_a$ and $mRNA_r$ production by a factor α_a and α_r, respectively. The protein R, acts as a represser for both genes by binding to A to form the inactive complex A_R. All gene products, mRNA and protein, are actively degraded. However, the heterodimer A_R protects the R subunit from degradation. The system consists of 9 chemical species and the following 14 biochemical reactions:
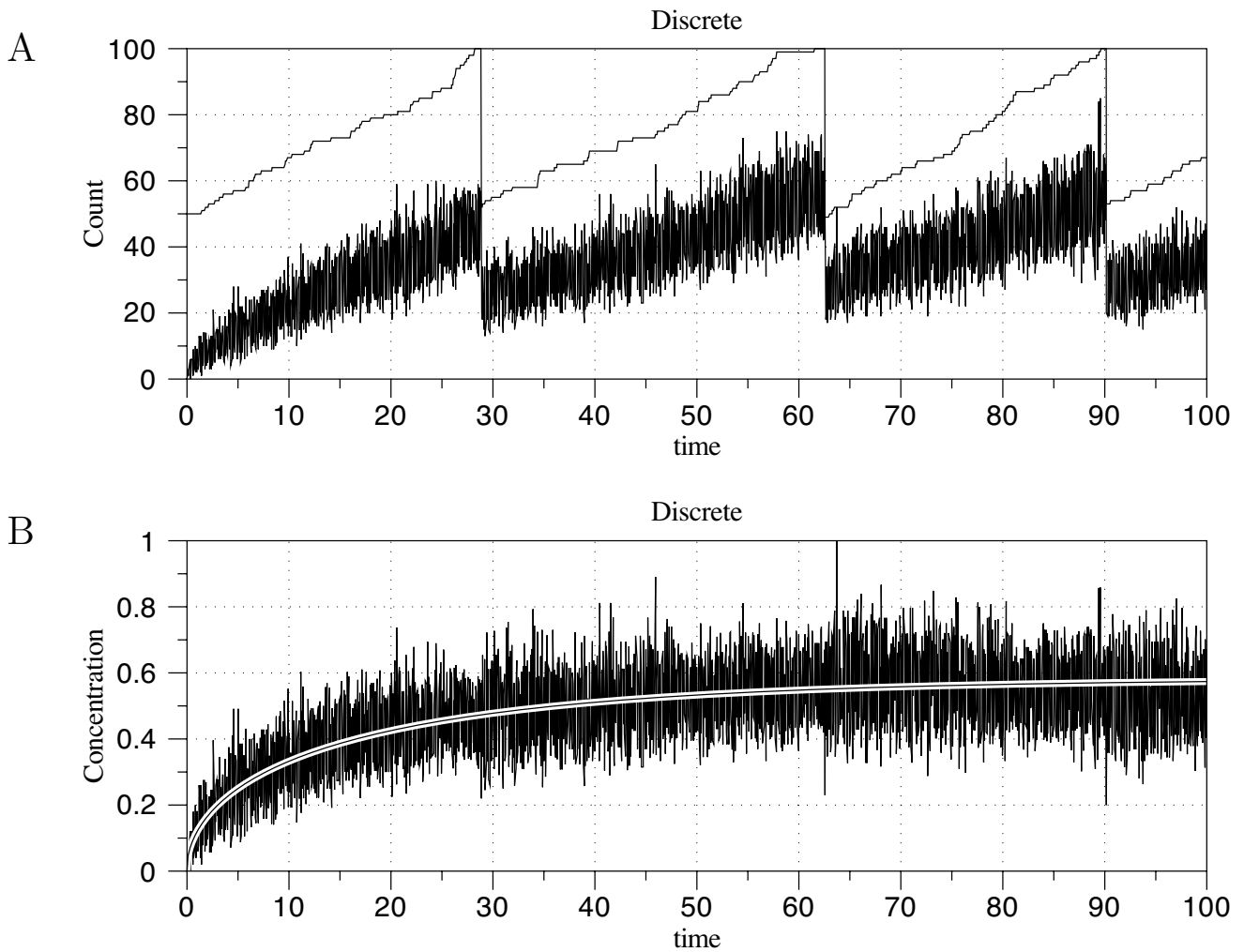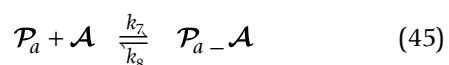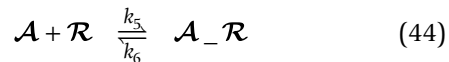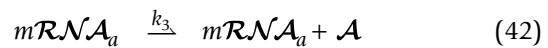
Figure 4
A) Time series for the volume and monomer number for the case of cell growth and division. The parameter values used to generate these figures were $\gamma = 50$, $\delta_m = 1.0$, $\delta_d = 0.0$, $k_1 = 0.01$, $k_2 = 0.1$, $k_3 = 0.02$, and $V_{max} = 100$. The initial conditions used were $M(0) = 0$, $D(0) = 0$ and $V(0) = 50$. B) The monomer concentration $M(t)/V(t)$ as a function of time. The solid line is the result from Eqs. 35–36.
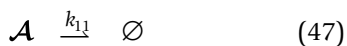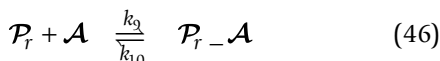
$$\mathcal{P}_a \xrightarrow{k_1} \mathcal{P}_a + m\mathcal{RNA}_a \qquad (38)$$

$$\mathcal{P}_a\_\mathcal{A} \xrightarrow{\alpha_a k_1} \mathcal{P}_a\_\mathcal{A} + m\mathcal{RNA}_a \qquad (39)$$

$$\mathcal{P}_r \xrightarrow{k_2} \mathcal{P}_r + m\mathcal{RNA}_r \qquad (40)$$

$$\mathcal{P}_r\_\mathcal{A} \xrightarrow{\alpha_r k_2} \mathcal{P}_r\_\mathcal{A} + m\mathcal{RNA}_r \qquad (41)$$

$$m\mathcal{RNA}_a \xrightarrow{k_3} m\mathcal{RNA}_a + \mathcal{A} \qquad (42)$$

$$m\mathcal{RNA}_r \xrightarrow{k_4} m\mathcal{RNA}_r + \mathcal{R} \qquad (43)$$

$$\mathcal{A} + \mathcal{R} \underset{k_6}{\overset{k_5}{\rightleftharpoons}} \mathcal{A}\_\mathcal{R} \qquad (44)$$

$$\mathcal{P}_a + \mathcal{A} \underset{k_8}{\overset{k_7}{\rightleftharpoons}} \mathcal{P}_a\_\mathcal{A} \qquad (45)$$

$$\mathcal{P}_r + \mathcal{A} \;\; \underset{k_{10}}{\overset{k_9}{\rightleftharpoons}} \;\; \mathcal{P}_r\_\mathcal{A} \qquad\qquad (46)$$

$$\mathcal{A} \;\; \overset{k_{11}}{\rightarrow} \;\; \varnothing \qquad\qquad (47)$$

$$\mathcal{R} \;\; \overset{k_{12}}{\rightarrow} \;\; \varnothing \qquad\qquad (48)$$

$$m\mathcal{RNA}_a \;\; \overset{k_{13}}{\rightarrow} \;\; \varnothing \qquad\qquad (49)$$

$$m\mathcal{RNA}_r \;\; \overset{k_{14}}{\rightarrow} \;\; \varnothing \qquad\qquad (50)$$

$$\mathcal{A}\_\mathcal{R} \;\; \overset{k_{15}}{\rightarrow} \;\; \mathcal{R} \qquad\qquad (51)$$

Figure 5 shows the way this system is entered into BioNetS.
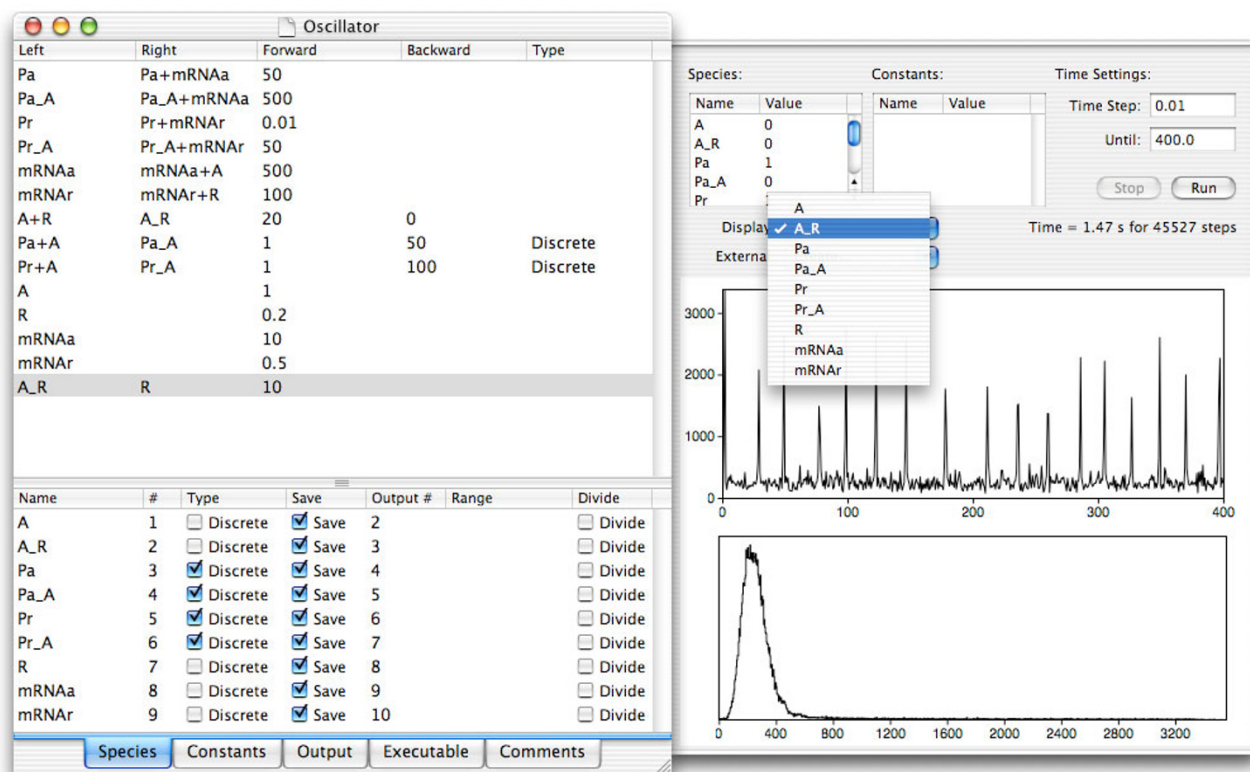
Figure 5
A screen shot of the BioNetS user interface for the chemical oscillator example. The parameter values shown in the figure are the ones used to produce the results shown in Figs. 6, 7, 8. The area to the right of the reaction entry interface is a a testing panel provided by BioNetS to allow the user to enter rate constants and initial conditions, then examine the results of a run.

An interesting feature of the system is that it is capable of producing sustained oscillations [37]. Figure 6A shows a times series for the repressor protein number when all the chemical species are treated as discrete random variables. The values of the rate constants used to generate this figure are listed in Fig. 5.

The chemical species P_a, P_r, P_r_A, and P_r_A are binary random variables: they can only take on the values 0 or 1.

Therefore, these species can not be approximated as continuous random variables. All the other chemical species appear in sufficient quantities to justify the continuum approximation. Figure 6B shows a time series corresponding to Fig. 6A using the hybrid model. The hybrid model was run using the semi-implicit Euler method, and for these parameter values, runs 3 times faster than full model. Visually, the agreement between the two methods appears good. To test the accuracy of the Euler method, we
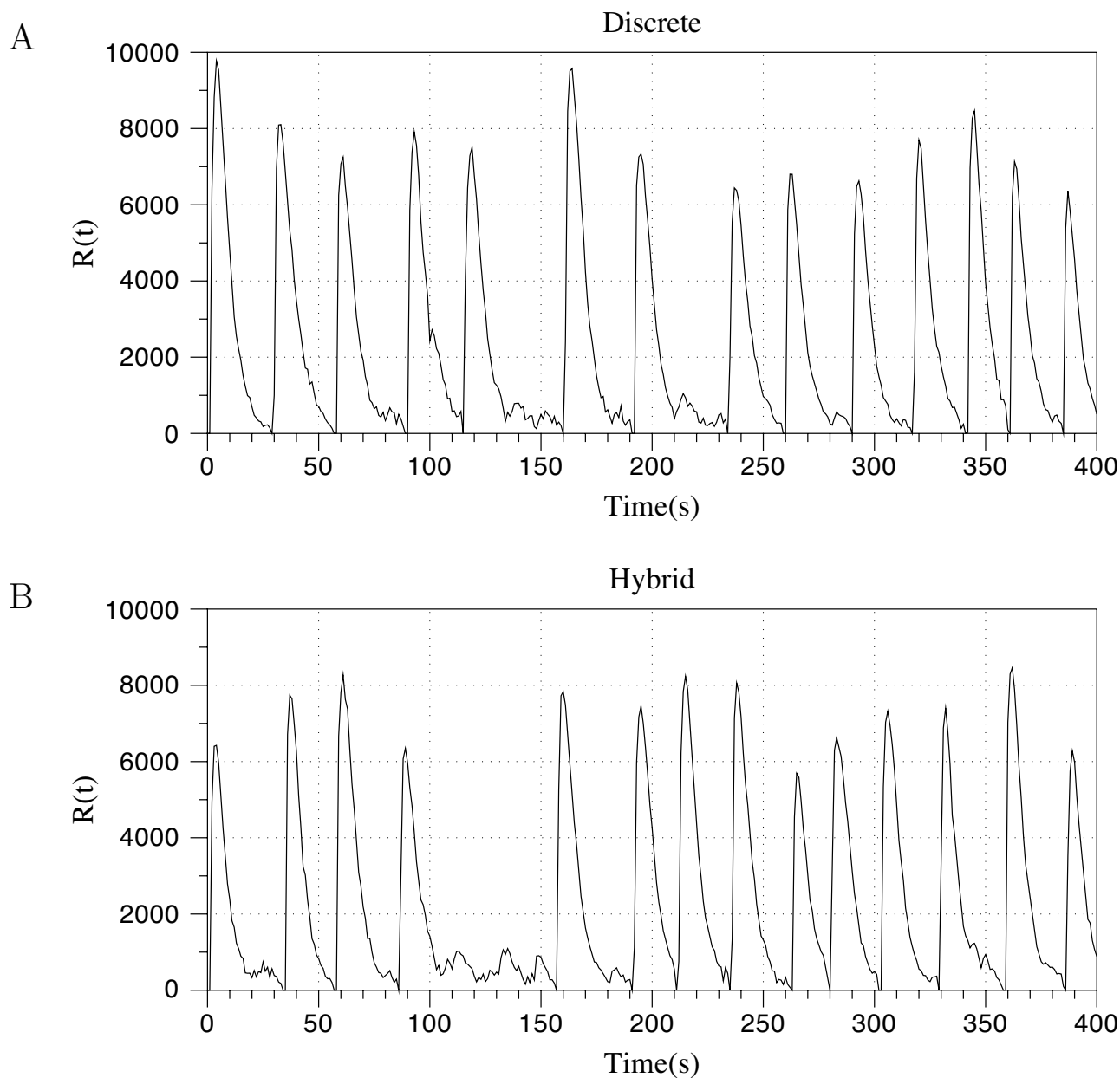
Figure 6
Sample paths for the repressor protein number. Panel A is the fully discrete case and Panel B is the hybrid model.

used BioNetS to construct 2-D histograms of R versus $mRN A_r$. The results for the discrete and hybrid models are shown in Figs. 7B and 7B. To construct these histograms 10, 000 oscillations were used. Excellent agreement between the discrete and hybrid model is seen. This indicates that the hybrid model is accurately sampling the steady-state distribution. To verify that the hybrid model faithfully captures the dynamics of the system, we computed the power spectra of both models. The results are shown in Figs. 8A and 8B. Again, excellent agreement is seen between the discrete and hybrid model.
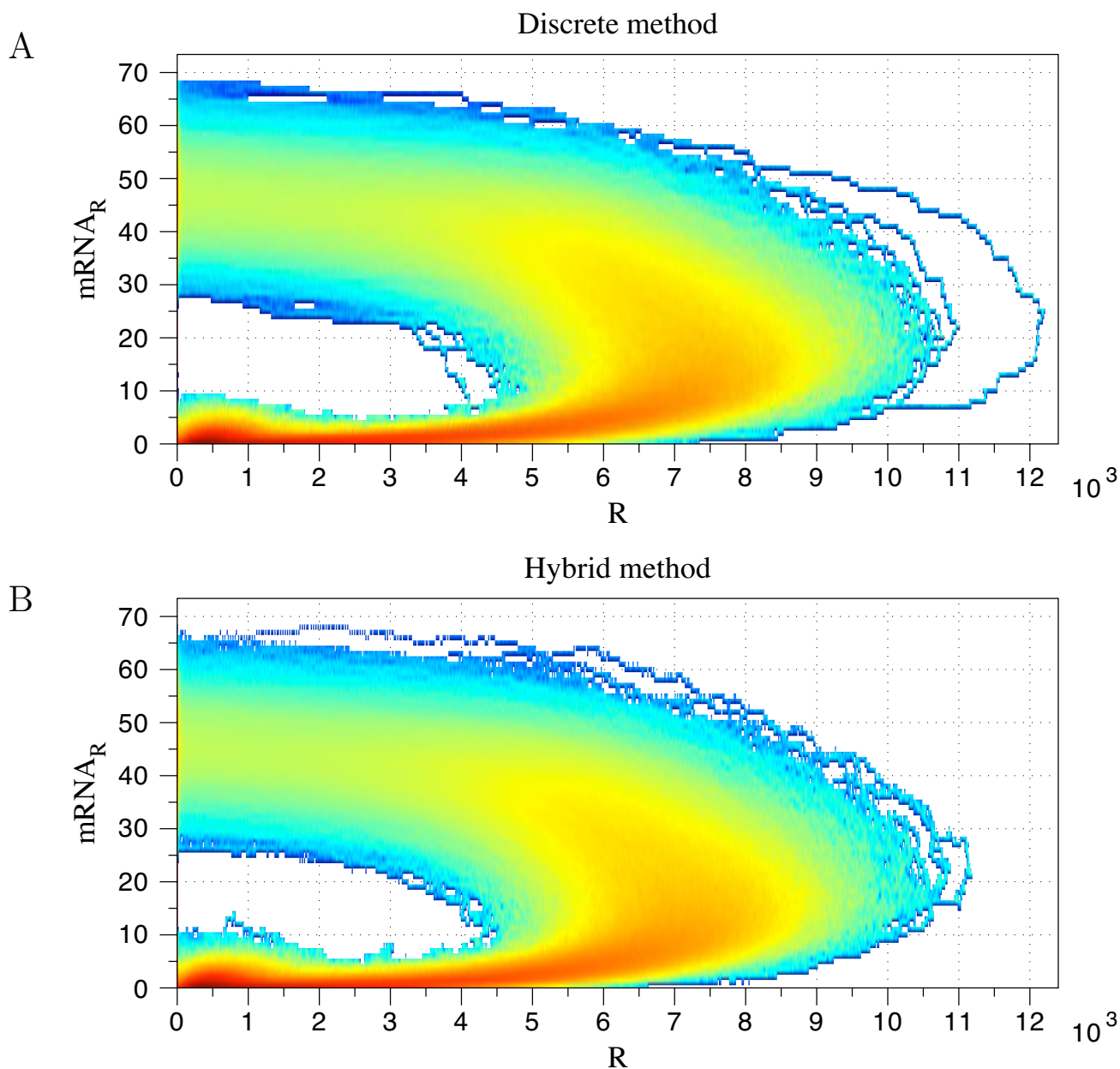
Figure 7
2-D histograms of repressor mRNA number versus repressor protein number. Red indicates regions of large frequency (i.e., where the system spends a lot of time) and blue indicates regions of low frequency. Panel A is for the discrete system and Panel B is the hybrid model. Roughly 10, 000 oscillations were used to produce both plots.

An engineered promoter system
Using standard techniques in modern molecular biology, it is possible to design novel systems of promoter-gene pairs, such that virtually any desired regulatory network architecture may be instantiated; such networks are often called "synthetic gene networks." Recent implementa-

tions have included direct negative [22] and positive [23] feedback, a bistable switch [12], an oscillator [11], an intercellular communication system [38], and a bimodal self-activating system [39].
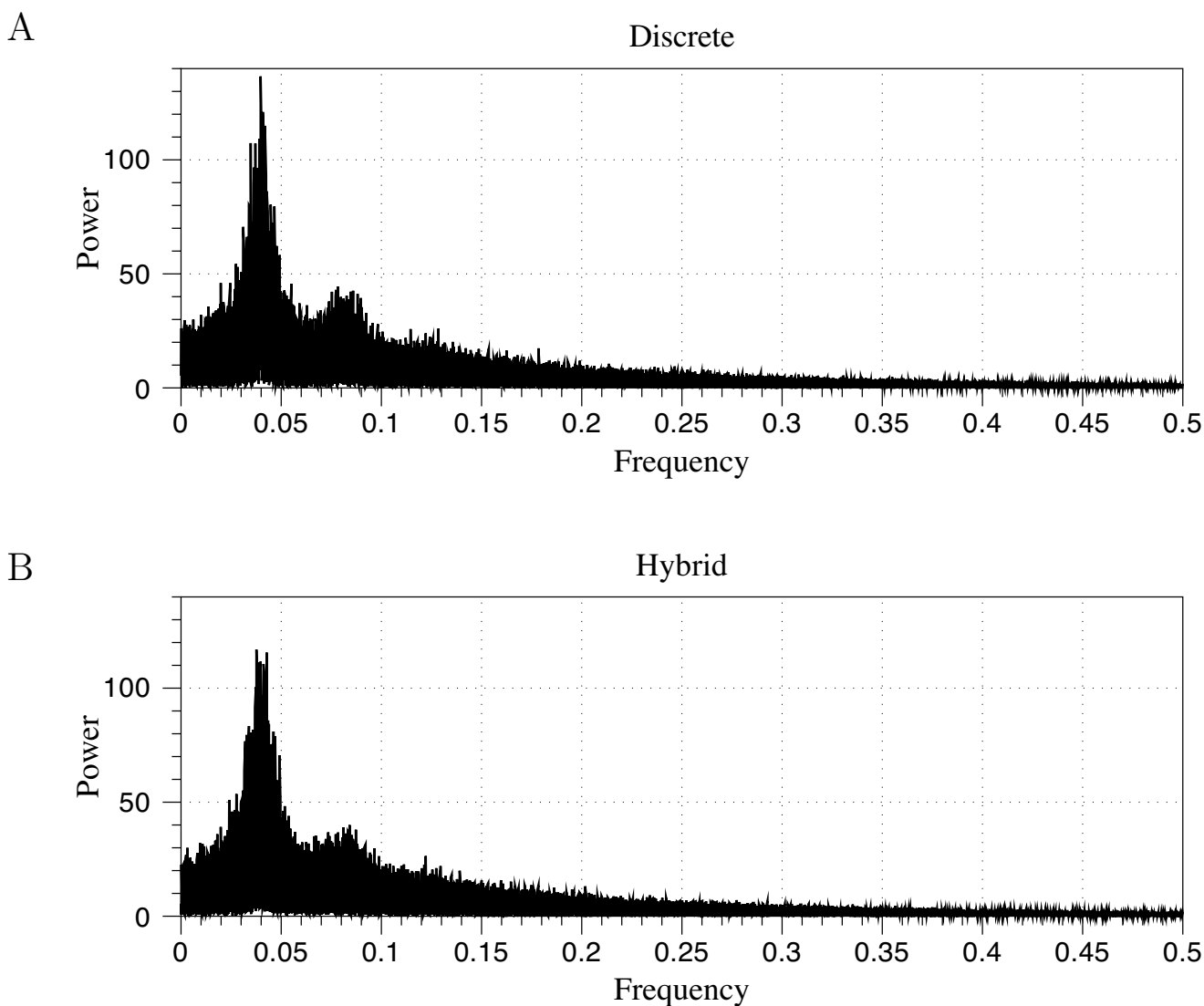
A   Discrete

B   Hybrid

Figure 8
Power spectra for the repressor protein number. Panel A is the discrete case and B is the hybrid model.

In this example, we use BioNetS to implement a model of a simple, open-loop network based around a novel engineered promoter, which has been designed and constructed by N. Guido and J. J. Collins at Boston University. The promoter, called $O_R O_{lac}$, combines the O_{lac}, $O_R 1$, and $O_R 2$ operator sites, so that it is repressed by the lac repressor protein (LacI) and activated by the lambda repressor protein (CI); see Fig. 9. Experiments have been conducted in which the promoter, along with other sites to produce the activator and repressor proteins, is integrated into a high copy number plasmid and inserted into a strain of *Escherichia coli*. The promoter's activity is observed using a fluorescent reporter, Green Fluorescent Protein (GFP). A detailed modeling study with direct comparisons to experimental results has been carried out using a fully discrete stochastic approach, and will be reported elsewhere (McMillen *et al.*, manuscript in preparation). Our goal here is to provide a reasonably complex test case to evaluate the performance of BioNetS.

The processes to be captured by the model are: transcription and degradation of mRNA strands; translation of mRNA into protein; degradation of protein; formation of protein multimers (dimers in the case of CI, tetramers in
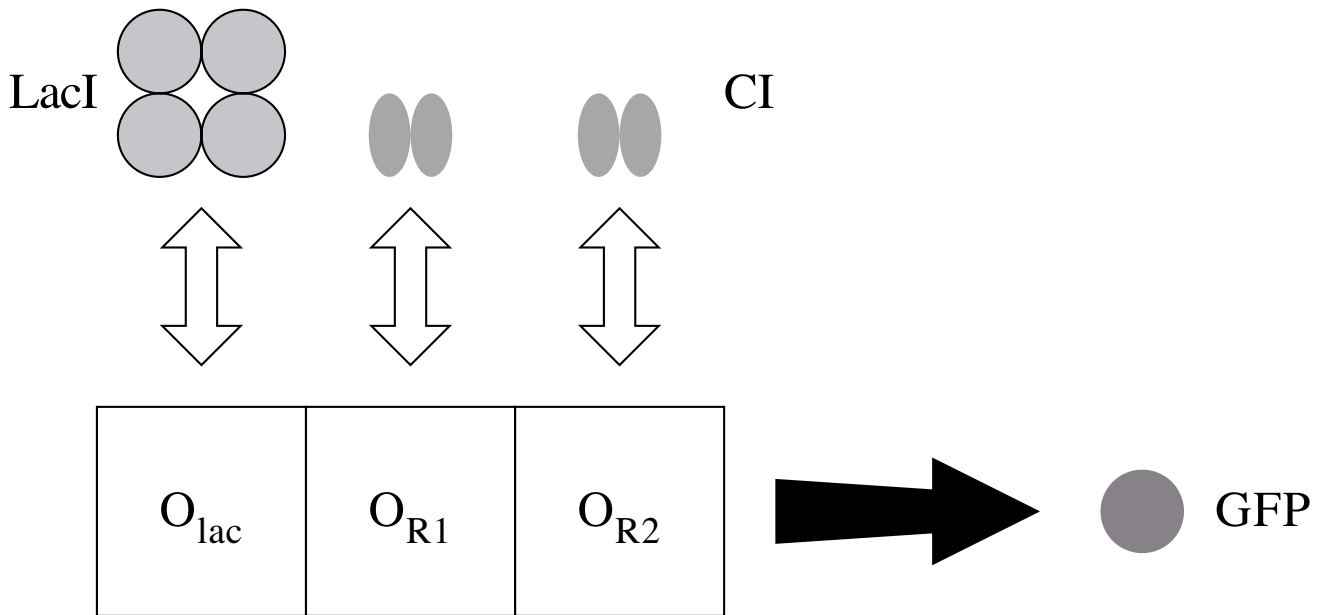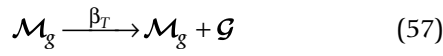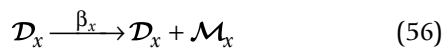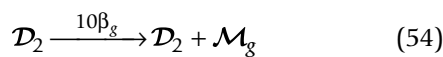
Figure 9
Schematic of the $O_R O_{lac}$ engineered promoter. The promoter fuses three operator sites, one of which (O_{lac}) yields repression when the LacI tetramer is bound to it, while binding of the CI dimer to $O_R 2$ yields approximately ten-fold activation (the $O_R 1$ site assists activation by cooperatively enhancing CI binding to $O_R 2$ when CI is bound to $O_R 1$). In the experimental system, CI and LacI are produced in the cell by external promoters, not shown here. The Green Fluorescent Protein (GFP) product is then used to monitor the output of the promoter by flow cytometry.
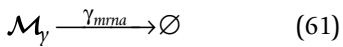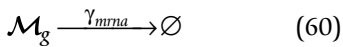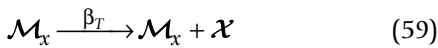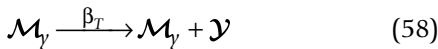
the case of LacI); LacI binding to isopropyl-β-D-thiogalactopyranoside (IPTG), a chemical inducer that reduces LacI's binding affinity for O_{lac}; and protein-DNA binding at the $O_R O_{lac}$ promoter's operator sites. We define the following chemical species: G, GFP; M_g, mRNA coding for GFP; X, CI monomer; X_2, CI dimer; M_x, mRNA coding for CI; D_x, the arabinose-inducible *pBAD* promoter site producing CI; Y, LacI monomer; Y_2, LacI dimer; Y_4, LacI tetramer; I_0, IPTG (present in massive excess and thus taken to be constant); Y_I, LacI tetramer bound to IPTG; M_y, mRNA coding for LacI; and D_y, the $P_L tetO1$ site constitutively producing LacI. In addition to these, we define species D_0 through D_8, representing the various permutations of proteins bound to the three operator sites in the $O_R O_{lac}$ promoter (see Table for a list). There are twelve combinatorial possibilities, but we eliminate three of them on the basis that CI (X_2) binding $O_R 2$ but not $O_R 1$ is unlikely, because of the low binding affinity of CI for $O_R 2$ compared to $O_R 1$. Table also lists the effect on the basal rate of production when the promoter is in each state. This reflects the regulatory effect of the proteins; for example, CI bound to $O_R 2$ leads to a 10-fold increase in transcription rate, while LacI bound to O_{lac} halts transcription completely (note that we assume in the event of simultaneous

binding of activator and repressor, repression "wins" and transcription is halted).

The following irreversible reactions represent the processes of transcription, translation, and degradation:

$$\mathcal{D}_0 \xrightarrow{\beta_g} \mathcal{D}_0 + \mathcal{M}_g \qquad (52)$$

$$\mathcal{D}_1 \xrightarrow{\beta_g} \mathcal{D}_1 + \mathcal{M}_g \qquad (53)$$

$$\mathcal{D}_2 \xrightarrow{10\beta_g} \mathcal{D}_2 + \mathcal{M}_g \qquad (54)$$

$$\mathcal{D}_y \xrightarrow{\beta_y} \mathcal{D}_y + \mathcal{M}_y \qquad (55)$$

$$\mathcal{D}_x \xrightarrow{\beta_x} \mathcal{D}_x + \mathcal{M}_x \qquad (56)$$

$$\mathcal{M}_g \xrightarrow{\beta_T} \mathcal{M}_g + \mathcal{G} \qquad (57)$$

$$\mathcal{M}_y \xrightarrow{\beta_T} \mathcal{M}_y + \mathcal{Y} \qquad (58)$$

$$\mathcal{M}_x \xrightarrow{\beta_T} \mathcal{M}_x + \mathcal{X} \qquad (59)$$

$$\mathcal{M}_g \xrightarrow{\gamma_{mrna}} \varnothing \qquad (60)$$

$$\mathcal{M}_y \xrightarrow{\gamma_{mrna}} \varnothing \qquad (61)$$

$$\mathcal{M}_x \xrightarrow{\gamma_{mrna}} \varnothing \qquad (62)$$

$$\mathcal{G} \xrightarrow{\gamma_{prot}} \varnothing \qquad (63)$$

$$\mathcal{Y} \xrightarrow{\gamma_{prot}} \varnothing \qquad (64)$$

$$\mathcal{X} \xrightarrow{\gamma_{prot}} \varnothing \qquad (65)$$
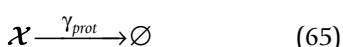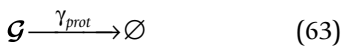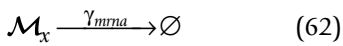
As in previous reactions, the caligraphic letters represent individual molecules of each species. We scale all times and rates by the cell division time.

Experimental measurements generally provide equilibrium rather than rate constants, and thus when writing reversible reactions we use the following notational convention: a reaction with equilibrium constant K has forward rate constant KR and backward rate constant R, where R is a scaling factor which sets the speed at which the reaction approaches equilibrium (we will consider three values of R: 1, 10, and 100). Using this notation, we represent protein-protein binding with the following set of reactions

$$\mathcal{Y} + \mathcal{Y} \;\overset{K_y}{\rightleftharpoons}\; \mathcal{Y}_2 \qquad (66)$$

$$\mathcal{Y}_2 + \mathcal{Y}_2 \;\overset{K_{y2}}{\rightleftharpoons}\; \mathcal{Y}_4 \qquad (67)$$

$$\mathcal{Y}_4 + I_0 \;\overset{K_{yI}}{\rightleftharpoons}\; \mathcal{Y}_I \qquad (68)$$

$$\mathcal{X} + \mathcal{X} \;\overset{K_x}{\rightleftharpoons}\; \mathcal{X}_2 \qquad (69)$$
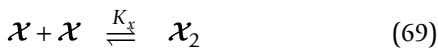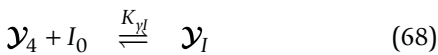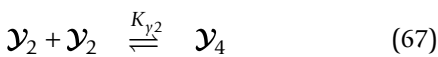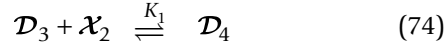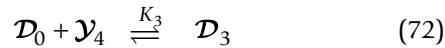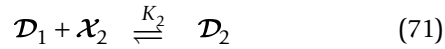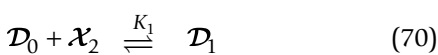
Finally, protein-DNA binding is given by:

$$\mathcal{D}_0 + \mathcal{X}_2 \;\overset{K_1}{\rightleftharpoons}\; \mathcal{D}_1 \qquad (70)$$

$$\mathcal{D}_1 + \mathcal{X}_2 \;\overset{K_2}{\rightleftharpoons}\; \mathcal{D}_2 \qquad (71)$$

$$\mathcal{D}_0 + \mathcal{Y}_4 \;\overset{K_3}{\rightleftharpoons}\; \mathcal{D}_3 \qquad (72)$$

$$\mathcal{D}_1 + \mathcal{Y}_4 \;\overset{K_3}{\rightleftharpoons}\; \mathcal{D}_4 \qquad (73)$$

$$\mathcal{D}_3 + \mathcal{X}_2 \;\overset{K_1}{\rightleftharpoons}\; \mathcal{D}_4 \qquad (74)$$

$$\mathcal{D}_2 + \mathcal{Y}_4 \;\overset{K_3}{\rightleftharpoons}\; \mathcal{D}_5 \qquad (75)$$

$$\mathcal{D}_4 + \mathcal{X}_2 \;\overset{K_2}{\rightleftharpoons}\; \mathcal{D}_5 \qquad (76)$$

$$\mathcal{D}_0 + \mathcal{Y}_I \;\overset{K_4}{\rightleftharpoons}\; \mathcal{D}_6 \qquad (77)$$

$$\mathcal{D}_1 + \mathcal{Y}_I \;\overset{K_4}{\rightleftharpoons}\; \mathcal{D}_7 \qquad (78)$$

$$\mathcal{D}_6 + \mathcal{X}_2 \;\overset{K_1}{\rightleftharpoons}\; \mathcal{D}_7 \qquad (79)$$

$$\mathcal{D}_2 + \mathcal{Y}_I \;\overset{K_4}{\rightleftharpoons}\; \mathcal{D}_8 \qquad (80)$$

$$\mathcal{D}_7 + \mathcal{X}_2 \;\overset{K_2}{\rightleftharpoons}\; \mathcal{D}_8 \qquad (81)$$
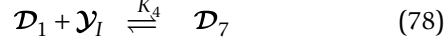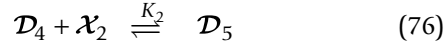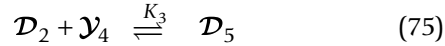
In all, the system consists of 21 species, participating in 34 reactions. The reactions are entered into BioNetS using the same method described in the previous examples. We use BioNetS' ability to represent individual species as either discrete or continuous to formulate three versions of the model: fully discrete, fully continuous, and a hybrid version in which the DNA species D_0 through D_8 are discrete while all other species are continuous. We vary the value of R, the scaling factor for reversible reactions, and keep all other parameters fixed at the following nondimensionalized values: $\beta_g = 0.1$, $\beta_y = 1$, $\beta_x = 0.5$, $\beta_T = 10$, $\gamma_{mrna} = 3.5$, $\gamma_{prot} = 0.7$, $K_y = 0.01$, $K_{y2} = 0.1$, $K_{yI} = 2 \times 10^{-6}$, $K_x = 0.05$, $K_1 = 0.3$, $K_2 = 2K_1$, $K_3 = 0.008$, $K_4 = 1.4 \times 10^{-4} K_3$, $I_0 = 1 \times 10^6$.

To evaluate the steady-state probability distributions produced by the reaction system, simulations 250000 cell cycles in length were used to accumulate histograms (a built-in feature of BioNetS) of the number of molecules of GFP (species G), for each of the three versions of the model. As Fig. 10A shows, the resulting distributions are essentially identical, indicating that the continuum approximations used in the fully continuous and hybrid forms of the model were valid. Not all species in the sys-

tem are well approximated as continuous variables: Fig. 10B shows the continuous probability distribution for species D_8, representing a promoter fully populated with two CI dimers and a LacI tetramer-IPTG complex. This situation is very rare in our parameter regime, and the system spends essentially all of its time with $D_8 = 0$. The fully continuous model, however, fluctuates into negative values, indicating that the continuum approximation has broken down. This does not significantly affect the distribution for GFP because the other, more common DNA states dominate the system's behavior; note, however, that if we were considering genomic DNA rather than a high copy number plasmid, we would not be able to employ a fully continuous model. The hybrid model, by treating the DNA species as continuous, eliminates the fluctuations into negative values. In general, the appropriate approximations will depend on both the system and the variables of interest: in the present example, if we were interested in the behavior of the operator sites themselves, we would not be able to use the fully continuous version of the model, but as a model solely of GFP expression the approximation suffices. Comparisons between types of models should be made to test the underlying assumptions, and BioNetS facilitates this process.

We used simulations 200 cell cycles in length to test the speed at which the three model versions ran. In each case, 200 simulations were run using a consistent set of 200 different random seeds; all runs were started with identical initial conditions. For the fully continuous and hybrid systems, the semi-implicit scheme was numerically stable and yielded consistent histograms for all time step sizes between $dt = 0.001$ and $dt = 0.5$, but the latter corresponds to just two time points per cell division cycle (recall that all times are scaled by the cell division time), and we chose instead to sample 20 points per cycle and set $dt = 0.05$. As shown in Table 2, the fully continuous method was always fastest, with the degree of improvement over the exact, fully discrete method depending strongly on the value of R, the scaling factor for the reversible reaction rates. For $R = 1$, the fully continuous method was only 1.4-fold faster than the fully discrete method, but as R is increased this speed advantage increases to over 4-fold at $R = 10$, then to over 30-fold at $R = 100$. (Note that the speed advantage of the fully continuous over the fully discrete method increases with the abundances of the chemical species. Shifting parameters to generate higher protein numbers can yield cases in which the continuum approximation is hundreds of times faster than the discrete approach; runs not shown here.) Use of a hybrid discrete/continuous method did not, for this particular model system, offer any speed gain over the fully discrete approach; the increased time involved in computing the Jacobian for the semi-implicit method is more time-consuming than simply simulating the reactions directly. Optimizing effi-

ciency requires testing various potential approaches, and BioNetS makes this a simple process.

Conclusions

We have developed BioNetS to be a reliable tool for studying the stochastic dynamics of large chemical networks. The software allows the user to specify which of the chemical species in the network should be treated as discrete random variables and which can be approximated as continuous random variables. The software is highly optimized for speed and should be be able to simulate networks consisting of hundreds of chemical species. We have verified the accuracy of the numerical methods by considering several test systems (a dimerization reaction, a chemical oscillator, and an engineered promoter), each of which shows excellent agreement between the fully discrete version and the fully or partially continuous versions. Our hope is that BioNetS, by providing a simple, user-friendly interface, will allow biological experimentalists to formulate biochemical reaction models of their systems quickly and easily, ideally increasing the number of systems in which direct comparisons are available between models and experimental results. Clearly, not every possible biological system can be captured in the current version of BioNetS, and its capabilities will continue to grow in the future. We wish to encourage users, or potential users, to contact us regarding which additional features would be most helpful to them.

Availability and requirements

• **Project name:** BIOchemical NETwork Stochastic Simulator (BioNetS)

• **Project home page:** http://x.amath.unc.edu/BioNetS

• **Operating system:**

* User interface: Macintosh OS X, version 10.2 or above.

* Generated source code: Ability to compile portable C++ code. Makefiles included for OS X and Linux.

• **Programming language:** C++.

• **Other requirements:** None.

• **License:** BSD license.

• **Restrictions on use by non-academics:** None.

Authors' contributions

DA wrote the BioNetS code in its entirety, providing the user interface, the numerical optimizations and the code generator. TE provided the mathematical derivations, carried out the dimer and oscillator examples, wrote an ini-
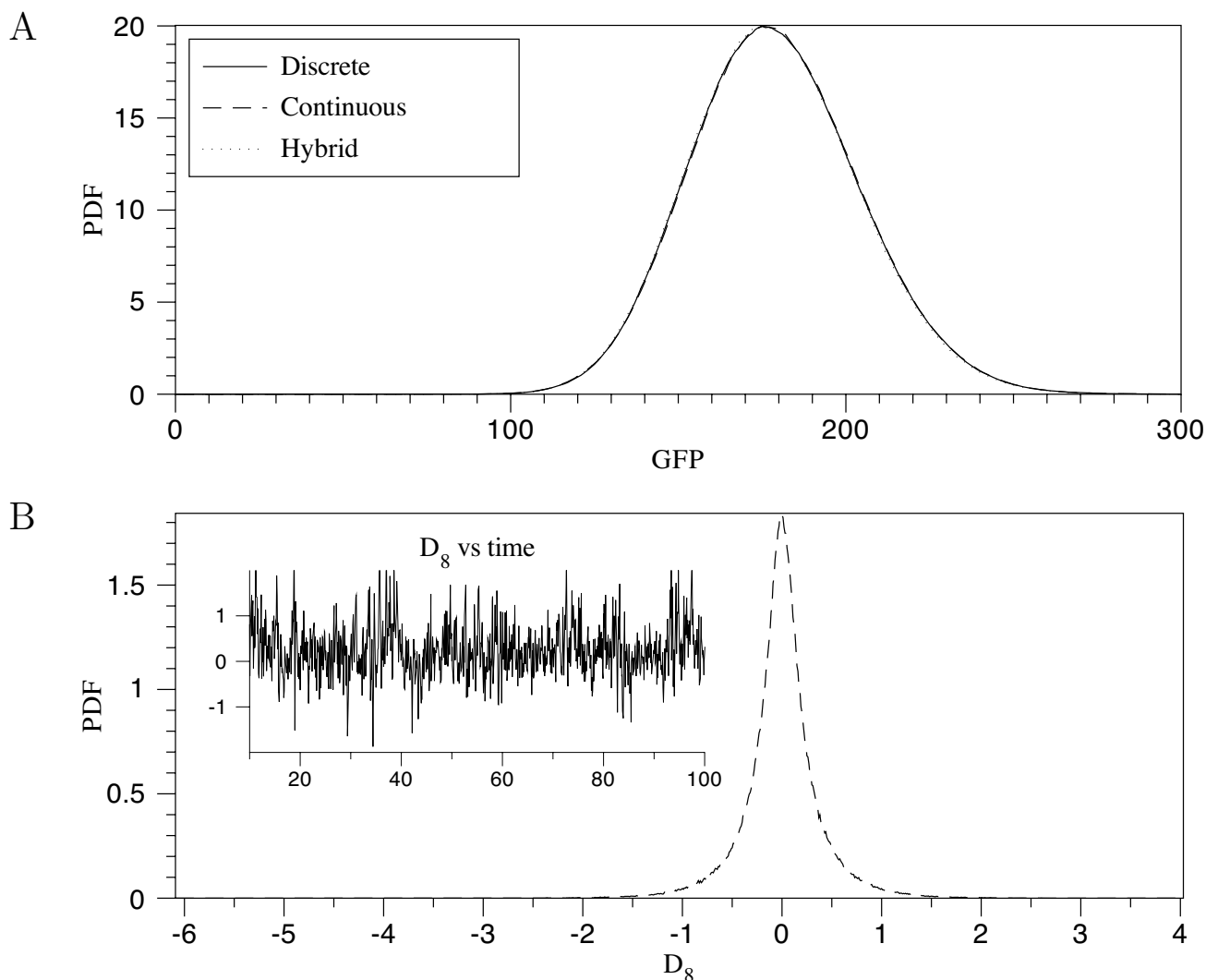
Figure 10
A) Probability densities of GFP molecules, for three versions of the O_RO_{lac} promoter model. Densities were generated by accumulating statistics in runs 250000 cell cycles in duration. (*Solid line*) Fully discrete model. (*Dashed line*) Fully continuous model. (*Dash-dotted line*) Hybrid model: the DNA binding states are represented by discrete variables, while all other species are continuous. All three methods produce virtually identical probability distributions. B) Probability density for species D_8, for the continuous model. The hybrid and fully discrete methods produce identical distributions in which D_8 spends 99.996% of the time at zero. This histogram shows that the continuous model artificially allows negative numbers (see inset time series).

tial draft of the paper, and worked with DA on the algorithms and numerical methods. DM revised and finalized the paper, provided the engineered promoter example, and worked extensively with DA on testing and debugging of the software. All authors read, edited, and approved the final version of the paper.

Table 1: List of DNA-binding states in the $O_R O_{lac}$ engineered promoter example.

Species	O_{lac}	O_R2	O_R1	Production Rate
D_0	-	-	-	1
D_1	-	-	X_2	1
D_2	-	X_2	X_2	10
D_3	Y_4	-	-	0
D_4	Y_4	-	X_2	0
D_5	Y_4	X_2	X_2	0
D_6	Y_1	-	-	0
D_7	Y_1	-	X_2	0
D_8	Y_1	X_2	X_2	0

Column O_{lac} represents the *lac* operator site, while columns O_R1 and O_R2 represent the two operator sites in the p_{rm} region of bacteriophage λ. Entries in these columns indicate which regulatory proteins are bound to the operator sites in each state. The production rate column indicates how quickly transcription from the promoter proceeds in each state: basal expression occurs in states D_0 and D_1; enhanced expression occurs in state D_2; and the remaining states display full repression (no expression).

Table 2: Execution times for three versions of the $O_R O_{lac}$ promoter model.

	Fully continuous		Fully discrete		Hybrid	
R	Mean (s)	Std (s)	Mean (s)	Std (s)	Mean (s)	Std (s)
1	0.26 [1]	0.01	0.36 [1.4]	0.01	0.69 [2.7]	0.01
10	0.27 [1]	0.01	1.15 [4.3]	0.01	6.91 [27]	0.12
100	0.27 [1]	0.01	9.11 [34]	0.28	69.43 [260]	1.13

The mean values represent the number of seconds of CPU time required to run a simulation of 200 cell cycles on an otherwise unloaded 700 MHz PowerPC G4 processor, averaged over a set of 200 different random seeds, identical for each version of the model. "Std" indicates the standard deviation of this same set of 200 runs. Three values of parameter *R*, a scaling factor giving the relative speed of the reversible reactions in the system, have been used. In each row, the values in square brackets are normalized by the shortest execution time. The fully continuous version runs substantially faster than the other methods, and its execution time does not depend on *R*. The fully continuous version produces identical histograms to the fully discrete and hybrid methods for GFP (see Fig. 10A), but for the low-number species such as D_8 it produces spurious negative values (see Fig. 10B). If accuracy is required for the small-number states, the fully discrete method should be used; note that for this system, the hybrid approach is consistently slower than the fully discrete method.

Table 3

• Initialize all species and rate constants
• Compute all reaction rates
• Loop:
    * Set μ = sum of rates for the discrete reactions
    * if ($p_t = \mu\Delta t > \varepsilon$), **use Gillespie algorithm:**
        * R = a uniform random number in (0,1)
        * Set timeStep = -log(R)/μ
        * Find which reaction occurred, update the species involved
    * else, **use small Δt approximation:**
        * R = a uniform random number in [0,1]
        * timeStep = continuousTimeStep
        * if ($R < p_t = \mu \times$ timeStep), **discrete transition has occurred:**
            • Determine which discrete transition occurred:

            • Find the first value of *k* for which $\sum_{i=1}^{k}[F_i + B_i] \geq R\mu$

            • If $\sum_{i=1}^{k-1}[F_i + B_i] + F_k \geq R\mu$, the forward reaction occurred, otherwise the backward reaction occurred
        * else, **no discrete transition:**
            • No discrete reaction occurs, update is entirely due to continuous reactions (below)
        * end if (*small Δt method, determination if discrete transition occurred*)
    * end if (*selection of Gillespie or small Δt method for discrete reactions*)
    * Update the continuous species using the Langevin equation, with step size timeStep (where timeStep is either equal to continuousTimeStep or to the step size found by the Gillespie algorithm), using a semi-implicit numerical method
    * Update any rates that have been changed by the continuous reactions and the single discrete reaction
    * Break when user-defined total simulation time is reached
• end loop

Acknowledgements

References

1. Glass L, Kauffman S: **The logical analysis of continuous, nonlinear biochemical control networks.** *J Theor Biol* 1973, **39:**103-129.
2. Kauffman S: **The large-scale structure and dynamics of gene control circuits: An ensemble approach.** *J Theor Biol* 1974, **44:**167-190.
3. Savageau M: **Comparison of classical and autogenous systems of regulation in inducible operons.** *Nature* 1974, **252:**546-549.
4. Glass L: **Classification of biological networks by their qualitative dynamics.** *J Theor Biol* 1975, **54:**85-107.
5. Tyson J, Othmer H: **The dynamics of feedback control circuits in biochemical pathways.** *Progr Theor Biol* 1978, **5:**1-60.
6. Ackers G, Johnson A, Shea M: **Quantitative model for gene regulation by λ phage repressor.** *Proc Natl Acad Sci USA* 1982, **79:**1129-1133.
7. Shea M, Ackers G: **The O_R control system of bacteriophage lambda: A physical-chemical model for gene regulation.** *J Mol Biol* 1985, **181:**211-230.
8. Reinitz J, Vaisnys J: **Theoretical and experimental analysis of the phage lambda genetic switch implies missing levels of cooperativity.** *J Theor Biol* 1990, **145:**295-318.
9. Keller A: **Model genetic circuits encoding autoregulatory transcription factors.** *J Theor Biol* 1995, **172:**169-185.
10. Mestl T, Lemay C, Glass L: **Chaos in high-dimensional neural and gene networks.** *Physica D* 1996, **98:**33-52.
11. Elowitz M, Leibler S: **A synthetic oscillatory network of transcriptonal regulators.** *Nature* 2000, **403:**335-338.
12. Gardner T, Cantor C, Collins J: **Construction of a genetic toggle switch in** *Escherichia coli.* *Nature* 2000, **403:**339-342.
13. Endy D, You L, Yin J, Molineux I: **Computation, prediction, and experimental tests of fitness for bacteriophage T7 mutants with permuted genomes.** *Proc Natl Acad Sci USA* 2000, **97:**5375-5380.
14. Hasty J, Isaacs F, Dolnik M, McMillen D, Collins J: **Designer gene networks: Towards fundamental cellular control.** *Chaos* 2001, **11:**207-220.
15. Santillán M, Mackey M: **Dynamic regulation of the tryptophan operon: A modeling study and comparison with experimental data.** *Proc Natl Acad Sci USA* 2001, **98:**1364-1369.
16. McMillen D, Kopell N, Hasty J, Collins J: **Synchronizing genetic relaxation oscillators by intercell signaling.** *Proc Natl Acad Sci USA* 2002, **99:**679-684.
17. Spudich J, Koshland DEJ: **Non-genetic individuality: Chance in the single cell.** *Nature* 1976, **262:**467-471.
18. Ross I, Browne C, Hume D: **Transcription of individual genes in eukaryotic cells occurs randomly and infrequently.** *Immunol Cell Biol* 1994, **72:**177-185.
19. Hume D: **Probability in transcriptional regulation and its implications for leukocyte differentiation and inducible gene expression.** *Blood* 2000, **96:**2323-2328.
20. Arkin A, Ross J, McAdams H: **Stochastic kinetic analysis of developmental pathway bifurcation in phage λ -infected** *Escherichia coli* **cells.** *Genetics* 1998, **149:**1633-1648.
21. Barkai N, Leibler S: **Biological rhythms: Circadian clocks limited by noise.** *Nature* 2000, **403:**267.
22. Becskei A, Serrano L: **Engineering stability in gene networks by autoregulation.** *Nature* 2000, **405:**590-593.
23. Becskei A, Séraphin B, Serrano L: **Positive feedback in eukaryotic gene networks: cell differentiation by graded to binary response conversion.** *EMBO J* 2001, **20:**2528-2535.
24. Bialek W: **Stability and noise in biochemical switches.** In *Advances in Neural Information Processing Systems Volume 13*. Cambridge: The MIT Press; 2001:103-109.
25. Elowitz M, Levine A, Siggia E, Swain P: **Stochastic gene expression in a single cell.** *Science* 2002, **297:**1183-1186.
26. Hasty J, Pradines J, Dolnik M, Collins J: **Noise-based switches and amplifiers for gene expression.** *Proc Natl Acad Sci USA* 2000, **97:**2075-2080.
27. Ko M: **A stochastic model for gene induction.** *J Theor Biol* 1991, **153:**181-194.
28. Kepler T, Elston T: **Stochasticity in transcriptional regulation: origins, consequences, and mathematical representations.** *Biophys J* 2001, **81:**3116-3136.
29. McAdams H, Arkin A: **Stochastic mechanisms in gene expression.** *Proc Natl Acad Sci USA* 1997, **94:**814-819.
30. Ozbudak E, Thattai M, Kurtser I, Grossman A, van Oudenaarden A: **Regulation of noise in the expression of a single gene.** *Nature Genet* 2002, **31:**69-73.
31. Thattai M, van Oudenaarden A: **Intrinsic noise in gene regulatory networks.** *Proc Natl Acad Sci USA* 2001, **98:**8614-8619.
32. Gillespie D: **Exact stochastic simulation of coupled chemical reactions.** *J Phys Chem* 1977, **81:**2340-2361.
33. Gibson M, Bruck J: **Efficient exact stochastic simulation of chemical systems with many species and many channels.** *J Phys Chem* 2000, **104:**1876-1889.
34. Gillespie D: **Approximate accelerated stochastic simulation of chemically reacting systems.** *J Phys Chem* 2001, **115:**1716-1733.
35. Gardiner C: *Handbook of stochastic methods: For physics, chemistry and the natural sciences* Berlin: Springer Verlag; 1996.
36. Kloeden P, Platen E: *Numerical solution of stochastic differential equations* Berlin: Springer Verlag; 1992.
37. Vilar J, Kueh H, Barkai N, Leibler S: **Mechanisms of noise-resistance in genetic oscillators.** *Proc Natl Acad Sci USA* 2002, **99:**5988-5992.
38. Weiss R, Knight T: **Engineered communications for microbial robotics.** In *DNA6: Sixth International Meeting on DNA Based Computers: June 13–17 2000; Leiden, Netherlands* Edited by: Condon A, Rozenberg G. Berlin: Springer Verlag; 2001.
39. Isaacs F, Hasty J, Cantor C, Collins J: **Prediction and measurement of an autoregulatory genetic module.** *Proc Natl Acad Sci USA* 2003, **100:**7714-7719.