BIOINFORMATICS

Vol. 18 no. 3 2002
Pages 470–481

STOCKS: STOChastic Kinetic Simulations of biochemical systems with Gillespie algorithm

Andrzej M. Kierzek

*Institute of Biochemistry and Biophysics, Polish Academy of Sciences,
Pawinskiego 5a, 02-106 Warszawa, Poland*

Received on June 5, 2001; revised on August 22, 2001; accepted on October 4, 2001

ABSTRACT

Motivation: The availability of a huge amount of molecular data concerning various biochemical reactions provoked numerous attempts to study the dynamics of cellular processes by means of kinetic models and computer simulations. Biochemical processes frequently involve small numbers of molecules (e.g. a few molecules of a transcriptional regulator binding to one ‘molecule’ of a DNA regulatory region). Such reactions are subject to significant stochastic fluctuations. Monte Carlo methods must be employed to study the functional consequences of the fluctuations and simulate processes that cannot be modelled by continuous fluxes of matter. This provides the motivation to develop software dedicated to Monte Carlo simulations of cellular processes with the rigorously proven Gillespie algorithm.

Results: STOCKS, software for the stochastic kinetic simulation of biochemical processes is presented. The program uses a rigorously derived Gillespie algorithm that has been shown to be applicable to the study of prokaryotic gene expression. Features dedicated to the study of cellular processes are implemented, such as the possibility to study a process in the range of several cell generations with the application of a simple cell division model. Taking expression of *Escherichia coli* beta-galactosidase as an example, it is shown that the program is able to simulate systems composed of reactions varying in several orders of magnitude by means of reaction rates and the numbers of molecules involved.

Availability: The software is available at ftp://ibbrain.ibb.waw.pl/stocks and http://www.ibb.waw.pl/stocks.

Supplementary information: Parameters of the model of prokaryotic gene expression are available in example files of software distribution.

Contact: andrzejk@ibb.waw.pl

INTRODUCTION

The determination of rates of reactions involved in cellular metabolism is one of the classic research topics in biochemistry. As data for the rates of single reactions accumulate, it becomes possible to study more complex biochemical pathways by means of kinetic models and computer simulations. The ultimate goal of these studies is to understand the dynamics of the living cell in terms of the interactions among its molecular components. The advances in genomics that yield unprecedented capabilities of controlled modifications of protein function and gene expression levels further motivate the development of models that are able to predict dynamic effects, within metabolic networks, resulting from these changes.

The kinetic model involves a set of substances interacting through a network of reactions. If the reactions are described by differential equations, the time evolution of the system can be simulated by numerical integration of the rate equations. From the rate equations, elasticity coefficients can also be computed that quantitatively describe the susceptibility of the system to the perturbation of the selected parameter of the model. The latter approach is also known as Metabolic Control Analysis (MCA). There are numerous computer programs that perform these calculations e.g. E-CELL (Tomita *et al.*, 1999); DBSolve (Goryanin *et al.*, 1999); GEPASI (Mendes, 1993, 1997); MEG (Mendes and Kell, 2001); KINSIM (Barshop *et al.*, 1983; Dang and Frieden, 1997); MIST (Ehlde and Zacchi, 1995); METAMODEL (Cornish-Bowden and Hofmeyr, 1991); SCAMP (Sauro, 1993). E-CELL software has been applied in an attempt to build a whole-cell kinetic model (Tomita *et al.*, 1999). The authors either collected from the literature or fitted rate constants describing metabolic reactions involving the products of 127 genes of *Mycoplasma genitalium*—the cell with the smallest known genome. The results of the computer simulations have been discussed in context of genome engineering.

The methods presented above use a deterministic formulation of chemical kinetics, i.e. they treat reactions as continuous fluxes of matter. This approach is correct if there is a very large number of molecules present in the system. The average outcome of a very large number of random molecular collisions is a continuous and deterministic process. Several authors argued that the deterministic approach is inappropriate for many biochemical processes involving very small numbers of molecules (McAdams and Arkin, 1997; Levin *et al.*, 1998; Hume, 2000; Kierzek *et al.*, 2001). An example is the regulation of gene expression when the number of transcription regulators present in the cell may be as low as ten molecules and regulators bind to a single 'molecule' of the DNA regulatory region (e.g. Lac repressor and LacZ promoter; Levin, 1999). In the case of a reaction involving such a small number of molecules stochastic fluctuations of time intervals between individual random molecular collisions become significant. This also implies significant random fluctuations in the numbers of various molecular species present in the system. The influence of the stochastic effects on the course of biological processes has been shown in works on the kinetics of phage lambda life cycle regulation McAdams and Arkin, 1997; Arkin *et al.*, 1998). The authors have shown that the lytic or lysogenic fate of the particular phage molecule can be determined by a random event due to stochastic fluctuations of the numbers of regulatory proteins. This implies that the complex regulatory networks of deterministic behaviour, which is crucial for cell function, must contain mechanisms that compensate for random changes in the numbers of regulatory proteins. It has been postulated that checkpoints in the eukaryotic cell cycle serve that purpose (Alberts *et al.*, 1994). The variability of cell behaviour in the isogenic population is another example of a phenomenon that can be explained by the stochastic fluctuations in biochemical processes. Individual cell responses to subsaturating inducer concentrations in the lactose and arabinose operons (Siegele and Hu, 1997) and individual swimming behaviour of *Escherichia coli* cells have been attributed to stochastic processes (Levin *et al.*, 1998).

The above examples show that, in order to correctly model the dynamics of many cellular processes, stochastic effects must be taken into account. In order to do so, Monte Carlo approaches to chemical kinetics must be employed. In these methods, individual molecular encounters are explicitly simulated with the use of computer generated random numbers following the appropriate probability distributions. There were several attempts to formulate Monte Carlo computer simulation protocols applicable to the studies of biochemical kinetics. Carrier and Keasling (1999) proposed an algorithm dedicated to the studies of prokaryotic gene expression. They applied the method to test various hypotheses concerning the role of mRNA degradation in prokaryotic gene expression and to the modeling of an all-or-none phenomena in lactose operon regulation. Morton-Firth and Bray (1998) applied their own Monte Carlo simulation algorithm to study signal transduction in bacterial chemotaxis. Simulations were able to explain individual swimming behaviour of *E. coli* cells. The algorithm has been implemented as the STOCHSIM program.

The Gillespie algorithm (Gillespie, 1977) is the general method for Monte Carlo simulation of the systems composed of coupled chemical reactions. The physical validity of the method is rigorously proven. The algorithm has already been applied to the simulations of various biochemical processes. Arkin *et al.* (1998) studied the role of stochastic phenomena in the bifurcation of the development pathway of bacteriophage λ. Garcia-Olivares *et al.* (2000) applied the Gillespie algorithm to stochastic simulations of the cyclic dynamics in glycolitic and gluconeogenetic cycle. Laurenzi and Diamond (1999) used the algorithm to investigate the aggregation kinetics of platelets and neutrophils. In a recent paper (Kierzek *et al.*, 2001), the Gillespie algorithm has been applied to study the relationship between transcription and translation initiation frequencies and the magnitude of stochastic fluctuations in prokaryotic gene expression. Taking into account the applications listed above, the method is worth implementing as publicly available software. To my best knowledge, the only software suitable for biochemical kinetics simulations with the Gillespie algorithm is SIMULAC (http://genomics.lbl.gov/∼aparkin) written by Adam Arkin. Here I present the software STOCKS which implements the Gillespie algorithm and new functions dedicated to the simulation of cellular processes. I show that the program is able to accurately compute the time evolution of systems composed of reactions with rates varying by several orders of magnitude (gene expression and the enzymatic reaction of synthesized protein). Another example shows that the method is computationally fast enough to allow for intensive parameter scanning. In this example, new results concerning magnitude of fluctuations in prokaryotic gene expression are also presented.

IMPLEMENTATION

STOCKS is software written in standard C^{++} language. It can be compiled on any platform with a C^{++} compiler. The program has been tested under Linux and Irix operating systems. The interface of STOCKS is best suited for running the program as a background job under UNIX operating systems. The following sections present the background of the Gillespie algorithm, details concerning the implementation of this algorithm and formats for input and job control files. Additional utility programs aiding the analysis of the results are also presented.

Gillespie algorithm

Let us consider a system composed of N chemical species $S_i (i = 1, \ldots, N)$ interacting through M reactions $R_\mu (\mu = 1, \ldots, M)$ in the volume V. Every reaction μ is characterized by its stochastic rate constant c_μ, which depends on the physical properties of the molecules taking part in the reaction and the temperature of the system.

The stochastic rate constant has the meaning of ‘reaction probability per unit time’ as the product $c_\mu \,\mathrm{d}t$ is the probability that one elementary reaction μ happens in the next infinitesimal time interval $\mathrm{d}t$. By an elementary reaction I mean a single, reactive molecular collision between the species taking part in the reaction. Taking a specific example, if the reaction formula is $A + B \rightarrow AB$ the elementary reaction will remove a single A and single B molecule from the reaction environment and add single molecule of AB.

There is a simple and intuitive relationship between the stochastic rate constant and deterministic rate constant used in chemical kinetics. It is important, as it allows the direct application of experimentally determined rate constants in the Gillespie algorithm simulation. For first order reactions, both constants are equal. In the case of second order reactions, the stochastic rate constant c_μ equals the deterministic rate constant k_μ divided by the volume of reaction environment:

$$c_\mu = k_\mu/(N_\mathrm{A} V) \tag{1}$$

(N_A is Avogadro’s number). In the case of second order reactions of two molecules of the same substance (e.g. $A + A \rightarrow AA$) stochastic rate constant is calculated as follows:

$$c_\mu = 2k_\mu/(N_\mathrm{A} V). \tag{2}$$

This is caused by the fact that the number of distinct pairs of molecules that can reactively collide is smaller in the case of (2). For example, in reaction $A + B \rightarrow AB$ the number of distinct molecular encounters is $X_\mathrm{A} X_\mathrm{B}$ where X_A, X_B denote the numbers of the molecules A and B. In the case of homodimer formation reaction $A + A \rightarrow AA$, the number of distinct molecular encounters is $X_\mathrm{A}(X_\mathrm{A} - 1)/2$. Let us denote the number of distinct reactant combinations available for the reaction R_μ at the given state of the system as h_μ. For the derivation of the above relations, see the original paper of Gillespie (1977).

For the system of reactions considered above, the Gillespie algorithm proceeds as follows. In every step of the simulation, two questions are answered: (i) what is the waiting time for the next reaction to occur and (ii) which one of all reactions in the system will occur. These questions are answered by generating two random numbers according to the following probability density function:

$$P(\tau, \mu) = a_\mu \exp(-a_0 \tau) \tag{3}$$

where

$$a_\mu = h_\mu c_\mu$$
$$a_0 = \Sigma a_\mu.$$

$P(\tau, \mu)\,\mathrm{d}\tau$ is the probability that the next reaction will occur in the system in the infinitesimal time interval $\mathrm{d}\tau$ and that it will be an R_μ reaction. After determination of the waiting time for the next reaction and the identity of this reaction, numbers of molecules in the system and the time of the simulation are adjusted accordingly and simulation proceeds. The practical procedure for performing simulations consistent with (3) is shown in Figure 1.

As the algorithm shows, the way in which the identity of the next reaction is determined is very intuitive. The larger the reaction rate is, or the larger are the numbers of substrate molecules, the greater is the chance that a given reaction will happen in the next step of the simulation. There is no constant timestep in the simulation. The timestep is determined in every iteration and it takes different values depending on the state of the system. The practical consequence of this fact is that it is difficult to determine in advance the computational cost of the simulation. As the timestep changes and depends on the numbers of reactant molecules, the number of program iterations that need to be executed in order to reach a preset maximal time of the simulation is unknown in advance.

The rigorous derivation of the algorithm has been given elsewhere (Gillespie, 1977). The author argued that the algorithm is ‘exact’ in the sense that it never approximates the infinitesimal time increment $\mathrm{d}t$ by discrete timesteps. The algorithm determines the exact times at which individual molecular reactive encounters occur. From the practical point of view, it is useful as one does not have to test the sensitivity of the simulations to the timestep values. In contrast to the deterministic formulation of chemical kinetics, the algorithm remains exact for arbitrary low numbers of the molecules. Repeated runs of the simulation can be used to study fluctuations in the numbers of molecules.

Implementation

STOCKS has an object oriented data structure. The reaction object contains pointers to substance objects which in turn held the names and numbers of molecules of particular reactants. Separate arrays, within reaction objects, contain pointers to substrate and products and their stochiometric coefficients. The reaction object also encapsulates the functions that compute a_μ, h_μ and update the numbers of substrate and product molecules according to single elementary reactions. The simulator object contains the list of all reactions in the system. While performing a step of the Gillespie algorithm, the simulator computes a_μ, h_μ for every reaction by calling its encapsulated functions. Then it generates the waiting time, chooses a reaction and executes a single elementary reaction by calling its encapsulated function. The data structure described above is dynamically built according to the input file defining the system.

There are three features added to the software that are

```
Initialisation:
    Load reactions and the values of their stochastic rate constants $c_i$ (i=1,..,M).
    Load initial values for the numbers of reactant molecules $X_i$ (i=1,..,N).
    Set time of the simulation t = 0.
Iteration:
    For every reaction calculate $a_\mu = h_\mu c_\mu$ ($\mu$=1,..,M).
    Calculate $a_0 = \Sigma\, a_\mu$
    Generate two random numbers $r_1$ and $r_2$ uniformly distributed over unit interval (0,1)
    Calculate the waiting time for the next reaction as $\tau = (1/a_0) \ln (1/r_1)$.
    Take the index $\mu$ of the next reaction so that $(a_1 + a_2 + ... + a_{\mu-1}) < r_2 a_0 < (a_1 + a_2 +...+ a_{\mu-1})$
    Change the numbers of molecules in the system by executing one elementary reaction $\mu$.
    Set time of the simulation $t = t + \tau$
Termination:
    Terminate simulation when time of the simulation t exceeds preset maximal time of the simulation or
    when all substrates of all reactions in the system are consumed ($a_0$=0).
```

Fig. 1. Gillespie algorithm.

dedicated to the simulation of biological systems: the growing volume of the reaction environment, simulation of cell division and random pools of reactants. The first two features allow simulation of cellular process in the time scale of several cellular generations. During a single generation, the cell doubles its volume. Then, cell division is simulated and the ‘attention’ of the program is switched to one of the new cells with the volume reset to its initial value.

In the current version of the software, only a linear volume change is implemented in the following way. The stochastic rate constants given in the input file must be specific for the initial volume of the system. Therefore, the initial volume is set to 1 and during the generation time T grows up to 2 according to the formula:

$$V(t) = (1 + t/T) \tag{4}$$

where t is the time of the simulation.

Before each step of the Gillespie algorithm, the rates of all second order reactions are divided by the current volume. Therefore, at the beginning of every generation the stochastic rate constants of second order reactions are equal to the values given in the input file and at the end of generation they are twice as small. In a similar way, any growth law for the volume can be added in future, although a linear one was so far sufficient to give reasonable results in the simulation of prokaryotic systems (Arkin *et al.*, 1998; Kierzek *et al.*, 2001).

Cell division has been modeled as follows. First, the numbers of all reactants that model DNA elements are doubled. This is implemented by a separate set of reaction objects that do not take a part in the Gillespie algorithm calculations and are executed only when the system reaches the generation time. Then, all the numbers of molecules present in the system are divided by 2 and the volume is reset to the initial value. In this way, the program continues the simulation with the system which has half of the molecules present at the end of the previous generation and the proper number of DNA elements. Future versions of the software will account for the fact that, in bacterial cells, genes are replicated at different times and expressed from two copies during bacterial generations. I also plan to add the possibility of setting random variation in generation times.

Random pools of reactants have been added in order to model the pools of cellular substances which are in dynamic equilibrium as a result of the large number of competing processes. If the number of molecules in the pool results from many small contributions of other processes and the fluctuations of this number are fast compared to the time-scale of the simulation, its distribution should be Gaussian. Let us take a specific example. In bacterial cells, the number of RNA polymerase molecules which are free to bind to the promoter region of a given gene is determined by the following processes: synthesis of polymerase subunits and their degradation, binding of RNA polymerase to the promoter region of other genes, engaging polymerase in the transcription of other genes and nonspecific binding to DNA (McClure, 1985). All these processes are in a delicate and dynamic balance keeping the number of polymerase molecules fluctuating around some constant mean value. Detailed modeling of all these processes would be extremely difficult, if possible at all. In this example, the STOCKS software allows modeling of the RNA polymerase pool as a random variable with Gaussian distribution. The mean of the distribution can be set according to experimental estimates and the sensitivity of the results to various values of standard deviation can be checked.

The pools are implemented as follows: before computing h_μ and a_μ values in the Gillespie algorithm, the number of molecules in a random pool is drawn from the Gaussian distribution with a specified mean and standard deviation. Then, the simulation continues as described above. The mean value of the number of molecules in the pool grows, together with the volume so the concentration of molecules remains constant. This simulation protocol has been justified in more detail in a previous paper (Kierzek *et al.*, 2001). Arkin *et al.* (1998) used a similar strategy to model equilibrated binding reactions. The number of molecules being in a free and bound state were drawn from an appropriate distribution before executing the step of the Gillespie algorithm. Random pools of reactants, although first implemented in order to model the numbers of ribosomes and RNA polymerase molecules for the purpose of modeling prokaryotic gene expression, will also be useful in the case of other processes in which the number of reactant molecules is buffered by a large number of the processes which are difficult to be modeled explicitly.

One should note at this point that, although the application of random pools can be justified in many cases, there is no rigorous, mathematical proof of this simulation protocol. Simulations with the rigorously proven Gillespie algorithm can be performed with STOCKS software if random pools are not included into the model.

The program uses the random number generator of Marsaglia and Zaman (1990) with the cycle of 2^{144}. Figure 2 shows the schema of the Gillespie algorithm implementation in STOCKS software.

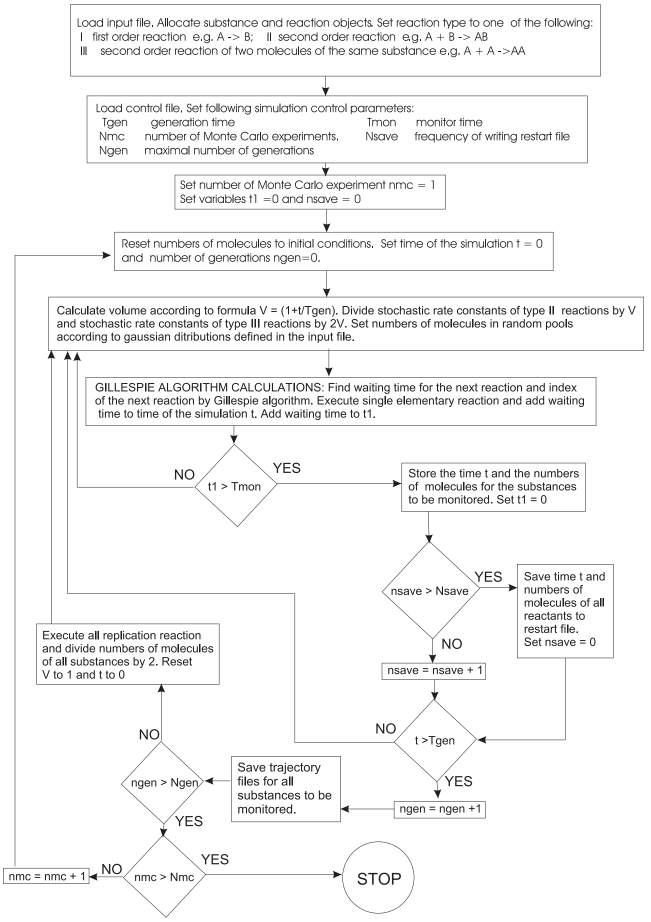

Load input file. Allocate substance and reaction objects. Set reaction type to one of the following:
I first order reaction e.g. A -> B; II second order reaction e.g. A + B -> AB
III second order reaction of two molecules of the same substance e.g. A + A ->AA
Load control file. Set following simulation control parameters:
Tgen generation time
Tmon monitor time
Nmc number of Monte Carlo experiments.
Nsave frequency of writing restart file
Ngen maximal number of generations
Set number of Monte Carlo experiment nmc = 1
Set variables t1 =0 and nsave = 0
Reset numbers of molecules to initial conditions. Set time of the simulation t = 0 and number of generations ngen=0.
Calculate volume according to formula V = (1+t/Tgen). Divide stochastic rate constants of type II reactions by V and stochastic rate constants of type III reactions by 2V. Set numbers of molecules in random pools according to gaussian ditributions defined in the input file.
GILLESPIE ALGORITHM CALCULATIONS: Find waiting time for the next reaction and index of the next reaction by Gillespie algorithm. Execute single elementary reaction and add waiting time to time of the simulation t. Add waiting time to t1.
NO
t1 > Tmon
YES
Store the time t and the numbers of molecules for the substances to be monitored. Set t1 = 0
nsave > Nsave
YES
Save time t and numbers of molecules of all reactants to restart file. Set nsave = 0
NO
nsave = nsave + 1
Execute all replication reaction and divide numbers of molecules of all substances by 2. Reset V to 1 and t to 0
NO
t >Tgen
NO
ngen > Ngen
Save trajectory files for all substances to be monitored.
YES
ngen = ngen +1
YES
nmc = nmc + 1
NO
nmc > Nmc
YES
STOP

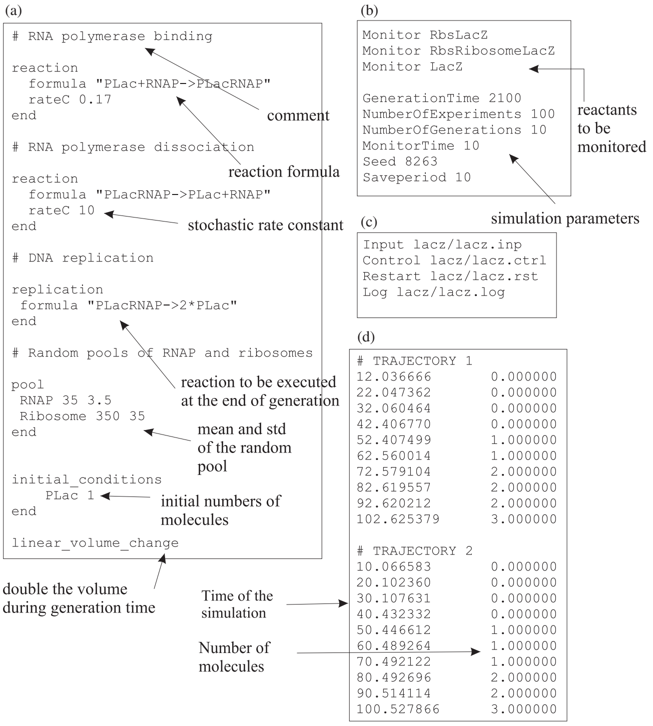
Fig. 2. Implementation of Gillespie algorithm in STOCKS software.

User interface and utility programs

As will follow in the next section, the tasks for which STOCKS software has been written may be computationally expensive and require execution times of a few hours or even days. I believe that this kind of computation are most conveniently executed as background jobs under UNIX operating systems. Therefore, although the software can be compiled and used on other operating systems, its user interface is best suited to the UNIX environment.

The simulation is specified by three text files. The first one specifies the names of the input, control, restart and log files and the directory in which the output is written. The input file contains the specification of the system. The control file contains job control variables—names of reactants to be monitored, the number of Monte Carlo experiments to be performed, etc.

Within the input file, the reaction formulas and stochastic rate constants are specified using a simple syntax which is shown in Figure 3 and described in detail in the pro-



(a)

```
# RNA polymerase binding

reaction
  formula "PLac+RNAP->PLacRNAP"
  rateC 0.17
end

# RNA polymerase dissociation

reaction
  formula "PLacRNAP->PLac+RNAP"
  rateC 10
end

# DNA replication

replication
 formula "PLacRNAP->2*PLac"
end

# Random pools of RNAP and ribosomes

pool
 RNAP 35 3.5
 Ribosome 350 35
end


initial_conditions
    PLac 1
end

linear_volume_change
```

(b)

```
Monitor RbsLacZ
Monitor RbsRibosomeLacZ
Monitor LacZ

GenerationTime 2100
NumberOfExperiments 100
NumberOfGenerations 10
MonitorTime 10
Seed 8263
Saveperiod 10
```

(c)

```
Input lacz/lacz.inp
Control lacz/lacz.ctrl
Restart lacz/lacz.rst
Log lacz/lacz.log
```

(d)

```
# TRAJECTORY 1
12.036666       0.000000
22.047362       0.000000
32.060464       0.000000
42.406770       0.000000
52.407499       1.000000
62.560014       1.000000
72.579104       2.000000
82.619557       2.000000
92.620212       2.000000
102.625379      3.000000

# TRAJECTORY 2
10.066583       0.000000
20.102360       0.000000
30.107631       0.000000
40.432332       0.000000
50.446612       1.000000
60.489264       1.000000
70.492122       1.000000
80.492696       2.000000
90.514114       2.000000
100.527866      3.000000
```

comment
reaction formula
stochastic rate constant
reaction to be executed at the end of generation
mean and std of the random pool
initial numbers of molecules
double the volume during generation time
reactants to be monitored
simulation parameters
Time of the simulation
Number of molecules

Fig. 3. Examples of input, output and control files. (a) Input file. (b) Control file. (c) The file containing names of input, control, restart and log files and the name of the output directory. (d) Example of trajectory files. Two 100-s long trajectories are shown.

gram's MANUAL file. The input file also contains specification of replication reactions. A single elementary reaction is executed for each replication reaction at the end of generation. Therefore, the number of molecules of every reactant which is interpreted as a DNA element must be doubled by specifying the appropriate replication reaction. After execution of all replication reactions, the numbers of all reactants in the system are divided by 2 (see Figure 2). The replication reaction entry in the input file allows the execution of an arbitrary elementary reaction at the end of a generation.

Initial conditions need to be defined in the input file by setting the initial number of all reactants for which this number is not equal to zero. Random pools of reactants can also be specified in the input file by setting the reactant name and two parameters for the Gaussian distribution.

The control file lists the names of reactants to be monitored. For each specified reactant, the program records the number of molecules at specified time intervals. Trajectory files, containing numbers of molecules as a function of time, are saved after every generation time which is also specified in the control file. One can set the time interval in which the restart file is written. This file contains the number of all reactant molecules in the system which allows the resumption of the job in case it has been interrupted. Other variables in the control file specify the number of Monte Carlo repetitions and the seed of the random number generator.

Trajectory files are saved in the output directory specified by the user. They are text files in a simple two-column format that can be imported into any plotting software. The files are optimized for GNUPLOT software as the trajectories are separated by blank lines, so they are treated as a separate data series in GNUPLOT.

Data analysis is aided by four utility programs. The first one calculates average trajectories. It reads trajectory files that contain results of repeated Monte Carlo experiments and computes the average number of molecules. Within each specified time interval, the program computes the mean number of molecules and the standard deviation. It outputs the mean and $+/-$ n trajectories where n is the number of standard deviations specified by the user. An alternative output format is the ratio of standard deviation and the mean at every time interval. This value expresses the magnitude of the random variation at a given time. The ratio of standard deviation to its mean will be referred to as the variation coefficient. In many cases, the user may want to know the sum of the numbers of some molecules (e.g. in Michaelis–Menten reactions the number of enzyme molecules is the sum of the numbers of free enzyme molecules and those with the bound substrate). One of the utility program can be used in such a case to add or subtract trajectories i.e. add or subtract numbers of molecules at corresponding times in two trajectory files. There are two additional programs that can compute the mean number of molecules and fit the linear function to the specified part of the trajectory. The first one can be applied in the case when the number of molecules of the given substance achieves a stationary level i.e. fluctuates around a constant value. This value can be estimated by taking the mean number of molecules in part of the trajectory. In the case in which the numbers of molecules increases or decreases with a constant rate that rate can be estimated by fitting the linear function to the appropriate part of the trajectory.

Both the main and utility programs are very well suited to be run under the control of PERL or UNIX shell scripts. This allows the user execution of complex simulations. In the distribution of the software, I include an example PERL script for the fully automatic calculation of a two dimensional phase plot in which the variation coefficient of a specified reactant is computed as the function of two specified stochastic rate constants. A very limited knowledge of PERL basics is required to modify this script for any other parameter-scanning task.

EXAMPLES OF PROGRAM APPLICATION

Example 1. Dependence between frequencies of transcription and translation initiation and stochastic fluctuations in prokaryotic gene expression—two dimensional phase plot

In a previous paper (Kierzek *et al.*, 2001), the kinetic model of prokaryotic gene expression was presented. Gillespie algorithm simulations were performed with a pre-release version of the STOCKS software. The model was tested against experimental data concerning the speed of protein synthesis and mRNA levels in the LacZ gene of *E. coli* (Kennell and Riezman, 1977). A good agreement with experimental data was achieved. In this paper, I present a refined version of this model with an improved quantitative agreement with experimental data. Parameters of the model and initial conditions for the simulations are presented in Table 1. Table 2 shows the comparison to experimental data. For a detailed discussion of the model's assumption and the way parameters have been derived and justified see Kierzek *et al.* (2001).

The model has already been applied to study the magnitude of random variation in the number of synthesized protein molecules as a function of promoter strength and the strength of the Ribosome Binding Site (RBS). Figure 4 presents 100 independent trajectories obtained for a very weak promoter with an effective frequency of transcription initiation of the order of 10^{-4} s. Every trajectory corresponds to a single cell in which the expression of the gene under investigation is monitored. One can see that the protein is expressed in 'bursts' rather then continuously. For such a weak promoter, the gene is, in most cases, inactive throughout the whole cell generation and slow decay of the number of molecules due to protein degradation is observed. In cells in which transcription events occur, there is a sudden increase in the number of protein molecules. The time intervals between 'transcription bursts' undergo significant random variation. Therefore, what, at the cell culture level, is observed as constitutive gene expression at a very low level is actually the average of the cases in which the gene is transcriptionally active and inactive. In the works of McAdams and Arkin (1997) and Kierzek *et al.* (2001) the consequences of such fluctuations in the expression of regulatory proteins have been discussed.

In the previous work, only two series of simulations were performed. In the first, the promoter strength was decreased with respect to the LacZ model by increasing the RNA polymerase dissociation rate. In the second, the translation initiation frequency was decreased by increasing the ribosome dissociation rate. In this work I present the two-dimensional phase plot in which the magnitude of the random variation in the number of protein molecules is plotted as the function of transcription and translation initiation frequencies. In order to do this it is necessary to define the quantitative measure of random variation. For that purpose I use the variation coefficient of the number of protein molecules i.e. the ratio of standard deviation to the mean number of protein molecules at the given time interval. The simulations previously performed showed that the variation coefficient converges to a constant value if the simulations are carried out in the timescale of several bacterial generations. Therefore, the value to which the variation coefficient converges can be used as the measure of the magnitude of the random variation for the given strength of promoter and RBS site.

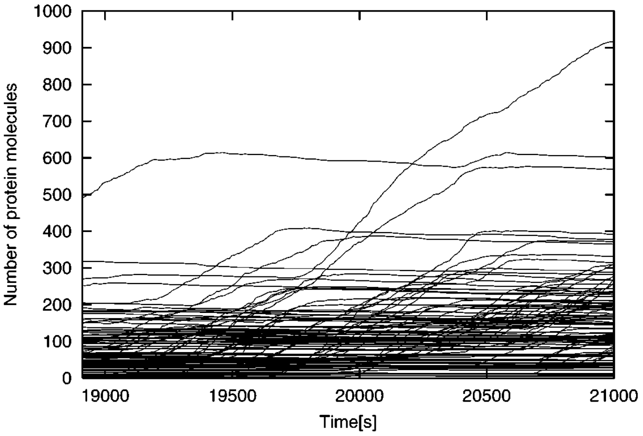


Fig. 4. Simulation of prokaryotic gene expression performed for the case of a very weak promoter. The model presented in Table 1 was used but the RNAP dissociation rate was set to 10 000 1/s. The plot shows 100 trajectories recorded in the 10th cellular generation (generation time 2100 s).

Figure 5 shows the variation coefficient of the number of protein molecules computed as a function of transcription and translation initiation frequencies. Each of the 256 points on this plot corresponds to one simulation with different rates of RNA polymerase dissociation (reaction 2 in Table 1) and ribosome dissociation (reaction 7). In every such simulation, 100 independent trajectories were computed and every trajectory spanned ten generations of the cells with a generation time of 2100 s. It has also been checked for several points on the plot (data not shown) that increasing the number of Monte Carlo runs to 1000 does not significantly change the results. After the results of the 100 program runs have accumulated, the variation coefficient of the number of protein molecules, as a function of time, was computed by a utility program. The value to which the variation coefficient converges was computed as the mean value from the last 1000 s of the simulation and plotted in Figure 5. The actual values of transcription and translation initiation frequencies were also calculated.

The phase plot shown in Figure 5 confirms, by a

Table 1. Kinetic model of LacZ gene expression

(a) Reaction formulas and stochastic rate constants

Reaction	Stochastic rate constant [1/s][a]	Meaning
PLac + RNAP → PLacRNAP	0.17	*RNA polymerase binding*/ RNAP—RNA polymerase. PLac—promoter, PLacRNAP closed RNAP/promoter complex
PLacRNAP → PLac + RNAP	10	*RNA polymerase dissociation*
PLacRNAP → TrLacZ1	1	*Closed complex isomerization* TrLacZ1—open RNAP/promoter complex
TrLacZ1 → RbsLacZ + Plac + TrLacZ2	1	*Promoter clearance*. RBSLacZ—RBS, TrLacZ2—RNA polymerase elongating LacZ mRNA
TrLacZ2 → RNAP	0.015	*mRNA chain elongation and RNAP release*
Ribosome + RbsLacZ → RbsRibosome	0.17	*Ribosome binding*. Ribosome—ribosome molecule, RbsRibosome—ribosome/RBS complex
RbsRibosome → Ribosome + RbsLacZ	0.45	*Ribosome dissociation*
RbsRibosome → TrRbsLacZ + RbsLacZ	0.4	*RBS clearance*. TrRbsLacZ—ribosome elongating LacZ protein chain
TrRbsLacZ → LacZ	0.015	*LacZ protein synthesis*
LacZ → dgrLacZ	6.42e−5	*Protein degradation* dgrLacZ—inactive LacZ protein
RbsLacZ → dgrRbsLacZ	0.3	*Functional mRNA degradation*. dgrRbsLacZ—inactive mRNA

(b) Initial conditions

Substance	Initial number of molecules
Plac	1
RNAP	The number of RNAP molecules available for the LacZ gene was modeled as a random pool with mean 35 and standard deviation 3.5 molecules. Therefore, the initial number of molecules was also drawn from this distribution
Ribosome	The number of ribosomes available for the LacZ gene was modeled as the random pool with mean 350 and standard deviation 35 molecules. Therefore, the initial number of molecules was also drawn from this distribution
Other substances	0

[a] Second order rate constants calculated for a volume of the cell equal to 10^{-15} l. Stochastic rate constants of two second order reactions equal to 0.17 1/s correspond to second order rate constants of 10^8 $\mathrm{M}^{-1}\mathrm{s}^{-1}$.

more systematic approach, the conclusions of the previous paper. One can see that fluctuations in the number of protein molecules grow along the x-axis corresponding to transcription initiation frequency. Translation initiation frequency can be decreased without introducing large fluctuations. It was also checked (data not shown) that the speed of protein synthesis (expression level) is comparable for points B and C on the phase plot. This shows that the same average magnitude of gene expression can be achieved by controlling it at either the promoter or RBS level, but control at the promoter level introduces significantly larger random fluctuations. Discussion of the biological consequences of this fact are given elsewhere (Kierzek *et al.*, 2001).

The calculations of the data shown in Figure 5 have been done fully automatically by executing a PERL script running STOCKS and utility programs. Execution of this task took about 22 h CPU time on a single Pentium III 800 MHz processor under the Linux operating system. The script is given in the distribution of the software and can be used as a framework for executing parameter-scanning simulation protocols.

Example 2. Simulation of LacZ and LacY genes expression and enzymatic/transport activities of LacZ and LacY proteins

The computational costs of the Gillespie algorithm are proportional to the number of elementary reactions that need to be simulated in order to cover the preset time of the simulation. For that reason, it is not possible to simulate reactions that involve macroscopic numbers of molecules as it would imply that the numbers of elementary reactions would have an order of magnitude

Table 2. Comparison of the LacZ gene expression model with experimental data[a]

Quantity	Experimentally determined value[b]	Calculated value[c]
Transcription initiation frequency	0.3 1/s	0.26 1/s
The speed of protein synthesis	20 1/s	22 1/s
Stationary number of mRNA molecules	62	61
Ribosome spacing	110 nucleotides	118 nucleotides

[a] Except quantitative agreement with the experimental data presented by Kennell and Riezman (1977), the model also reproduces decrease of mRNA level resulting from a decrease of the strength of RBS (experimental data in Yarchuk *et al.*, 1992; see Kierzek *et al.*, 2001 for details).

[b] Experimental data from Kennell and Riezman (1977).

[c] Calculations has been performed with the model presented in Table 1. Results for the first bacterial generation (2100 s), were taken as measurements of Kennell and Riezman, were done immediately after LacZ gene induction.

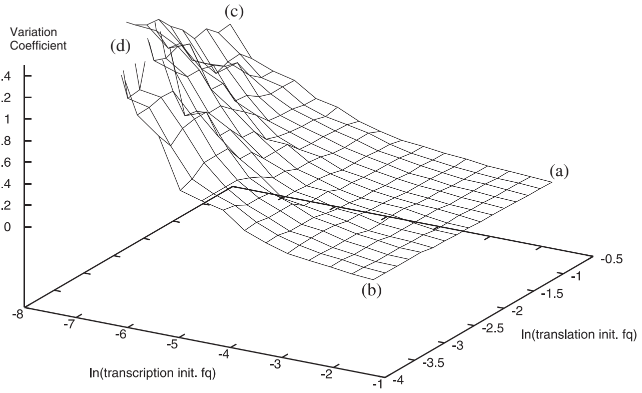


Fig. 5. Variation coefficient of the number of protein molecules as the function of transcription and translation initiation frequencies calculated for the model of prokaryotic gene expression presented in Table 1. For each 10 s time interval, the mean number of protein molecules and its standard deviation were computed from 100 independent runs. In every simulation the ratio of standard deviation to the mean (variation coefficient) converged to the constant value shown in the plot. (a) Parameters of LacZ gene. (b) Gene with weak RBS and strong promoter. (c) Strong promoter and weak RBS. (d) For very low both transcription and translation initiation frequencies, very high values of variation coefficient (2.94) were obtained. These values are not shown on the plot.

of Avogadro's number. In the previous example, it was shown that for numbers of molecules characteristic for gene expression phenomena, the algorithm is fast enough to allow parameter scanning tasks that involve long-timescale simulations (several bacterial generations). The purpose of this example is to test the applicability of the software to biochemical systems involving enzymatic processes and small-molecule reactants. The numbers of elementary reactions in such a systems are much greater than in systems composed exclusively of macromolecular reactants.

As an example, I took the expression and activity of LacZ and LacY proteins in *E. coli*. As the purpose of the calculations is to test the computational limits of the software rather than building a detailed model of the lactose operon, regulation by the lac repressor was unaccounted for. Therefore, the example corresponds to the LacI$^-$ strain of *E. coli*—the mutant lacking active lac repressor and expressing LacZ and LacY proteins constitutively.

Transcription initiation and LacZ expression were modeled using reactions and parameters listed in the Table 1. Transcription of LacY mRNA was modeled in the following way. Reaction 5 in Table 1 was modified so that at the end of LacZ transcription, a new 'reactant' (TrLacY1) appears which models RNA polymerase transcribing LacY mRNA:

Formula: TrLacZ2 $\rightarrow$ TrLacY1
Stochastic rate constant: 0.015 1/s.

Then, RBS synthesis, mRNA degradation and ribosome binding/dissociation have been modeled by the same reactions as in the case of LacZ protein. The reaction modeling protein chain elongation has the stochastic rate constant of 0.36 1/s.

The Michaelis constant and turnover number of *E. coli* beta-galactosidase were assigned values of 7.52 mM and 431 1/s respectively according to the BRENDA database entry for EC 3.2.1.23. The Michaelis constant was expressed as the number of molecules ($7.52 \cdot 10^5$) in the volume of the cell (10^{-15} l). The dissociation of the ligand was neglected and the stochastic rate constant of ligand binding was computed as the ratio of turnover number and Michaelis constant. Therefore, the enzymatic activity of beta galactosidase was modeled by the following reactions:

Formula: LacZ + lactose $\rightarrow$ LacZlactose (5a)
Stochastic rate constant: $5.731 \cdot 10^{-4}$ 1/s
Formula: LacZlactose $\rightarrow$ product + LacZ (5b)
Stochastic rate constant: 431 1/s.

The turnover number of lactose permease was assigned as 14 1/s, according to the measurements of Lolkema *et al.* (1991). I assumed that the external lactose concentration is saturating and lactose is transported into the cell with a maximal rate during the whole time of the simulation. Thus, the permease action was modeled by a single

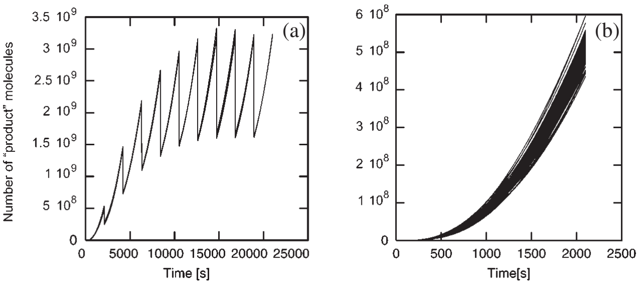


Fig. 6. The number of enzymatic reactions performed as a function of time. The number of reactions were determined by counting ‘product’ molecules (each product molecule corresponds to one digested lactose molecule; see (5b). (a) Results of 3 independent simulations spanning 10 bacterial generations. After every generation the number of product molecules is divided by 2, as for all other molecules in the system. As one can see, the system reaches a stationary state in which approximately $2 \cdot 10^9$ enzymatic reactions are performed in one generation. (b) Results of 100 independent simulations spanning a single generation.

reaction:

$$\text{Formula:} \qquad \mathrm{LacY} \rightarrow \mathrm{lactose} + \mathrm{LacY} \qquad (6)$$

$$\text{Stochastic rate constant:} \qquad 14\ 1/\mathrm{s}.$$

This treatment of the permease reaction exaggerated the number of lactose molecules present in the cell which serves the purpose of our benchmark.

The results of simulations for the system described above are presented in Figure 6. As expected, the computations were much more time consuming than in the case of Example 3. Computation of the single trajectory for one bacterial generation (2100 s) took approximately 2.5 h on a Pentium III 800 MHz processor. Simulation of a single trajectory spanning ten bacterial generations took approximately 90 h. This is caused by the fact that the number of reactions needed to be executed in the first generation is much smaller than the number of reactions appearing in subsequent generations. In Figure 6, one can see that in the last generations the number of ‘product’ molecules produced per generation was about $2 \cdot 10^9$, whereas in the first generation $5 \cdot 10^8$ molecules were produced.

DISCUSSION

The software for stochastic simulations of biochemical processes that implements the rigorously justified Gillespie algorithm have been presented. The algorithm remains correct for reactions involving arbitrary small numbers of molecules, which is very important in the modeling of biochemical reactions. It allows the study, not only of the average course of the process, but also fluctuations in the number of molecules influencing various cellular processes. The simulation algorithm does not approximate infinitesimal time increments by finite timesteps so the user need not to be concerned in setting arbitrary timesteps. This property of the algorithm is also advantageous from the point of view of numerical stability. The calculations are numerically stable even in the case of a system that is composed of reactions that differ by 8 orders of magnitude in the number of reactant molecules involved and reaction rates (Example 3).

Arbitrary reaction networks can be defined and simulated by STOCKS software, provided that they are composed exclusively of first and second order reactions. The Gillespie algorithm can be applied only if the system is defined in terms of elementary reactions. Therefore, if kinetic parameters of a complex mechanism (e.g. Michaelis–Menten reaction in Example 3) are available, the user must express them in terms of elementary first and second order reactions. This usually needs to be done at the expense of additional assumptions, e.g. assuming the irreversibility of ligand binding in the case of reaction (6). On the other hand, application of the complex mechanism is usually correct only for a system in the stationary state (this is also the case in reaction (6)). If parameters of elementary reactions can be found/approximated and the time evolution of the system is numerically simulated, the severe assumption of the stationary state can be avoided. This is especially important if regulated processes are under investigation. When, for instance, the gene changes its expression level as the result of induction or repression, the system is far from being in the stationary state.

As was mentioned in the introduction, computer simulations of biochemical systems with the Gillespie algorithm can also be performed by SIMULAC software. The major difference between STOCKS and SIMULAC is the implementation of a simple model of cell divisions that allows application of STOCKS to simulate several cellular generations. This feature of the program was applied, for instance, to show that, in timescales longer than a single bacterial generation, the variation coefficient of constitutively expressed gene converges to a stationary level (Kierzek *et al.*, 2001). Another difference between STOCKS and SIMULAC software are the utility programs that aid analysis of trajectories calculated by STOCKS. One feature of SIMULAC that is not implemented in STOCKS is a dedicated mechanism to model the binding of transcription factors to DNA sites. It is assumed that the binding of transcription factors to regulatory regions is much faster than transcription initiation at the promoter. Using this rapid equilibrium assumption, the promoter state is chosen randomly at each instant using probabilities calculated by partition function formulated by Shea and Ackers (1985). In the input file of SIMULAC, the binding of protein to a DNA site is described by free energies rather than rate constants.

In the previous paper, it was shown that application of a pre-release version of the software allowed building of a reliable, kinetic model of prokaryotic gene expression and yielded insights into stochastic phenomena involved in this process. Here I show that the software is computationally fast enough to allow for parameter scanning tasks in modeling systems involving macromolecular reactants with realistic numbers of molecules and reaction rates. Modeling of processes involving intensive metabolic reactions is much more computationally demanding. Example 3 shows that the computational cost of simulating metabolic reactions together with gene expression processes is very high. Calculations of this kind are affordable if the user limits the simulation time scale to a single generation. If several generations need to be computed, a multiple processor platform would be necessary. From the point of view of parallelization, Monte Carlo simulations are convenient, as independent Monte Carlo experiments can be run in parallel on different processors/computers. If, in the case of Example 3, independent Monte Carlo experiments would be run on a few independent processors the statistics could be collected within 1 week even in the case of a simulation spanning 10 bacterial generations. Therefore, I conclude that the current version of the software can be applied to exact simulations of large metabolic networks if run, for instance, on a PC cluster—the platform which becomes affordable for most laboratories.

There is an ongoing effort towards improving computational efficiency of exact algorithms for simulation of kinetics of coupled chemical reactions. Gibson and Bruck (2000) formulated the next reaction method—an exact algorithm to simulate coupled chemical reactions that use only one random number per elementary reaction event and takes a time proportional to the logarithm of the number of reactions instead of the time proportional to the number of reactions itself. The authors (Gibson, 2000) have also formally analyzed the problem of kinetic parameter estimation in stochastic simulations. In his recent paper, Gillespie (2001) presented ideas of how to significantly decrease computational costs by ‘linking’ stochastic and deterministic regimes with an acceptable loss of accuracy. Further development of STOCKS software will be directed towards implementation of these theoretical concepts.

STOCKS software is available under a GNU GPL licence from an anonymous ftp site ftp://ibbrain.ibb.waw.pl/stocks. Distribution of the program includes parameters of the model of LacZ gene expression and all examples presented in this paper including the PERL script which can be easily customized for complex simulation tasks.

ACKNOWLEDGEMENTS

I am grateful to Professor P.Zielenkiewicz for critical comments on the manuscript. I am also very grateful to an anonymous referee for his efforts in improving this paper and for some good ideas for future developments.

REFERENCES

Alberts,B., Bray,D., Lewis,J., Raff,M., Roberts,K. and Watson,J. (1994) *Molecular Biology of the Cell*, 3rd edn, Garland, New York.

Arkin,A., Ross,J. and McAdams,H.H. (1998) Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected *Escherichia coli* cells. *Genetics*, **149**, 1633–1648.

Barshop,B.A., Wrenn,R.F. and Frieden,C. (1983) Analysis of numerical methods for computer simulation of kinetic processes: development of KINSIM—a flexible, portable system. *Anal. Biochem.*, **130**, 134–145.

Carrier,T.A. and Keasling,J.D. (1999) Investigating autocatalytic gene expression systems through mechanistic modeling. *J. Theor. Biol.*, **201**, 25–36.

Cornish-Bowden,A. and Hofmeyr,J.H. (1991) MetaModel: a program for modelling and control analysis of metabolic pathways on the IBM PC and compatibles. *Comput. Appl. Biosci.*, **7**, 89–93.

Dang,Q. and Frieden,C. (1997) New PC versions of the kinetic-simulation and fitting programs, KINSIM and FITSIM. *Trends Biochem. Sci.*, **22**, 317.

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

Garcia-Olivares,A., Villarroel,M. and Marijuan,P.C. (2000) Enzymes as molecular automata: a stochastic model of self-oscillatory glycolytic cycles in cellular metabolism. *Biosystems*, **56**, 121–129.

Gibson,M.A. (2000) *Computational Methods for Stochastic Biological Systems*, PhD Thesis, California Institute of Technology, Pasadena, California.

Gibson,M.A. and Bruck,J. (2000) Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem.* A, **104**, 1876–1889.

Gillespie,D.T. (1977) Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, **81**, 2340–2361.

Gillespie,D.T. (2001) Approximate accelerated stochastic simulation of chemically reacting systems. *J. Phys. Chem.*, **115**, 1716–1733.

Goryanin,I., Hodgman,T.C. and Selkov,E. (1999) Mathematical simulation and analysis of cellular metabolism and regulation. *Bioinformatics*, **15**, 749–758.

Hume,D.A. (2000) Probability in transcriptional regulation and its implications for leukocyte differentiation and inducible gene expression. *Blood*, **96**, 2323–2328.

Kierzek,A.M., Zaim,J. and Zielenkiewicz,P. (2001) The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression. *J. Biol. Chem.*, **276**, 8165–8172.

Kennell,D. and Riezman,H. (1977) Transcription and translation initiation frequencies of the *Escherichia coli* lac operon. *J. Mol. Biol.*, **114**, 1–21.

Laurenzi,I.J. and Diamond,S.L. (1999) Monte Carlo simulation of the heterotypic aggregation kinetics of platelets and neutrophils. *Biophys. J.*, **77**, 1733–1746.

Levin,B. (1999) *Genes VII*. Oxford University Press, Oxford.

Levin,M.D., Morton-Firth,C.J., Abouhamad,W.N., Bourret,R.B. and Bray,D. (1998) Origins of individual swimming behavior in bacteria. *Biophys. J.*, **74**, 175–181.

Lolkema,J.S., Carrasco,N. and Kaback,H.R. (1991) Kinetic analysis of lactose exchange in proteoliposomes reconstituted with purified lac permease. *Biochemistry*, **30**, 1284–1290.

Marsaglia,G. and Zaman,A. (1990) Toward a universal random number generator. *Stat. Prob. Lett.*, **8**, 35–39.

McAdams,H.H. and Arkin,A. (1997) Stochastic mechanisms in gene expression. *Proc. Natl Acad. Sci. USA*, **94**, 814–819.

McClure,W.R. (1985) Mechanism and control of transcription initiation in prokaryotes. *Annu. Rev. Biochem.*, **54**, 71–204.

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

Mendes,P. (1997) Biochemistry by numbers: simulation of biochemical pathways with Gepasi 3. *Trends Biochem. Sci.*, **22**, 361–363.

Mendes,P. and Kell,D.B. (2001) MEG (Model Extender for Gepasi): a program for the modelling of complex, heterogeneous, cellular systems. *Bioinformatics*, **17**, 288–289.

Morton-Firth,C.J. and Bray,D. (1998) Predicting temporal fluctuations in an intracellular signalling pathway. *J. Theor. Biol.*, **192**, 117–128.

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

Shea,M.A. and Ackers,G.K. (1985) The OR control system of bacteriophage lambda. A physical-chemical model for gene regulation. *J. Mol. Biol.*, **181**, 211–230.

Siegele,D.A. and Hu,J.C. (1997) Gene expression from plasmids containing the araBAD promoter at subsaturating inducer concentrations represents mixed populations. *Proc. Natl Acad. Sci. USA*, **94**, 8168–8172.

Tomita,M., Hashimoto,K., Takahashi,K., Shimizu,T.S., Matsuzaki,Y., Miyoshi,F., Saito,K., Tanida,S., Yugi,K., Venter,J.C. and Hutchison,C.A. 3rd. (1999) E-CELL: software environment for whole-cell simulation. *Bioinformatics*, **15**, 72–84.

Yarchuk,O., Jacques,N., Guillerez,J. and Dreyfus,M. (1992) Interdependence of translation, transcription and mRNA degradation in the lacZ gene. *J. Mol. Biol.*, **226**, 581–596.