Proceedings of Department of Defense High Performance Computing Modernization Program Users Group Conference 2007,
June 18-22, 2007, Pittsburgh, PA

Stochastic Simulations of Cellular Biological Processes

Yaroslav Chushak
US Army Medical Research and Materiel Command, Biotechnology HPC Software Applications Institute (MRMC/BHSIA), Ft. Detrick, MD
schushak@bioanalysis.org

Brent Foy
Wright State University, Dayton, OH
brent.foy@wright.edu

John Frazier
US Air Force Research Laboratory (AFRL), Wright-Patterson AFB, OH
john.frazier@wpafb.af.mil

Abstract

At the functional level, all biological processes in cells can be represented as a series of biochemical reactions that are stochastic in nature. We have developed a software package called Biomolecular Network Simulator (BNS) that uses a stochastic approach to model and simulate complex biomolecular reaction networks. Two simulation algorithms—the exact Gillespie stochastic simulation algorithm and the approximate adaptive tau-leaping algorithm—are implemented for generating Monte Carlo trajectories that describe the evolution of a system of biochemical reactions. The software uses a combination of MATLAB and C-coded functions and is parallelized with the Message Passing Interface (MPI) library to run on multiprocessor architectures.

1. Introduction

All chemical reactions require the interaction of the reactants with sufficient energy to overcome the activation energy barrier and, fundamentally, they are stochastic in nature. The best way to model kinetics of a system of chemical reactions is to use a stochastic approach in terms of the Chemical Master Equation (CME), with the number of molecules of each molecular species as variables. The CME describes transitions of the system from one state to another state based on probabilistic methods. Gillespie proposed a method to simulate probabilistically-correct trajectories based on the CME through the use of Monte Carlo methods[1].

Although the Gillespie stochastic algorithm is a method for exact simulations, as it explicitly counts each reaction event that occurs in the system, the accuracy of the method comes at a high computational cost. This is especially true for systems with a large number of molecular species, where reactions occur numerous times in a short period of time. To accelerate discrete stochastic simulation, Gillespie proposed the tau-leaping method as an approximate simulation strategy[2]. By using Poisson random numbers, the tau-leaping method can leap over many reactions without a significant loss of accuracy.

We developed a software package—the BNS—that can use the Gillespie stochastic algorithm or the tau-leaping algorithm to simulate the behavior of a system of biochemical reactions. It allows scientists to build a synthetic biomolecular network and explore its performance utilizing the capacities of high performance computing (HPC). BNS contains tools for both simulating the system and for analyzing the results of the simulation. Since the simulations are stochastic in nature, the user must run thousands of simulations to characterize the "ensemble" behavior of biological systems. The parallelized BNS code allows users to run multiple simulations and store results on multiprocessor platforms. In this paper, we present a brief description of the BNS software along with some examples.

2. Stochastic Simulation Algorithm

Let us consider a system composed of N well-mixed chemical species, S_i (i = 1,...,N), in a fixed volume V, which are involved in M reactions, R_j(j= 1,...,M). The dynamical state of the system can be specified by the state vector X(t) = (X_1(t), X_2(t),...,X_N(t)), where X_i(t) is the number of molecules of species S_i at time t.

0-7695-3088-5/07 $25.00 © 2007 IEEE

The probability for each reaction R_j is defined by a propensity function $a_j(X) = c_j \cdot H_j(X)$, where c_j is the stochastic probability constant and $H_j(X)$ represents the number of possible combinations of reactants. The probability that reaction R_j will occur in the time interval dt is defined as $a_j(X)dt$. Each reaction is also characterized by its state-change vector $v_j = (v_{1j}, v_{2j}, ..., v_{Nj})$, where v_{ij} is the change in the number of molecules S_i caused by one reaction R_j.

To study the evolution of the state vector X(t), Gillespie proposed an algorithm for Monte Carlo generation of stochastic trajectories[1]. The direct simulation algorithm, which is implemented in BNS, answers two questions: (1) which reaction will occur next? and (2) what is the waiting time for the next reaction to occur?

To answer these questions, two random numbers, r_1 and r_2, uniformly distributed over the interval (0,1) are generated. The first random number is used to determine the next reaction R_j, such that

$$\sum_{i=1}^{j-1} a_i < r_1 \cdot a_0 < \sum_{i=1}^{j} a_i. \quad (1)$$

where

$$a_0 = \sum_{j=1}^{M} a_j . \quad (2)$$

The distribution of the waiting time is given by following probability density function

$$P(\tau, j) = a_j \exp(-a_0 \tau). \quad (3)$$

Here, $P(\tau, j)$ is the probability that the waiting time for the reaction is τ and that it will be an R_j reaction. The waiting time for the next reaction is calculated as[1]

$$\tau = \frac{1}{a_0} \log \frac{1}{r_2}. \quad (4)$$

After the next reaction and its waiting time are determined, the reaction is executed and the state of the system is changed according to the state-change vector v_j. The simulation time is increased by τ and the next simulation step is generated.

The Gillespie stochastic algorithm is exact for the elementary reactions, uni-uni, uni-bi, and bi-uni, with any number of molecules in the system. For a system with large numbers of molecules, the trajectories generated by stochastic Monte Carlo simulations converge to the trajectory generated by deterministic differential equations[1].

3. TAU-Leaping Algorithm

The tau-leaping method calculates a time interval τ, which encompasses more than one reaction event and satisfies the Leap Condition, i.e., the expected state change induced by the leap must be sufficiently small that no propensity function changes its value by a significant amount. Several methods have been proposed recently to choose the size of the time interval for tau-leaping[3-5]. We implemented the tau-leaping method proposed by Cao et al.[6]. In that method, a tau selection formula is given by

$$\tau' = \min_{i \in I_{rs}} \left\{ \frac{\max\{\varepsilon x_i / g_i, 1\}}{|\mu_i(x)|}, \frac{\max\{\varepsilon x_i / g_i, 1\}^2}{\sigma_i^2(x)} \right\}, \quad (5)$$

where g_i is the highest order of reaction in which species S_i appears as reactant, ε is an error control parameter, I_{rs} is the set of indices of all reactant species, and

$$\mu_i(x) = \sum_j v_{ij} a_j(x), \quad (6)$$

$$\sigma_i^2(x) = \sum_j v_{ij}^2 a_j(x). \quad (7)$$

After a time interval τ' has been selected, the number of firings of each reaction channel during this time interval is approximated as a Poisson random variable. The Poisson random variable can have an arbitrarly large value and the population of some of the molecular species is allowed to temporarily attain negative values. To avoid negative populations we used a modified tau-leaping algorithm proposed in Reference 7. With this approach a second control parameter n_c is introduced; a positive integer typically in the range of 5–20. If a number of molecular species becomes less than n_c, all reactions in which that species appears as a reactant are defined as critical reactions. These reactions are simulated by the stochastic simulation algorithm. We calculate the sum of propensity functions of all the critical reactions a_0^c and generate a second candidate time τ'' according to

$$\tau'' = \frac{1}{a_o^c} \log \frac{1}{r}. \quad (8)$$

The actual time leap τ is selected as the smaller of τ' and τ''. If $\tau = \tau'$, a number of firings k_i is generated as a Poisson random variable with mean $a_i(x)\tau$ for all noncritical reactions and the reactions are executed; no critical reaction is executed in this case. If $\tau = \tau''$, we generate k_i and fire noncritical reactions as in the previous case; also, another random number with uniform distribution is generated to find which critical reaction needs to be fired according to Eq. (1).

Evaluations of the performance of the tau-leaping algorithm shows a 2–3 fold speedup in simulations compared to the exact stochastic algorithm, while maintaining excellent accuracy with regards to both rare and frequent events.

4. Biomolecular Network Simulator

The Biomolecular Network Simulator uses a combination of MATLAB and C-coded modules. The front-end, graphical user interface (GUI) and analysis tools of BNS are written in MATLAB, while the simulation core engine is written in C. Such a combination allows one to use the interactive features and visualization tools of MATLAB, while achieving high speed for the computationally intensive part of the software. The parallelization of the code is done with the help of a MPI library. The BNS can be run on any computer platform where MATLAB 6.5 or newer is installed.

4.1. Input Data

A model is a set of mathematical relationships that describe the behavior of biochemical reactions that control cellular biological processes. Each of the 'model' directories contains one or more subdirectories with model description files and/or different sets of parameters for the same model and an 'output' directory where the results of the simulations are stored. There are two types of model directories: one for models defined in the Systems Biology Markup Language (SBML) format[8] and one for models defined as a set of MATLAB m-files, referred to as the BNS format. In addition, BNS allows one to perform simulations with multiple parameter sets, with each parameter set being run multiple times. Simulations with multiple parameter sets can be used for optimization and sensitivity analysis of the model.

4.2. Output Data

There are two types of output files: snapshot data and event log data. Snapshot data files contain the state of the system (number of molecules of each molecular species) at user-specified time intervals. The information stored in the snapshot files is used to create runtime interactive graphics and for *post hoc* analysis of the data. The second type of output files, the event log files, contain the record of every discrete event that occurs during the simulation. The user should be aware that event log files may require considerable memory or hard disk space and, therefore, may create memory management problems for simulations involving a large number of long runs or for large reaction networks.

4.3. Parallelization

The parallelization of the BNS code is accomplished using the MPI library. In our parallelization scheme, the 'master' processor divides the total number of user-specified runs among the available processors, sends a set of jobs to each of the 'workers', and performs some of the simulation runs itself. In this approach to parallelization, sometimes called "embarrassingly parallel", we reduce the communication among nodes and increase the speedup of multiple simulations. To test BNS scaling with the number of processors, we ran 1,000 simulations on an SGI Origin 3900 machine at the Aeronautical Systems Center Major Shared Resource Center, which has a shared memory architecture. A simulation is defined as a single run of the mathematical model for a particular biochemical reaction network. A 92-fold speedup was observed when running BNS on 100 processors (Figure 1).

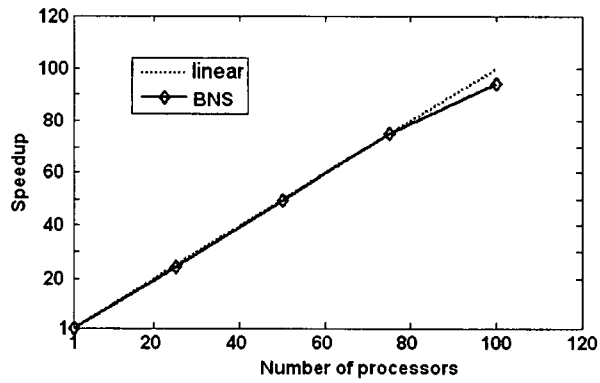

Figure 1. Scaling of BNS with the number of processors

4.4. Running the Simulations

The BNS can be run either in command line mode or via a GUI. The GUI allows the user to modify model parameters at runtime and to execute simulations in the interactive or batch mode on HPC resources.

The main dialog window of the BNS GUI is shown in Figure 2. It allows the user to select the appropriate 'Model' and 'Parameters' directories and set the 'Run' mode. A click on the 'Details' button next to the 'Parameters' directory opens a new window, shown in Figure 2(b). This dialog window allows the user to modify model parameters and to set parameters for the simulation.

If simulations are run in an interactive mode, partial results of the simulations will appear on the screen during the run. Usually, HPC centers allocate limited resources (in number of processors and running time) for interactive simulations. Therefore, in addition to running in an interactive mode, BNS can also run in 'batch' mode. In

this mode, all output data are stored on the hard drive for further analysis.

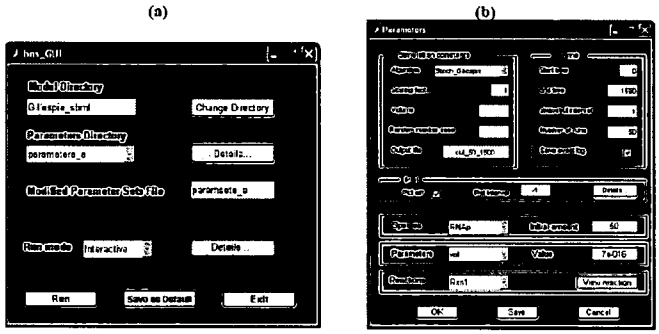

Figure 2. The screen shots of the BNS GUI dialog windows. (a) The main BNS dialog window. (b) The parameters dialog window, which allows users to modify the model parameters and to set simulation parameters.

4.5. Analysis

BNS has a comprehensive set of tools for post-simulation analysis. A GUI for the analysis tools allows the user to easily select the data and to set conditions for the analysis. Multiple types of post-simulation analyses are available.

4.5.1. Plots of Number of Molecules vs. Time

The most frequently used type of analysis is a plot of the number of molecules vs. time. Such plots are available in the interactive mode or as post-simulation analysis. There are two ways to create plots: each compound is plotted on a separate figure or multiple compounds are plotted on the same figure window (grouping mode). The number of molecules vs. time plots can be created with both types of output files: snapshot data or event log data.

4.5.2. Time-Weighted Average Analysis

A time-weighted average analysis refers to the calculations of the average number of molecules of a particular molecular species during a user-selected time-averaging interval. The average is weighted according to the amount of time the compound exists in each state during the selected time-averaging interval. The time-weighted average is then plotted versus time. The averaging analysis can be performed for a single run or for a selected set of runs. When the analysis is applied to multiple runs, the plot shows the between run average (the average of each individual time-weighted average) and standard deviation.

The graphs in Figure 3 show the behavior of two molecular species, S1 and S2, over the time interval of 2,000 seconds for a biomolecular reaction network, containing transcription, translation and metabolic reactions. The simulations were carried out using two different algorithms implemented in BNS: Gillespie stochastic algorithm and tau-leaping algorithm with parameters $\varepsilon = 0.1$ and $n_c = 5$. Figure 3 shows the between-run average of the time-weighted average number of molecules for 200 runs using a time-averaging interval of 50 sec. The average number of molecules of species S1 changed in the range of ~0–20 molecules, while the average number of molecules of species S2 changed ~0–2,500 molecules. Results obtained using an approximate tau-leaping algorithm are almost indistinguishable from the results of the exact Gillespie algorithm for both types of species. On the other hand, using the tau-leaping algorithm speeds up simulations more than 3-fold compared to the exact stochastic algorithm.

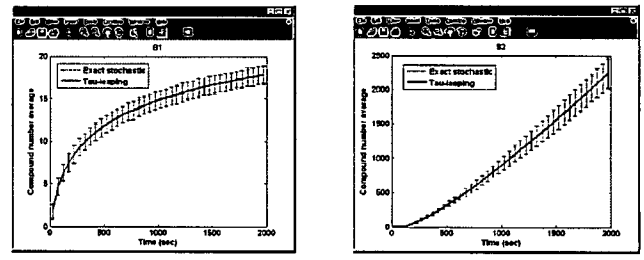

Figure 3. The averaged number of compounds S1 and S2 in the simulation interval (0, 2,000) obtained using the exact Gillespie stochastic algorithm and approximate tau-leaping algorithm (ε=0.1, n_c=5). The time-weighted average was calculated using a 50-sec time-averaging interval and the individual time-weighted averages were averaged over the 200 simulation runs. Data for the mean ± SD are shown.

4.5.3. Reaction Frequency Analysis

Complex biomolecular reaction networks usually contain reactions that occur on different time scales: some reactions have a low propensity and occur rarely, while other reactions are very fast and occur frequently. The data stored in the event log files allow the user to perform various reaction-frequency analyses of the simulation data to learn more about the basic nature of the system. One type of analysis creates plots of the total number of times each reaction in the network occurs during the simulation.

A second type of analysis is the average and standard deviation of the time-averaged reaction frequency in each reaction channel. Figure 4 shows an example of the time-averaged reaction frequency for two reaction channels averaged over the 200 runs obtained by running simulations with the two algorithms. The reaction rxn21 belongs to the group of "slow" reactions with the frequency of occurrences in the range of 0–20 firings per second. The reaction rxn24 is a "fast" reaction and reached 500–600 occurrences per second. As in the case

of average number of molecules, the tau-leaping algorithm shows an excellent agreement with the results from exact stochastic simulations for this particular biomolecular reaction network.

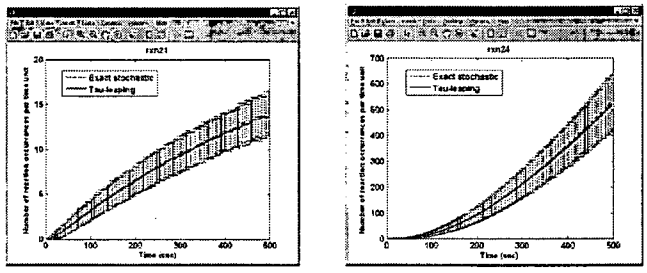

Figure 4. The averaged number of reaction occurrences per time-averaging interval for the reaction channels rxn21 and rxn24 obtained using the exact Gillespie stochastic algorithm and approximate tau-leaping algorithm ($\varepsilon = 0.1$, $n_c = 5$). The time-averaged rate was calculated using a 5-sec time-averaging interval and the individual time-averaged rates were averaged over the 200 simulation runs. Data for the mean ± SD are shown.

5. Conclusions

The BNS allows the users to simulate the behavior of complex biological processes utilizing the capabilities of HPC systems. Some of the features that distinguish BNS from similar tools include:

- usage of MATLAB and C-coded functions allows the user to combine intensive visualization of data with high speed computations;
- parallelized code for multiple simultaneous simulations allows the user to run BNS on multiprocessor machines;
- options to run the code in the interactive or batch mode;
- user friendly graphical user interface allows the user to easily set and modify parameters of the model, simulation and analysis; and
- comprehensive tool sets provide for post-simulation analysis of results.

Acknowledgments

The work of Yaroslav Chushak was sponsored by the US Department of Defense High Performance Computing Modernization Program, under the High Performance Computing Software Applications Institutes initiative. The work of Brent Foy was made possible by a grant from the Air Force Office of Scientific Research and by the Air Force Summer Faculty Fellowship Program.

Disclaimer

The opinions or assertions contained herein are the private views of the authors and are not to be construed as official or as reflecting the views of the US Army or of the US Department of Defense. This paper has been approved for public release; distribution is unlimited.

References

1. Gillespie, D., "Exact Stochastic Simulation of Coupled Chemical Reactions." *J. Phys. Chem.*, v. 81, p. 2340, 1977.

2. Gillespie, D., "Approximate accelerated stochastic simulations of chemically reacting systems." *J. Chem. Phys.*, v. 115, p. 1716, 2001.

3. Gillespie, D. and L. Petzold, "Improved leap-size selection for accelerated stochastic simulations." *J. Chem. Phys.*, v. 119, p. 8229, 2003.

4. Tian, T. and K. Burrage, "Binomial leap methods for simulation chemical kinetics." *J. Chem. Phys.*, v. 121, p. 10356, 2004.

5. Chatterjee, A., D. Vlachos, and M. Katsoulakis, "Binomial distribution based τ-leap accelerated stochastic simulation." *J. Chem. Phys.*, v. 122, 024112, 2005.

6. Cao, Y., D. Gillespie, and L. Petzold, "Efficient step size selection for the tau-leaping simulation method." *J. Chem. Phys.*, v. 124, 044109, 2006.

7. Cao, Y., D. Gillespie, and L. Petzold, "Avoiding negative populations in explicit Poisson tau-leaping." *J. Chem. Phys.*, v. 123, 054104, 2005.

8. Hucka, M., A. Finney, H.M. Sauro, and H. Bolouri, et al., "The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models." *Bioinformatics*, v. 19, p. 524, 2003.