Hybrid simulation of cellular behavior

Thomas R. Kiehl, Robert M. Mattheyses and Melvin K. Simmons

Bioinformatics Laboratory, GE Global Research, 1 Research Circle, Schenectady NY 12309, USA

ABSTRACT

Motivation: To be valuable to biological or biomedical research, *in silico* methods must be scaled to complex pathways and large numbers of interacting molecular species. The correct method for performing such simulations, discrete event simulation by Monte Carlo generation, is computationally costly for large complex systems. Approximation of molecular behavior by continuous models fails to capture stochastic behavior that is essential to many biological phenomena.

Results: We present a novel approach to building hybrid simulations in which some processes are simulated discretely, while other processes are handled in a continuous simulation by differential equations. This approach preserves the stochastic behavior of cellular pathways, yet enables scaling to large populations of molecules. We present an algorithm for synchronizing data in a hybrid simulation and discuss the trade-offs in such simulation. We have implemented the hybrid simulation algorithm and have validated it by simulating the statistical behavior of the well-known lambda phage switch. Hybrid simulation provides a new method for exploring the sources and nature of stochastic behavior in cells.

Contact: simmons@research.ge.com

Supplementary information: The SBML file for the lambda phage tests will be made available at the OUP site.

INTRODUCTION

New data from high-throughput biology will enable more ambitious, more complete and better-validated simulation models, allowing *in silico* methods to take an important place in biological research. Indeed, understanding the functional significance of protein pathways (Biocarta, 2002, http://www.biocarta.com; KEGG, 2003, http://www.genome.ad.jp/kegg/, Li *et al*., 2002; Bader, 2003; Demir *et al*., 2002) will require the extensive use of simulations. Critical to this program is the ability to capture and analyze stochastic processes that arise from the low copy number of molecules participating in pathways, and the complex interactions that amplify molecular variability to create stochastic behavior at the cellular level.

MODELING OF CELLULAR PROCESSES

Computer simulations can now replicate important biological behavior (Kitano, 2002). Early success was achieved with the simplest organisms: the phages. The lambda phage 'switch' has been a target of several efforts (Shea and Ackers, 1985; Arkin *et al*., 1998). The T-7 phage has been studied in a fruitful combination of simulation and laboratory measurements (Endy *et al*., 1997; You *et al*., 2002). However eukaryotes in general, and mammalian cells in particular, are much more challenging targets (Asthagiri and Lauffenburger, 2000). The scope of models considered here is intended to cover cellular processes with stochastic properties of importance in biomedical studies. This necessarily includes stochastic regulation of gene expression, cellular signaling and related pathways. Some ambitious efforts are underway to create *whole cell* models, most notably E-Cell (Tomita *et al*., 1999; Bioinformatics.org, 2002, http://www.bioinformatics.orgle-cell), Virtual Cell (Loew and Schaff, 2001; NRCAM, 2002, http://www.nrcam.uchc.edu/vcell_development/vcell_dev.html), MCell (MCell, 2002, http://www.mcell.cnl.salk.edu) and IECA (Holden, 2002). We note that such large simulations when done completely through Monte Carlo methods require immense computational resources. This is one motivation for the hybrid method. We also note that large discrete simulation models are not very transparent: it is difficult to understand the sources of stochastic behavior. The hybrid method allows smaller and more transparent stochastic models since non-stochastic parts of the system can be abstracted as conventional deterministic equations.

A model is an integral part of the research process, and as such it should be sharable in a standard format. The major standards relevant to cellular simulation are Systems Biology Markup Language (SBML, 2002, http://www.cds.caltech.edu/erato/sbml/docs/) and CellML (CellML.org, 2002, http://www.cellml.org/public/about/what_is_cellml.html). We have utilized SBML (Level 1) with annotations that allow for adding optional information to a standards-compliant SBML model.

Stochastic simulation algorithms

Biochemical processes within cells (other than metabolic processes) are generally characterized by interactions of molecular species at low concentrations in a small volume, and hence at very low numbers. Thus in modeling such systems, we are working within the regime of statistical

mechanics, rather than in the 'large N', limit of thermodynamics. The usual method for simulation of such systems is detailed Monte Carlo simulation of the statistical form of chemical rate equations.

We will confine our discussion here to a single compartment model, though our method is also valid for multiple compartments. Within the compartment, it is assumed that all molecular species are well mixed, so that the techniques of statistical mechanics are applicable. The species in the compartment engage in a fixed set of reactions with known stoichiometry and reaction rates. The state of the system at any moment is represented by the number of molecules of each species, a vector X of n non-negative integers. Similarly, the stoichiometry of a reaction r is represented by a vector Y_r of n integers, where $Y_{r,i}$ represents the number of molecules of species i produced by the reaction r. A negative value indicates consumption of the corresponding species. The propensity of a reaction channel $\alpha_r[X(t)]\,dt$ is the probability, given that the system is in state $X(t)$, that a reaction will occur on channel r in the time interval $[t, t + dt)$. Using statistical mechanics arguments, Gillespie derives $\alpha_r[X(t)]$ as a function of the standard chemical reaction rate constants, the populations of reactants and a combinatorial measure of the number of reacting configurations (Gillespie 77).

Gillespie proposed two algorithms for Monte Carlo generation of stochastic trajectories of such a chemical system (Gillespie 77). The *first reaction* algorithm proceeds by generating a random event time for each reaction channel from the exponential distributions of each. The alternative *direct method* generates the time of the next reaction from an exponential distribution with the sum of the channel propensities as the characteristic time. In either method, the system is updated according to the stoichiometry of the chosen reaction channel, and the simulation time is updated to the generated event time. The equivalence of these two methods has been established (Gillespie, 1977). Monte Carlo methods become expensive for systems where reactions run at very high rates. When the high rates are a result of high concentrations, it is possible to take time steps that are longer than a single reaction in a method called τ-leap (Gillespie, 2001).

Gibson (2000) has addressed the key algorithmic issues, repeated recalculation of propensities and generation of many random numbers. Algorithmic and analytical improvements combined with advances in computing platforms over the years have made feasible simulation of realistically sized systems (Endy and Brent, 2001; MCell, 2002). However even with these improvements, the cost of discrete event simulation has motivated the development of alternative or hybrid methods for approximating the fully discrete simulation. In particular, efforts have been made to include some stochastic effects within deterministic models, as described in the next section.

Stochastic modeling with differential equations

Ordinary Differential Equations (ODEs) are the most common approach to modeling systems of chemical reactions. Any systems other than the simplest require a numerical solver. These range from the trivial Euler method, to more stable mid-point methods, to the 'workhorse' Runge–Kutta family of methods (Press *et al.*, 2002). ODEs are relatively easy both conceptually and computationally. However their underlying model is completely deterministic.

The Langevin equation, originally formulated for Brownian motion (van Kampen, 1992), extends the ODE formulation by adding random noise terms, and thus allowing the representation of some stochastic behavior. The noise terms must be statistically independent. Gillespie (2000) has adapted the Langevin equation for application to chemical reaction channels. There are two assumptions required in his derivation (the assumptions are similar to the τ-leap requirements).

- The propensities cannot change too rapidly over small time intervals.
- Over those same intervals, there must be significant activity in all reaction channels.

The requirements are jointly satisfiable in systems where all species populations are large.

We note that the lambda phage does not meet the conditions of either the τ-leap or the Langevin methods. There are low copy numbers of some species (1–10), low rates of some reactions (once per min), and the reactions are strongly coupled to each other. We expect the same to be true of other models of biological and biomedical significance. This motivates our development of another method for integrating stochastic behavior into systems of differential equations.

Simulation of high-frequency reactions

In the approaches described above, reactions with high frequency cause a severe computational burden. Approximate methods can reduce this burden for two specific types of reactions: fast statistical-thermodynamic reactions that maintain a quasi-steady state (i.e. average occupancy of a promoter binding site), and reaction sequences that consist of a large fixed number of identical steps (i.e. gene transcription and translation). In this section, we review these methods as a background to the next section in which we describe our hybrid method for approximate simulation of other types of reactions.

The binding of promoters and repressors controls stochastic gene expression events. Ackers *et al.* developed a methodology for approximating such statistical-thermodynamic binding events and applied it to simulation of the lambda phage (Ackers *et al.*, 1982; Shea and Ackers, 1985). This approach has since been used to reduce the computational burden of discrete event simulation of regulated gene expression (Arkin *et al.*, 1998; Gibson, 2000).

Transcription and translation, once initiated, typically continue by repetition of molecular binding and reaction steps until a stopping point is reached. If the steps can be treated as identical and statistically independent, then an approximation can be used for the probability of completion and for the time to completion (Gibson, 2000). If the time duration of each step is exponentially distributed, then the duration of the entire process has a Gamma distribution.

Combining deterministic and stochastic solvers

The 'gold standard' in simulation of cellular processes is a full microscopic discrete event simulation. Anything else is an approximation. In the section above, we gave two examples of methods for treating special types of high frequency events within a discrete simulation without simulating each individual event. We now formulate an approach for a more general class of reactions. Our approach preserves probabilistic behavior but provides greater computational efficiency for those reaction types that occur with high rates. This approach merges the Monte Carlo algorithm of discrete simulation with the time-step integration of ordinary differential equations. This avoids the approximations necessary in adding stochastic noise terms to differential equations.

We partition the set of reaction channels in a simulation into two regimes: continuous and discrete. Reactions placed in the continuous regime are implemented in differential equations. The remainder of the model, with events of lower frequency, such as transcription, translation and molecular signaling, are retained in the stochastic regime so that important consequences of their stochastic behavior are preserved. The continuous regime must satisfy conditions for accuracy and stability of the numerical solution technique: the number of instances of each molecular species in a reaction in the continuous regime must be large relative to one, and the number of reaction events of each reaction occurring within one time step of the numerical solver must be large relative to one. If either condition is not satisfied for a reaction, then it must be handled in the discrete regime.

This partitioning leaves some molecular species (those participating in reactions assigned to both regimes) being represented in both the discrete and the continuous regimes. These 'bridging' species are represented simultaneously by two variables: one an integer and one a floating point number. Both regimes can 'read' this species (in the data type required) and both can 'write' updates to the species from reactions that produce or consume the species. In linking the two solvers for these variables, we must assure that

- information in the two solvers is properly synchronized,
- conversions between continuous and discrete variables are well behaved and
- correctness and stability of each solver are not compromised.

Synchronization must cope with the fact that the discrete solver is essentially asynchronous—the time of the next event is a random variable—while in the continuous solver time is usually synchronous—clocked by the step size of the numerical algorithm. The synchronization algorithm we present below links these two clocks.

In converting between a discrete variable (the number of molecules of a given species) and a continuous variable (the chemical concentration), we will need to assure that round trip conversions do not introduce errors that could accumulate and thus compromise stability and accuracy of the algorithms.

Information passing from the discrete regime to the continuous includes the number of molecules of each of the bridging species. This number will remain constant in the discrete regime until the next occurrence of a reaction that alters that species. Thus, the numerical algorithm in the continuous regime can treat the concentrations of bridging species from the discrete regime as being constant for at least some interval of time. However the same is not true in the opposite direction. The predictions of the continuous regime for the concentrations of bridging species depend upon the details of the numerical algorithm. Thus Runge–Kutta methods will produce polynomial predictions of concentrations over the current time step of the algorithm. These predictions will in general produce time-dependent propensities in the discrete regime Monte Carlo process. It is possible to generate Monte Carlo events according to a time-dependent propensity. We have explored several methods for generating Monte Carlo events according to the concentrations generated in Runge–Kutta methods (up to fourth order). A power series expansion over the uniform Monte Carlo variable can give acceptable approximations to the event time distributions.

Synchronization of the two regimes is accomplished by the following algorithm. While this description refers to only one discrete regime and one continuous regime, the algorithm extends directly to multiple regimes of each type. There is one global time, t. The molecular species x_i, are converted to floating point numbers in the continuous regime and to integers in the discrete regime. The time step (possibly adaptive) of the numerical algorithm in the continuous regime is δt. This version of the algorithm uses the Gillespie direct method, but other methods could be used in the discrete regime.

Set time $t = 0$ and set $x_i(0) =$ initial values.
While t is less than the simulation duration
  Calculate the propensity for each discrete reaction from
    the values $x_i(t)$.
  From the propensities of the discrete reactions, select
    a discrete time step size $\delta\tau$.
  Set t_0 to the minimum of $t + \delta\tau$ and $t + \delta t$.
  While there is no discrete event
    In the continuous regime, generate predicted
      populations $x_{c,i}(t)$ over the time step $[t, t_0]$.
    Communicate $x_{c,i}(t)$ to the discrete regime.

```
In the discrete regime, compute the (time-varying)
    propensities over [t, t0]
Generate a candidate next event with its time of
    occurrence t1.
If t1 > t0 then   // No discrete events occurred in
    this time step.
    t ← t0
    xi(t) ← xc,i(t).
else   // There was a discrete event in this time step.
    t ← t1
    xi(t) ← xc,i(t) + Yr,i
EndIf
End
End
```

A reasonable choice for the discrete time step size $\delta\tau$ is the expected time to the next event, thus taking the two branches of the if statement approximately equally often. This version of the algorithm is conservative in that it forces a complete recomputation of all parameters in the continuous regime after each discrete event. This should not always be necessary. A threshold could be used to determine when the discrete event makes a large enough change in a bridging species that a significant change in Runge–Kutta parameters is necessary. This should improve the speed of computation with little effect on accuracy. It is not necessary that Runge–Kutta be used in the continuous regime: this algorithm is compatible with other methods. The step size in the continuous regime need not be fixed: adaptive time step methods can be used so long as the current step size is used in the comparisons with the predicted time of the next discrete event.

This version of the algorithm uses time-varying propensities in the discrete regime, which is not necessary if the size of the time step in the continuous regime is kept small. This is a trade-off of increased complexity in generating Monte Carlo events against smaller step size and increased number of steps in the continuous solver. The best approach will depend upon the relative efficiencies of implementation of the two regimes. There are many more opportunities for reducing the amount of recomputation in the algorithm.

Errors and accuracy trade-offs

The sources of errors on the continuous regime consist of the usual errors from the Runge–Kutta or similar numerical solution algorithm, plus additional error terms introduced by the bridging variables being altered in integer increments by discrete events. In the conservative algorithm presented above, we force a recomputation of the Runge–Kutta parameters after every discrete event. If this were not done, we would have a small error term in the continuous regime (corresponding to changes of plus or minus one molecule) that would appear as noise in the differential equations.

We have measured the errors introduced by the continuous solver and by discrete event simulation (data not shown).

With appropriate choice of numerical algorithms, errors are much smaller than the statistical variation of concentration due to the discrete nature of the processes being simulated (i.e. small numbers of molecules and small number of events). We have also used simpler algorithms and approximations, with a small step size, and obtained good results. Our tests with lambda phage presented below used a Runge–Kutta algorithm, constant propensity and an adaptive step size.

Implementation and modeling standards

Our primary aim in implementation has been flexibility and extensibility for a program of experiments on the hybrid algorithm. A simulator was built using object oriented design practices in C++. This simulator has been used for reproducing results of several published models and to explore variations on the hybrid algorithm. In addition to the Gamma processes for transcription and translations and the statistical-thermodynamic reactions to govern the initiation of transcription, we can add other abstracted types of reactions. We can experiment with various continuous and discrete solvers and with assigning reactions to either a stochastic or continuous reaction engine.

We chose to utilize Level 1 SBML for storing models (SBML, 2002). One advantage of SBML is the allowance for *annotations*, supporting the attachment of additional information to the standard representation. We used this annotation mechanism for the abstract reaction types. The lambda phage model used annotations for both the fast statistical-thermodynamic abstract reaction of gene promoters, and for the Gamma processes of transcription and translation.

Tests with lambda phage

The lambda phage 'switch' is a thoroughly studied and well-documented mechanism, which has been eloquently elucidated by Ptashne (1992). Briefly stated, the lambda phage is a virus that infects *Escherichia coli*. Infection can result in two states. One possibility is that the lambda phage DNA is integrated into the DNA of the host *E.coli*, where it remains dormant and is passed from one generation of *E.coli* to the next. When the *E.coli* enters this state, it is said to be in a state of *lysogeny*. Alternatively, the phage DNA can be expressed, leading to the production of the protein components for more phage particles. This eventually leads to the demise of the cell and a release of many new phage progeny into the environment. This is known as the *lytic* state. The state decision is binary—the cell cannot remain between these two. This switching behavior is the result of the complex interactions between repressors and promoters present in the lambda phage DNA. The fact that there are two stable states of the lambda system can be seen in the structure of a deterministic differential equation models. However deterministic models cannot generate correctly the probability of the two states being taken, nor predict how this probability will depend upon attributes of the infected cell.

This system has been analyzed in several simulations and the results compared to experimental measurements (Ackers *et al.*, 1982; Arkin *et al.*, 1998; Gibson, 2000). Our model was built using parameters from the models by Arkin *et al.* and by Gibson. We focus on one specific stochastic phenomenon, the dependence of the probability of lysogeny upon of the multiplicity of infection (MOI). The MOI is the number of phage particles that have infected a single *E.coli* cell. The probability of lysogeny increases with MOI, reaching nearly 100% with MOI above 10. The dependence of probability on MOI cannot be predicted by a deterministic model, and would be expected to be sensitive to any failure to maintain important stochastic effects in the simulation.

Implementing the model

In order to establish the validity of our simulation engine and the model itself we first executed the entire model using only the stochastic reaction engine. The model consists of 40 species interacting through 71 reaction channels. With populations of the individual species corresponding to a moderate MOI, simulating 35 min of the cell life cycle, with 3.4×10^5 reaction events, required 4 s. Further improvements could be had by optimizing the code and by streamlining the logging. Numerous runs were made to obtain results for comparison with published results (Arkin *et al.*, 1998; Gibson, 2000). This validated our implementation of the direct method in the stochastic reaction engine.

Once the stochastic engine had been verified, we reallocated some of the reactions in the model. The criteria outlined above led us to move the following reactions to the continuous regime:

- Zeroth order reactions: Those introducing new material into the system at a constant rate.
- Dimerization reactions: Those governing the steady-state ratio of monomer to dimer for several key proteins in the system (cro, cI and N).
- Degradation reactions: Those responsible for degrading those same key proteins.

In all, only 10 of the 71 reactions were allocated to the continuous regime, but these represented the great bulk of the reaction events. In runs with MOI = 6, the number of discrete events was reduced from 337 100 to 10 836, a savings of better than 95%. We observed that variance was reduced for those species residing partially in the continuous regime (Fig. 1). This would be expected, since some of the reactions impacting those species were now computing only mean rates and mean populations. Our hypothesis was that the loss of stochastic properties of these reactions would not be important in the global stochastic behavior of the system, of which the most notable is the probability of lysogeny.

Comparison with previous work

To determine the dependence of probability of lysogeny on MOI, many runs were executed at each value of MOI. The runs were then partitioned into two sets by comparing the concentrations of the cI dimer and cro dimer proteins at the end of the 35 min simulation time. If cI dimer was greater than the cro dimer, the cell was lysogenic. Likewise, if there was more cro dimer than cI dimer, the cell was lytic. The estimated probability of lysogeny was then the fraction of runs in the lygsogenic set. Our results from these runs (Fig. 2) show that the distinctive bifurcation behavior of the system had been preserved, thus indicating that the necessary stochastic effects had been preserved in the hybrid simulation.

Conclusions and future directions

We have presented a method to integrate discrete and continuous simulation of cellular processes. This approach offers advantages in the scale-up of models to include high-frequency reactions and molecular species in high concentrations, and yet maintains the stochastic behavior that is the hallmark of many biological processes. This approach is similar in spirit and method to the means previously used for including high-frequency reactions in stochastic models, but extends to a broad class of reaction types. We have examined some of the trade-offs in using this approach with different numerical algorithms. In tests this hybrid simulation has produced good results, including reproducing results of other simulation engines.

We have not done a formal analysis of the accuracy of the hybrid simulations. This will be difficult because the effects of the stochastic regime are not easily characterized as noise terms in the continuous regime. The low population and infrequent events of the model, and the lack of independence in stochastic behavior do not satisfy the requirements for the Langevin approach. The correct question is the accuracy relative the stochastic simulation methods that have been shown to correctly implement the Chemical Master Equation. An important challenge for further work is to develop criteria for knowing which reactions are most important to global stochastic behavior of the system. This would provide a basis for deciding how to assign reactions between regimes. One would like to know the minimum set of reactions that must be handled stochastically, but we do not yet have a basis for such a judgment.

Hybrid simulation can serve as a tool for exploring the sources and nature of stochastic behavior. With this approach, it is possible to easily experiment with moving one or more reactions from the stochastic to the continuous regime, and thus determine the global system consequences of stochastic events in those reactions. It should also be feasible to dynamically and automatically move reactions based on computed rates and concentrations. As developed thus far, this approach handles only reactions within one or a few 'well-mixed' chambers. We intend development of extensions to include
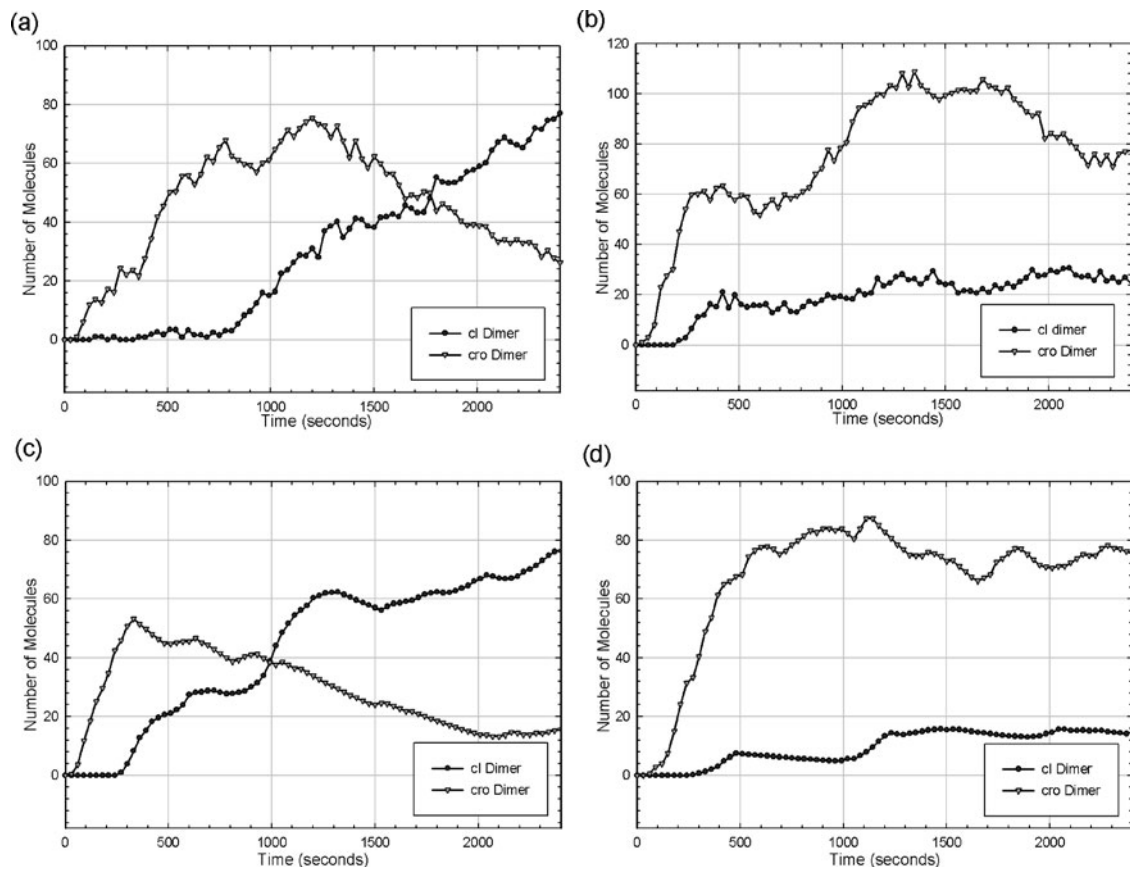
Fig. 1. Examples of individual runs of the fully stochastic and hybrid implementations of the lambda phage model. (**a**) Stochastic lysogenic run, (**b**) stochastic lytic run, (**c**) hybrid lysogenic run, and (**d**) hybrid lytic run. The hybrid runs show less short-term variance of the populations, but have the same long-term behavior in choosing one or the other fate for the cell.
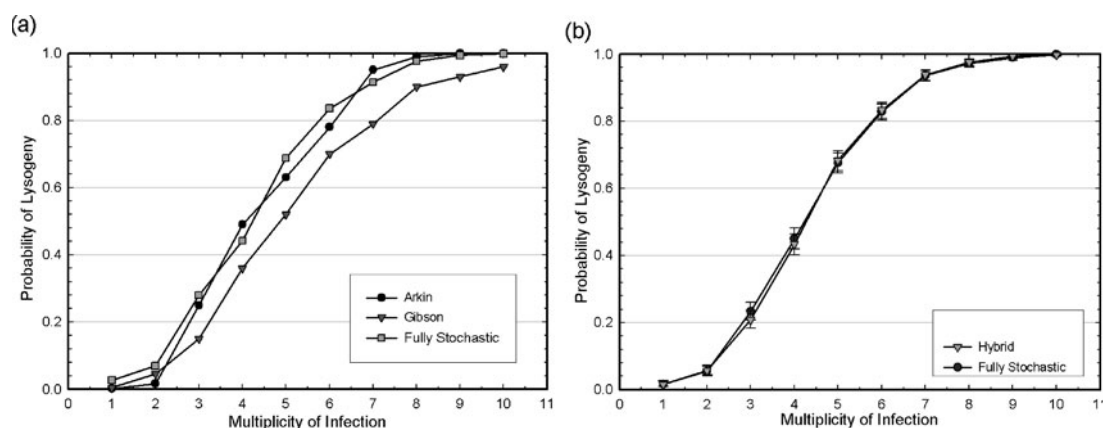

Fig. 2. Probability of lysogeny as a function of MOI. (**a**) Comparison of the published results of Arkin (closed circles) and Gibson (inverted closed triangles) with the fully stochastic implementation, average of 1000 runs at each value of MOI (closed squares). In the few parameters in which Arkin and Gibson differed in their models, we have followed Arkin. (**b**) Comparison of the same 1000 runs of the fully stochastic implementation (closed circles) at each MOI with 1000 runs of the hybrid implementation (inverted closed triangles) at each MOI. The error bars show the 95% confidence interval from a binomial distribution of 1000 samples.

diffusion processes and geometry through integration of partial differential equations and discrete simulation of diffusion. The approach as developed thus far handles some issues of widely disparate time scales, but much yet remains to be done. Our interest is to eventually be able to combine models of processes across widely varying scales of time and space.

ACKNOWLEDGEMENTS

REFERENCES

Ackers,G.K., Johnson,A.D. and Shea,M.A. (1982) Quantitative model for gene regulation by a lambda phage repressor. *Proc. Natl Acad. Sci.*, **79**, 1129–1133.

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

Asthagiri,A. and Lauffenburger,D.A. (2000) Bioengineering models of cell signaling. *Ann. Rev. Biomed. Eng.*, 31–53.

Bader,G.D., Betel,D. and Hogue,C.W. (2003) BIND: the biomolecular interaction network database. *Nucleic Acids Res.*, **31**, 248–250.

Biocarta (2002).

Bioinformatics.org (2002).

CellML.org (2002).

Demir,E., Babur,O., Dogrusoz,U., Gursoy,A., Nisanci,G., Cetin-Atalay,R. and Ozturk,M. (2002) PATIKA: an integrated visual environment for collaborative construction and analysis of cellular pathways. *Bioinformatics*, **18**, 996–1003.

Endy,D. and Brent,R. (2001) Modelling cellular behavior. *Nature*, **409**, 391–395.

Endy,D., Kong,D. and Yin,J. (1997) Intracellular kinetics of a growing virus: a genetically structured simulation for bacteriophage T7. *Biotechnol. Bioeng.* **55**, 375–389.

Gibson,M.A. (2000) Computational methods for stochastic biological systems. Ph.D. Thesis, California Institute of Technology, Pasadena, CA.

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

Gillespie,D.T. (2000) The chemical Langevin equation. *J Chem. Phys.*, **113**, 297–306.

Gillespie,D.T. (2001) Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.*, **115**, 1716–1732.

Holden,C. (2002) Alliance launched to model *E.coli. Science*, **297**, 1459–1460.

KEGG (2003) Kyoto Encyclopedia of Genes and Genomes.

Kitano,H. (2002) Computational systems biology. *Nature*, **420**, 206–210.

Li,J., Ning,Y., Hedley,W., Saunders,B., Chen,Y., Tindill,N., Hannay,T. and Subramaniam,S. (2002) The Molecule Pages database. *Nature*, **420**, 716–717.

Loew,L.M. and Schaff,J.C. (2001) The virtual cell: a software environment for computational cell biology. *Trends Biotechnol.*, **19**, 401–406.

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

MCell (2002).

NRCAM (2002) National Resource for Cell Analysis and Modeling.

Press,W.H., *et al.* (2002) *Numerical Recipes in C++ The Art of Scientific Computing*, 2nd edn. Cambridge University Press, Cambridge, UK.

Ptashne,M. (1992) *A Genetic Switch: Phage λ and Higher Organisms,* 2nd edn. Blackwell Science, Malden MA and Cell Press, Cambridge, MA.

SBML (2002) California Institute of Technology Systems Biology Workbench Development Group.

Shea,M.A. and Ackers,G.K. (1985) The O_R control system of bacteriophage lambda. *J. Mol. Biol.*, **181**, 211–230.

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

van Kampen,N.G. (1992) *Stochastic Processes in Physics and Chemistry.* Elsevier Science B.V., Amsterdam.

You,L., Suthers,P.F. and Yin,J. (2002) Effects of *Escherichia coli* physiology on growth of phage T7 *in vivo* and *in silico. J. Bacteriol.*, **184**, 1888–1894.