BIOINFORMATICS APPLICATIONS NOTE Vol. 25 no. 17 2009, pages 2289–2291
doi:10.1093/bioinformatics/btp387

Systems biology

Efficient stochastic simulation of reaction–diffusion processes via direct compilation

Mieszko Lis[1,*], Maxim N. Artyomov[2], Srinivas Devadas[1] and Arup K. Chakraborty[2,3,4]
[1]Computer Science and Artificial Intelligence Laboratory and the Departments of [2]Chemistry,
[3]Chemical Engineering and [4]Biological Engineering, Massachusetts Institute of Technology,
Cambridge, MA 02139, USA

Received on February 2, 2009; revised on June 13, 2009; accepted on June 19, 2009

Advance Access publication July 3, 2009

Associate Editor: Olga Troyanskaya

ABSTRACT

We present the Stochastic Simulator Compiler (SSC), a tool for exact stochastic simulations of well-mixed and spatially heterogeneous systems. SSC is the first tool to allow a readable high-level description with spatially heterogeneous simulation algorithms and complex geometries; this permits large systems to be expressed concisely. Meanwhile, direct native-code compilation allows SSC to generate very fast simulations.

Availability: SSC currently runs on Linux and Mac OS X, and is freely available at http://web.mit.edu/irc/ssc/.

Contact: mieszko@csail.mit.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 BACKGROUND

Cells interact with their environment via receptors that bind to extracellular molecules; these events are then translated into functions by biochemical signaling networks. Non-linearities arising from the complex topology of such networks often make it difficult to intuit qualitative behavior of signaling modules. Moreover, recent imaging experiments have revealed that signaling components are organized into spatial patterns that modulate signaling (Grakoui *et al.*, 1999; Lee *et al.*, 2003). Finally, extrinsic and intrinsic stochastic effects, which make each cell's response unique, can be important when small numbers of signaling molecules are involved (Artyomov *et al.*, 2007). As computational studies are increasingly becoming necessary complements to genetic, biochemical and imaging experiments in unraveling this non-intuitive behavior of cell signaling networks, efficient and easy to use tools that can carry out stochastic simulations of biochemical networks, both in well-mixed and spatially inhomogeneous approximations, have become key technologies.

Since the original stochastic simulation algorithm (Gillespie, 1977), basic computer science techniques have reduced the rate at which the per-step computation time grows with the number of possible reactions to logarithmic growth (Gibson and Bruck, 2000; Li and Petzold, 2006; Wylie *et al.*, 2006), or optimized performance by noting that a few reactions account for most events (Cao *et al.*, 2004; McCollum *et al.*, 2006); more recently, Slepoy *et al.* (2008) have reduced per-step computation to expected constant time via an elegant composition–rejection algorithm. Similar techniques have been applied to reduce spatially heterogeneous simulation time to logarithmic (Elf and Ehrenberg, 2004). The combinatorial growth of the instantiated reaction network size, another limiting factor for complex systems, has been addressed either by generating species and reactions on the fly (Faeder *et al.*, 2005; Lok and Brent, 2005) during a Gillespie-based simulation, by representing each molecule separately (Morton-Firth and Bray, 1998), or ingeniously do away with explicit counts altogether by adjusting the sampling distribution (Danos *et al.*, 2007; Yang *et al.*, 2008).

Efficient formulation of such simulations in a general programming language like C or FORTRAN, however, is not a trivial task: while simulating a few reactions is fast even with a simple implementation, a system with thousands of reactions and subvolumes demands more complex algorithms which are much more tricky to code. The programming burden has been reduced by libraries (e.g. Li *et al.*, 2008) as well as by simulators for well-mixed (e.g. Gillespie *et al.*, 2006; Mauch, 2009) and spatially inhomogeneous (e.g. Hattne *et al.*, 2005; Meier-Schellersheim *et al.*, 2006) models. File formats like SBML (Hucka *et al.*, 2008), developed to express biochemical models, can be read by several simulators.

The modeling task is further complicated by the explosion in combinatorial complexity which arises when modeling post-translational modification or reactions local to one molecule in a complex (Hlavacek *et al.*, 2006): in SBML (and, indeed, in most simulators) all possible species and each combination of every possible reacting complex must be written out as a separate reaction, which renders expressing even modestly complex reaction networks impractical. To mitigate these limitations, BioNetGen (Faeder *et al.*, 2009) and κ (Danos and Laneve, 2004) have proposed higher level specifications where the reactants in each reaction are written as patterns covering many possible species; such descriptions not only naturally correspond to the intuitive concept of a biochemical reaction, but are significantly smaller and therefore more readable as well as much less error-prone.

The main contribution of the Stochastic Simulation Compiler (SSC) that we present here lies in combining a higher level specification required for modeling larger systems with the ability to model spatially heterogeneous systems. It differs from BioNetGen and κ because their syntax and expansion algorithms offer no

[*]To whom correspondence should be addressed.

support for spatially inhomogeneous containers, while SSC supports multiple regions with arbitrarily complex shapes specified using Constructive Solid Geometry (CSG); meanwhile, while MesoRD allows such regions and geometries, it suffers from the combinatorial complexity limitations described above. In addition, SSC produces fast simulations (cf. Supplementary Material) by directly generating machine code tailored to a specific architecture.

2 IMPLEMENTATION

2.1 Tool flow

The tool flow resembles a programming language compiler. The user writes a high-level description of the reaction system (see Supplementary Material for examples), using patterns to select and change specific parts of compounds (similar to how a cell biologist would describe a known or hypothesized cell signaling network). Regions are specified using CSG, a technique that employs simple operations (e.g. union, intersection, difference and scale) on basic shapes (such as spheres, cubes, cylinders, etc.) to describe arbitrarily complex geometries and widely used in solid modeling and computer graphics (see, e.g. Requicha and Voelcker, 1977). Any reaction can be restricted to a subset of regions, and diffusions within and among the regions are written using the same high-level pattern syntax as reactions. The compiler then expands the model starting with initial species and reaction patterns, creating the necessary instances with specific properties and connections as well as specific reactions operating on each of those compounds in each region; meanwhile, the regions are discretized into cubic subvolumes. This intermediate representation is used to produce a simulator executable, which, in turn, simulates the model signaling pathway.

2.2 Reaction expansion

Most biologically relevant signaling reactions are conceptually *local*, that is, they ‘see’ only a part of a larger molecule or complex (say, a single phosphorylation site). Therefore, we write reactions and diffusions locally, using pattern matching to recognize and modify parts of complexes, and rely on the compiler to derive all the possible cases in all regions. Similarly, only initially present compounds are specified; the compiler generates the rest from the initial set and the reactions.

Formally, the reactions and diffusion form a graph term-rewriting system, which is fully evaluated to generate the simulator. Briefly, each expansion step considers a rule in the system, finding all combinations of substrates in the relevant region that match the rule. The rule is then applied to each match, possibly resulting in new compounds, and a compound-specific reaction is created for the specific substrate combination. Any new compounds not excluded by predefined limits (used to prevent infinite expansion) are added to the region where the reaction took place and any regions reachable by following the given diffusion patterns; the cycle then repeats until no more new compounds have been created. (See Supplementary Material for details of the expansion process).

2.3 Direct code generation

We obtain the efficiency of hand-optimized code by directly generating assembly code from the fully expanded set of reachable species and reactions. This allows us to avoid the interpretive overhead of consulting dependency graphs to determine which copy counts and propensities must be recomputed.

The generated code is also tailored for model complexity and processor architecture. For most sizes, the compiler creates a separate, straight-line segment of code for each possible reaction in a region; each segment is parameterized only on the subvolume (or, in the case of diffusion, two subvolumes), and directly updates and propagates the affected propensities (see Section 2.4). This avoids pipeline stalls and cache flushes caused by mispredicted branches, and reduces the number of data memory reads and writes (which are the performance bottleneck) to the absolute minimum. (See Supplementary Material for a detailed description of the code generation method).

2.4 Reaction–diffusion simulation algorithm

The simulation algorithm is similar to the logarithmic-time versions of the direct stochastic simulation algorithm (Li and Petzold, 2006; Wylie *et al.*, 2006). The simulation-time representation details may be found in the Supplementary Material; briefly, the reactions in each subvolume (or on each boundary between subvolumes) are arranged in an n-ary heap with the leaves corresponding to individual reaction propensities and each node carrying the combined propensity of the reactions underneath—the topmost node for each subvolume is, then, the propensity of any reaction taking place within. The subvolume and boundary reaction propensities are, in turn, themselves arranged in a heap where each leaf is either a subvolume or a boundary propensity; the topmost node is the propensity of any reaction in the system taking place (and, hence, the range from which the random number should be selected).

Simulation proceeds as follows: a random number r is selected from range $[0,R)$ where R is the propensity of any reaction taking place; then the subvolume and reaction corresponding to r is selected by n-ary search in the heap. Next, the reaction is ‘executed’, that is, the copy numbers of the affected species are adjusted as the reaction dictates. Finally, the propensity of each reaction whose substrate copy counts were altered is recomputed, and the partial propensities are propagated up the propensity heap until the new R is recomputed and the cycle can be repeated.

Since the propensity heap in each subvolume (or boundary) has height logarithmic in the number of reactions within, and the heap above is logarithmic in the number of subvolumes and boundaries, the total tree depth scales roughly logarithmically in the number of reactions in the system. Both the reaction selection/search and copy number/propensity update step, therefore, run in time logarithmic in the number of reactions.

3 PERFORMANCE

We compared spatially homogeneous SSC against BioNetGen 2.0.46 (Faeder *et al.*, 2009) (since, like SSC, it builds reaction networks from pattern-matching rules), and against simulators built with the StochKit library (Li *et al.*, 2008); because of the complexity of the larger models, we had SSC automatically generate the required StochKit C++ configurations. To test real-world performance, we selected two toy systems and two more realistic systems with various reaction counts: a dimer decay model (Gillespie, 2001) with four reactions, a simplified EGFR signaling model (Blinov *et al.*, 2006) with 64 reactions, a model for the earliest events in

T-cell signaling (Wylie *et al.*, 2006) with 1120 reactions, and an enhanced version of the same with 2422 reactions. To test spatially heterogeneous models, we compared with the latest development revision of MesoRD (Hattne *et al.*, 2005), SVN r559; we used the T-cell signaling model above where single molecules (but not compounds) were permitted to diffuse around a membrane interface, which was divided into 100, 10 000, and 50 000 subvolumes. All simulations produced the same results (modulo random seed variation and precision loss during floating point arithmetic). To focus on measuring only the simulation time, we disabled all output except the final species counts, and repeated each experiment 5-fold to account for initial random seed variation and possible effects of other processes executing on the system.

We found that SSC consistently outperformed the faster of the two spatially homogeneous simulators we tested by 2 × to 6 ×, with the advantage growing with the size of the model (see Supplementary Fig. 3). For spatially heterogeneous simulation, we found that SSC was ∼50 × faster than MesoRD, although both scaled very well with the number of subvolumes (see Supplementary Fig. 4).

4 CONCLUSIONS

We have described the SSC, a new tool for exact stochastic simulations of biochemical reaction networks. SSC is, to our knowledge, the first tool to combine a succinct high-level description (which avoids combinatorial complexity explosion) with spatially resolved simulation where species and reactions may be restricted to specific regions of arbitrarily complex shapes, and unique in employing direct native machine code generation to produce fast simulators.

ACKNOWLEDGEMENTS

This research was funded by NIH Grant #1PO1/AI071195/01.

Conflict of Interest: none declared.

REFERENCES

Artyomov,M.N. *et al.* (2007) Purely stochastic binary decisions in cell signaling models without underlying deterministic bistabilities. *Proc. Natl Acad. Sci. USA*, **104**, 18958.

Blinov,M.L. *et al.* (2006) A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity. *Biosystems*, **83**, 136.

Cao,Y. *et al.* (2004) Efficient formulation of the stochastic simulation algorithm for chemically reacting systems. *J. Chem. Phys.*, **121**, 4059.

Danos,V. and Laneve,C. (2004) Formal molecular biology. *Theor. Comput. Sci.*, **325**, 69.

Danos,V. *et al.* (2007) Scalable simulation of cellular signaling networks. In *Proceedings of APLAS, Singapore, Nov–Dec, 2007*, Springer.

Elf,J. and Ehrenberg,M. (2004) Spontaneous separation of bi-stable biochemical systems into spatial domains of opposite phases. *Syst. Biol.*, **1**, 230.

Faeder,J.R. *et al.* (2005) Rule-based modeling of biochemical networks. *Complexity*, **10**, 22–41.

Faeder,J.R. *et al.* (2009) Rule-based modeling of biochemical systems with BioNetGen. In Maly,I.V. (ed), *Methods in Molecular Biology: Systems Biology*, vol. 500. Humana Press, Clifton, NJ.

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.

Gillespie,C.S. *et al.* (2006) Tools for the SBML community. *Bioinformatics*, **22**, 628.

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

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

Grakoui,A. *et al.* (1999) The Immunological synapse: a molecular machine controlling T cell activation. *Science*, **285**, 221.

Hattne,J. *et al.* (2005) Stochastic reaction-diffusion simulation with MesoRD. *Bioinformatics*, **21**, 2923.

Hlavacek,W.S. *et al.* (2006) Rules for modeling signal-transduction systems. *Science STKE*, **344**, re6.

Hucka,M. *et al.* (2008) Systems biology markup language (SBML) level 2: structures and facilities for model definitions. *Nat. Prec.* [Epub ahead of print, doi: 10.1038/npre.2008.2715.1, November 5, 2007].

Lee,K.-H. *et al.* (2003) The immunological synapse balances T cell receptor signaling and degradation. *Science*, **302**, 1218.

Li,H. and Petzold,L.R. (2006) Logarithmic direct method for discrete stochastic simulation of chemically reacting systems. *Technical report*, UCSB Computer Science and Engineering Group, Santa Barbara, CA.

Li,H. *et al.* (2008) Algorithms and software for stochastic simulation of biochemical reacting sytems. *Biotechnol. Prog.*, **24**, 56–61.

Lok,L. and Brent,R. (2005) Automatic generation of cellular reaction networks with Moleculizer 1.0. *Nat. Biotechnol.*, **23**, 131–136.

Mauch,S. (2009) Cain: stochastic simulations for chemical kinetics. Available at http://cain.sourceforge.net/ (last accessed date May 10, 2009).

McCollum,J.M. *et al.* (2006) The sorting direct method for stochastic simulation of biochemical systems with varying reaction execution behavior. *Comput. Biol. Chem.*, **30**, 39.

Meier-Schellersheim,M. *et al.* (2006) Key role of local regulation in chemosensing revealed by a new molecular interaction-based modeling method. *PLOS Comput. Biol.*, **2**, e82.

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

Requicha,A. and Voelcker,H. (1977) Constructive solid geometry. *Technical Memorandum no. 25*, Production Automation Project, University of Rochester, Rochester, NY.

Slepoy,A. *et al.* (2008). A constant-time kinetic Monte Carlo algorithm for simulation of large biochemical reaction networks. *J. Chem. Phys.*, **128**, 205101.

Wylie,D.C. *et al.* (2006) A hybrid deterministic-stochastic algorithm for modeling cell signaling dynamics in spatially inhomogeneous environments and under the influence of external fields. *J. Phys. Chem. B*, **110**, 12749.

Yang,J. *et al.* (2008) Kinetic Monte Carlo method for rule-based modeling of biochemical networks. *Phys. Rev. E*, **78**, 031910.