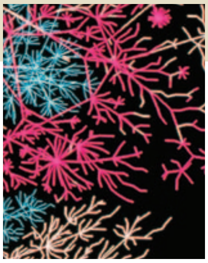

Automatic generation of cellular reaction networks with Moleculizer 1.0

Larry Lok & Roger Brent

Accurate simulation of intracellular biochemical networks is essential to furthering our understanding of biological system behavior. The number of protein complexes and of chemical interactions among them has traditionally posed significant problems for simulation algorithms. Here we describe an approach to the exact stochastic simulation of biochemical networks that emphasizes the contribution of protein complexes to these systems. This simulation approach starts from a description of monomeric proteins and specifications for binding, unbinding and other reactions. This manageable specification is reasonably intuitive for biologists. Rather than requiring the inclusion of all possible complexes and reactions from the outset, our approach incorporates new complexes and reactions only when needed as the simulation proceeds. As a result, the simulation generates much smaller reaction networks, which can be exported to other simulators for further analysis. We apply this approach to the automatic generation of reaction systems for the study of signal transduction networks.

A dynamical theory models the behavior of a system over a period of time. A useful dynamical theory is usually predictive in the sense of telling how the system will behave in the future given its present state. A simulation is an application, usually computerized, of a dynamical theory to some particular system to display its behavior over a span of time. A simulator is a computer program that performs simulations.

Biologists can simulate the behavior of a cellular pathway if they can describe it in terms compatible with the simulation's underlying dynamical theory. Such a description amounts to a hypothesis. Running the simulation amounts to drawing a nonobvious conclusion, the predicted behavior, from the hypothesis and the dynamical theory. If the biologist can observe the results, then the simulation provides a nontrivial link between the hypothesis and a potential experiment. Numerous hypotheses can be tried out computationally before subjecting a few of them to experimental verification. Thus, a simulator can provide a platform for 'sanity checking' hypotheses and comparing their outcomes before experiments are done.

Two forms of simulation that molecular biologists use are molecular dynamics and simulation of chemical reaction systems. Molecular dynamics[1] predicts the behavior of one or a few (usually large, complex) molecules using the physical theory of atomic interaction and bonding[2]. Use of molecular dynamic simulation has become central to computational structural biology, and we do not review it here (the reader is referred to ref. 3).

Chemical reaction system simulations are aimed at biological function. They display the behavior of systems of molecular species and reactions. Since the 1990s, their dynamical theory has been derived from statistical mechanics[4]. Simulators based on this theory can describe reaction systems at different levels of detail. Here, we describe previous and current efforts to simulate intracellular reaction systems according to the amount and kind of physical detail they represent.

Simulating biochemical reaction networks

Two properties are useful for classifying chemical reaction system simulators:

1. If the simulator represents species amounts as integer populations changing by probabilistic rules in response to random molecular interactions, then we will call it stochastic. Alternatively, if it approximates species populations as real number concentrations changing by a deterministic formula, then we will call it deterministic.
2. If the simulator represents the spatial location of species, then we will call it spatial. Alternatively, if it treats the reaction volume as homogeneous, then we will call it nonspatial.

We now describe the alternatives for each property more fully and then give examples of simulators satisfying each combination of alternatives.

For cellular systems, simulators frequently assume that diffusion is sufficiently fast and the reaction volume is sufficiently small that the system is well mixed. In fact, for many protein species, diffusion rates are fairly high[5]. In the well-mixed case, the dynamical theory derived from statistical mechanics is summed up in the chemical master equation (or CME)[4,6,7]. The CME describes the random reaction events between individual molecules, predicting event times, but not locations. Stochastic simulators that assume mixture by diffusion use algorithms derived directly from the CME. Deterministic simulators may also treat the reaction volume as well mixed. If so, their dynamical theory is summed up in the mass-action equation[8], an ordinary differential equation (ODE). Once an empirical formula, it is now grounded in statistical mechanics as the (large population, large volume) 'thermodynamic limit' of the CME[4].

Assuming uniform mixing results in drastic computational simplification because of the CME. Without it, the simulator must be spatial. The state of a spatial stochastic simulator consists of the locations of all the molecules, instead of a simple list of the species and their populations. A spatial deterministic simulator generally must solve partial differential equations (PDEs) instead of the mass-action ODE.

The Molecular Sciences Institute, 2168 Shattuck Avenue, Berkeley, California, 94704, USA. Correspondence should be addressed to L.L. (lok@molsci.org).

Published online 6 January 2005; doi:10.1038/nbt1054

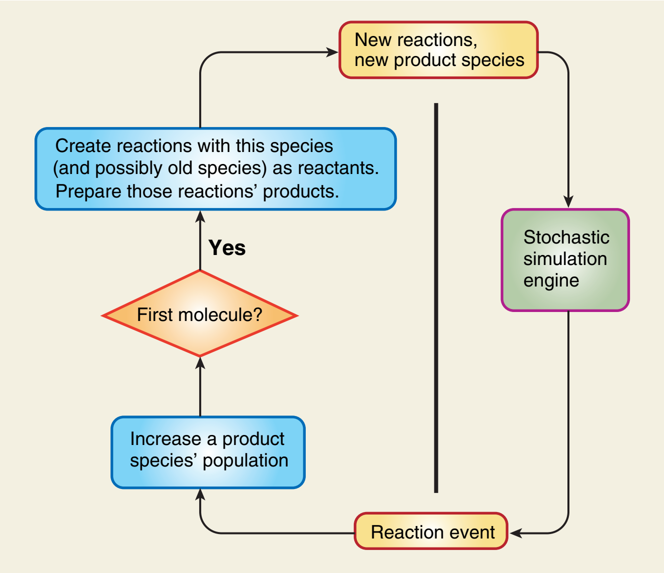

Figure 1 Reaction network generation cycle. Moleculizer creates reactions involving a new species when the first molecule of the new species appears. If the new reactions have new product species, it enters them into a growing database of species known to the simulation. Later, when the first molecule of one of these new product species appears because a reaction event occurs, Moleculizer triggers the reaction generation cycle again. The bold vertical line between reaction generation and the stochastic simulation engine is intended to indicate that Moleculizer's species and reaction generation machinery and the basic stochastic simulation machinery are not deeply intertwined, so that Moleculizer's species and reaction generation machinery could be coupled easily to other kinds of stochastic simulation algorithms.

Several investigators have asked whether stochastic effects in cellular reaction systems are biologically significant. Elowitz *et al.*[9] classified the causes of stochastic cell-to-cell phenotypic variation between genetically identical cells. They demonstrated experimentally that random variation in gene expression, or 'intrinsic noise,' contributes to cell-to-cell phenotypic variation. McAdams and Arkin[10] gave a simulation-based explanation for phenotypic consequences of stochastic gene expression. Rao, Wolf and Arkin[11] surveyed the significance of intracellular noise and the use of stochastic simulation to investigate it. Recently, G. Colman-Lerner, R. Brent *et al.* (R.B., personal communication) have done a detailed analysis of random variation in a signaling pathway, the transcription of associated genes and translation of the generated RNA into protein. They determined the relative contributions to cell-to-cell phenotypic variation of randomness in these layers of cellular response.

Deterministic simulators

We provide below examples of simulators having all four combinations of the spatial/non-spatial and stochastic/deterministic dichotomies. Relevant biological results are indicated for some of the examples.

Non-spatial deterministic simulators are typically ODE solvers applied to the mass-action equation. One widely-used simulator of this type is Gepasi[12]. It comes with a set of accessory programs (http://www.gepasi.org/), including tools to fit reaction rates to experimental time courses and tools to optimize metabolic throughput. Cross *et al.*[13] used ODE-based simulation to analyze the budding yeast cell cycle by comparing measured protein abundances with predictions of a preexisting model reported by Chen *et al*[14].

Spatial, deterministic simulators approximate the amount of each species as a concentration but permit different concentrations at different points in space. They typically solve a partial differential equation, the reaction-diffusion equation[15]. The Virtual Cell simulator (http://www.nrcam.uchc.edu) uses the reaction-diffusion equation as its underlying dynamical theory.

Stochastic simulators

A great deal of current work involves non-spatial stochastic simulators. These stochastic simulators were developed first, mainly because of Gillespie's pioneering work on the so-called first reaction algorithm[6,7,16]. The first reaction algorithm makes a schedule of reactions' tentative times of next occurrence and updates this schedule during the simulation as species populations change. The tentative time of next occurrence of the first reaction in the schedule is its actual time of next occurrence, and the simulation cycle proceeds by extracting this first reaction event from the schedule, performing the reaction and updating the schedule of tentative reaction times appropriately. Gibson and Bruck[17,18] accelerated the first reaction algorithm by avoiding unnecessary generation of random numbers. Other non-spatial stochastic simulators include StochSim[19] and Stochastirator (D. Endy & E. Lyons, Stochastirator. http://opnsrcbio.molsci.org/stochastirator/stoch-main.

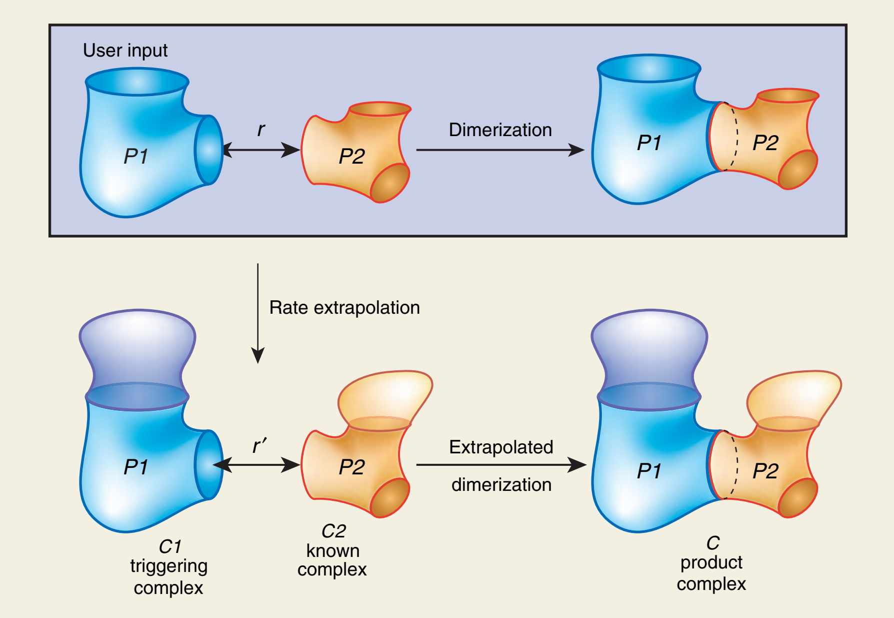

Figure 2 Dimerization example. When the first molecule of a new complex species C1 appears, Moleculizer creates dimerization reactions for free binding sites exposed by C1 and free binding sites on already known complexes such as C2 that display a compatible binding site. It extrapolates the rate of the new dimerization reaction from the rate of a user-provided prototype P1-P2 dimerization by correcting for the molecular weights of the new reactants C1 and C2. It enters the product complex C into its database of complex species when it constructs the new dimerization reaction.

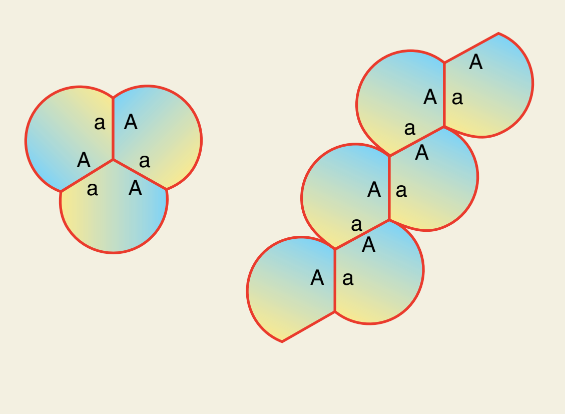

Figure 3 'Runaway' polymerization. Moleculizer's method for generating reactions can produce species and reactions that are unlikely to occur in the living cell. The figure shows a simple protein with two binding sites forming 120° angles, so as to naturally form the illustrated homo-trimeric complex. The user has told the program only that there are two binding sites on the protein and that they can bind one another, but Moleculizer knows no geometry. Thus, it 'unrolls' the three-member loop in the correct homo-trimer to generate polymeric chains, as illustrated. This phenomenon, also noted by Bray and Lay[29], points out the limitations of Moleculizer's rate extrapolation formula based entirely on molecular weight. The user can sometimes avoid 'runaway polymerization' by carefully modifying the input specification to avoid the possibility of a complex containing a loop, a closed chain of simple proteins, each bound to the next, that starts and ends with the same simple protein. By avoiding loops, the user avoids the unrolling of loops to form 'runaway' polymers. This is an unsatisfying solution, because cyclic complexes are not unknown in living cells. Handling this difficulty and others connected with the geometry of complexes is an important next step in Moleculizer's development and the eventual development of realistic reaction rate extrapolation.

html). McAdams and Arkin[10] applied non-spatial stochastic simulation to study gene expression. They showed that random variation in expression of competitive effector proteins can lead to phenotypic variation among genetically identical cells.

We further divide non-spatial stochastic simulators into two subclasses, exact and approximate. To maintain exact counts of molecular species at all times, exact stochastic simulators must simulate every single reaction event. This requirement represents a substantial computational burden. To diminish it, Gillespie proposed[16] and has subsequently refined[20] an accelerated, but approximate, method of stochastic simulation. This technique, called tau-leaping, moves time forward in steps, or 'leaps.' The integer changes in species populations from leap to leap are summaries of the effects of many intervening reaction events of different kinds. The numbers and kinds of reaction events that occur during a leap are approximated by applying the methods of the ODE solution.

Another kind of approximate, non-spatial, stochastic simulator uses a 'hybrid' approach to reduce the computational burden of stochastic simulation. This approach divides the reaction system into 'slow' and 'fast' sections. It then uses stochastic simulation to predict reaction events in the 'slow' section and ODEs to model the 'fast' section. Versions of this technique are offered by Haseltine and Rawlings[21], Rao and Arkin[22] and Cao, Gillespie, and Petzold (D. Gillespie, personal communication).

Yet another approximate way of accelerating non-spatial stochastic simulation is to use special-purpose hardware. In two recent papers[23,24], a technique has been forwarded that uses parallel computing devices called field programmable gate arrays (FPGAs). Rather than attempting to summarize reaction events, these simulators use their special hardware to perform individual reaction events more rapidly, within one clock cycle of the FPGA device. Approximation enters in at least two ways: the times of reaction events are approximate because they occur essentially on FPGA clock cycle boundaries, and some approximation is introduced to avoid more than one reaction event in any FPGA clock cycle.

The most detailed simulators are spatial and stochastic. These track molecules' locations and the places and times at which reaction events occur. Reaction events only occur when molecules that can react collide. Fricke and Wendt[25] described an early approach to spatial stochastic simulation. A more recent stochastic simulator, MCell[26], has been used, among other applications, to generate highly detailed simulations of neuromuscular junctions. MCell uses ray-tracing algorithms to determine when ligand molecules collide productively to receptors displayed on a cell membrane. Another, ChemCell (S. Plimpton, http://www.cs.sandia.gov/~sjplimp), is being used to simulate carbon fixation in photosynthesis. It uses the spatial aspect of the simulation to distribute its execution on multiprocessor supercomputer hardware.

Coping with large cellular reaction networks

Perhaps the largest challenge in developing simulations of cellular events is the proliferation of species and reactions. Cells contain thousands of different proteins[27] that frequently interact with one another (e.g., see Database of Interacting Proteins, http://dip.doe-mbi.ucla.edu;

Table 1 Moleculizer modules

	Module	Purpose
General purpose	Dimer	Generators for dimerization and decomposition reactions of complexes
	Kinase phosphatase	Supports 'generic' kinases and phosphatases, as well as simple nucleotide binding and unbinding
Alpha project	Nucleotide exchange (receptor)	Supports the special nucleotide binding and auto-hydrolysis properties of G-proteins
	Scaffold kinase cascade	Supports the localization of the action of scaffolding kinases to the scaffolding complex

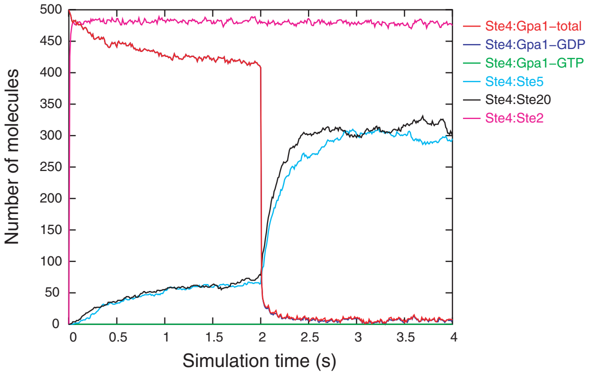

Figure 4 Illustrative simulation output. This Moleculizer output, plotted using gnuplot, shows how multiple species can be combined in a single trace. The alpha pathway simulation that produced this output generated more than 16,000 complex species in all. The system equilibrates until time 2.0, when α factor is added. The light blue trace 'Ste4:Ste5' gives the total population of the 1,300 complex species containing the scaffold protein Ste5 bound to the beta subunit Ste4 of the G-protein complex. This trace rises rapidly when α factor is introduced, causing dissociation of Gpa1 from Ste4 and thereby permitting binding of Ste4 to the scaffold protein Ste5.

Table 2 Moleculizer species and reactions

	Type	How specified	Purpose
Species	Stoch-species (see **Supplementary Notes** online)	Name and weight	Small molecules; importing, exporting reaction networks
	Complexes	Combinatorial structure and states of constituents	Represent protein complexes (and simple proteins, too)
Reactions	Explicit (see **Supplementary Notes** online)	Individually, stoichiometrically: $2H_2 + O_2 \xrightarrow{r} 2H_2O$	'Unique' reactions; importing, exporting reaction networks
	Generated	In bulk via reaction generators	Reduce user's burden of specification due to 'combinatorial explosion'

R.B., unpublished data, http://www.molsci.org/samizdat1.html). On the genomic DNA, there are thousands of sites for regulatory protein binding[28]. Many simple constituents combining in many different ways lead to a proliferation of complex species (see **Supplementary Fig. 1** online). This proliferation was noted by Morton-Firth[19], who termed it a 'combinatorial explosion.'

Large combinatorial networks of related protein complexes pose two problems. The first problem is that they are difficult for the researcher to understand. Their reactions are not only numerous, but also redundant: what a biologist might regard as a single reaction, say the dimerization of two proteins, may be replicated as dimerizations between multitudinous protein complexes containing the two monomeric proteins and exposing the appropriate binding sites. Most reaction system simulators require the researcher to enumerate this repetitive system of reactions completely. Later, when the simulation is done, the same proliferation of species and reactions makes it difficult for the researcher to extract meaning from its output. The second problem is practical, rather than cognitive: the full network of possible species of complexes may exhaust computer memory.

Here, we describe a non-spatial stochastic simulator, Moleculizer, that uses a queued form of Gillespie's 'first reaction' algorithm[6]. Moleculizer provides a tool to help handle the 'combinatorial explosion' problem described above. It reduces the simulation specification to what biologists normally regard as distinct reactions by automatically extrapolating the large network of elementary reactions required for the simulation. Later, when the simulation is finished, Moleculizer allows the researcher to bundle output about elementary reactions and species into the same 'biological' level of abstraction as the input. For example, a biologist can easily arrange that a single trace on an output plot give the total population of all those species of complexes that contain a particular protein. For the researcher, Moleculizer's parallel simplifications in simulation setup and output provide protection from the full, unintelligible blast of the explosion of species and reactions. For the machine, Moleculizer reduces the memory problem by using only species demanded by the simulation, a small fraction of the possible species of complexes that could be formed. Although the 'memory footprint' of a Moleculizer job is still sometimes large, the memory required for unrequired species and reactions would be much larger.

Box 1 Moleculizer in the Alpha Project

Moleculizer was developed in the context of the Alpha Project (http://www.molsci.org/), a comprehensive experimental and computational program to understand the quantitative behavior of the yeast mating pheromone signal transduction pathway, or briefly, the alpha pathway. This pathway was reviewed by Dohlman and Thorner[31] and is summarized in the **Supplementary Notes** and **Supplementary Figure 3** online. We take a modular view of the pathway. Each alpha pathway module has a parallel Moleculizer module providing dedicated reaction generators. We expect these reaction generators to be widely adaptable because the pathway is prototypical of many in higher eukaryotes. Two of Moleculizer's pathway-specific modules are described below.

The receptor-G-protein complex. One module provides special reactions for the receptor–G-protein complex. When pheromone binds the receptor, Ste2, it causes a conformational change in the receptor. This conformational change causes the G-protein subunit Gpa1 to increase its affinity for GTP and to decrease its affinity for GDP, leading it to bind GTP. GTP-bound Gpa1 dissociates from another G-protein component, Ste4. Moleculizer conditions the GTP binding on the presence of an 'enabling subcomplex' (Gpa1, Ste4, Ste2, pheromone) that can propagate the conformational change in the receptor. Moleculizer provides two special generators for these reactions: one, the 'more precise' version, changes the relative affinity of Gpa1 for GTP and GDP depending on whether or not Gpa1 is in an 'enabling' complex. To actually affect the nucleotide exchange, this generator depends on nucleotide binding and unbinding reactions produced by other reaction generators. These reaction generators are packaged in the (general-purpose) kinase-phosphatase module, because kinase function generally depends on nucleotide (ATP/ADP) binding and unbinding reactions. The other 'less precise' version does not depend on separate reactions to unbind GDP and bind GTP, but simply releases GDP and binds GTP if Gpa1 is in an 'enabling' complex.

The MAP kinase cascade and scaffold complex. A second module supports the mitogen-activated protein (MAP) kinase cascade and the scaffold complex. In this level of the pathway, a sequence of three protein kinases are juxtaposed by a 'scaffold protein,' Ste5. Close proximity increases the ability of each kinase to phosphorylate and activate the next. Moleculizer supports the kinase cascade with special phosphorylation reactions that operate only in species of complexes where a kinase and its substrate are both bound to the scaffold. Just as in the receptor module, the special programming for these reactions allows them to be conditioned on a particular ambient subcomplex. Using a combination of these 'localizing' reactions with generic kinase reactions, provided by a non-alpha-specific module, the user can arrange for the kinase cascade reactions to also take place outside of the scaffolding complex, perhaps at lower rates.

Box 2 Interaction of Moleculizer with other simulators

Moleculizer enables the exact stochastic simulation of biochemical networks and improves the treatment of the contribution of protein complexes to these networks. Because this simulation approach incorporates new complexes and reactions only when needed as the simulation progresses, much smaller reaction networks are generated. This has the key advantage of enabling Moleculizer's output to be exported to other simulators for further work.

We have facilitated this linkage with several translators from Moleculizer's state output into input formats for other simulators.

One translator generates input for rk4tau, an experimental stochastic simulator. Rk4tau is based on Gillespie's 'tau-leaping' idea[16], described previously, incorporating parts of a standard fifth-order adaptive Runge-Kutta solver for ODEs. The accuracy of the Runge-Kutta scheme's fifth order of approximation is not directly reflected in rk4tau's accuracy of stochastic approximation, because rk4tau makes other critical approximations. The order of the scheme was chosen to facilitate step adaption, rather than to improve the accuracy of stochastic simulation. Rk4tau is still experimental; it succumbs frequently to the same 'stiffness' phenomenon[8,33] that hinders using standard 'explicit' ODE solvers on chemical reaction systems. Rk4tau accompanies Moleculizer mainly as a demonstration.

Another translator converts Moleculizer's state documents into input for odie, a simple simulator based on solving ODEs by the Bulirsch-Stoer algorithm, an 'implicit' ODE solver that does not suffer from stiffness.

Finally, a third translator converts Moleculizer's state documents into SBML Level 2 (ref. 34). Because SBML Level 2 does not handle complexes, it is necessary to refer back to the Moleculizer's state file to get the structure of complex species put into the SBML Level 2 file. SBML Level 3 will convey nearly all of the content of a Moleculizer's state file network, including the structures of complex species and modifications of their constituents.

How Moleculizer generates reaction networks

Moleculizer generates reaction networks by a cyclic process attached to, but largely independent of the core stochastic simulation machinery that generates reaction events (**Fig. 1**). We explain this process with an example, the generation of a family of dimerization reactions and their reaction products, illustrated in **Figure 2**. Reaction generation starts when the first molecule of a species appears in the run. The triggering molecule may appear in the initial population of the simulation or when some reaction produces it. Suppose that the molecule is a complex *C1*, and that this complex contains a simple protein *P1*. Suppose that there is already a known complex *C2* containing another simple protein *P2*. Also, suppose that the user has specified on-rates and off-rates for *P1* and *P2* at binding sites exposed in the complexes *C1* and *C2*. Moleculizer asserts that the dimerization between *P1* and *P2* implies a dimerization between *C1* and *C2* because these two complexes expose 'compatible' binding sites. Moleculizer constructs the asserted reaction in two steps, estimating the dimerization rate, then preparing the dimerization product species *C*.

Moleculizer estimates the reaction rate by correcting the rate at which the simple proteins *P1* and *P2* dimerize for the larger molecular weights of the complexes *C1* and *C2*. It performs this correction by reference to the formula

$$c_\mu = V^{-1}\pi d_{12}^2(8kT/\pi m_{12})^{½}\exp(-u_\mu^*/kT)$$

from Gillespie's original exposition[7] and further treatment[6] of the Stochastic Simulation Algorithm. We will neither derive this formula here nor describe all its parts. But c_μ is proportional to the conventional dimerization rate *r*, and the formula relates it to physical properties of the two reacting molecules. We focus on their masses, which appear in the factor

$$m_{12}^{½} = \sqrt{\frac{m_1 m_2}{m_1 + m_2}},$$

where m_1 and m_2 are the molecular weights of *P1* and *P2*. Moleculizer estimates the dimerization rate *r′* between the complexes *C1* and *C2*, of mass m'_1 and m'_2 respectively, by assuming that

$$r\sqrt{\frac{m_1 m_2}{m_1 + m_2}} = r'\sqrt{\frac{m'_1 m'_2}{m'_1 + m'_2}}.$$

This amounts to assuming that the other factors in Gillespie's formula above remain the same for the new reaction. We realize that, because the diameters of *C1* and *C2* are greater than those of *P1* and *P2*, this assumption is unwarranted for the ideal molecular diameters involved in the factor d_{12}. In fact, Moleculizer does not represent or use the geometry of molecules at all, an issue we intend to address in future development. **Figure 3** shows an example of the difficulties to which the 'geometric ignorance' of this way of extrapolating reaction rates can lead. For this reason, we must recognize that this rate extrapolation formula is at best a placeholder. The **Supplementary Notes** online indicates how to engage or disengage this formula. The difficult problem of estimating reaction rates from the physical properties of reactant molecules is central to automatic generation of reaction networks. Any more sophisticated attempt at it will require Moleculizer to address molecular geometry at the very least.

The second step in building the new reaction is 'preparing' the dimerization product species *C*. This means making an entry for *C* in the growing database of all species known to the simulation. The program forms a two-part description of *C*, giving its structure and the states of its simple protein constituents. The structure is derived from the structures of *C1* and *C2*, and the states are the same as they were in *C1* and *C2*. Moleculizer uses this two-part description as a key to search the database of known species. If *C* has already appeared, the program will locate it in the database as detailed in the **Supplementary Notes** and **Supplementary Figure 2** online. If *C* is new, then Moleculizer enters it into the database with a population of zero. But Moleculizer does not generate new reactions having *C* as a reactant until the first triggering molecule of *C* appears. This might happen, for example, when the just constructed dimerization reaction of *C1* and *C2* occurs for the first time. If Moleculizer did not temporize in this way, it would generate the network of all possible reactions and reactants at simulation startup. Instead, it generates reactions at the last instant before the simulation might demand them. By analogy with industrial production, we call this 'just-in-time' reaction generation.

Dimerization reaction construction is typical of the way the program builds all automatically generated reactions. Moleculizer constructs reactions of a particular type, such as the dimerizations in the example, with a reaction generator. Moleculizer packages reaction generators in modules, listed in **Table 1**. A general-purpose module provides the dimerization generator. Reactions may also be entered explicitly, as described in **Table 2** and in the **Supplementary Notes** online. Some performance implications of reaction network generation are also addressed in the **Supplementary Notes** and **Supplementary Tables 1** and **2** online.

Illustrative simulation output

Figure 4 shows an example of Moleculizer output that combines multiple species in a single trace. This plot was produced from an alpha pathway simulation (see **Box 1**) that generated 41,033 reactions and 16,886 species. Reaction rates and molecular populations were not realistic.

The plot focuses on the receptor complex. When α factor is added at time 2.0, Gpa1 undergoes nucleotide exchange and dissociates from the receptor complex. The dark blue trace 'Ste4:Gpa1-GDP' shows the total population of the six complex species where Ste4 is bound to Gpa1-GDP. Gpa1 dissociation is marked by its rapid drop after time 2. The red trace, 'Ste4:Gpa1-total', shows the total population of species with Ste4 bound to Gpa1 in any modification state. Because Ste4 and GTP-bound Gpa1 interact only rarely, dark blue 'Ste4:Gpa1-GDP' and red 'Ste4:Gpa1-total' almost coincide. The light blue trace 'Ste4:Ste5' shows the total population of 1,300 species containing Ste4 bound to Ste5. Its rapid rise after α factor addition reflects the Ste5's recruitment to the membrane and its binding to Ste4, made possible by Gpa1's dissociation from the receptor.

Conclusions

We have described Moleculizer 1.0, a stochastic simulator for intracellular biochemical systems, with special treatment for protein complexes. The strengths of this approach include the fact that the program automatically generates the networks of reactions that arise from the very large numbers of similar protein complexes that occur inside simulations (and cells). To find out more about Moleculizer, including details of its treatment of complexes and allostery, how to obtain it and use it, and our future plans for its development, please consult the **Supplementary Notes** and **Supplementary Tables 3** and **4** online.

The networks of reactions generated by Moleculizer are by-products of its simulation, and include only those complexes and reactions that arise during the simulation. Once generated, they can be exported to other simulators (see **Box 2**). In contrast, several other programs generate larger networks of all the possible reactions that can arise from combinations of individual molecules. Bray and Lay[29] have described OLIGO, a program for constructing the subcomplexes of a user-provided complex, together with reactions between them. Blinov, Faeder and Hlavacek developed BioNetGen (http://cellsignaling.lanl.gov), which enumerates the full set of complexes formed from given constituents. Like Moleculizer, BioNetGen allows protein modifications to figure into the description of complexes.

For all systems, including biological ones, one can assert that the degree to which one can predict their quantitative behavior in response to defined perturbations defines operationally how well they are understood[30,31]. For many intracellular biological systems, we believe that computational abilities like those afforded by Moleculizer, including generation of networks of biochemical reactions, and their solution by stochastic methods, will be helpful in attaining such a predictive understanding.

Note: Supplementary information is available on the Nature Biotechnology website.

ACKNOWLEDGMENTS
L.L. conceived, designed and programmed Moleculizer. R.B. started and fostered the program of quantitative biological inquiry that enabled the development of Moleculizer and helped frame explanations of the work for biologists. L.L. and R.B. wrote the manuscript and stand as guarantors of its veracity. The work was supported by a grant from the US Defense Advanced Research Projects Agency and by the "Alpha Project" at the Center for Genomic Experimentation and Computation, a National Institutes of Health Center of Excellence in Genomic Science. The Alpha Project is supported by grant P50 HG02370 to R.B. from the National Human Genome Research Institute.

COMPETING INTERESTS STATEMENT
The authors declare that they have no competing financial interests.

Published online at http://www.nature.com/naturebiotechnology/

1. Frenkel, D. & Smit, B. *Understanding Molecular Simulation* (Academic Press, San Diego, California, 1996).
2. Pauling, L. The Nature of the Chemical Bond and the Structure of Molecules and Crystals (Cornell University Press, Ithaca, New York, 1960).
3. Vaidehi, N. & Goddard, W. Atomic-level simulation and modeling of biomacromolecules. in *Computational Modeling of Genetic and Biochemical Networks.* (eds. Bower, J. & Bolouri, H.) 161–188 (MIT Press, Cambridge, Massachusetts, 2001).
4. Gillespie, D. A rigorous derivation of the chemical master equation. *Physica A* **188**, 404–425 (1992).
5. Elowitz, M., Surrete, M., Wolf, P., Stock, J. & Leibler, S. Protein mobility in the cytoplasm of *Escherichia coli. J. Bacteriol.* **181**, 197–203 (1999).
6. Gillespie, D. *Markov processes: an introduction for physical scientists* (Academic Press, Boston, Massachusetts, 1992).
7. Gillespie, D. A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *J. Comp. Phys.* **22**, 403–434 (1976).
8. Deuflhard, P. & Bornemann, F. *Scientific Computing with Ordinary Differential Equations* (Springer-Verlag, New York, 2002).
9. Elowitz, M. *et al.* Stochastic gene expression in a single cell. *Science* **297**, 1183–1186 (2002).
10. McAdams, H. & Arkin, A. Stochastic mechanisms in gene expression. *Proc. Natl. Acad. Sci. USA* **94**, 814–819 (1997).
11. Rao, C., Wolf, D. & Arkin, A. Control, exploitation and tolerance of intracellular noise. *Nature* **420**, 231–237 (2002).
12. Mendes, P. Computer simulation of the dynamics of biochemical pathways. PhD thesis, University of Wales Aberystwyth (1994).
13. Cross, F., Archambault, V., Miler, M. & Klovstad, M. Testing a mathematical model of the yeast cell cycle. *Mol. Biol. Cell* **13**, 52–70 (2002).
14. Chen, K.C. *et al.* Kinetic analysis of a molecular model of the budding yeast cell cycle. *Mol. Biol. Cell* **11**, 369–391 (2000).
15. Bormann, G., Brosens, F. & De Schutter, E. Diffusion. in *Computational Modeling of Genetic and Biochemical Networks.* (eds. Bower, J. & Bolouri, H.) 189–224 (MIT Press, Cambridge, Massachusetts, 2001).
16. Gillespie, D. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.* **115**, 1716–1733 (2001).
17. Gibson, M. Computational methods for stochastic biological systems. PhD Thesis, California Institute of Technology (2000).
18. Gibson, M. & Bruck, J. Efficient exact stochastic simulation of chemical systems with many species and many channels. *J. Phys. Chem.* **104**, 1876–1889 (1999).
19. Morton-Firth, C. Stochastic simulation of cell signalling pathways. PhD thesis, University of Cambridge (1998).
20. Gillespie, D. & Petzold, L. Improved leap-size selection for accelerated stochastic simulation. *J. Chem. Phys.* **119**, 8229–8234 (2003).
21. Haseltine, E. & Rawlings, J. Approximate simulation of coupled fast and slow reactions for stochastic chemical kinetics. *J. Chem. Phys.* **117**, 6959–6969 (2002).
22. Rao, C. & Arkin, A. Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm. *J. Chem. Phys.* **118**, 4999–5010 (2003).
23. Keane, J., Bradley, C. & Eberling, C. A compiled accelerator for biological cell signaling simulations. *ACM SIGDA Int. Symp. Field Program Gate Arrays FPGA* **12**, 233–241 (2004).
24. Salwinski, L. & Eisenberg, D. *In silico* simulation of biological network dynamics. *Nat. Biotechnol.* **22**, 1017–1019 (2004).
25. Fricke, T. & Wendt, D. The Markoff automaton: a new algorithm for simulating the time-evolution of large stochastic dynamic systems. *Int. J. Mod. Phys.* **6**, 277–306 (1995).
26. Stiles, J. & Bartol, T. Monte Carlo methods for simulating realistic synaptic microphysiology using MCell. in *Computational Neuroscience: Realistic Modeling for Experimentalists.* (ed. de Schutter, E.) 87–127 (CRC Press, Boca Raton, Florida, 2000).
27. Hodges, P., Payne, W. & Garrels, J. The yeast protein database (YPD): a curated proteome database for *Saccharomyces cerevisiae. Nucleic Acids Res.* **26**, 68–72 (1998).
28. Ptashne, M. *A genetic switch: phage λ and higher organisms* (Blackwell Scientific Publications, Cambridge, Massachusetts, 1992).
29. Bray, D. & Lay, S. Computer-based analysis of the binding steps in protein complex formation. *Proc. Natl. Acad. Sci. USA* **94**, 13493–13498 (1997).
30. Brent, R. Genomic biology. *Cell* **100**, 169–183 (2000).
31. Endy, D. & Brent, R. Modelling cellular behavior. *Nature* **409**, 391–395 (2001).
32. Dohlman, H. & Thorner, J. Regulation of G-protein initiated signal transduction in yeast: Paradigms and principles. *Annu. Rev. Biochem.* **70** (2001).
33. Press, W., Teukolsky, S., Vetterling, W. & Flannery, B. *Numerical Recipes in C,* edn. 2 (Cambridge University Press, Cambridge, 1992).
34. Hucka, M. *et al.* The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics* **19**, 524–531 (2003).