The Microsoft Research - University of Trento
Centre for Computational
and Systems Biology

Technical Report CoSBi 06/2008

Cyto-Sim: A Formal Language Model and Stochastic Simulator of Membrane-Enclosed Biochemical Processes

Sean Sedwards
The Microsoft Research – University of Trento Centre for Computational and Systems Biology
sedwards@cosbi.eu

Tommaso Mazza
The Microsoft Research – University of Trento Centre for Computational and Systems Biology
mazza@cosbi.eu

This is a preliminary version of a paper that will appear in

Bioinformatics

Available at

http://bioinformatics.oxfordjournals.org/

Abstract

Compartments and membranes are the basis of cell topology and more than 30% of the human genome codes for membrane proteins. It is possible to represent compartments and membrane proteins in a nominal way with many mathematical formalisms used in systems biology, however few explicitly model the topology of the membranes themselves.

Discrete stochastic simulation of molecular kinetics potentially offers the most accurate representation of cell dynamics. Since the details of every molecular interaction in a pathway are often not known, the relationship between chemical species in not necessarily best described by simple mass action chemistry. Moreover, modelling every individual molecular interaction in the cell is probably unnecessary and currently impractical.

Simulation is a form of *computer aided analysis*, relying on human interpretation to derive meaning. To improve efficiency and gain meaning in an automatic way, it is necessary to have a formalism based on a model which has *decidable* properties.

We present Cyto-Sim, a stochastic simulator of membrane-enclosed hierarchies of biochemical processes, where the membranes comprise an inner, outer and integral layer. The underlying model is based on well-established formal language theory and has been shown to have decidable properties [1], allowing formal analysis in addition to simulation. The simulator provides arbitrary levels of abstraction based on chemical kinetics and ordinary differential equations; these latter providing a further dimension of analysability.

The paradigm is flexible and extensible, permitting adaptation to other types of simulation and analysis and integration within standard platforms. In addition to its compact native syntax, based on stoichiometric equations and reaction kinetics, Cyto-Sim currently supports models described as Petri nets, can import all versions of SBML and can export SBML and MATLAB® m-files.

Cyto-Sim is available free, either as an applet or a standalone Java program via the web page [6]. Other versions can be made available upon request.

1 Introduction

The function of membranes in cells is fundamental to their activity, separating them from other cells to permit differentiation of function and separating organelles within cells for similar purposes. Membrane proteins regulate the communication between the membrane-enclosed compartments and play a statistically important role in cell activity: more than 30% of the human genome codes for membrane proteins. Since membrane proteins control the entry of susbtances to cells, it is no surprise that in 2000 almost 50% of the drugs prescribed in the USA targetted one class of membrane proteins alone (GPCRs).

In computational terms, compartments correspond to *scope*, that is, regions where calculations can be performed in a local context. In computer programs, scope is the basis of functions, which are a means to perform the same calculation on different data without creating new code - computational differentiation. Biology has a finite repertoire of molecules to perform its *calculations* and uses compartments to increase computational power in the same way, i.e., by having a multiplicity of membrane-enclosed cells.

Our aim is to create predictive or otherwise useful models of biological processes, with our current focus on iter- and intracellular pathways. Recognising the important roles of membrane proteins and biological scope, we require a model that allows us to simultaneously represent different (types of) cells which communicate via ligands and receptors. In addition to the obvious biological analogues, membranes can also be used to model notional compartments in order to represent, for example, diffusion or localised behaviour. Further, compartments can be used like assay plate wells, enabling several experiments to be run simultaneously and efficiently.

2 Approach

Our simulator model is based on that of *P systems* [7] (a.k.a. *Membrane Systems*), a computational paradigm which draws its inspiration from biological cells: *multisets* of objects are enclosed in a nested hierarchy of compartments and acted upon by local *rewriting* rules. This translates to: collections of molecules in a nested hierarchy of cellular compartments acted upon by local chemical reactions. We have extended the basic Membrane Systems model to include peripheral and integral membrane proteins [1] and incorporate a Markov chain Monte Carlo algorithm to simulate the time evolution of the system.

Our default level of abstraction is chemistry: objects interact governed by stoichiometric rules at a rate defined by mass action kinetics. A variable level of abstraction is facilitated by the use of arbitrary kinetic laws: objects are consumed and produced at rates defined by arbitrary functions of reactants, thus modelling behaviour rather than mechanism. This allows complex systems to be reduced to single equations that nevertheless remain in the formal language and Markov framework for analysis. Such reduction may be a necessity if the structure of the system is unknown, or may be used to gain efficiency when the structure is unimportant.

The native simulator syntax language aims to be intuitive and uncluttered [1]. Objects, rules and compartments are defined, then (a subset of) these are composed to create the final system to simulate. In this way it is not necessary to explicitly define rules and objects for each compartment, as is necessary, for example, in SBML.

Petri nets are an intuitive graphical representation of logical flows, which have been successfully applied to biology (e.g., [4]). Hence, in addition to the native rule syntax, the simulator supports rule definitions in the form of Petri net incidence matrices. There is strong correspondence between stochastic Petri nets and stochastic chemical rules so, by default, the transitions adopt mass action kinetics, i.e., the rate of the transition is proportional to a constant multiplied by the numbers of tokens in the incoming places. Other Petri net dynamics are also possible by explicitly defining appropriate kinetic laws.

3 Methods

The software is written in J#, a Java-like language which is part of the .NET framework. This choice allows easy porting to both Java and C# and hence maximises cross-platform compatibility. There are currently two versions available on the internet [6]; an applet and a standalone version, both implemented in Java and having a graphical user interface (GUI). Other versions can be made available upon request to the authors.

The core simulation engine is an efficient implementation of a Markov chain Monte Carlo algorithm, based on that of Gillespie [2]. Performance is optimised for mass action chemical kinetics, however the simulator also supports chemical reactions with arbitrary kinetic laws based on functions of the reactants. This latter restriction guarantees the Markov property but permits the use of the many alternative kinetic laws used in systems biology. The addition of arbitrary kinetic laws is achieved in an efficient way, based on compiler technology and a virtual machine, and does not adversely affect the performance of the default kinetics.

4 Discussion

Much of the modelling in systems biology is done in the framework of deterministic differential equations, which are usually *solved* by numerical methods. Such solutions might more accurately be described as deterministic *simulations*. Recognising that molecular interactions are discrete stochastic events and that this stochasticity has a significant effect on the behaviour of models which have neutral or unstable manifolds [2], discrete stochastic simulations are now often used to give a more accurate representation of such behaviour. The principal qualitative difference between the two approaches is that, for a given set of simulation parameters, a deterministic simulation will have a unique, *average*, solution, whereas a discrete stochastic simulation is a single trajectory through the solution space. By using Monte Carlo techniques, the stochastic trajectory is guaranteed to be statistically consistent with the master equation describing the system, however this does not guarantee that a particular behaviour will be observed in any given simulation. Hence, a deterministic simulation gives a single average characterisation of the system but may not characterise all the behaviour, whereas a discrete stochastic simulation is only guaranteed to display all the behaviour in the limit of simulations. The choice between these two techniques is therefore dependent on the quality and robustness of the behaviour and the types of manifold present. Given that some robust behaviour may not be observed in a deterministic simulation, e.g., behaviour that relies on a stochastic divergence at a bifurcation point, performing many stochastic simulations is usually preferable. The efficiency of the Cyto-Sim algorithm, which can outperform the deterministic simulation of some models (particularly those with many reactions and low molecular concentrations, see Table 1) makes such statistical analysis feasible. In Cyto-Sim, models can also be exported as sets of differential equations in the form of a MATLAB$^{®}$ m-file. So, having performed initial stochastic simulations, it is thus possible to perform algebraic or numerical analysis or, at least, deterministic simulation, to gain further insight about the system.

Simulator	Yeast G prot. cycle	N-R oscillator
Cyto-Sim	**1.1s**	**0.41s**
CellDesigner[1]	1.5s	2.2s
MATLAB®[2] (deterministic)	617s	6.3s
Dizzy[3]	10^{13}s (Dizzy estd.)	19.3s

(Windows XP, JRE 1.6, Centrino M, 1.6GHz, 2MB L2 cache, 512 MB RAM)
[1]www.celldesigner.org [2]www.mathworks.com [3]www.systemsbiology.org

Table 1: Timings of Cyto-Sim and other simulators with models of yeast G protein coupled cycle [3] and a noise-resistant oscillator [5]. Simulalted times were 600 seconds and 250 hours, respectively.

5 Conclusion

We have developed a tool to provide stochastic simulation of biological processes in compartments, based on Markov Chain Monte Carlo techniques and which can be used at arbitrary and mixed levels of abstraction. We can simulate micro- and macroscopic biological processes using arbitrary kinetic laws best suited to each defined interaction. Our use of hierarchical compartments has wide application in biological modelling, while the way in which we have modelled membranes is specifically aimed at molecular cell biology. The explicit modelling of compartments and membranes facilitates the construction of models by composing instances of the variously defined parts, making them compact and transparent. With a common root in Markov theory, the simulator accepts models defined with combinations of stoichiometric-like rules, Petri net matrices and kinetic functions. Analysis can thus be performed on the models in the frameworks of formal language, Petri nets, Markov theory and differential equations, as required.

References

[1] Cavaliere, M., Sedwards, S. (2006) Modelling Cellular Processes Using Membrane Systems with Peripheral and Integral Proteins, *Lect. Notes in Comp. Sci.*, **4210**, 108-126.

[2] Gillespie, D. T. (1977) Exact Stochastic Simulation of Coupled Chemical Reactions, *Journal of Physical Chemistry*, **81**, 2340-2361.

[3] Yi, T-M., Kitano, H., Simon, M. I. (2003) A quantitative characterization of the yeast heterotrimeric G protein cycle, *Procs. of the Nat. Acad. of Sci.*, **100**, 19.

[4] Nutsch, T., D. Oesterhelt, E.-D. Gilles, and W. Marwan (2005) A quantitative model of the switch cycle of an archael flagellar motor and its sensory control, *Biophysical Journal*, **89**, 2307-2323.

[5] Vilar, M. G., Kueh, H. Y., Barkai, N., Leibler, S. (2003) Mechanisms of noise-resistance in genetic oscillators, *Procs. of the Nat. Acad. of Sci.*, **99**, 9.

[6] Cyto-Sim web page, http://www.cosbi.eu/Rpty_Soft_CytoSim.php/

[7] P Systems web page, http://psystems.disco.unimib.it/

Revised 30/9/09