BIOINFORMATICS APPLICATIONS NOTE Vol. 19 no. 5 2003, pages 677–678
DOI: 10.1093/bioinformatics/btg042

Cellerator: extending a computer algebra system to include biochemical arrows for signal transduction simulations

Bruce E. Shapiro[1,], Andre Levchenko[2], Elliot M. Meyerowitz[3], Barbara J. Wold[3] and Eric D. Mjolsness[4]*

[1]Jet Propulsion Laboratory, California Institute of Technology, Mail Stop 126-347, 4800 Oak Grove Drive, Pasadena, CA 91109, USA, [2]Department of Biomedical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA, [3]Division of Biology, California Institute of Biology, Pasadena, CA 91125, USA and [4]Department of Information and Computer Science, University of California, Irvine 92647, USA

Received on May 30, 2002; revised on September 26, 2002; accepted on November 4, 2002

ABSTRACT

Summary: Cellerator describes single and multi-cellular signal transduction networks (STN) with a compact, optionally palette-driven, arrow-based notation to represent biochemical reactions and transcriptional activation. Multi-compartment systems are represented as graphs with STNs embedded in each node. Interactions include mass-action, enzymatic, allosteric and connectionist models. Reactions are translated into differential equations and can be solved numerically to generate predictive time courses or output as systems of equations that can be read by other programs. Cellerator simulations are fully extensible and portable to any operating system that supports Mathematica, and can be indefinitely nested within larger data structures to produce highly scaleable models.

Availability: Cellerator can be licensed free of charge to noncommercial academic, U.S. government, and nonprofit users. Details and sample notebooks are available at http://www-aig.jpl.nasa.gov/public/mls/cellerator.

Contact: bshapiro@jpl.nasa.gov

Cellerator is a Mathematica package that facilitates biological modeling via automated equation generation, with the intent of simulating at least the following essential biological processes: (a) signal transduction networks (STNs); (b) cells that are represented by interacting signal transduction networks; and (c) multi-cellular tissues that are represented by interacting networks of cells that may themselves contain internal STNs. STNs are specified using an arrow-based language (Shapiro *et al.*, 2001) to represent biochemical interactions, including simplified representations of transcriptional regulation. The general input canonical form is {*reaction*, *reaction*, ...} where

$$reaction = \{rlist \overset{catalyst}{\underset{catalyst}{arrow}} rlist, clist\} \quad (1)$$

where *rlist* is a list of reactants (e.g. $A + B + C$); *arrow* is one of the arrows in Table 1 (e.g. $\rightarrow, \rightleftharpoons, \mapsto$); *catalyst* is an optional species that catalyzes the reaction (such as an enzyme; the upper and lower catalysts affect forward and reverse reactions); and *clist* is an optional list of comma-delimited rate constants (symbolic names or values). For example, $S \overset{A}{\rightleftharpoons} P$ represents the sequence of chemical reactions $S + A \rightleftharpoons X \rightarrow P + A$ (e.g. S, P, A, and X are source, product, catalyst and intermediate compound in an enzymatic reaction). Reactions are symbolically translated and collected into differential equations using symbolic algebra. Thus the expression `interpret[{{S`$\overset{A}{\rightleftharpoons}$`P,a,d,k},{A+B`$\rightarrow$`C,k1}}]` returns lists of ODEs and chemical species

```
{{A'[t]==-k1*A[t]*B[t]-a*A[t]*S[t]
         +d*SPlusA[t]+K*SPlusA[t],
  B'[t]==-K1*A[t]*B[t],
  C'[t]==K1*A[t]*B[t],
  P'[t]==K*SPlusA[t],
  S'[t]==-a*A[t]*S[t]+d*SPlusA[t],
  SPlusA'[t]==a*A[t]*S[t]-d*SPlusA[t]
             -k*SPlusA[t]},
  {A, B, C, P, S, SPlusA}}
```

The names of intermediate complexes are automatically generated. Unlimited additional reactions and explicit differential equations may also be included. Cascades and inter-compartmental diffusion can be specified by adding indices to variables, e.g. as, `A[n]`$\rightleftharpoons$`A[n]`$\rightleftharpoons$`...`$\rightleftharpoons$`A[n]`. It is not necessary to explicitly use the `interpret` command; users enter `run`[*reactions, options*], to translate, numerically solve, and if desired, plot the predicted time course of the system. Alternatively, multiple `interpret` commands can be combined and edited before running a simulation.

*To whom correspondence should be addressed.

Table 1. Cellerator arrow notation

Cellerator arrow	Terms in ODES
$S \to P$	$\dot{S} = -\dot{P} = -k$
$A + B \to C$	$\dot{A} = \dot{B} = -\dot{C} = -kAB$
$A + B^n \to C$	$\dot{A} = \dot{B} = -\dot{C} = -kAB^n$
$A \rightleftharpoons B$	$\dot{A} = -\dot{B} = -k_f A + k_r B$
$A + B \rightleftharpoons C$	$\dot{A} = \dot{B} = -\dot{C} = -k_f AB + k_r C$
$\emptyset \to A$	$\dot{A} = k$
$B \to \emptyset$	$\dot{B} = -kB$
$S \overset{E}{\rightleftharpoons} P$	$\dot{S} = -a \cdot E \cdot S + d \cdot S, P\cdot = k \cdot (SE)$
	$\dot{E} = -a \cdot E \cdot S + (d + k) \cdot (SE) = -(\dot{SE})$
$S \underset{F}{\overset{E}{\rightleftharpoons}} P$	Equivalent to $S \overset{E}{\rightleftharpoons} P$ and $P \overset{F}{\rightleftharpoons} S$
$S \overset{E}{\to} P$	$\dot{S} = -k \cdot E \cdot S = -\dot{P}$
$S \overset{E}{\mapsto} P$	$\dot{P} = \frac{(k+vE)S^n}{K^n+S^n} = -\dot{S}$
$A \mapsto B$(Hill)	$\dot{B} = r_0 + \frac{(r_1+\sum_{i=1}^{p} v_i A_i)^n}{K^n+(r_1+\sum_{i=1}^{p} v_i A_i)^n}$
$A \mapsto B$(GRN)	$\dot{B} = R\left[1 + \exp\left(-\sum_{i=1}^{p} T_i A_i^{n_i} + h_i\right)\right]^{-1}$
$A \mapsto B$(NHCA)	$\dot{B} = \frac{1+(\sum_{i=1}^{p} T_i^{+} A_i^{n_i})^m}{k(\sum_{i=1}^{p} T_i^{-} A_i^{n_i})^m+(\sum_{i=1}^{p} T_i^{+} A_i^{n_i})^m}$

The column on the left gives the exact syntax of the arrow in equation (1). Arrows can be typed (a) explicitly in Mathematica (e.g. $S \to P$ typed from the keyboard as `S\[ShortRightArrow]P`); (b) using standard Mathematica palettes; (c) using the Cellerator Palette (which has buttons for each arrow and templates fore each reaction); (d) using Mathematica text-equivalents of the command (e.g. $S \to P$ can also be written as `ShortRightArrow[S,P]`); or (e) by using Cellerator text-equivalents of each command (e.g. $S \to P$ can be written as `arrow[conversion,S,P,k]`. See the web site for details and examples. The column on the right gives the differential equations that a single reaction produces. Reactions are combined and converted to ODEs. In addition to mass action, the right-tee arrow $\mapsto$ can be used for transcriptional activation (Hill, sigmoidal and allosteric models)

Other functions allow the user to automatically determine input/output relationships, i.e. the concentration of one (or more) protein(s) after, say, 10 minutes, as a function of the concentration of a signal protein. Reactions, differential equations, and all Cellerator commands can be embedded within Mathematica functions, giving 'power-users' nearly unlimited flexibility in controlling and extending simulations. Output can be expressed in a variety of formats: as Mathematica ODEs, in C, FORTRAN, SBML, MATHML, or HTML files. High-resolution images of the Cellerator command palette and screen are available on the Cellerator web page.

Multicellular systems are represented by graphs containing a list of *nodes* (representing cells), a list of *links* (representing intercellular interaction), and a *lineage tree* (a familial history of cell birth). Cell division occurs (optionally) whenever a user-specified variable passes a threshold, and new cells are added to the graph in corresponding locations, within nodes; genetic regulatory networks are represented by a generalization of the connectionist model (Mjolsness *et al.*, 1991). If cell j contains n species with concentrations v_a, $a = 1, 2, \ldots, n$, then

$$\tau_a \dot{v}_a = g(u_a + h_a) - \lambda_a v_a + \tau_a \dot{v}_{a,Cellerator}$$

where τ_a, h_a and λ_a are constants, $\dot{v}_{a,Cellerator}$ is the sum of terms from Table 1, $g(x) = [1 + x/(1 + x^2)^{1/2}]/2$, and

$$u_a = \sum_b T_{ab} v_b + \sum_{i \in Nbrs} \Lambda^i \sum_b \hat{T}_{ab} \hat{v}_b^i + \sum_{i \in Nbrs} \Lambda^i \sum_b \sum_c \tilde{T}_{ac}^{(1)} \tilde{T}_{cb}^{(2)} v_c \hat{v}_b^i.$$

Here hats indicate concentrations in neighboring cell i; Λ^i is the geometric connectivity; and T_{ab}, $\hat{T}_{ab}$, $\tilde{T}_{ac}^{(1)}$ $\tilde{T}_{cb}^{(2)}$ give the production rate of a by b locally; of a by b in a neighboring cell; of a by activation of receptor c; and the activation of receptor c by ligand b, respectively.

To describe growth, a spring potential $V_{ij} = \frac{1}{2}\sum_j k_{ij}[(|\mathbf{x}_i - \mathbf{x}_j| - d_{ij})^2 - \mu]$, k_{ij} constant, is associated with each link. The desired separations d_{ij} between spherical cells increase as cells grow; a gradient towards local minimum ($\dot{\mathbf{x}}_i = -\nabla V_i$) ensures an exponential relaxation of actual towards desired separations in response to intercellular fluidic and tensile forces. V_{ij} is set to zero when the interaction distance becomes too large ($d_{ij} > d$ for some constant d).

The tedious process of manually translating chemical networks from cartoon-diagrams to chemical and differential equations is highly error prone and impractical for all but the simplest of systems because of the combinatoric increase in the number of equations with chemical species. Cellerator provides a framework for generating, translating, and numerically solving a potentially unlimited number of biochemical interactions. To our knowledge it is the first (and only) extensible program that does this by adding biochemical notation to a currently existing modeling language, thereby inheriting (and extending) the full power of that language.

ACKNOWLEDGEMENTS

Portions of this research were carried out by the Jet Propulsion Laboratory, California Institute of Technology, under contract with NASA.

REFERENCES

Mjolsness,E., Sharp,D.H. and Reinitz,J. (1991) A connectionist model of development. *J. Theor. Biol.*, **152**, 429–453.

Shapiro,B.E., Levchenko,A. and Mjolsness,E. (2001) Automatic model generation for signal transduction a to MAP-Kinase pathways. In Kitano,H. (ed.), *Foundations of Systems Biology*. MIT Press.