AUTOMATIC EQUATION GENERATION FOR SIGNAL TRANSDUCTION MODELING

Bruce E. Shapiro*, Andre Levchenko**†, Eric Mjolsness*†,
*Machine Learning Systems Group, Jet Propulsion Laboratory, and
Divisions of †Biology and **Engineering and Applied Science, California Institute of Technology

The Problem

The rapid growth of information about intracellular signal transduction and genetic networks has led to the view of regulatory biomolecular circuits as highly structured multi-component systems evolved to perform optimally in very uncertain environments.

This emergent complexity of biochemical regulation necessitates the development of new tools for analysis, especially computer assisted mathematical models. Computer modeling has proved to be of crucial importance in the analysis of genomic DNA sequences and molecular dynamics simulations and is likely to become an indispensable tool in biochemical and genetic research.

Several platforms exist that perform complex simulations of various aspects of cellular signaling and genetic regulatory networks. Despite their promise, these new modeling environments have not been widely utilized in the biological research community, possibly because of the relative inaccessibility of the modeling interface.

Instead of cartoon representations of signaling pathways, in which activation can be represented by an arrow connecting two molecular species, users are often asked to write specific differential equations or choose among different modeling approximations. Even for fairly modest biomolecular circuits this involves explicitly writing dozens to hundreds of differential equations, a job that can be highly error prone, even for an experienced modeler.

The Combinatoric Explosion

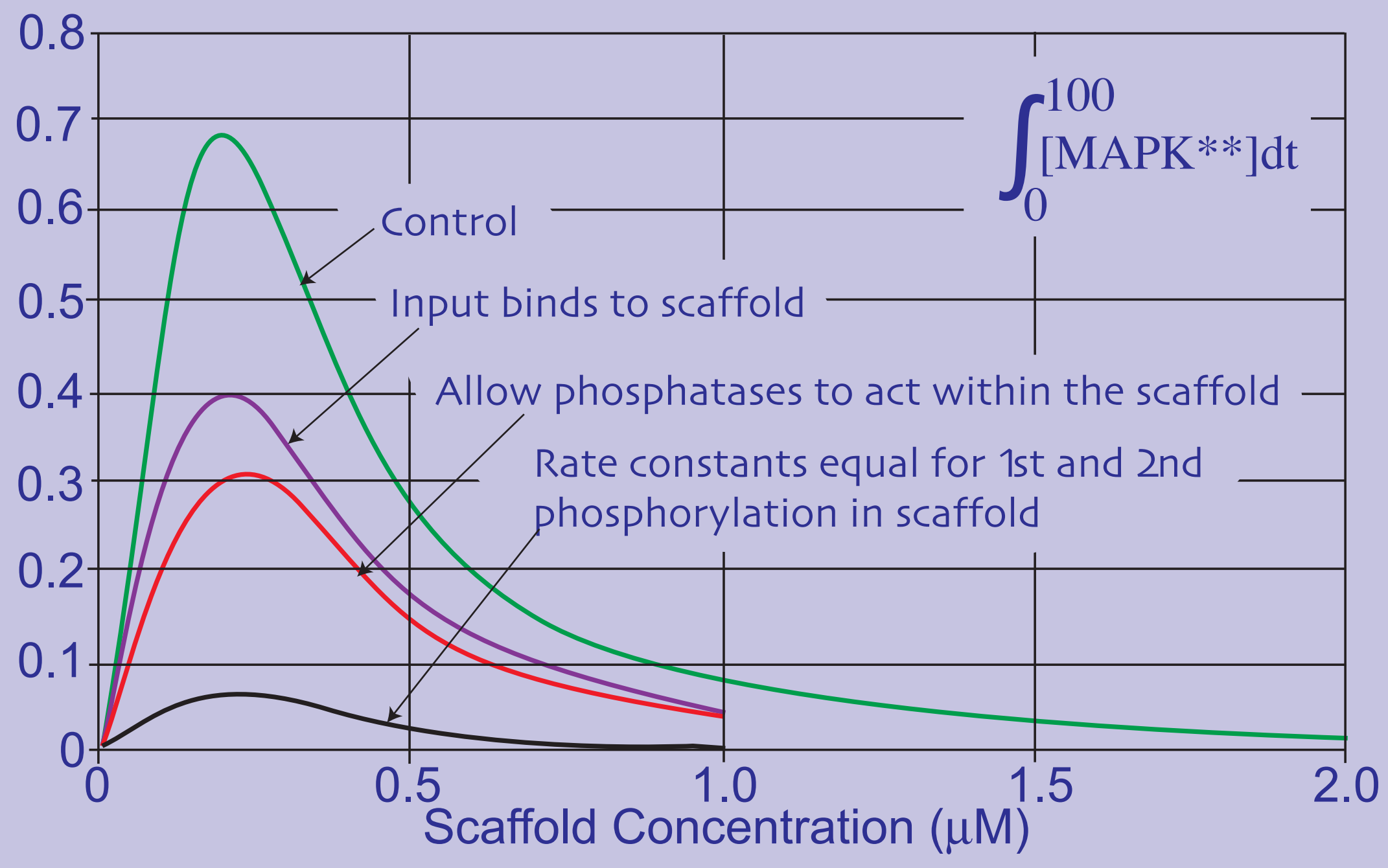
It is not uncommon to find multi-molecular complexes of modifiable proteins in which the number of states increases exponentially with the number of molecules or classes of molecules. The following table illustrates this effect for a typical example, the MAPK cascade.

cascade in solution	phosphatase (solution)	scaffold	phosphatase (scaffold)	separate on signal	high level reactions	low level reactions	species (equations)	terms (all equations)
●					5	15	14	46
●	●				5	30	22	90
●	●			●	6	36	26	105
●	●	●			139	300	85	778
●	●	●	●		139	348	101	922
●	●	●	●	●	140	354	105	991

Notes: "separate on signal" means add one additional reversible (enzymatic) stage to the input; "species" is the total number of molecular species required to describe the process; "terms" gives the total number of terms in all differental equations added together. There is one differential equation for each molecular species.

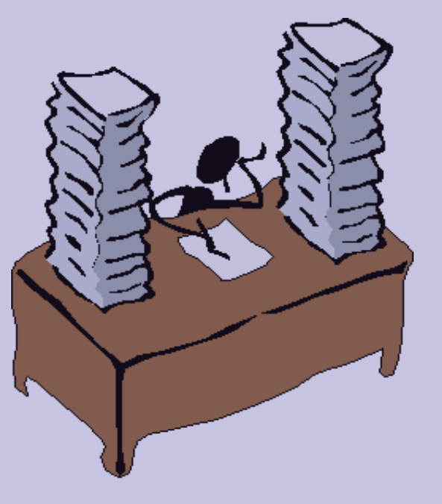
Science with Cellerator

Results of a study of parametric variation in a model of the MAPK cascade using Cellerator. The time integral of free dually phosphorylated MAPK over the first 100 seconds after stimulation is pl0tted vs. scaffold concentration. The "control" curve reproduces the results of an earlier model (Levchenko et al, PNAS 97:5818-23, 2000). The remaining curves relax various assumptions made in the earlier model.

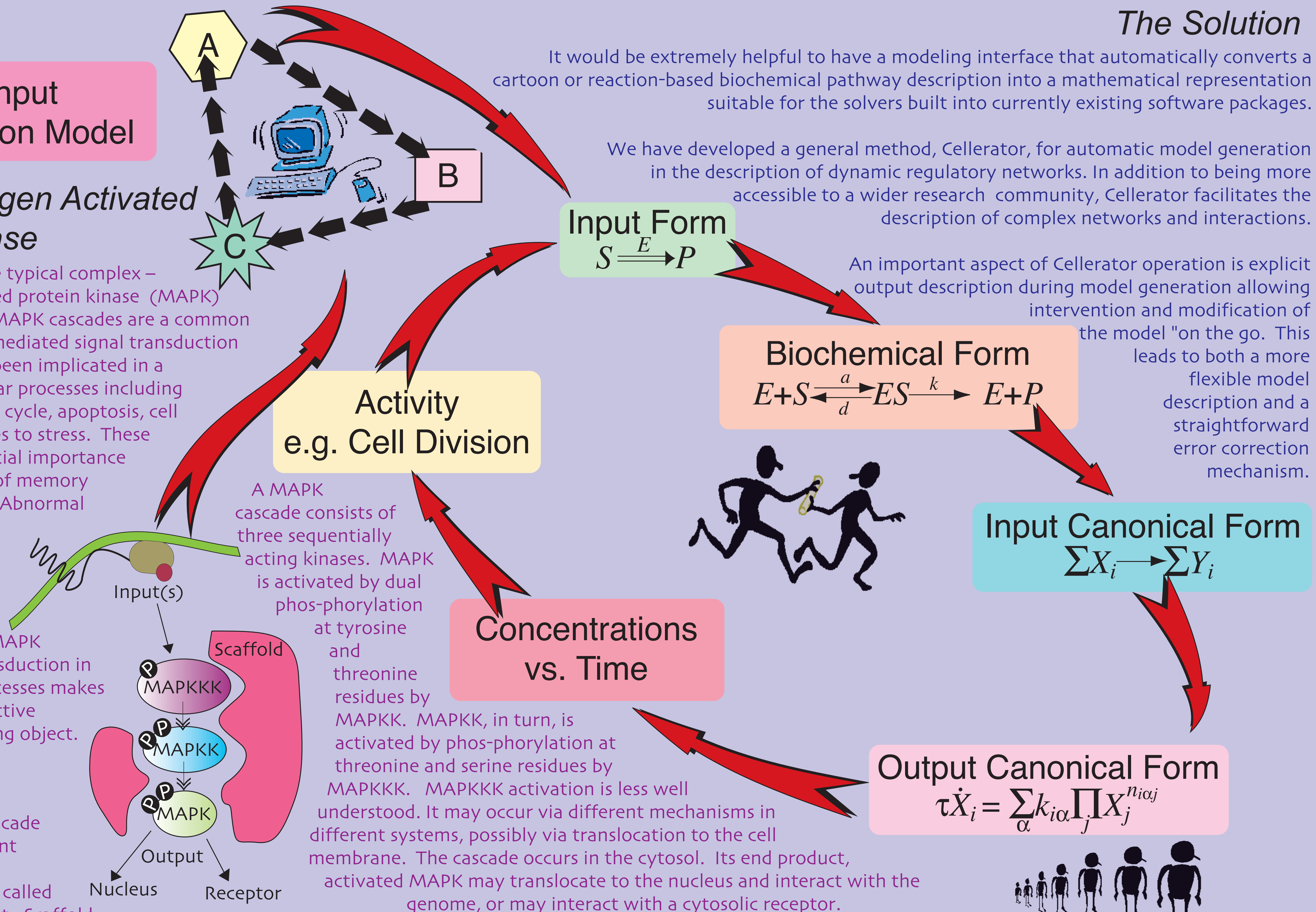


For more information about Cellerator, contact:

Bruce E. Shapiro
M/S 126-347, Jet Propulsion Laboratory
4800 Oak Grove Drive, Pasadena, CA 91109
Electronic mail: bshapiro@jpl.nasa.gov
Telephone: 818-393-0980



For more information about machine learning at JPL, visit our web page at http://www-aig.jpl.nasa.gov/mls



MAPK: Mitogen Activated Protein Kinase

We have studied one typical complex – the mitogen-activated protein kinase (MAPK) cascade – in detail. MAPK cascades are a common feature of receptor mediated signal transduction pathways and have been implicated in a variety of intercellular processes including regulation of the cell cycle, apoptosis, cell growth and responses to stress. These molecules are of crucial importance in the development of memory and wound healing. Abnormal changes in MAPK pathway regulation often mediate various pathologies, most notably cancer. This central role of MAPK mediated signal transduction in most regulatory processes makes it an especially attractive research and modeling object.

A MAPK cascade consists of three sequentially acting kinases. MAPK is activated by dual phos-phorylation at tyrosine and threonine residues by MAPKK. MAPKK, in turn, is activated by phos-phorylation at threonine and serine residues by MAPKKK. MAPKKK activation is less well understood. It may occur via different mechanisms in different systems, possibly via translocation to the cell membrane. The cascade occurs in the cytosol. Its end product, activated MAPK may translocate to the nucleus and interact with the genome, or may interact with a cytosolic receptor.

Scaffolds

Signal transduction through a MAPK cascade can be very inefficient unless additional regulatory proteins, called scaffolds, are present. Scaffold proteins nucleate signaling by binding two or more MAP kinases into a single multi-molecular complex. Scaffolds can change the efficiency of signaling in a concentration dependent manner as well as reduce the non-linear activation characteristics of the cascade. These properties may be crucial for global and local activation of MAPK as scaffold proteins may selectively translocate to small sub-cellular compartments, thus locally facilitating or inhibiting MAPK activation. Cellerator includes both types of MAPK module - in solution and when bound to a scaffold.

MAPK Formal Description

Molecules:

$\{K_{ij}: i=1,..,n,\ j = 0,1,...,a_i\}$ (free Kinase)

$\{S_{p_1,p_2,...,p_n}: p_i = \varepsilon,0,1,...,a_i\}$(on Scaffold)

Initial Conditions:

$\{K_{ij}(0) : i = 1,...,n,\ j = 0,...,a_i\}$

I/O:

Input: $K_4(\mathrm{t})=K_{40}\Theta(t\text{-}t_{on})$

Output: $\int K_{i,a_1}(t)dt$

Scaffold Binding Reactions:

$$\{S_{p_1,...,p_i=\varepsilon,...,p_n} + K_{ij} \longleftrightarrow S_{p_1,...,p_i=j,...,p_n}\}$$

Phosphorylation:

In Solution

$$\{K_{ij} \Longleftrightarrow K_{i,j+1}: i = 1,...,n\text{-}1,\ j = 0,...,a_j\text{-}1\}$$

On Scaffold by Free Kinase

$$\{S_{p_1,...,p_{i-1}=j<a_{i-1},p_i=a_i,...,p_n} + K \Longleftrightarrow S_{p_1,...,p_{i-1}=j+1,p_i=a_i,...,p_n}\}$$

On Scaffold by Bound Kinase

$$\{S_{p_1,...,p_{i-1}=j<a_{i-1},p_i=a_i,...,p_n} \longrightarrow S_{p_1...,p_{i-1}=j+1<a_{i-1},p_i=a_i,...,p_n}\}$$

Canonical Forms

Cellerator classifies each input as instantiating a particular "input canonical form (ICF)" , chosen from a library of fundamental biochemical processes. An ICF is a single chemical formula typical of a wide class of reactions. Examples of ICFs include association, dissociation, degradation, conversion, creation, enzymatic, transcriptional, and translational reactions. Corresponding to each ICF there is a unique "output canonical form (OCF)" - a mathematical description, usually terms in differential equations - into which the reaction is translated. This process requires formal mathematical grammars at several levels: input form (see below); detailed biochemical description (example at left); ICFs; and OCFs . An overview of this process is illustrated in the large figure at the top right of this poster.

Cellerator Input Forms

Canonical forms are not the same as input forms. For example, the input

$$S \overset{E}{\Rightarrow} P$$

denotes the conversion of molecule S into P via an enzymatic reaction facilitated by E. Hidden within this notation are three chemical reactions

$$S+E \xrightarrow{k_1} SE,\quad SE \xrightarrow{k_2} S+E,\quad SE \xrightarrow{k_3} S+P$$

where the k are rate constants. Two-way reactions are input as

$$S \underset{G}{\overset{E}{\Longleftrightarrow}} P$$

where G is the reverse enzyme. This two way reaction corresponds to six simple chemical reactions.

Indices describe steps in a cascade. A 3-step MAPK cascade in solution in the absence of phosphatase is

$$\mathrm{K}_3 \overset{\mathrm{K}_4^*}{\Longrightarrow} \mathrm{K}_3^*$$
$$\mathrm{K}_2 \overset{\mathrm{K}_3^*}{\Longrightarrow} \mathrm{K}_2^* \overset{\mathrm{K}_3^*}{\Longrightarrow} \mathrm{K}_2^{**}$$
$$\mathrm{K}_1 \overset{\mathrm{K}_2^{**}}{\Longrightarrow} \mathrm{K}_1^* \overset{\mathrm{K}_2^{**}}{\Longrightarrow} \mathrm{K}_1^{**}$$

The indexed notation can also be used to indicate scaffold occupancy (see figure to the right). We have implemented cascades-on-command so the user is not required to actually type in all of the chemical reactions.

Cellerator input is via Mathematica; menu and GUI/cartoon driven interfaces are planned.

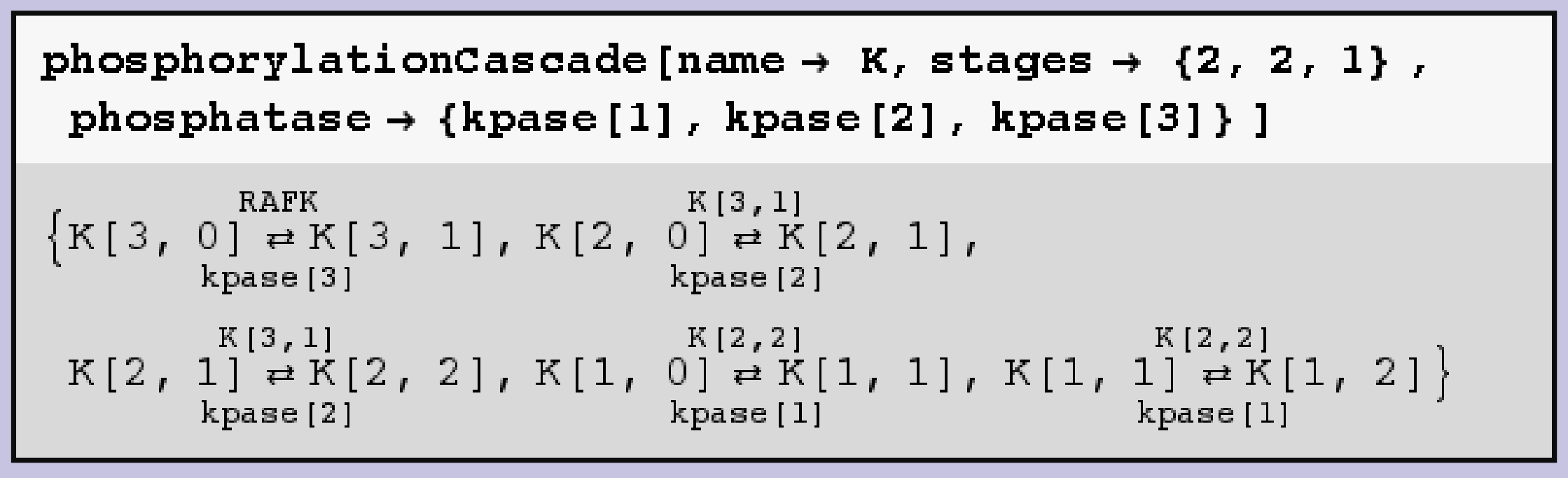
The Solution

It would be extremely helpful to have a modeling interface that automatically converts a cartoon or reaction-based biochemical pathway description into a mathematical representation suitable for the solvers built into currently existing software packages.

We have developed a general method, Cellerator, for automatic model generation in the description of dynamic regulatory networks. In addition to being more accessible to a wider research community, Cellerator facilitates the description of complex networks and interactions.

An important aspect of Cellerator operation is explicit output description during model generation allowing intervention and modification of the model "on the go. This leads to both a more flexible model description and a straightforward error correction mechanism.

Automatic Equation Generation



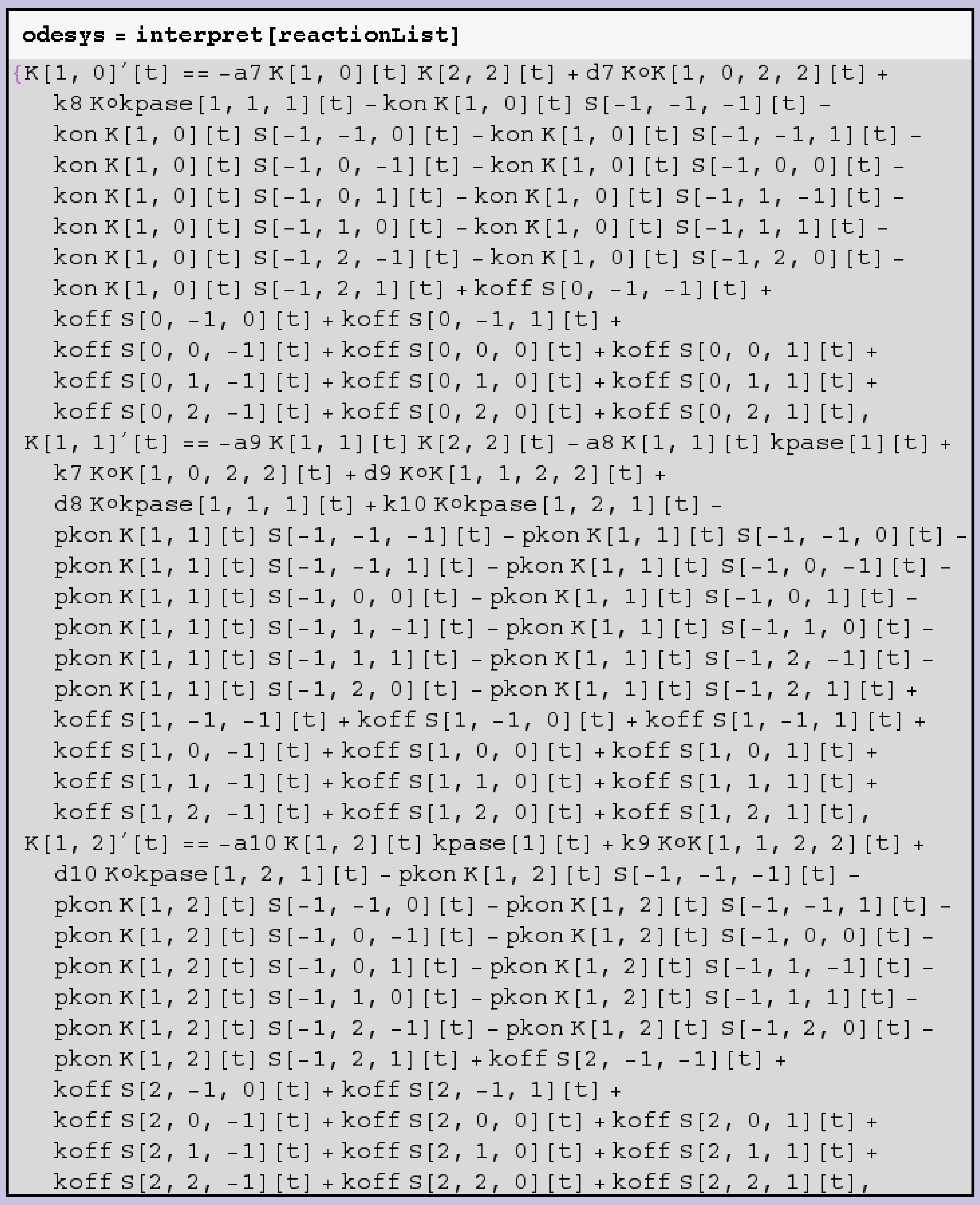
```
phosphorylationCascade[name → K, stages → {2, 2, 1} ,
 phosphatase → {kpase[1], kpase[2], kpase[3]} ]

        RAFK                K[3,1]
{K[3, 0] ⇄ K[3, 1], K[2, 0] ⇄ K[2, 1],
       kpase[3]            kpase[2]

        K[3,1]              K[2,2]              K[2,2]
 K[2, 1] ⇄ K[2, 2], K[1, 0] ⇄ K[1, 1], K[1, 1] ⇄ K[1, 2]}
       kpase[2]            kpase[1]            kpase[1]
```

Cellerator can be used to generate the chemical reactions for an entire signaling module - MAPK in solution - using the arrow-based Mathematica interface, as illustrated above. The user can optionally modify the biochemical equations manually before translating them into differential equations. The indices account for the kinase and phosphorylation states.

The first 3 of 105 differential equations that Cellerator generates with a single command to describe MAPK on scaffold are shown below. S[i,j,k] is the concentration of scaffold protein; indices indicate the phosphorylation levels of each bound enzyme. A value of -1 indicates an empty slot.



```
odesys = interpret[reactionList]
{K[1, 0]′[t] == -a7 K[1, 0][t] K[2, 2][t] + d7 K∘K[1, 0, 2, 2][t] +
   k8 K∘kpase[1, 1, 1][t] - kon K[1, 0][t] S[-1, -1, -1][t] -
   kon K[1, 0][t] S[-1, -1, 0][t] - kon K[1, 0][t] S[-1, -1, 1][t] -
   kon K[1, 0][t] S[-1, 0, -1][t] - kon K[1, 0][t] S[-1, 0, 0][t] -
   kon K[1, 0][t] S[-1, 0, 1][t] - kon K[1, 0][t] S[-1, 1, -1][t] -
   kon K[1, 0][t] S[-1, 1, 0][t] - kon K[1, 0][t] S[-1, 1, 1][t] -
   kon K[1, 0][t] S[-1, 2, -1][t] - kon K[1, 0][t] S[-1, 2, 0][t] -
   kon K[1, 0][t] S[-1, 2, 1][t] + koff S[0, -1, -1][t] +
   koff S[0, -1, 0][t] + koff S[0, -1, 1][t] +
   koff S[0, 0, -1][t] + koff S[0, 0, 0][t] + koff S[0, 0, 1][t] +
   koff S[0, 1, -1][t] + koff S[0, 1, 0][t] + koff S[0, 1, 1][t] +
   koff S[0, 2, -1][t] + koff S[0, 2, 0][t] + koff S[0, 2, 1][t],
 K[1, 1]′[t] == -a9 K[1, 1][t] K[2, 2][t] - a8 K[1, 1][t] kpase[1][t] +
   k7 K∘K[1, 0, 2, 2][t] + d9 K∘K[1, 1, 2, 2][t] +
   d8 K∘kpase[1, 1, 1][t] + k10 K∘kpase[1, 2, 1][t] -
   pkon K[1, 1][t] S[-1, -1, -1][t] - pkon K[1, 1][t] S[-1, -1, 0][t] -
   pkon K[1, 1][t] S[-1, -1, 1][t] - pkon K[1, 1][t] S[-1, 0, -1][t] -
   pkon K[1, 1][t] S[-1, 0, 0][t] - pkon K[1, 1][t] S[-1, 0, 1][t] -
   pkon K[1, 1][t] S[-1, 1, -1][t] - pkon K[1, 1][t] S[-1, 1, 0][t] -
   pkon K[1, 1][t] S[-1, 1, 1][t] - pkon K[1, 1][t] S[-1, 2, -1][t] -
   pkon K[1, 1][t] S[-1, 2, 0][t] - pkon K[1, 1][t] S[-1, 2, 1][t] +
   koff S[1, -1, -1][t] + koff S[1, -1, 0][t] + koff S[1, -1, 1][t] +
   koff S[1, 0, -1][t] + koff S[1, 0, 0][t] + koff S[1, 0, 1][t] +
   koff S[1, 1, -1][t] + koff S[1, 1, 0][t] + koff S[1, 1, 1][t] +
   koff S[1, 2, -1][t] + koff S[1, 2, 0][t] + koff S[1, 2, 1][t],
 K[1, 2]′[t] == -a10 K[1, 2][t] kpase[1][t] + k9 K∘K[1, 1, 2, 2][t] +
   d10 K∘kpase[1, 2, 1][t] - pkon K[1, 2][t] S[-1, -1, -1][t] -
   pkon K[1, 2][t] S[-1, -1, 0][t] - pkon K[1, 2][t] S[-1, -1, 1][t] -
   pkon K[1, 2][t] S[-1, 0, -1][t] - pkon K[1, 2][t] S[-1, 0, 0][t] -
   pkon K[1, 2][t] S[-1, 0, 1][t] - pkon K[1, 2][t] S[-1, 1, -1][t] -
   pkon K[1, 2][t] S[-1, 1, 0][t] - pkon K[1, 2][t] S[-1, 1, 1][t] -
   pkon K[1, 2][t] S[-1, 2, -1][t] - pkon K[1, 2][t] S[-1, 2, 0][t] -
   pkon K[1, 2][t] S[-1, 2, 1][t] + koff S[2, -1, -1][t] +
   koff S[2, -1, 0][t] + koff S[2, -1, 1][t] +
   koff S[2, 0, -1][t] + koff S[2, 0, 0][t] + koff S[2, 0, 1][t] +
   koff S[2, 1, -1][t] + koff S[2, 1, 0][t] + koff S[2, 1, 1][t] +
   koff S[2, 2, -1][t] + koff S[2, 2, 0][t] + koff S[2, 2, 1][t],
```