Abstract Machines of Systems Biology

Luca Cardelli

Microsoft Research
Cambridge UK

2006-03-23 Max Planck ISS

www.luca.demon.co.uk
50 Years of Molecular Cell Biology

- **Genes are made of DNA**
  - Store digital information as sequences of 4 different nucleotides
  - Direct protein assembly through RNA and the Genetic Code

- **Proteins (>10000) are made of amino acids**
  - Process signals
  - Activate genes
  - Move materials
  - Catalyze reactions to produce substances
  - Control energy production and consumption

- **Bootstraping still a mystery**
  - DNA, RNA, proteins, membranes are today interdependent. Not clear who came first
  - Separation of tasks happened a long time ago
  - Not understood, not essential
Towards **Systems Biology**

- **Biologists now understand many of the cellular components**
  - A whole team of biologists will typically study a single protein for years
  - Reductionism: understand the components in order to understand the system

- **But this has not led to understand how “the system” works**
  - Behavior comes from complex patterns of interactions between components
  - Predictive biology and pharmacology still rare
  - Synthetic biology still unreliable

- **New approach: try to understand “the system”**
  - Experimentally: massive data gathering and data mining (e.g. Genome projects)
  - Conceptually: modeling and analyzing networks (i.e. interactions) of components

- **What kind of a system?**
  - Just beyond the basic chemistry of energy and materials processing...
  - Built right out of digital information (DNA)
  - Based on information processing for both survival and evolution
  - *Highly* concurrent

- **Can we fix it when it breaks?**
  - Really becomes: How is information structured and processed?
Eukaryotic Cell

(10~100 trillion in human body)

Membranes everywhere

H.Lodish et al. Molecular Cell Biology fourth edition p.1
Abstract Machines of Systems Biology

The “hardware” (biochemistry) is fairly well understood. But what is the “software” that runs on these machines?

- Gene Machine Nucleotides
  - Makes proteins, where/when/howmuch
  - Directs membrane construction and protein embedding
  - Holds genome(s), confines regulators
  - Model Integration Different time and space scales
  - Holds receptors, actuators, hosts reactions
  - Implants fusion, fission

- Protein Machine Aminoacids
  - Metabolism, Propulsion
  - Signal Processing
  - Molecular Transport
  - Functional Architecture Diverse
    - chemical toolkits
    - instruction sets
    - programming models
    - notations

- Membrane Machine Phospholipids
  - Confinement Storage
  - Bulk Transport
  - hierarchical multisets

Biochemical Networks, Transport Networks, glycans, strings, records, trees
Today we represent, store, search, and analyze:

- Gene sequence data
- Protein structure data
- Metabolic network data
- Signaling pathway data
- ...

How can we represent, store, and analyze biological processes?

- Scalable, precise, dynamic, highly structured, maintainable representations for systems biology.
- Not just huge lists of chemical reactions or differential equations.

In computing...

- There are well-established scalable representations of dynamic reactive processes.
- They look more or less like little, mathematically based, programming languages.
Reactive Systems

- **Modeling biological systems**
  - Not as continuous systems (often highly nonlinear)
  - But as discrete *reactive systems*; abstract machines where:
    - *States* represent situations
    - Event-driven *transitions* between states represent dynamics
  - The adequacy of describing (discrete) complex systems as reactive systems has been argued convincingly [Harel]

- **Many biological systems exhibit features of reactive systems:**
  - Discrete transitions between states
  - Deep layering of abstractions ("steps" at multiple levels)
  - Complexity from combinatorial interaction of simple components
  - High degree of concurrency and nondeterminism
  - "Emergent behavior" not obvious from part list

- **Still, needs quantitative semantics**
  - Stochastic, hybrid, etc. to talk about *rates* (and geometry).
Methods

- **Model Construction** *(writing things down precisely)*
  - Formalizing the notations used in systems biology.
  - Formulating description languages.
  - Studying their kinetics (semantics).

- **Model Validation** *(using models for postdiction and prediction)*
  - Simulation from compositional descriptions
    - Stochastic: quantitative concurrent semantics.
    - Hybrid: discrete transitions between continuously evolving states.
  - "Program" Analysis
    - Control flow analysis
    - Causality analysis
  - Model checking
    - Standard, Quantitative, Probabilistic
**Chemistry vs. \( \pi \)-calculus**

**A process calculus (chemistry)**

\[ \begin{align*}
\text{r: } & A + B \rightarrow_{k1} C + D \\
\text{s: } & C + D \rightarrow_{k2} A + B
\end{align*} \]

Does A become C or D?

**A different process calculus (\( \pi \))**

\[ \begin{align*}
A &= !r_{k1}; C \\
C &= ?s_{k2}; A \\
B &= ?r_{k1}; D \\
D &= !s_{k2}; B
\end{align*} \]

A Petri-Net-like representation. Precise and dynamic, but not modular, scalable, or maintainable.

A compositional graphical representation (precise, dynamic and modular) and the corresponding calculus.
Why $\pi$-calculus, in particular

- **Well studied, compact, precise, and general**
  - A “programming language” first, a mathematical model second
  - Semantics (reactions): $(P' + !n(m).P) | (Q' + ?n(m').Q) \rightarrow P | Q{m'\leftarrow m}$

- **Binary interactions**
  - I.e., “collisions”

- **Reactive and compositional**
  - Each subsystem is a separate (composition of) nondeterministic automata interacting with the environment (more automata)

- **Dynamic network evolution and species evolution**
  - Each subsystem can create fresh connections or spawn new subsystems

- **Compact description of combinatorics (like any programming language)**
  - $(\text{Bit}_1 | \text{Bit}_2 | \ldots | \text{Bit}_n)$ size-n description, where Bit is a 2-state subsystem
    - E.g. a protein with with $2^n$ phosphorylation configurations (i.e. “different chemical species”)

- **Complexation/polymerization**
  - The most characteristic feature of $\pi$-calculus (fresh names) models “sticking”
A quantitative variant of π-calculus:
- Channels have stochastic “firing” rates with exponential distribution.
- Nondeterministic choice becomes stochastic race.
- Cuts down to CTMCs (Continuous Time Markov Chains) in the finite case (not always). Then, standard analytical tools are applicable.
- Can be given friendly automata-like scalable graphical syntax (Andrew Phillips et al.).
- Is directly executable (e.g. via the Gillespie algorithm from physical chemistry).
- Is analyzable (large body of literature, at least in the non-stochastic case).

Figure 2. Regulating Gene Expression by Positive Feedback [9]

Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of TF A. Phillips, L. Cardelli. BioConcur’04.
The Rate of What?

- **In chemistry:**
  - Each reaction involves 2 molecules, and each reaction has a rate. Rates belong to reactions. Molecules do not have rates.

- **In process algebras:**
  - Should rates belong to:
    - each individual action? only outputs? delays only?
  - The rate of a synchronization of two actions should be the:
    - max? product? undefined if different? infinite (except for delays)?
  - All that has been tried.

- **We go back to chemistry**
  - Rates belong to channels. (This is called the "biochemical" stochastic π-calculus by Priami-Regev-Shapiro-Silverman)

- **Issues:**
  - Multiple activities on the same channel (concentrations of molecules involved in a reaction: mass action law of chemistry).
  - Choices between different channels (molecules involved in multiple reactions: still standard chemistry).
  - In biochemistry, rates of homodimerization (a molecule can interact with a copy of itself, but not with itself).
The speed of interaction \(\dagger\) is proportional to the number of possible interactions \(\ddagger\).

\[ d[C]/dt = -\lambda [C] \]
\[ d[D]/dt = \lambda [C] \]

\[ d[A]/dt = -\lambda [A] [B] \]
\[ d[B]/dt = -\lambda [A] [B] \]
\[ d[AB]/dt = \lambda [A] [B] \]

**Decay**
- Exponential Decay Law
  - Rate of change proportional to number of possible decays.

**Mass Interaction**
- Interaction Law generalizes Decay Law
  - Rate of change proportional to number of possible interactions

\(N\) interactions on the same channel are “faster” than \(N\) delays at the same rate (on \(N\) independent channels).

**Chemical Law of Mass Action**
The speed of a chemical reaction is proportional to the activity of the reacting substances.
(Activity = concentration, for well-stirred aqueous medium)
(Concentration = number of moles per liter of solution)
(Mole = \(6.022141\times10^{23}\) particles)
Activity and Apparent Rate
stochastic algebras disagree!

The speed of interaction is proportional to the number of possible interactions.

The mass interaction law [Buchholz] [Priami-Regev-Shapiro-Silverman] is compatible with chemistry [Gillespie] and incompatible with any other stochastic algebra in the literature! (including [Priami]; see [Hermanns])

Other algebras assign rates to actions, not channels, with apparent rates:
2\lambda \cdot 2\lambda = 4\lambda^2
\text{max}(2\lambda, 2\lambda) = 2\lambda \ [\text{Goetz}]
\text{min}(2\lambda, 2\lambda) = 2\lambda \ [\text{Priami}]
1/(1/(2\lambda)+1/(2\lambda)) = \lambda \ [\text{PEPA}]
2\lambda \cdot 1 = 2\lambda \ (\text{passive inputs})

Power Actions [Buchholz]

Activity = k
(similarly for ?^k a)

Can generalize k to a real number, and to a dynamically bound variables.
1. The Protein Machine

- **Complex folded-up shapes that:**
  - Fit together, dock, undock.
  - Excite/unexcite, warp each other.
  - Bring together, catalyze, transform materials.
  - Form complex aggregates and networks.

- **Mapping out such networks:**
  - In principle, it’s “just” a very large set of chemical equations.
  - Notations have been developed to summarize and abstract.

*An actual molecular interaction network.*

(Nodes are distinct protein kinds, arcs mean that two kinds of proteins interact.)
Protein Structure

Primary
The 20 Aminoacids
Tryptophan

Secondary
Alpha Helix, Beta Sheet

Tertiary
Green Fluorescent Protein

Quaternary
Triose Phosphate Isomerase

http://www.cmbi.kun.nl/gvteach/bioinformatica1/
Some Allosteric Switches

<table>
<thead>
<tr>
<th>Domain architecture</th>
<th>Repressed state</th>
<th>Activated state</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td><img src="image1" alt="Diagram" /></td>
<td><img src="image2" alt="Diagram" /></td>
</tr>
<tr>
<td>(b)</td>
<td><img src="image3" alt="Diagram" /></td>
<td><img src="image4" alt="Diagram" /></td>
</tr>
<tr>
<td>(c)</td>
<td><img src="image5" alt="Diagram" /></td>
<td><img src="image6" alt="Diagram" /></td>
</tr>
</tbody>
</table>

Domain architecture and autoinhibitory interactions in modular switch proteins. (a) Src family kinases contain N-terminal SH3 and SH2 domains, and a kinase domain flanked by intramolecular SH3-binding and SH2-binding sites (when the C-terminal motif tyrosine is phosphorylated by Csk). The crystal structures of several family members show that both intramolecular domain interactions function in concert to lock the kinase in an inactive conformation. Activating stimuli (red) include external SH2 or SH3 ligands. After initial activation, the kinase is maintained in an active state by autophosphorylation of its activation loop. (b) SHP-2 phosphatase contains two SH2 domains and a phosphatase domain. The crystal structure of the phosphatase shows that the N-terminal SH2 domain participates in an autoinhibitory interaction that directly blocks the phosphatase active site. Binding of external SH2 ligands activates by disrupting the autoinhibitory interaction. (c) N-WASP contains an Enabled Wiskott-Aldrich Syndrome Protein homology 1 (EVH1) domain, a B motif, a GBD, a proline-rich segment (pro) and an output region (VCA) that alone binds the Arp2/3 complex and stimulates its actin nucleation activity. The B and GBD motifs are required to repress activity and, by current models, are thought to participate in intracomplex interactions (only the structure of the GBD intramolecular complex for WASP is known). GTP-bound Cdc42 and PI(3,4,5)P3 synergistically activate N-WASP.

Allosteric ("other shape") reactions modify accessibility.

- **Kinase**
  - donates phosphate P
  - phosphorilates other proteins

- **Phosphatase**
  - accepts phosphate P
  - dephosphorilates other proteins

**Logical AND**

at equal concentrations of the individual input stimuli, activation is much higher if both stimuli are present

"Phosphatase Kinase Kinase" = a kinase that activates a kinase that activates a phosphatase that deactivates a protein.

Humans have the same number of modular protein domains (building blocks) as worms, but twice the number of multi-domain proteins.
**MIM: Molecular Interaction Maps (Kohn)**

The double-arrowed line indicates that proteins A and B can bind to each other. The "node" placed on the line represents the A:B complex.

Asymmetric binding where protein A donates a peptide that binds to a receptor site or pocket on protein B.

Representation of multimolecular complexes: x is A:B; y is (A:B):C. This notation is extensible to any number of components in a complex.

Covalet modification of protein A. The single-arrowed line indicates that A can exist in a phosphorylated state. The node represents the phosphorylated species.

Cleavage of a covalent bond: dephosphorylation of A by a phosphatase.

Proteolytic cleavage at a specific site within a protein.

Stoichiometric conversion of A into B.

Transport of A from cytosol to nucleus. The node represents A after it has been transported into the nucleus.

Formation of a homodimer. Filled circle on the right represents another copy of A. The node on the line represents the homodimer A:A.

z is the combination of states defined by x and y.

Enzymatic stimulation of a reaction.

General symbol for stimulation. A bar behind the arrowhead signifies necessity.

General symbol for inhibition.

Shorthand symbol for transcriptional activation.

Shorthand symbol for transcriptional inhibition.

Degradation products

*Taken from Kurt W. Kohn*
Molecular Interaction Maps

The p53-Mdm2 and DNA Repair Regulatory Network

The Protein Machine “Instruction Set”

cf. BioCalcules [Kitano&Nagasaki], κ-calculus [Danos&Laneve]

Each protein has a structure of binary switches and binding sites. But not all may be always accessible.

Switching of accessible switches.
- May cause other switches and binding sites to become (in)accessible.
- May be triggered or inhibited by nearby specific proteins in specific states.

Binding on accessible sites.
- May cause other switches and binding sites to become (in)accessible.
- May be triggered or inhibited by nearby specific proteins in specific states.

Each protein has a structure of binary switches and binding sites. But not all may be always accessible.

Switching of accessible switches.
- May cause other switches and binding sites to become (in)accessible.
- May be triggered or inhibited by nearby specific proteins in specific states.

Binding on accessible sites.
- May cause other switches and binding sites to become (in)accessible.
- May be triggered or inhibited by nearby specific proteins in specific states.
Notations for the Protein Machine

- **Stochastic \( \pi \)-Calculus**
  - Priami (following Hillston’s PEPA) formalizes a stochastic version of \( p \)-calculus where channels have communication rates.

- **BioSPI**
  - Regev-Shapiro-Silverman propose modeling chemical interactions (exchange of electrons and small molecules) as “communication”.
  - Standard stochastic simulation algorithms (Gillespie) can be used to run in-silico experiments.
  - Complex formation is encoded via \( p \)-restriction.

- **PEPA**
  - Calder Gilmore and Hillston model the ERK pathway.

- **\( k \)-calculus**
  - Danos and Laneve (following Kitano’s BioCalculation) define a calculus where complex formation is primitive.

- **(Stochastic) Petri Nets**
  - S.Reddy’94 modeling pathways.
  - Srivastava Perterson and Bentley analyze and simulate E.coli stress response circuit.

- **Bio State Charts**
  - Harel uses State Charts to model biological interactions via a semi-graphical FSM notation.

- **Pathway Logic**
  - Talcott-Eker-Knapp-Lincoln use term-rewriting.

- **BioCham**
  - Chabrier-Rivier-Fages-Soliman use term-rewriting and CLT modelchecking.

- **Kohn Diagrams, Kitano Diagrams**

- **SBML (Systems Biology Markup Language)**
  - XML dialect for MIM’s:
    - Compartments (statically nested)
    - Reagents with concentrations
    - Reactions with various rate laws
  - Read and written by many tools via the Systems Biology Workbench protocol
MAPK Cascade


Table 2. Predicted Hill coefficients for MAP kinase cascade components: Varying the assumed \( K_m \) values

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Range of assumed ( K_m ) values</th>
<th>MAPKK</th>
<th>MAPKK</th>
<th>MAPK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MAPKK → MAPKK*</td>
<td>60–1500 nM</td>
<td>1.0</td>
<td>1.7</td>
<td>4.9</td>
</tr>
<tr>
<td>2. MAPKK* → MAPKK</td>
<td>60–1500 nM</td>
<td>1.0</td>
<td>1.7</td>
<td>4.9</td>
</tr>
<tr>
<td>3. MAPKK → MAPKK-P</td>
<td>60–1500 nM</td>
<td>1.0</td>
<td>1.3–2.3</td>
<td>4.0–5.1</td>
</tr>
<tr>
<td>4. MAPKK-P → MAPKK</td>
<td>60–1500 nM</td>
<td>1.0</td>
<td>1.5–1.9</td>
<td>3.6–6.7</td>
</tr>
<tr>
<td>5. MAPKK-P → MAPKK-PP</td>
<td>60–1500 nM</td>
<td>1.0</td>
<td>1.2–2.4</td>
<td>3.8–5.2</td>
</tr>
<tr>
<td>6. MAPKK-PP → MAPKK-P</td>
<td>60–1500 nM</td>
<td>1.0</td>
<td>1.7–1.8</td>
<td>4.1–6.4</td>
</tr>
<tr>
<td>7. MAPK → MAPKK-P</td>
<td>60–1500 nM (300 nM)</td>
<td>1.0</td>
<td>1.7</td>
<td>3.7–6.2</td>
</tr>
<tr>
<td>8. MAPK → MAPKK</td>
<td>60–1500 nM</td>
<td>1.0</td>
<td>1.7</td>
<td>4.3–5.2</td>
</tr>
<tr>
<td>9. MAPK → MAPKK-PP</td>
<td>60–1500 nM</td>
<td>1.0</td>
<td>1.7</td>
<td>3.4–6.1</td>
</tr>
<tr>
<td>10. MAPK → MAPKK-P</td>
<td>60–1500 nM</td>
<td>1.0</td>
<td>1.7</td>
<td>4.7–5.1</td>
</tr>
</tbody>
</table>

The assumed \( K_m \) values for each reaction were individually varied over the ranges shown, with the assumed \( K_m \) values for the other nine reactions held constant. The effective Hill coefficients were calculated from the steepness of the predicted stimulus/response curves, as described in the text.

The \( K_m \) value for reaction 7 has been measured to be 300 nM for the phosphorylation of a mammalian MAPK by a MAPKK (N. Ahn, personal communication). All of the other \( K_m \) values were initially assumed to be 300 nM as well.

**Fig. 1.** Schematic view of the MAPK cascade. Activation of MAPK depends upon the phosphorylation of two conserved sites [Thr-183 and Tyr-185 in rat p42 MAPK/Erk2 (4, 5)]. Full activation of MAPKK also requires phosphorylation of two sites [Ser-218 and Ser-222 in mouse Mek-1/MKK1 (6–10)]. Detailed mechanisms for the activation of various MAPKKs (e.g., Raf-1, B-Raf, Mos) are not yet established; here we assume that MAPKKs are activated and inactivated by enzymes we denote E1 and E2. MAPKK* denotes activated MAPKK. MAPKK-P and MAPKK-PP denote singly and doubly phosphorylated MAPKK, respectively. MAPK-P and MAPK-PP denote singly and doubly phosphorylated MAPK. P'ase denotes phosphatase.
As 18 Ordinary Differential Equations
Plus 7 conservation equations

\[ \frac{d}{dt} [KKK] = -a_4[KKK][E1] + 6a_6[KKK-E1] \\
+ k_1[KKK^* - E2] \quad (11) \]

\[ \frac{d}{dt} [KKK-E1] = a_2[KKK][E1] - (d_1 + k_1)[KKK-E1] \quad (12) \]

\[ \frac{d}{dt} [KKK^*] = -a_4[KKK][E1] + 6a_6[KKK-E1] \\
+ k_1[KKK^* - E2] + (d_1 + k_1)[KKK-E1] \\
+ k_2[KKK^* - KK-P] - a_4[KKK^*][KKK] \quad (13) \]

\[ \frac{d}{dt} [KKK^* - E2] = a_4[KKK]^* [E2] - (d_1 + k_1)[KKK^* - E2] \quad (14) \]

\[ \frac{d}{dt} [KK] = -a_1[KK][KKK^*] + 6a_6[KKK-E1] \\
+ k_1[KK-P] + k_2[KK-P] \quad (15) \]

\[ \frac{d}{dt} [KK-KK^*] = a_4[KK][KKK^*] \\
+ (d_1 + k_1)[KK-KK^*] \quad (16) \]

\[ \frac{d}{dt} [KK-P] = -a_1[KK][KK-P] + 6a_6[KKK-E1] \\
+ k_1[KK-P-KK^*] + k_2[KK-P-KK^*] \\
+ a_4[KK-P][KKK^*] \quad (17) \]

\[ \frac{d}{dt} [KK-P-KK^*] = -a_1[KK-P][KK^*] \\
+ (d_1 + k_1)[KK-P-KK^*] \quad (18) \]

\[ \frac{d}{dt} [KK-P-KK^*] = a_4[KK-P][KKK^*] \\
+ (d_1 + k_1)[KK-P-KK^*] \quad (19) \]

\[ \frac{d}{dt} [KK-P-KK^*] = a_4[KK-P][KKK^*] \\
+ (d_1 + k_1)[KK-P-KK^*] \\
+ a_4[KK-P][KKK^*] \quad (20) \]

\[ \frac{d}{dt} [KK-P-KK^*] = a_4[KK-P][KKK^*] \\
+ (d_1 + k_1)[KK-P-KK^*] \quad (21) \]

\[ \frac{d}{dt} [KK] = -a_1[KK][KKK^*] + 6a_6[KKK-E1] \\
+ k_1[KK-P-KK^*] \quad (22) \]

\[ \frac{d}{dt} [KK-P] = a_4[KK-P][KKK^*] - (d_1 + k_1)[KK-P-KK^*] \quad (23) \]

\[ \frac{d}{dt} [KK-P-KK^*] = a_4[KK-P][KKK^*] - (d_1 + k_1)[KK-P-KK^*] \quad (24) \]

\[ \frac{d}{dt} [KK-P-KK^* - K-P] = a_4[KK-P][KKK^*] - (d_1 + k_1)[KK-P-KK^*] \quad (25) \]

\[ \frac{d}{dt} [KK-P-KK^* - K-P] = a_4[KK-P][KKK^*] - (d_1 + k_1)[KK-P-KK^*] \quad (26) \]

\[ \frac{d}{dt} [KK-P-KK^* - K-P] = a_4[KK-P][KKK^*] - (d_1 + k_1)[KK-P-KK^*] \quad (27) \]

\[ \frac{d}{dt} [KK-P-KK^* - K-P] = a_4[KK-P][KKK^*] - (d_1 + k_1)[KK-P-KK^*] \quad (28) \]

The 10 reactions described above give rise to 18 rate equations.

One equation for each species (8) and complex (10), but not for constant concentration enzymes (4).

In addition, there are seven conservation equations (Eqs. 29-35).

\[ [KKK_{tot}] = [KKK] + [KKK^*] + [KKK-E1] \]

\[ + [KKK^* \cdot E2] \]

\[ + [KKK^* \cdot K] + [KKK^* \cdot K-P] \quad (29) \]

These equations were solved numerically using the Rump–Kutta-based NDSolve algorithm in Mathematica (Wolfram Research, Champaign, II). An annotated copy of the Mathematica code for the MAPK cascade rate equations can be obtained from J.E.F.
The Circuit

E1

KKK \xrightarrow{E2} KKK^* \xrightarrow{KK-P'ase} KK-P \xrightarrow{KK-P'ase} KK-PP \xrightarrow{KK-P'ase} K-PP

K \xrightarrow{K-P'ase} K-P \rightarrow K-PP

(output)
Enzymatic Reactions

**Reaction View**

\[ \begin{align*}
S & \rightarrow P \\
E & \rightarrow E+S \\
E+S & \rightarrow ES \\
ES & \rightarrow P+E
\end{align*} \]

**Interaction View**

\[ \begin{align*}
S() & \triangleq \text{new } u@d \text{ new } k@e \\
& \quad !a_c(u,k); (!u_d; S() + !k_e; P()) \\
E() & \triangleq \ ?a_c(u,k); (?u_d; E() + ?k_e; E()) \\
P() & \triangleq \ ...
\end{align*} \]
As 12 processes (in SPiM)

let KKK() =
  (new u1@d1:Release new k1@r1:React
   !a1(u1,k1); (do !u1;KKK() or !k1;KKKst()))
and KKKst() =
  (new u2@d2:Release new k2@r2:React
   do !a2(u2,k2); (do !u2;KKKst() or !k2;KKK())
   or ?a3(u3,k3); (do ?u3;KKKst() or ?k3;KKKst())
   or ?a5(u5,k5); (do ?u5;KKKst() or ?k5;KKKst()))

let E1() =
  ?a1(u1,k1); (do ?u1;E1() or ?k1;E1())

let E2() =
  ?a2(u2,k2); (do ?u2;E2() or ?k2;E2())

let KK() =
  (new u3@d3:Release new k3@r3:React
   !a3(u3,k3); (do !u3;KK() or !k3;KK_P())))
and KK_P() =
  (new u4@d4:Release new k4@r4:React
   do !a4(u4,k4); (do !u4;KK_P() or !k4;KK_P())
   or ?a7(u7,k7); (do ?u7;KK_P() or ?k7;KK_PP())
   or ?a9(u9,k9); (do ?u9;KK_P() or ?k9;KK_PP()))

let KKPse() =
  do ?a4(u4,k4); (do ?u4;KKPse() or ?k4;KKPse())

let K() =
  (new u7@d7:Release new k7@r7:React
   !a7(u7,k7); (do !u7;K() or !k7;K_P())))
and K_P() =
  (new u8@d8:Release new k8@r8:React
   new u9@d9:Release new k9@r9:React
   do !a8(u8,k8); (do !u8;K_P() or !k8;K())
   or !a9(u9,k9); (do !u9;K_P() or !k9;K_PP()))

and K_PP() =
  (new u10@d10:Release new k10@r10:React
   !a10(u10,k10); (do !u10;K_PP() or !k10;K_P())))
and KPse() =
  do ?a8(u8,k8); (do ?u8;KPse() or ?k8;KPse())

One process for each component (12) including enzymes, but not for complexes.

No need for conservation equations: implicit in “choice” operator in the calculus.
... and 30 Interaction Channels

type Release = chan()

type React = chan()

type Bond = chan(Release, React)

new a1@1.0:Bond val d1=1.0 val r1=1.0
new a2@1.0:Bond val d2=1.0 val r2=1.0
new a3@1.0:Bond val d3=1.0 val r3=1.0
new a4@1.0:Bond val d4=1.0 val r4=1.0
new a5@1.0:Bond val d5=1.0 val r5=1.0
new a6@1.0:Bond val d6=1.0 val r6=1.0
new a7@1.0:Bond val d7=1.0 val r7=1.0
new a8@1.0:Bond val d8=1.0 val r8=1.0
new a9@1.0:Bond val d9=1.0 val r9=1.0
new a10@1.0:Bond val d10=1.0 val r10=1.0

... 

run 100 of KKK() run 100 of KK() run 100 of K()
run 1 of E2() run 1 of KKPse() run 1 of KPse()
run 1 of E1()
MAPK Cascade Simulation in SPiM

1st stage:
KKK* barely rises

2nd stage:
KK-PP rises, but is not stable

3rd stage:
K-PP flips up to max even anticipating 2nd stage

Rates and concentrations from paper:

1xE2 (0.3 nM)
1xKKPase (0.3 nM)
120xK (120 nM)
3xKKK (3 nM)
1200xKK (1.2 uM)
1200xK (1.2 uM)

dx = rx = 150, ax = 1
(KmK = (dx + rx) / ax, Km = 300 nM)

1xE1 injected
MAPK Cascade Simulation in SPiM

All coefficients 1.0 !!!
100xKKK, 100xKK, 100xK,
13xE2, 13xKKPse, 13xKPse.
nxE1 as indicated
(1xE1 is not sufficient to produce an output)
MAPK Cascade Simulation in SPiM

1st stage:
KKK* barely rises

2nd stage:
KK-PP rises, but is not stable

3rd stage:
K-PP flips up to max
even anticipating 2nd stage

All coefficients 1.0 !!!
100xKKK, 100xKK, 100xK,
5xE2, 5xKKPse, 5xKPse.

Input is 1xE1.
Output is 90xK-PP (ultrasensitivity).
2. The Gene Machine

The “Central Dogma” of Molecular Biology

4-letter digital code
4-letter digital code
20-letter digital code
50.000(?) shapes

regulation
transcription
translation
interaction

Taken from Pedro Mendes
DNA Tutorial

Metabolic space
Protein space
Gene space
Regulation of a gene (positive and negative) influences transcription. The regulatory region has precise DNA sequences, but not meant for coding proteins: meant for binding regulators.

Transcription produces molecules (RNA or, through RNA, proteins) that bind to regulatory region of other genes (or that are end-products).

**Human (and mammalian) Genome Size**
- 3Gb (Giga base pairs) @ 750MB @ 4bp/Byte (CD)
- Non-repetitive: 1Gb 250MB
- In genes: 320Mbp 80MB
- Coding: 160Mbp 40MB
- Protein-coding genes: 30,000-40,000

**M.Genitalium (smallest true organism)**
- 580,073bp 145KB (eBook)

**E.Coli (bacteria):**
- 4Mbp 1MB (floppy)

**Yeast (eukarya):**
- 12Mbp 3MB (MP3 song)

**Wheat**
- 17Gbp 4.25GB (DVD)
Gene Composition

Under the assumptions [Kim & Tidor]
1) The solution is well-stirred
   (no spatial dependence on concentrations or rates).
2) There is no regulation cross-talk.
3) Control of expression is at transcription level only
   (no RNA-RNA or RNA-protein effects)
4) Transcriptions and translation rates monotonically
   affect mRNA and protein concentrations resp.

Ex: Bistable Switch

Ex: Oscillator

Expressed
Repressed
Expressing
Indirect Gene Effects

No combination of standard high-throughput experiments can reconstruct an a-priori known gene/protein network [Wagner].

One of many bistable switches that cannot be described by pure gene regulatory networks [Francois & Hakim].
RNA is not just an intermediary; it can:
- Fold-up like a protein
- Act like an enzyme
- Regulate other transcribed RNA
- Direct protein editing
- ...

97-98% of the transcriptional output of the human genome is non-protein-coding RNA.
30-40,000 “protein genes” (1.5% of genome)
60-100,000 “transcription units” (>30% of genome is transcribed)
Structure of a Regulatory Region

Proteins

DNA

Protein binding sites

DNA Sequence

2300bp!

> average protein

Fig. 1. *Endo16* cis-regulatory system and interactive roles of module A. (A) Diversity of protein binding sites and organization into modular subregions [modified from (7)]. Specific DNA binding sites are indicated as red blocks; modular subregions are denoted by letters G to A (Bp, basal promoter). Proteins binding at the target sites considered in this work are indicated: Otx, SpOtx-1 (12); SpGCF1 (14); the proteins C, Z, and P, which are not yet cloned; and protein C [a CREB family protein (18)] in subregion F. Proteins for which sites occur in multiple regions of the DNA sequence (indicated by the black line) are shown beneath. (B) Sequence of module A and location of protein binding sites. Sites are indicated in the same colors as in (A). A fragment containing CG5 and CG4 sites as well as Bp has no endoderm-specific activity and serves other upstream cis-regulatory systems promiscuously; similarly, the *Endo16* cis-regulatory system functions specifically with heterologous promoters substituted for Bp (5, 8, 19). Boxed sequences indicate conserved core elements of the target sites (7, 12, 14), not the complete target site sequences. (C) Integrative and interactive functions of module A (8, 18). Module A communicates the output of all upstream modules to the basal transcription apparatus. It also initiates endoderm expression, increases the output of modules B and C, and is required for functions of the upstream modules E, F, and DC. These functions are repression of expression in nonendodermal domains and enhancement of expression in response to Lh3.
Function of a Regulatory Region

B
If \((F = 1 \text{ or } E = 1 \text{ or } CD = 1) \text{ and } (Z = 1)\)
\(\alpha = 1\)
else \(\alpha = 0\)
if \((P = 1 \text{ and } CG_1 = 1)\)
\(\beta = 2\)
else \(\beta = 0\)
if \((CG_2 = 1 \text{ and } CG_3 = 1 \text{ and } CG_4 = 1)\)
\(\gamma = 2\)
else \(\gamma = 1\)
\(\delta(t) = B(t) + G(t)\)
\(\epsilon(t) = \beta \cdot \delta(t)\)
if \((\epsilon(t) = 0)\)
\(\zeta(t) = \text{Otx}(t)\)
else \(\xi(t) = o(t)\)
If \(\alpha = 1\)
\(\eta(t) = 0\)
else \(\eta(t) = \xi(t)\)
\(\xi(t) = \gamma \cdot \eta(t)\)
Final output communicated to BTA

Gene Regulatory Networks

http://strc.herts.ac.uk/bio/maria/NetBuilder/


The Programming Model

- **Strange facts about genetic networks:**
  - **Not an operator algebra.** The output of each gate is fixed and pre-determined; it is never a function of the input!
  - **Not term-rewriting, nor Petri nets.** Inhibition is widespread.
  - **Not Communicating Sequential Processes.** Feedback is widespread: asynchronous communication needed to avoid immediate self-deadlocks. Even the simplest gates cannot be modeled as a single synchronous automata.
  - **Not Message-Passing between genes.** Messages themselves have behavior (e.g., they stochastically decay and combine), hence messages are processes as well.
  - **Not Data-Flow.** Any attempt to use data-flow-style modeling seems doomed because of widespread loops that lead to deadlocks or unbounded queues. Data-flow tokens do not “decay” like proteins.

- **How can it possibly work?**
  - **Stochastic broadcasting.** The apparently crude idea of broadcasting a whole bunch of asynchronous decaying messages to activate a future gate, means there are never any “pipeline full” deadlocks, even in presence of abundant feedback loops.
  - **Stochastic degradation.** Degradation is fundamental for system stability, and at the same time can lead to sudden instability and detection of concentration levels.
Notations for the Gene Machine

- Many of the same techniques as for the Protein Machine apply.
  - Process Calculi, Petri Nets, Term-Rewriting Systems...

- But the “programming model” is different.
  - Asynchronous stochastic control.
  - Biologically poorly understood.
  - Network “motifs” are being analyzed.

- Specific techniques:
  - Hybrid Petri Nets
    - [Matsuno, Doi, Nagasaki, Miyano] Gene Regulation
    - Genomic Object Net www.genomicobject.net

- Gene Regulation Diagrams

- Mixed Gene-Protein Diagrams
Gene Gates and Circuits

A gene gate

\[ \text{neg}(a, b) \triangleq \quad ?a; \tau_\eta; \text{neg}(a, b) + \tau_\epsilon; (\text{tr}(b) \mid \text{neg}(a, b)) \]
\[ \text{tr}(p) \triangleq (\lnot p_r; \text{tr}(p)) + \tau_\delta \]

A genetic circuit (engineered in E.Coli)

\[ \text{neg}(a, b) \mid \text{neg}(b, c) \mid \text{neg}(c, a) \]

The stochastic-\(\pi\) program

\[
\begin{align*}
\text{val } dk &= 0.001 \quad (* \text{ Decay rate } *) \\
\text{val } inh &= 0.001 \quad (* \text{ Inhibition rate } *) \\
\text{val } cst &= 0.1 \quad (* \text{ Constitutive rate } *) \\
\text{let } \text{tr}(p: \text{chan}) &= \\
& \quad \text{do } !p; \text{tr}(p) \text{ or delay}\text{dk} \\
\text{let } \text{neg}(a: \text{chan}, b: \text{chan}) &= \\
& \quad \text{do } ?a; \text{delay}\text{inh}; \text{neg}(a, b) \text{ or delay}\text{cst}; (\text{tr}(b) \mid \text{neg}(a, b)) \\
\end{align*}
\]

(* The circuit *)
\[
\begin{align*}
\text{val } bnd &= 1.0 \quad (* \text{ Protein binding rate } *) \\
\text{new } a@\text{bnd: chan} &\text{ new } b@\text{bnd: chan} \text{ new } c@\text{bnd: chan} \text{ run } (\text{neg}(c, a) \mid \text{neg}(a, b) \mid \text{neg}(b, c))
\end{align*}
\]

A stochastic simulation (in SPiM)

\(r=1.0, \epsilon=0.1, h=0.001, \delta=0.001\)
3. The Membrane Machine

Molecular transport and transformation through dynamic compartment fusion and fission.

Well, what is all that for?

“Given the complicated pathways that have evolved to synthesize them, it seems likely that these modified proteins have important functions, but for the most part these functions are not known” [MBC p.609]
Membrane Fusion

Positive curvature to Negative curvature transition in 3D

Proposed sequence of events in pH sensitive hemagglutinin membrane fusion

Aggressive fusion (virus)

By unknown mechanisms, the exoplasmic leaflets of the two membranes fuse” [MCB p745]

Cooperative fusion (vesicle)

“Fusion of the two membranes immediately follows prefusion, but precisely how this occurs is not known” [MCB p742]
Membrane Fission

Assembly and disassembly of the clathrin coat

Vesicle Formation

“Nonetheless, the actual process whereby a segment of phospholipid bilayer is ‘pinched off’ to form a pit and eventually a new vesicle is still not understood” [MCB p.746]

Vesicle Formation

Cytokinesis (Mitosis)

Movie by Allison Bruce
The Membrane Machine “Instruction Set”

Arbitrary subsystem

Mate

Mito

Pino

Phago

Arbitrary subsystem

Zero case

One case

P

Q

Mate

Mito

Zero case

Drip

Bud

Mito: special cases

P

Q

Fusion

Fission

P

Q

Endo: special cases

Pino

Phago

Endo

Arbitrary subsystem

Fusion

Fission

P

Q

Fusion

Fission

P

Q

P

R

One case

P

R

Exo

Endo

One case

Zero case
... in 3D

S-Exo  Fission
S-Endo  Fission
T-Exo  Fission
T-Endo  Fusion
S-Mito  Fission
S-Mate  Fusion
T-Mito  Fusion
T-Mate  Fission
Locally Implementable!

Global Views

Mito (Fission)
Mate (Fusion)
Endo (Fission)
Exo (Fusion)

Same Local View!
Mito/Mate by 3 Endo/Exo

[Diagram showing the process of Mito/Mate by 3 Endo/Exo with labeled arrows indicating Endo and Exo.]
Notations for the Membrane Machine

- **“Snapshot” diagrams**
  - In biology literature.

- **P-Systems**
    http://psystems.disco.unimib.it/.

- **BioAmbients**
  - An extension of BioSPI along Ambient Calculus lines (with more bio-relevant mobility primitives) to model dynamic compartments.

- **Brane Calculi**
  - Computation on the membrane...
Membrane Algorithms

Protein Production and Secretion

Viral Replication

LDL-Cholesterol Degradation


Abstract Machines of Systems Biology

The “hardware” (biochemistry) is fairly well understood. But what is the “software” that runs on these machines?

Functional Architecture
- chemical toolkits
- instruction sets
- programming models
- notations

Biochemical Networks
- Metabolism, Propulsion
- Signal Processing
- Molecular Transport

Gene Machine
- Nucleotides
- Holds genome(s), confines regulators
- Model Integration
  - Different time and space scales
- Directs membrane construction and protein embedding
- Holds receptors, actuators, hosts reactions

Protein Machine
- Aminoacids
- Makes proteins, where/when/howmuch
- Signals conditions and events

Membrane Machine
- Phospholipids
- Implements fusion, fission
- Confinement, Storage, Bulk Transport

Transport Networks

Gene Regulatory Networks

Surface and Extracellular Features
- Glycan Machine
  - Sugars
- Glycan Machine
  - Glycans

Glycan Machine
- Surface and Extracellular Features

The function of these machines is to integrate different time and space scales, confine genome(s), and implement fusion and fission.
Conclusions

**Q:** “The data are accumulating and the computers are humming, what we are lacking are the words, the grammar and the syntax of a new language...”

D. Bray (TIBS 22(9):325-326, 1997)

**A:** “The most advanced tools for computer process description seem to be also the best tools for the description of biomolecular systems.”

E. Shapiro (Lecture Notes)
References


Papers

BioAmbients
a stochastic calculus with compartments.

Brane Calculi
process calculi with computation “on” the membranes, not inside them.

Bitonal Systems
membrane reactions and their connections to “local” patch reactions.

Abstract Machines of Systems Biology
the abstract machines implemented by biochemical toolkits.

www.luca.demon.co.uk/BioComputing.htm