Abstract Machines of Systems Biology

Luca Cardelli
Microsoft Research
Cambridge UK

2005-06-03 Lausanne

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

• Bootstrapping 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
  - When each component and each reaction is understood, the system is understood (?)

- But this has not led to understand how “the system” works
  - Behavior comes from complex chains 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

- Can we fix it when it breaks?
  - The question really becomes: How is information structured and processed?
Structural Architecture

Eukaryotic Cell
(10~100 trillion in human body)
Membranes everywhere

Nuclear membrane
Mitochondria
Golgi
Vesicles
E.R.
Plasma membrane (<10% of all membranes)

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

Regulation

Gene Regulatory Networks

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

Protein Machine
Aminoacids

Metabolism, Propulsion
Signal Processing
Molecular Transport

Membrane Machine
Phospholipids

Confinement
Storage
Bulk Transport

Biochemical Networks

Transport Networks

Glycan Machine
Surface and Extracellular Features

Model Integration
Different time and space scales

Holds receptors, actuators hosts reactions

Directs membrane construction and protein embedding

Holds genome(s), confines regulators

Makes proteins, where/when/howmuch

Signals conditions and events

P Q

A x y

C P

A

B

C

D


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

Secondary

Tryptophan

Alpha Helix, Beta Sheet

Tertiary

Green Fluorescent Protein

Quaternary

Triose Phosphate Isomerase

http://www.cmbi.kun.nl/gvteach/bioinformatica1/
Protein Function

- Regulation
- Degradation
- Metabolism
- Movement
- Assembly
- Transport
- Structure
- Signalling
Some Allosteric Switches

<table>
<thead>
<tr>
<th>Domain architecture</th>
<th>Repressed state</th>
<th>Activated state</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) Input SH3- SH2→ Output Kinase</td>
<td><a href="#">Diagram</a></td>
<td><a href="#">Diagram</a></td>
</tr>
<tr>
<td>(b) Input SH2→ Output Phosphatase</td>
<td><a href="#">Diagram</a></td>
<td><a href="#">Diagram</a></td>
</tr>
<tr>
<td>(c) Input EVH1-GBD-pro</td>
<td><a href="#">Diagram</a></td>
<td><a href="#">Diagram</a></td>
</tr>
</tbody>
</table>

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.

---

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 Ena/VASP 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 intramolecular interactions (only the structure of the GBD intramolecular complex for WASP is known). GTP-bound Cdc42 and FIP2 synergistically activate N-WASP.

[Diagram](#)
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 multicomponent complexes: \( x \) is \( A:B \); \( y \) is \( (A:B):C \). This notation is extensible to any number of components in a complex.

Covariant 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 covariant bond: dephosphorylation of \( A \) by a phosphatase.

Proteolytic cleavage at a specific site within a protein.
The Protein Machine “Instruction Set”

On/Off switches

Protein

Binding Sites

Inaccessible

Inaccessible

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

cf. BioCalculus [Kitano&Nagasaki], $\kappa$-calculus [Danos&Laneve]

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 π-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 BioCalculus) 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**
  - ChabrierRivier-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 )</th>
<th>MAPKK</th>
<th>MAPKK*</th>
<th>MAPK</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MAPKKE ( \rightarrow ) MAPKK*</td>
<td>60-1500 nM</td>
<td>1.0</td>
<td>1.7</td>
<td>4.9</td>
</tr>
<tr>
<td>2. MAPKK* ( \rightarrow ) MAPKK</td>
<td>60-1500 nM</td>
<td>1.0</td>
<td>1.7</td>
<td>4.9</td>
</tr>
<tr>
<td>3. MAPKK ( \rightarrow ) 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 ( \rightarrow ) 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 ( \rightarrow ) MAPK-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. MAPK-PP ( \rightarrow ) 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 ( \rightarrow ) MAPKK</td>
<td>60-1500 nM (300 nM(^3))</td>
<td>1.0</td>
<td>1.7</td>
<td>3.7-6.2</td>
</tr>
<tr>
<td>8. MAPKK ( \rightarrow ) MAPKK-P</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-P ( \rightarrow ) MAPKK-P</td>
<td>60-1500 nM</td>
<td>1.0</td>
<td>1.7</td>
<td>3.4-6.1</td>
</tr>
<tr>
<td>10. MAPKK-P ( \rightarrow ) MAPK-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

\[ \frac{d}{dt} [K] = -a_{[K]} [K] [E1] + b_{i} [KKK^* - E2] \quad (11) \]

\[ \frac{d}{dt} [KKK^* - E2] = a_{1} [KKK^* - E2] - (d_i + k_i) [KKK^* - E2] \quad (12) \]

\[ \frac{d}{dt} [KKK] = -a_{[KK]} [KK] [E1] + b_{i} [KKK] [KKK^* - E2] + b_{i} [KKK^* - E2] + k_{i} [KK] [KKK^* - E2] - a_{[KK]} [KK] [KKK^* - E2] \quad (13) \]

\[ \frac{d}{dt} \Delta [KKK^* - E2] = \Delta a_{[KKK^* - E2]} [KKK^* - E2] - (d_i + k_i) [KKK^* - E2] \quad (14) \]

\[ \frac{d}{dt} [KKK^* - E2] = a_{i} [KKK^* - E2] - (d_i + k_i) [KKK^* - E2] \quad (15) \]

\[ \frac{d}{dt} [K] = -a_{[KK]} [KKK] [KKK^* - E2] + k_{i} [K] [KKK^* - E2] - (d_i + k_i) [KKK^* - E2] \quad (16) \]

\[ \frac{d}{dt} [KK + K*K - KPP] = a_{i} [KK + K*K - KPP] - (d_{j} + k_{j}) [KK + K*K - KPP] \quad (17) \]

\[ \frac{d}{dt} [KPP + KK + K*K - KPP] = a_{i} [KPP + KK + K*K - KPP] - (d_{j} + k_{j}) [KPP + KK + K*K - KPP] \quad (18) \]

\[ \frac{d}{dt} [KKK] = a_{i} [KKK + K*K - KPP] - (d_{j} + k_{j}) [KKK + K*K - KPP] \quad (19) \]

\[ \frac{d}{dt} [KKK + K*K - KPP] = a_{i} [KKK + K*K - KPP] + d_{i} [KKK + K*K - KPP] - a_{i} [KKK + K*K - KPP] \quad (20) \]

\[ \frac{d}{dt} [KKK + K*K - KPP] = a_{i} [KKK + K*K - KPP] + d_{i} [KKK + K*K - KPP] - a_{i} [KKK + K*K - KPP] \quad (21) \]

\[ \frac{d}{dt} [KPP + KK + K*K - KPP] = a_{i} [KPP + KK + K*K - KPP] - (d_{j} + k_{j}) [KPP + KK + K*K - KPP] \quad (22) \]

\[ \frac{d}{dt} [KKK + K*K - KPP] = a_{i} [KKK + K*K - KPP] + d_{i} [KKK + K*K - KPP] - a_{i} [KKK + K*K - KPP] \quad (23) \]

\[ \frac{d}{dt} [KPP + KK + K*K - KPP] = a_{i} [KPP + KK + K*K - KPP] - (d_{j} + k_{j}) [KPP + KK + K*K - KPP] \quad (24) \]

\[ \frac{d}{dt} [KPP + KK + K*K - KPP] = a_{i} [KPP + KK + K*K - KPP] + d_{i} [KPP + KK + K*K - KPP] - a_{i} [KPP + KK + K*K - KPP] \quad (25) \]

\[ \frac{d}{dt} [KPP + KK + K*K - KPP] = a_{i} [KPP + KK + K*K - KPP] + d_{i} [KPP + KK + K*K - KPP] - a_{i} [KPP + KK + K*K - KPP] \quad (26) \]

\[ \frac{d}{dt} [KPP + KK + K*K - KPP] = a_{i} [KPP + KK + K*K - KPP] + d_{i} [KPP + KK + K*K - KPP] - a_{i} [KPP + KK + K*K - KPP] \quad (27) \]

\[ \frac{d}{dt} [KPP + KK + K*K - KPP] = a_{i} [KPP + KK + K*K - KPP] + d_{i} [KPP + KK + K*K - KPP] - a_{i} [KPP + KK + K*K - KPP] \quad (28) \]

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).

\[ [KKK_{tot}] = [KKK] + [KKK^*] + [KKK \cdot E1] + [KKK^* \cdot E2] + [KKK^* \cdot K] + [KKK^* \cdot K^*] + [KKK^* \cdot K^* \cdot K - P] \quad (29) \]

These equations were solved numerically using the Runga–Kutta-based NDSoSolve algorithm in Mathematica (Wolfram Research, Champaign, IL). An annotated copy of the Mathematica code for the MAPK cascade rate equations can be obtained from J.E.F.
As 12 processes (in SPiM)

let KKK() =
(new u1@d1:Release !a1(u1); (do !u1;KKK() or !k1;KKKst()))

and KKKst() =
(new u2@d2:Release
do !a2(u2); (do !u2;KKKst() or !k2;KKK())
or ?a3(u3); (do ?u3;KKKst() or ?k3;KKKst())
or ?a5(u5); (do ?u5;KKKst() or ?k5;KKKst()))

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

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

let KK() =
(new u3@d3:Release !a3(u3); (do !u3;KK() or !k3;KK_P()))

and KK_P() =
(new u4@d4:Release new u5@d5:Release
do !a4(u4); (do !u4;KK_P() or !k4;KK_P())
or !a5(u5); (do !u5;KK_P() or !k5;KK_P()))

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

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

let KK() =
(new u3@d3:Release !a3(u3); (do !u3;KK() or !k3;KK_P()))

and KK_P() =
(new u6@d6:Release
do !a6(u6); (do !u6;KK_P() or !k6;KK_P())
or ?a7(u7); (do ?a7;KK_P() or ?k7;KK_P())
or ?a9(u9); (do ?u9;KK_P() or ?k9;KK_P()))

let K() =
(new u7@d7:Release !a7(u7); (do !u7;K() or !k7;K_P()))

and K_P() =
(new u8@d8:Release new u9@d9:Release
do !a8(u8); (do !u8;K_P() or !k8;K_P())
or !a9(u9); (do !u9;K_P() or !k9;K_PP()))

and K_PP() =
(new u10@d10:Release !a10(u10); (do !u10;K_PP() or !k10;K_PP()))

and KPse() =
d0 ?a8(u8); (do ?u8;KPse() or ?k8;KPse())
or ?a10(u10); (do ?u10;KPse() or ?k10;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.
type Release = chan()
type Bond = chan(Release)
type React = chan()

new a1@1.0:Bond val d1=1.0 new k1@1.0:React
new a2@1.0:Bond val d2=1.0 new k2@1.0:React
new a3@1.0:Bond val d3=1.0 new k3@1.0:React
new a4@1.0:Bond val d4=1.0 new k4@1.0:React
new a5@1.0:Bond val d5=1.0 new k5@1.0:React
new a6@1.0:Bond val d6=1.0 new k6@1.0:React
new a7@1.0:Bond val d7=1.0 new k7@1.0:React
new a8@1.0:Bond val d8=1.0 new k8@1.0:React
new a9@1.0:Bond val d9=1.0 new k9@1.0:React
new a10@1.0:Bond val d10=1.0 new k10@1.0:React

... 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()

\(a_i, k_i\): Two channels for each reversible chemical reaction of 2 molecules.
(No behavior attached to channels except interaction rate.)
MAPK Cascade Simulation in SPiM

All coefficients 1.0 !!!
100xKKK, 100xKK, 100xK,
1xE2, 1xKKPse, 1xKPse.
Input is 1xE1.
Output is 100xK-PP (ultrasensitivity).

KKK* reacts mildly to E1
KK-PP rises quicker
K-PP flips up to 100!

1xE1 injected

K-KK

E2 (input)

KKK → KKK*

KK → KK-P

KK-ase → KK-PP

K → K-P

K-P'ase → K-PP

(output)
MAPK Cascade Simulation in SPiM

All coefficients 1.0!!!
100xKKK, 100xKK, 100xK,
10xK, 10xKP, 10xKPse.
(so 1xE1 is no longer sufficient to produce an output)
2. The Gene Machine

The “Central Dogma” of Molecular Biology

- **DNA**
  - 4-letter digital code
  - regulation

- **messenger RNA**
  - 4-letter digital code
  - transcription

- **PROTEIN**
  - 20-letter digital code
  - translation

- **SYSTEMS**
  - 50,000 (?) shapes
  - interaction

---

**Lactose Operon**

- Operator
- Structural genes
- RNA polymerase
- Z gene
- V gene
- A gene
- Ribosome
- mRNA
- β-galactosidase
- Permease
- Transacetylase

---

**Metabolic space**

- Metabolite 1 → Metabolite 2

**Protein space**

- Protein 1 → Protein 2 → Complex 34 → Protein 4

**Gene space**

- Gene 1 → Gene 2
- Gene 3 → Gene 4

---

**DNA Tutorial**

*Pretty far from the atoms.*
The Gene Machine “Instruction Set”

**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: 320Mb 80MB
  - Coding: 160Mb 40MB
  - Protein-coding genes: 30,000-40,000

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

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

**Yeast (eukarya):** 12Mb 3MB (MP3 song)

**Wheat:** 17Gb 4.25GB (DVD)
Gene Composition

Is a shorthand for:

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].
Structure of the Coding Region

The Central Dogma

DNA  transcription  mRNA  translation  Protein

Challenging the Dogma (in higher organisms)

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)
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 CG, 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 CG3 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 (5, 8). 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 G, and is required for functions of the upstream modules F, E, and DC. These functions are repression of expression in nonendodermal domains and enhancement of expression in response to L11.

2300bp!
> average protein
Function of a Regulatory Region

B
If (F = 1 or E = 1 or CD = 1) and (Z = 1) Repression functions of modules F, E, and
\[ \alpha = 1 \]
DC mediated by Z site
else \[ \alpha = 0 \]
if (P = 1 and CG, = 1) Both P and CG, needed for synergistic link
\[ \beta = 2 \]
with module B
else \[ \beta = 0 \]
if (CG, = 1 and CG, = 1 and CG, = 1) Final step up of system output
\[ \gamma = 2 \]
else \[ \gamma = 1 \]
A(t) = B(t) + G(t)
E(t) = \beta \cdot A(t)
if (\alpha(t) = 0) Switch determining whether Otx site in
\[ \zeta(t) = Otx(t) \]
module A, or upstream modules (i.e.,
else \[ \zeta(t) = e(t) \]
mainly module B), will control level of
if (\alpha = 1) Repression function inoperative in
\[ \eta(t) = 0 \]
endoderm but blocks activity elsewhere
else \[ \eta(t) = \zeta(t) \]
\[ \theta(t) = \gamma \cdot \eta(t) \]
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

\[ \text{neg}(a,b) \triangleq ?a_r; \tau_{\eta}; \text{neg}(a,b) + \tau_{\tau}; (\text{tr}(b) \mid \text{neg}(a,b)) \]

\[ \text{tr}(p) \triangleq (p_r; \text{tr}(p)) + \tau_{\delta} \]

A genetic circuit (engineered in E.Coli)

A stochastic simulation (in SPiM)
3. The Membrane Machine

Very far from the atoms.

Molecular transport and transformation through dynamic compartment fusion and fission.

The Instruction Set

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

1. Cell membrane
2. Virus membrane

Proposed sequence of events in pH sensitive hemagglutinin membrane fusion

3. Aggressive fusion (virus)

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

5. Cooperative fusion (vesicle)

6. “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]

Cytokinesis (Mitosis)
The Membrane Machine “Instruction Set”

**Mito** (special cases):
- **Mate**
- **Fusion**
- **Fission**

**Endo** (special cases):
- **Exo**
- **Fusion**
- **Fission**

**Zero case**:
- **Drip**
- **Bud**

**One case**:
- **Pino**
- **Phago**
... in 3D

- S-Exo → Fusion
- 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
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

LDL-Cholesterol Degradation

Viral Replication


Brane Calculi

systems

\[ P, Q ::= \diamond \mid P \circ Q \mid !P \mid \sigma(P) \]

- nests of membranes

branes

\[ \sigma, \tau ::= 0 \mid \sigma \mid \tau \mid !\sigma \mid a.\sigma \]

- combinations of actions

actions

\[ a ::= 1 \mid \ldots \]

(fill in as needed)

1D fluids (\(\sigma\)) inside a 2D fluid (\(P\))

TWO commutative monoids instead of ONE of normal process calculi

\[ \sigma(P) \]

\[ \sigma|\tau(P) \]

\[ a.\sigma|\tau = (a.\sigma)|\tau \]

N.B. Restriction (\(\cdot\)) could be added to both systems and branes. It usually would originate in branes, but would extrude to whole systems.
Brane Reactions (Cartoons)

A Turing-Complete language
[Busi Gorrieri]
Brane-Molecule Reactions (Cartoons)

With *molecule multisets* $p,q$:

$$
\begin{array}
\text{p}_1 & \text{B&R} & \text{q}_1 \\
\text{p}_2 & \text{ } & \text{q}_2
\end{array}
$$
Phago \(\varnothing_n.\sigma|\sigma'(P) \circ \varnothing_n^-(\rho).\tau|\tau'(Q) \Rightarrow \tau|\tau'(\rho(\sigma|\sigma'(P))\circ Q)\)

Exo \(\varnothing_n^-.\tau|\tau'(\varnothing_n.\sigma|\sigma'(P)\circ Q) \Rightarrow P \circ \sigma|\sigma'|\tau|\tau'(Q)\)

Pino \(\circ(\rho).\sigma|\sigma'(P) \Rightarrow \sigma|\sigma'(\rho(D\circ)\circ P)\)

N.B.: the parity of nesting of P and Q is preserved; this makes the reactions preserve bitonality.

B&R \(p_1 \circ p_1(p_2) \Rightarrow q_1(q_2).\alpha|\sigma(p_2 \circ P) \Rightarrow q_1 \circ \alpha|\sigma(q_2 \circ P)\)

(mutiset rewriting, inside and outside membranes)
Derivable Reactions (Cartoons)

A Decidable-Termination language
[Busi Gorrieri]
Ex: Viral Infection

virus

\( nucap \)

\( \text{membrane} \)

\( \text{endosome} \)

\( \text{cell} \)

\( \text{cytosol} \)

\( \text{mate} \)

\( \text{vesicle} \)

\( \text{endosome} \)

\( \text{Phago} \)

\( \text{Mate} \)

\( \text{Exo} \)

\( \text{membrane} \)

\( \text{endosome} \)
Ex: Viral Progeny

Assume:
\[ \text{nucap} \circ \text{cytosol} \rightarrow \rightarrow \text{nucap}^n \circ \text{envelope-vesicle}^m \circ \text{cytosol}' \]
by available cellular machinery

Then:

\[ \text{cell} \]
\[ \text{envelope-vesicle} \quad \text{nucap} \]
\[ \text{envelope} \quad \text{nucap} \]
\[ \text{cell} \quad \text{virus} \]
Ex: Autophagic Process

Lysosome and target don’t just merge.

Biologically, Mito/Mate clearly happens. However, weird sequences of Endo/Exo are also common.
"On Brane" vs. "In Brane"

Original "on brane" Exo of Brane Calculus

"In brane" encoding (e.g. in BioAmbients or SMBL) goes wrong

"Ball bearing" encoding; best we can do "in brane"

Awkward encoding. And all kinds of things can go wrong in the intermediate state.

- One cannot easily represent the Exo reaction in BioAmbients or any such compartment-based calculus, nor can one easily add it as a new primitive!

- But we can add BioAmbients-like In/Out out to Brane Calculi if we want to.
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
- Models function, creates, regulates
- Directs membrane construction
- Contains regulations
- Signals conditions and events

**Protein Machine**
- Aminocids
- Holds receptors, actuator hosts reactions
- Implements fusion, fission
- Metabolism, Propulsion
- Signal Processing
- Molecular Transport

**Membrane Machine**
- Phospholipids
- Holds genome(s)
- Different time and space scales
- Works with receptor(s) and protein embedding

Functional Architecture
- Diverse
- Chemical toolkits
- Instruction sets
- Programming models
- Notations

Biochemical Networks

Transport Networks
Importance of Stochastic Effects

- **A deterministic system:**
  - May get “stuck in a fixpoint”.
  - And hence never oscillate.

- **A similar stochastic system:**
  - May be “thrown off the fixpoint” by stochastic noise, entering a long orbit that will later bring it back to the fixpoint.
  - And hence oscillate.

---

**Fig. 5.** Time evolution of \( \dot{R} \) for the deterministic Eq. (1) (a) and stochastic (b) versions of the model. The values of the parameters are as in the caption of Fig. 1, except that now we set \( \Delta = 0.05 \) h\(^{-1} \). For these parameter values, \( \tau < 0 \), so that the fixed point is stable.

---

**Fig. 6.** Phase portrait as in Fig. 4 but for a situation in which the system falls into the stable fixed point \((R_0, C_0)\). The dotted arrow to the left of the fixed point illustrates a perturbation that would initiate a single sweep of the (former) oscillatory trajectory.

---

**Mechanisms of noise-resistance in genetic oscillators**

José M. G. Vilar, Hao Yuan Kueh, Naama Barkai, Stanislas Leibler

PNAS April 30, 2002 vol. 99 no. 9 p. 5991

Surprisingly enough, we have found that parameter values that give rise to a stable steady state in the deterministic limit continue to produce reliable oscillations in the stochastic case, as shown in Fig. 5. Therefore, the presence of noise not only changes the behavior of the system by adding more disorder but can also lead to marked qualitative differences.
Model Construction and Validation
Reactive Systems

- **Modeling biological systems**
  - Not as continuous systems (often highly nonlinear)
  - But as discrete reactive systems; abstract machines with:
    - *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**:
  - Deep layering of abstractions
  - Complex composition of simple components
  - Discrete transitions between states
  - Digital coding and processing of information
  - Reactive information-driven behavior
  - High degree of concurrency and nondeterminism
  - “Emergent behavior” not obvious from part list
Chemistry vs. \(\pi\)-calculus

A process calculus (chemistry, or SBML)

\[
\begin{align*}
Na + Cl & \rightarrow_{k_1} Na^+ + Cl^- \\
Na^+ + Cl^- & \rightarrow_{k_2} Na + Cl
\end{align*}
\]

This Petri-Net-like graphical representation degenerates into spaghetti diagrams: precise and dynamic, but not scalable, structured, or maintainable.

A compositional graphical representation, and the corresponding calculus.

\[
\begin{align*}
Na = !r_{k_1}; ?s_{k_2}; Na \\
Cl = ?r_{k_1}; !s_{k_2}; Cl
\end{align*}
\]

A different process calculus (\(\pi\))
Model Validation: Simulation

- **Basic stochastic algorithm: Gillespie**
  - Exact (i.e. based on physics) stochastic simulation of chemical kinetics.
  - Can compute concentrations and reaction times for biochemical networks.

- **Stochastic Process Calculi**
  - **BioSPI** [Shapiro, Regev, Priami, et. al.]
    - Stochastic process calculus based on Gillespie.
  - **BioAmbients** [Regev, Panina, Silverma, Cardelli, Shapiro]
    - Extension of BioSpi for membranes.
  - **Case study: Lymphocytes in Inflamed Blood Vessels** [Lecca, Priami, Quaglia]
    - Original analysis of lymphocyte rolling in blood vessels of different diameters.
  - **Case study: Lambda Switch** [Celine Kuttler, IRI Lille]
    - Model of phage lambda genome (well-studied system).
  - **Case study: VICE** [U. Pisa]
    - Minimal prokaryote genome (180 genes) and metabolism of whole VIRTUAL CELL, in stochastic $\pi$-calculus, simulated under stable conditions for 40K transitions.

- **Hybrid approaches**
  - **Charon language** [UPenn]
    - Hybrid systems: continuous differential equations + discrete/stochastic mode switching.
  - Etc.
Model Validation: “Program” Analysis

- **Causality Analysis**
  - Biochemical pathways, (“concurrent traces” such as the one here), are found in biology publications, summarizing known facts.
  - This one, however, was automatically generated from a program written in BioSpi by comparing traces of all possible interactions. [Curti, Priami, Degano, Baldari]
  - One can play with the program to investigate various hypotheses about the pathways.

- **Control Flow Analysis**
  - Flow analysis techniques applied to process calculi.
  - Overapproximation of behavior used to answer questions about what “cannot happen”.
  - Analysis of positive feedback transcription regulation in BioAmbients [Flemming Nielson].

- **Probabilistic Abstract Interpretation**
  - [DiPierro Wickicky].
Model Validation: Modelchecking

- **Temporal**
  - Software verification of biomolecular systems (NA pump) [Ciobanu]
  - Analysis of mammalian cell cycle (after Kohn) in CTL.
    - E.g. is state $S_1$ a necessary checkpoint for reaching state $S_2$?

- **Quantitative: Simpathica/xssys** [Antioniotti Park Policriti Ugel Mishra]
  - Quantitative temporal logic queries of human Purine metabolism model.
    
    $\text{Eventually(Always } (PRPP = 1.7 * PRPP1) \implies steady\text{\_state()}}$
    
    and $\text{Eventually(Always(IMP < 2 * IMP1))}$
    
    and $\text{Eventually(Always(hx\_pool < 10*hx\_pool1)))}$

- **Stochastic: Spring** [Parker Normal Kwiatkowska]
  - Designed for stochastic (computer) network analysis
    - Discrete and Continuous Markov Processes.
    - Process input language.
    - Modelchecking of probabilistic queries.
What Reactive Systems Do For Us

We can write things down precisely
- We can modularly describe high structural and combinatorial complexity (“do programming”).

We can calculate and analyze
- Directly support simulation.
- Support analysis (e.g. control flow, causality, nondeterminism).
- Support state exploration (model checking).

We can visualize
- Automata-like presentations.
- Petri-Net-like presentations.
- State Charts, Live Sequence Charts [Harel]
  • Hierarchical automata.
  • Scenario composition.

We can reason
- Suitable equivalences on processes induce algebraic laws.
- We can relate different systems (e.g. equivalent behaviors).
- We can relate different abstraction levels.
- We can use equivalences for state minimization (symmetries).

Disclaimers
- Some of these technologies are basically ready (medium-scale stochastic simulation and analysis, medium-scale nondeterministic and stochastic model checking).
- Others need to scale up significantly to be really useful. This is (has been) the challenge for computer scientists.

Many approaches, same basic philosophy, tools being built:
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