Abstract Machines of Systems Biology

Luca Cardelli

Microsoft Research
Cambridge UK

2005-03-21 Brighton

www.luca.demon.co.uk
• 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?
Eukaryotic Cell
(10~100 trillion in human body)

Membranes everywhere

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

Biochemical Networks - The Protein Machine
Gene Regulatory Networks - The Gene Machine
Transport Networks - The Membrane Machine

Systems Biology
1. “How do components interact?”
2. “Gather high-throughput data.”

Gene Machine
Nucleotides

Regulation

Diverse:
- chemical toolkits
- instruction sets
- programming models
- notations

Gene Regulation
Metabolism, Propulsion
Signal Processing
Molecular Transport

Protein Machine
Aminoacids

Makes proteins:
where/when/howmuch

Directs membrane construction
and protein embedding

Holds genome(s):
confines regulators

Model Integration
Different time
and space scales

Surface and Extracellular Features

Membrane Machine
Phospholipids

Confinement
Storage
Bulk Transport

Holds receptors, actuators
hosts reactions

Implements fusion, fission

Glycan Machine
Sugars

Diverse: - chemical toolkits - instruction sets - programming models - notations

Metabolism, Propulsion
Signal Processing
Molecular Transport

Surface and Extracellular Features
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

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 Wiscott Aldrich Syndrome Protein (EVH1) domain, a B motif, a GBD, a proline-rich segment (prn) 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 Wiscott is known). CTP-bound Cdc42 and PIP2 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 \( A:A \).

\( x \) 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


JDesigner

http://www.cds.caltech.edu/~hsauro/index.htm
The Protein Machine “Instruction Set”

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

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

On/Off switches

Protein

Binding Sites

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.
Notations for the Protein Machine

- **Stochastic π-Calculus**
  - Priami (following Hillston’s PEPA) formalizes a stochastic version of π-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 π-restriction.

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

- **k-calculus**
  - Danos and Laveve (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**
  - Chabrier Rvier-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. Values are the assumed Hill coefficients for each reaction. The Hill coefficients were calculated from the triplicate data. See Materials and Methods for details of the calculations. The Hill coefficients were assumed to be 3.0 for the phosphorylation of a maximally activated MAPK by a MAPK.

<table>
<thead>
<tr>
<th>Reactant</th>
<th>Range of $n_H$</th>
<th>Mean $n_H$</th>
<th>SD $n_H$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. ERK3 - ERK2</td>
<td>0.5-3.0</td>
<td>2.0</td>
<td>1.0</td>
</tr>
<tr>
<td>2. ERK1 - ERK2</td>
<td>0.5-3.0</td>
<td>2.0</td>
<td>1.0</td>
</tr>
<tr>
<td>3. ERK2 + MAPK</td>
<td>0.5-3.0</td>
<td>2.0</td>
<td>1.0</td>
</tr>
<tr>
<td>4. MAPK + MAPK</td>
<td>0.5-3.0</td>
<td>2.0</td>
<td>1.0</td>
</tr>
</tbody>
</table>

10 chemical reactions

Fig. 1. Schematic view of the MAPK cascade. Activation of MAPK depends upon the phosphorylation of two conserved sites of MAPK. The SH2 and Tyr-185 in rat p2 MAPK/ERK2 (4, 5). Full activation of MAPK also requires phosphorylation of two sites (Ser-218 and Thr-222 in mouse ERK1/2). Detailed mechanisms for the activation of various MAPKs (e.g., ERK, JNK) and MAPK-activated proteins (e.g., p38, JNK) are not yet established. By analogy, we denote that MAPK/ERK2 and MAPK/ERK2 are activated by MAPK/ERK2 and MAPK/ERK2, respectively. MAPK/ERK2 in the phosphorylated state is indicated by a red circle.
As 18 Ordinary Differential Equations
Plus 7 conservation equations

\[
\frac{d}{dt} [KKK] = -a_{kk} [KKK][E1] + b_{kk} [KKK]+E2]
\]

\[
\frac{d}{dt} [KKK-E1] = a_{kk} [KKK][E1] + (d_1 + k_1) [KKK-E1] + k_2 [KKK-E2]
\]

\[
\frac{d}{dt} [KKK'] = -a_{kk} [KKK][E1] + b_{kk} [KKK]+E2]
\]

\[
\frac{d}{dt} [KKK-E2] = a_{kk} [KKK][E2] + (d_1 + k_1) [KKK-E2]
\]

\[
\frac{d}{dt} [KK] = -a_{kk} [KK][KKK'] + b_{kk} [KK-KP-KP']
\]

\[
\frac{d}{dt} [KK-KP-KP'] = a_{kk} [KK][KKK'] - (d_1 + k_1) [KK-KP-KP']
\]

\[
\frac{d}{dt} [KKP] = a_{kk} [KKP][E1] + b_{kk} [KKP]+E2]
\]

\[
\frac{d}{dt} [KKP-E1] = a_{kk} [KKP][E1] + (d_1 + k_1) [KKP-E1] + k_2 [KKP-E2]
\]

\[
\frac{d}{dt} [KKP-E2] = a_{kk} [KKP][E2] + (d_1 + k_1) [KKP-E2]
\]

\[
\frac{d}{dt} [KKP-KP-KP'] = a_{kk} [KKP][KKP'] - (d_1 + k_1) [KKP-KP-KP']
\]

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

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

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

\[
+ [KKK*·E2]
\]

\[
+ [KKK*·K] + [KKK*·K/P]
\]

These equations were solved numerically using the Runge–Kutta-based NDSolve 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_PPP())))

and KKPse() =

do ?a4(u4); (do ?u4;KKPse() or ?k4;KKPse())
or ?a6(u6); (do ?u6;KKPse() or ?k6;KKPse())

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())
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_P())))

and KPse() =

do ?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, 1xKpse, 1xKpse.

Input is 1xE1. Output is 100xK-PP (ultrasensitivity).
MAPK Cascade Simulation in SPiM

All coefficients 1.0 !!!
100xKKK, 100xKK, 100xK, 10xE2, 10xKKPse, 10xKPse.
(so 1xE1 is no longer sufficient to produce an output)
2. The Gene Machine

The “Central Dogma” of Molecular Biology

- DNA
- messenger RNA
- PROTEIN
- SYSTEMS

4-letter digital code ➔ 4-letter digital code ➔ 20-letter digital code ➔ 50,000(? shapes)

regulation ➔ transcription ➔ translation ➔ interaction

Lactose Operon

Metabolic space

Protein space

Gene space

Pretty far from the atoms.
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**
- 3Gbp (Giga base pairs) = 750MB @ 4bp/Byte (CD)
  - Non-repetitive: 1Gbp = 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
Repessed
Expressing
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**

**C Module A functions:**

Vegetal plate expression in early development:

Synergism with modules B and G enhancing endoderm expression in later development:

Repression in ectoderm (modules E and F) and skeletogenic mesenchyme (module DC):

Modules E, F and DC with LiCl treatment:

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 E, F, and DC. These functions are repression of expression in nonendodermal domains and enhancement of expression in response to LiCl.
Function of a Regulatory Region

DNA

Begin coding region

And

Or

Sum

Amplify

Gate

Begin coding region

\[
\begin{align*}
B & \\
\text{If } (F = 1 \text{ or } E = 1 \text{ or } CD = 1) \text{ and } (Z = 1) & \text{ Repression functions of modules } F, E, \text{ and } CD \text{ mediated by } Z \\
\alpha & = 1 \\
\text{else} & \alpha = 0 \\
\text{if } (P = 1 \text{ and } CG_2 = 1) & \text{ Both } P \text{ and } CG_2 \text{ needed for synergistic link with module } B \\
\beta & = 2 \\
\text{else} & \beta = 0 \\
\text{if } (CG_1 = 1 \text{ and } CG_2 = 1 \text{ and } CG_4 = 1) & \text{ Final step up of system output} \\
\gamma & = 2 \\
\text{else} & \gamma = 1 \\
\xi(t) &= B(t) + G(t) \\
\zeta(t) &= \beta \xi(t) \\
\text{if } (\xi(t) = 0) & \text{ Switch determining whether Otx site in module } A, \text{ or upstream modules (i.e., mainly module } B), \text{ will control level of activity} \\
\zeta(t) &= \text{Otx}(t) \\
\text{else} & \zeta(t) = \eta(t) \\
\text{if } (\alpha = 1) & \text{ Repression function inoperative in endoderm but blocks activity elsewhere} \\
\eta(t) &= 0 \\
\text{else} & \eta(t) = \xi(t) \\
\xi(t) &= \gamma^* \eta(t) \\
\text{Final output communicated to } BTA
\end{align*}
\]

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.

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

A genetic circuit (engineered in E.Coli)
\[ \neg(a, b) \mid \neg(b, c) \mid \neg(c, a) \]

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

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

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]
Positive curvature to Negative curvature transition in 3D

Membrane Fusion

Cell membrane
Virus membrane

1

2

3

4

5

6

Proposed sequence of events in pH sensitive hemagglutinin membrane fusion

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

Aggressive fusion (virus)

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]

Cytokinesis (Mitosis)
The Membrane Machine “Instruction Set”

Arbitrary subsystem

Mito: special cases

Zero case

One case

Endo: special cases

Zero case

One case

Arbitrary subsystem
Locally Implementable!

Global Views

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

Viral Replication

LDL-Cholesterol Degradation


Brane Calculi

**Systems**
\[ P, Q := \diamond | P \circ Q | !P | \sigma(P) \]

Nests of membranes

**Branes**
\[ \sigma, \tau := 0 | \sigma|\tau | !\sigma | a.\sigma \]

Combinations of actions

**Actions**
\[ a := 1 | ... \]

(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 (\(\nu n\)) 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]

Exo

Pino

Phago
Brane-Molecule Reactions (Cartoons)

With molecule multisets $p, q$:

$$p_1 \xrightarrow{B&R} p_2 \quad \text{and} \quad q_1 \xrightarrow{B&R} q_2$$
\[
\text{Phago} \quad \forall_n \sigma | \sigma' (P) \circ \forall_n (\rho). \tau | \tau' (Q) \rightarrow \tau | \tau' (\rho (\sigma | \sigma' (P)) \circ Q)
\]

\[
\text{Exo} \quad \forall_n \tau | \tau' (\forall_n \sigma | \sigma' (P) \circ Q) \rightarrow P \circ \sigma | \sigma' | \tau | \tau' (Q)
\]

\[
\text{Pino} \quad \ominus (\rho). \sigma | \sigma' (P) \rightarrow \sigma | \sigma' (\rho (\circ) \circ P)
\]

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

\[
\text{B\&R} \quad p_1 \circ p_1 (p_2) \Rightarrow q_1 (q_2). \alpha | \sigma (p_2 \circ P) \rightarrow q_1 \circ \alpha | \sigma (q_2 \circ P)
\]

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

A Decidable-Termination language
[Busi Gorrieri]
**Viral Reproduction**

- **Virus**
- **Endosome**
- **Endoplasmic Reticulum**
  - (via Golgi)
- **Infection**
- **Cytosol**
  - **Nucleocapsid**
    - RNA Capsid Membrane Envelope protein
  - **Translation**
- **Nucleus**
  - **Endoplasmic Reticulum**
  - **RNA Replication**
  - **Budding**
  - **Drip**
  - **Vesicle**
- **Assembly**
- **Progeny**
Ex: Viral Infection

virus

cell

membrane endosome

membrane vesicle endosome

membrane endosome

membrane endosome
Assume:
\[ \text{nucap} \circ \text{cytosol} \rightarrow \text{nucap}^n \circ \text{envelope-vesicle}^m \circ \text{cytosol}' \]

by available cellular machinery

Then:

\[ \text{cell} \]

\[ \text{envelope-vesicle} \quad \text{nucap} \]

\[ \text{envvelope} \quad \text{nucap} \]

\[ \text{cytosol}'' \quad \text{nucap} \]

Ex: Viral Progeny
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?
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.

---

**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

---

**Fig. 5.** Time evolution of $R$ 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 $\sigma = 0.05 \, h^{-1}$. For these parameter values, $\tau < 0$, so that the fixed point is stable.

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.
Scaling up to Big Systems: ODE's vs Processes
Stochastic $\pi$-calculus Executive Summary

- A process calculus:
  - The modular representation of concurrent (and stochastic) processes of all kinds.
  - Cuts down to CTMCs (Continuous Time Markov Chains) in the finite case (not always). Then, standard tools are applicable.
  - Can be given friendly automata-like scalable graphical syntax (work in progress: Andrew Phillips).
  - Is directly executable (e.g. via Gillespie).
  - Is analyzable (large body of literature, at least in the non-stochastic case).

Figure 2. Regulating Gene Expression by Positive Feedback [6]

Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of $TF$
Chemistry vs. $\pi$-calculus

A process calculus (chemistry, or SBML)

$$\text{Na} + \text{Cl} \rightarrow_{k_1} \text{Na}^+ + \text{Cl}^-$$

$$\text{Na}^+ + \text{Cl}^- \rightarrow_{k_2} \text{Na} + \text{Cl}$$

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.

$$\text{Na} = !r_{k_1}; ?s_{k_2}; \text{Na}$$

$$\text{Cl} = ?r_{k_1}; !s_{k_2}; \text{Cl}$$

A different process calculus ($\pi$)

Maps to a CTMC

Maps to a CTMC

The same “model”

Interaction oriented

Reaction oriented

Maps to a CTMC

Maps to a CTMC

Maps to a CTMC
From Reactions to ODE's

Write the coefficients by columns

\[
\frac{d[x]}{dt} = N \cdot v
\]

Read the concentration changes from the rows

\[
\begin{align*}
\frac{d[A]}{dt} &= -v_1 - v_2 \\
\frac{d[B]}{dt} &= -v_1 + v_4 \\
\frac{d[C]}{dt} &= 2 \cdot v_1 - v_2 - v_3 \\
\frac{d[D]}{dt} &= v_2 \\
\frac{d[E]}{dt} &= v_3 \\
\frac{d[F]}{dt} &= v_3 - v_4
\end{align*}
\]

\[
\begin{array}{ccccc}
\text{species} & \text{r}_1 & \text{r}_2 & \text{r}_3 & \text{r}_4 \\
\hline
A & -1 & -1 & & \\
B & -1 & 1 & & \\
C & 2 & -1 & -1 & \\
D & 1 & & & \\
E & 1 & & & \\
F & 1 & 1 & -1 & \\
\end{array}
\]

Read the rate laws from the columns

\[
v_i(x, e_i, k_i)
\]

E.g. \[
\frac{d[A]}{dt} = -k_1 [A][B] - k_2 [A][C]
\]

x: chemical species
[-]: concentrations
v: rate laws
k: kinetic parameters
N: stoichiometric matrix
e: catalysts (if any)
From Reactions to Processes

For binary reactions, first species in the column does an input and produces result, second species does an output. For unary reactions, species does a tau action and produces result. No ternary reactions.

\[ r_1: A+B \rightarrow k_1 C+C \]
\[ r_2: A+C \rightarrow k_2 D \]
\[ r_3: C \rightarrow k_3 E+F \]
\[ r_4: F \rightarrow k_4 B \]

Write the coefficients by columns

<table>
<thead>
<tr>
<th></th>
<th>( r_1 )</th>
<th>( r_2 )</th>
<th>( r_3 )</th>
<th>( r_4 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td>D</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>E</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

A = \( ?v_1 k_1 (C|C) + ?v_2 k_2 . D + ?a \)
B = \( !v_1 k_1 + ?b \)
C = \( !v_2 k_2 + \tau k_3 (E|F) + ?c \)
D = \( 0 + ?d \)
E = \( 0 + ?e \)
F = \( \tau k_3 . B + ?f \)

Stoichiometric Matrix

Read the process interactions from the rows

Add a barb for counting and plotting

(Rate laws are implicit in stochastic semantics)
Stoichiometric Matrices Blow Up

- **We can translate Chemistry to ODE’s or Processes**
  - It is standard to go from chemical equations to ODE’s via a stoichiometric matrix.
  - It is similarly possible to go from chemical equations to processes via a stoichiometric matrix.

- **But there is a better way:**
  - Stoichiometric matrices blow-up exponentially for biochemical systems (unlike for ordinary chemical systems) because proteins have combinatorial state and complexed states are common.
  - To avoid this explosion, we should describe biochemical systems compositionally without going through a stoichiometric matrix (and hence without ODE’s).
Complexes: The ODE Way

- **n** domains: A, B, C
  - A ≅ A_p
  - B ≅ B_p
  - C ≅ C_p

- **2n** domain reactions:
  - ABC
  - A_pBC
  - AB_pC
  - ABC_p
  - A_pB_pC
  - AB_pC_p
  - A_pB_pC_p

- **1** complex: ABC
  - ABC ≅ A_pBC
  - ABC ≅ AB_pC
  - ABC ≅ ABC_p
  - A_pBC ≅ A_pB_pC
  - A_pBC ≅ A_pBC_p
  - A_pBC ≅ A_pB_pC_p
  - A_pB_pC ≅ A_pB_pC_p
  - A_pB_pC ≅ A_pB_pC_p

The matrix is very sparse, so the corresponding ODE system is not dense. But it still has $2^n$ equations, one per species, plus conservation equations ([ABC]+[A_pBC]=constant, etc.).

System description is exponential in the number of basic components.

### Stoichiometric Matrix

<table>
<thead>
<tr>
<th></th>
<th>v_1</th>
<th>v_2</th>
<th>v_3</th>
<th>v_4</th>
<th>v_5</th>
<th>v_6</th>
<th>v_7</th>
<th>v_8</th>
<th>v_9</th>
<th>v_{10}</th>
<th>v_{11}</th>
<th>v_{12}</th>
<th>v_{13}</th>
<th>v_{14}</th>
<th>v_{15}</th>
<th>v_{16}</th>
<th>v_{17}</th>
<th>v_{18}</th>
<th>v_{19}</th>
<th>v_{20}</th>
<th>v_{21}</th>
<th>v_{22}</th>
<th>v_{23}</th>
<th>v_{24}</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A_pBC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AB_pC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ABC_p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A_pB_pC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A_pB_pC_p</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
When the local domain reactions are not independent, we can use lateral communication so that each component is aware of the relevant others.

System description is linear in the number of basic components.

(Its “run-time” behavior or analysis potentially blows-up just as in the previous case.)
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