Abstract Machines of Systems Biology

Luca Cardelli
Microsoft Research
Cambridge UK

2005-06-12 Dobbiaco

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

---

**Bioinformatics:** storing and analyzing experimental data.

**Molecular Biology:** figuring out the components of living things.

**Systems Biology:** figuring out their connectivity.
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.
**Structural Architecture**

**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
- Regulation
- Holds genome(s), confines regulators

**Protein Machine** (Aminoacids)
- Metabolism, Propulsion
- Signal Processing
- Molecular Transport
- Holds receptors, actuators
- Hosts reactions

**Membrane Machine** (Phospholipids)
- Confinement
- Storage
- Bulk Transport
- Implants fusion, fission

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

**Biochemical Networks**

**Transport Networks**

**Model Integration**
- Different time and space scales

**Glycan Machine**
- Surface and Extracellular Features

**Hierarchical Multisets**

**Functional Notations**
- Strings
- Records
- Trees
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)

$Na + Cl \rightarrow_{k1} Na^+ + Cl^-$

$Na^+ + Cl^- \rightarrow_{k2} Na + Cl$

A compositional graphical representation, and the corresponding calculus.

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

A different process calculus ($\pi$)
Methods

- **Model Construction** *(writing things down precisely)*
  - Formalizing the notations used in systems biology.
  - Formulating modeling 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
Basic Modeling Guidelines

- Regev-Shapiro: “Molecules as Processes”:

<table>
<thead>
<tr>
<th>Molecule</th>
<th>Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interaction capability</td>
<td>Channel</td>
</tr>
<tr>
<td>Interaction</td>
<td>Communication</td>
</tr>
<tr>
<td>Modification</td>
<td>State change</td>
</tr>
<tr>
<td>(of chemical components)</td>
<td>(state-transition systems)</td>
</tr>
</tbody>
</table>

Cellular Abstractions: Cells as Computation
Regev&Shapiro NATURE vol 419, 2002-09-26, 343

- They chose $\pi$-calculus and adapted it with stochastic features
  - To match the stochastic aspects of (bio)chemistry
  - Many probabilistic process calculi predate them, but only Hillston (CSP) and Priami ($\pi$) had already studied stochastic calculi.
**π-calculus Executive Summary**

- **It's for:**
  - The modular description of concurrent, nondeterministic systems
  - Study of such systems based on their descriptions

- **It's got:**
  - Processes
  - Channels
  - A minimal syntax *(it's a language and also a model)*

- **You can:**
  - Fork new processes
  - Create new channels
  - Do I/O over channels (synchronous and asynchronous)
    - including passing channels over channels
  - Make nondeterministic choices
  - Define processes recursively

- **That's it.**
  - Except for extensive model theory and metatheory.
  - Cannot pass processes over channels *(simulated by passing channels to them)*
  - Cannot define procedures *(simulated by supplying reply channels)*
\textbf{\(\pi\)-calculus (a Process Algebra)}

- **Processes** \(P, Q, \ldots\) - components of a system
- **Channels** \(a, b, \ldots\) - interactions between components

\begin{itemize}
  \item \textbf{0} the process that does nothing
  \item !a(b); P the process that outputs \(b\) on channel \(a\) (and then does \(P\))
  \item ?a(x); P the process that inputs \(b\) on channel \(a\) (and then does \(P\{x\}\))
  \item \(P | Q\) the process made of subprocesses \(P\) and \(Q\) running concurrently
  \item \(P + Q\) the process that behaves like either \(P\) or \(Q\) nondeterministically
  \item *\(P\) the process that behaves like unboundedly many copies of \(P\)
      \hspace{1cm} \Rightarrow recursive processes
      \hspace{1cm} \Rightarrow unbounded number and species of processes
  \item new \(x\); P the process that creates a new channel \(x\) (and then does \(P\{x\}\))
      \hspace{1cm} \Rightarrow private interactions
      \hspace{1cm} \Rightarrow unbounded number and species of interactions
\end{itemize}
π-calculus (a Process Algebra)

- **Dynamics**

\[
(la(b); P) + P' | (\?a(x); Q(x)) + Q' \rightarrow P | Q{b} 
\]

Ex.

\[
\begin{align*}
l(a(b); ?b) & \rightarrow \ ?b & | \ ?a(x); (!x + ?b) \\
\rightarrow & \ ?b & | \ !b + ?b \\
\rightarrow & \ 0 & | \ 0 \\
= & \ 0
\end{align*}
\]

- **“Compositional” descriptions**
  - Describe how the individual components behave
    - i.e. how they interact with any environment they may be placed in
  - Build systems by combining components
    - each components is part of the environment for the other components
  - Behavior (and its analysis) arises from the combinatorics of interactions
    - state space can be arbitrarily larger than its compositional description

- **For concurrent, nondeterministic, unbounded-state systems**
  - Dynamic creation of new channels (e.g. binding sites)
  - Dynamic creation of new processes (e.g. proteins)
π-calculus

Syntax

\[ \pi ::= x(y) \text{ receive } y \text{ along } x \]
\[ \overline{x}(y) \text{ send } y \text{ along } x \]

\[ P ::= 0 \mid \sum_{i \in I} \pi_i.P_i \mid [x = y] P \mid P_1 | P_2 \mid (\text{new } x)P \mid !P \]

Structural congruence

Renaming of bound variables

\[ x(y).P = x(z).(\{z/y\} P) \text{ if } z \notin FN(P) \]
\[ (\text{new } y).P = (\text{new } z).(\{z/y\} P) \text{ if } z \notin FN(P) \]

Structural congruence laws

\[ P|Q \equiv Q|P \text{ commutativity of parallel composition} \]
\[ (P|Q)|R \equiv P|(Q|R) \text{ associativity of parallel composition} \]
\[ P + Q \equiv Q + P \text{ commutativity of summation} \]
\[ P + (Q + R) \equiv (P + Q) + R \text{ associativity of summation} \]
\[ (\text{new } x)0 \equiv 0 \text{ restriction of inert processes} \]
\[ (\text{new } x)(\text{new } y)P \equiv (\text{new } y)(\text{new } x)P \text{ polyadic restriction} \]
\[ ((\text{new } x)P)|Q \equiv (\text{new } x)(P|Q) \text{ if } x \notin FN(Q) \text{ scope extrusion} \]
\[ !P \equiv P!!P \text{ replication} \]

Reaction rules

\[ (\cdots + \overline{x}(z).Q)|(\cdots + x(y).P) \rightarrow Q|P\{z/y\} \text{ communication (COMM)} \]

\[ \frac{P \rightarrow P'}{P|Q \rightarrow P'|Q} \text{ reaction under parallel composition (PAR)} \]

\[ \frac{P \rightarrow P'}{(\text{new } x)P \rightarrow (\text{new } x)P'} \text{ reaction under restriction (RES)} \]

\[ Q \equiv P \quad P \rightarrow P' \quad P' \equiv Q' \quad Q \rightarrow Q' \text{ structural congruence (STRUCT)} \]
**Stochastic π-calculus Executive Summary**

- A simple 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 (work in progress: Andrew Phillips).
  - 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 [6]

Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of TF A. Phillips, L. Cardelli. BioConcur’04.
Stochastic $\pi$-calculus

- **Stochastic extension of $\pi$-calculus.** [C.Priami]
  - Associate a single parameter $r$ (rate) in $(0, \infty]$ to each activity $a$.
  - The rate and the associated exponential distribution describes the stochastic behavior of the activity.

  $$a.P \text{ is replaced by } a@r.P$$

- **Exponential distribution**
  - guarantees the memoryless property: the time at which a change of state occurs is independent of the time at which the last change of state occurred.

- **Race condition**
  - is defined in a probabilistic competitive context: all the activities that are enabled in a state compete and the fastest one (stochastically) succeeds.

- **New implementation: SPiM.** [A.Phillips]. Paper at BioConcur
Stochastic Approach

- Relatively recent development on Process Calculi
  - For computer networking simulation and analysis
  - Now for biochemical simulation and analysis

- Continuous Time Markov Chains
  - Finite State Machines, with state transition times exponentially distributed (memoryless)
  - Well studied class of stochastic processes
  - Efficient analysis algorithms for stationary and transient analysis

- High level formalisms mapping to CTMCs
  - Stochastic Petri Nets [Molloy]
  - Markovian Queuing Networks [Muppala & Triverdi]
  - Stochastic Automata Networks [Plateau]
  - Probabilistic I/O Automata [Wu et al.]
  - Stochastic Process Algebras [Herzog et al.] [Hillston]
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 $\dot{R}$ 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 $l_0 = 0.05 \text{ hr}^{-1}$. For these parameter values, $r < 0$, 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.

---

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.
Protein Networks

- Gene Machine (Nucleotides)
  - Makes proteins, where/when/how much
  - Directs membrane construction and protein embedding
  - Signals conditions and events
  - Holds genome(s), confines regulators
  - Implements fusion, fission

- Membrane Machine (Phospholipids)
  - Holds receptors, actuators hosts reactions
  - Implies fusion, fission

- Protein Machine (Aminoacids)
  - Holds nucleotides
  - Hosts biochemical networks

Metabolism, Propulsion, Signal Processing, Molecular Transport

Diagram:
- A pathway from Gene Machine to Membrane Machine includes:
  - Metabolism
  - Propulsion
  - Signal Processing
  - Molecular Transport

Biochemical Networks diagram:
- A, B, C, P nodes connected by arrows indicating interactions.

The diagram illustrates the interconnectedness of biochemical networks, highlighting the roles of gene, protein, and membrane machines in cellular processes.
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>![Diagram of SH3-SH2 domain]</td>
<td>![Diagram of SH3-SH2 domain]</td>
</tr>
<tr>
<td>(b)</td>
<td>![Diagram of SH2-SH1 domain]</td>
<td>![Diagram of SH2-SH1 domain]</td>
</tr>
<tr>
<td>(c)</td>
<td>![Diagram of EVH1-B-GBD-PD domain]</td>
<td>![Diagram of EVH1-B-GBD-PD domain]</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 (EWS) domain, a B motif, a GBD, a polyproline-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 intra- and intermolecular interactions (only the structure of the GBD intramolecular complex for Wiskott-Aldrich Syndrome protein is known). CTP-bound Cdc42 and PI(4,5)P2 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.

Taken from Wendell Lim.
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.

Cova lent 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.

\[ x \rightarrow z \rightarrow y \]

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


JDesigner

http://www.cds.caltech.edu/~hsauro/index.htm

Figure 6B: The p53-Mdm2 and DNA repair regulatory network (version 2) - May 19, 1999.
FIG. 3. Simple one-way enzymatic reaction. (If there is an energy source, such as ATP hydrolysis, it can be omitted when ATP concentration is not an important factor.) In explicit formulations, the reaction identifiers or rate constant designations can be placed on the enzyme reaction line, and the node ES can identify the enzyme-substrate species.

FIG. 4. Interconversions between the GTP- and GDP-bound states of Ras. (1) GDP and GTP compete with each other for binding to a site on Ras (this binding is only slowly reversible). (2) GEF (guanine nucleotide exchange factor) facilitates the binding or dissociation of GDP or GTP (the concentration of GTP normally far exceeds that of GDP). (Implicit is the reversible binding between GEF and Ras which opens the binding site for GDP/GTP exchange.) (3) Ras has an intrinsic GTPase activity that slowly converts bound GTP to bound GDP (stoichiometric conversion arrow points from the node representing Ras.GTP to the node representing Ras.GDP). (4) RasGAP (a GTPase activating protein) enhances the GTPase activity of Ras. (Implicit is the reversible enzyme–substrate binding between RasGAP and Ras.)
Figure 1. Representation of fission yeast Cdc2 protein in (a) the original MIM and (b) proposed improvements. Both diagrams represent interactions involving fission yeast Cdc2. Wee1 phosphorylates Thr14 and Tyr15, Mik1 phosphorylates Tyr15, Mcs6 phosphorylates Thr167, and Cdc25 dephosphorylates Thr14 and Tyr15. Cdc2 binds to either Cdc13, Cig1, or Cig2. When Cdc2 is forming a complex with Cdc13 and only Thr167 is phosphorylated, the complex interacts with Lamina. Phosphorylation of either Thr14 or Tyr15 inhibits activation of Cdc2 due to phosphorylation of Thr167. The complex auto-phosphorylates Tyr15 of its Cdc2. The complex of Cdc2 and Cig1 interacts with Rum1. Cdc2-Cdc13 complex and Cdc-Cig2 complex form heterotrimers involving Rum1.
The Protein Machine “Instruction Set”

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.

cf. BioCalculus [Kitano&Nagasaki], κ-calculus [Danos&Laneve]
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
Design of genetic networks with specified functions by evolution in silico

Reactions | Constants | Stability
--- | --- | ---
\( a \rightarrow a+A \) | 0.20 | 0.9 - 1.1
\( A \rightarrow \text{Nothing} \) | 0.0085 | 0.0 - 1.5
\( b \rightarrow b+B \) | 0.37 | 0.7 - 1.3
\( B \rightarrow \text{Nothing} \) | 0.034 | 0.0 - 8.9
\( A+B \rightarrow A:B \) | 0.72 | 0.1 - > 10
\( A:B \rightarrow \text{Nothing} \) | 0.53 | Irrelevant
\( b+A \rightarrow b:A \) | 0.19 | 0.7 - 7.6
\( b:A \rightarrow b+b+A \) | 0.42 | 0.2 - 1.5
\( b:A \rightarrow b:A+B \) | 0.027 | 0.0 - 2.3

Fig 3A

Reaction oriented

A

Protein A

Number of proteins

Pulse of A

Pulse of B

time (min)

Fig 14A

Free evolution

Protein A

Number of proteins

time (min)
François & Hakim 3A

rates
\[ \begin{array}{llll}
\text{dkA} & \triangleq 0.0085 \\
\text{dkB} & \triangleq 0.034 \\
\text{dkAB} & \triangleq 0.53 \\
\text{pntAunb} & \triangleq 0.42 \\
\text{geneACst} & \triangleq 0.20 \\
\text{geneBCst} & \triangleq 0.37 \\
\text{geneBInh} & \triangleq 0.027 \\
\text{new bA} & \triangleq 0.19 \\
\text{new AB} & \triangleq 0.72
\end{array} \]

deploy

ptnB() \triangleq
\begin{align*}
\tau_{\text{dkB}} \\
+ ?AB: \text{cpxAB()}
\end{align*}

ptnA() \triangleq
\begin{align*}
\tau_{\text{dkA}} \\
+ !AB \\
+ \text{new unb}_{\text{pntAunb}} \\
\text{lbA(unb); ?unb; ptnA()}
\end{align*}

cpxAB() \triangleq \tau_{\text{dkAB}}

degradation

geneA() \triangleq \tau_{\text{geneACst}}: (\text{ptnA()} | \text{geneA()})

constitutive make prot A

geneBfree() \triangleq \tau_{\text{geneBCst}}: (\text{ptnB()} | \text{geneBfree()})

constitutive make prot B

bind to prot A (inhibit)

geneBbound(unb) \triangleq \tau_{\text{geneBInh}}: (\text{ptnB()} | \text{geneBbound(unb)})

inhibit make prot B

unbind from prot A

geneA() | geneBfree() \quad 1 \text{ gene a and 1 gene b}
One State of the Simulation

6 possible reactions on AB

1 possible reactions on each τ

Use Gillespie to pick a reaction and move on
François & Hakim 3A in SPiM

(* Francois and Hakim circuit 3A *)

let ptnA() =
  do delay@dkA
  or !AB
  or (new unb@pntAunb
      lbA(unb);(?unb; ptnA()))

let ptnB() =
  do delay@dkB
  or ?AB;cpxAB()
  complex with prot A

let cpxAB() = delay@dkAB
  degrade

let geneA() =
  do delay@geneACst (ptnA() | geneA())
  constit. make prot A

let geneBfree() =
  do delay@geneBCst (ptnB() | geneBfree())
  constit. make prot B
  or ?ba(unb); geneBbound(unb)
  bind to prot A (inhibit)

and geneBbound(unb:ch()) =
  do delay@geneBInh (ptnB() | geneBbound(unb))
  inhib. make prot B
  or l unb; geneBfree()
  unbind from prot A

run (geneA() | geneBfree())
  1 gene a and 1 gene b
François & Hakim Fig3A, SPiM simulation

Parameters as in paper

Free evolution

3 copies of each gene.

Modified for stability: \(dk_A = 0.02, dk_B = 0.02\)
François & Hakim Fig3Ast8

Circuit of Fig 3A with parameters from SupportingText Fig 8, plotted in Fig 13A

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a \rightarrow a + A )</td>
<td>0.52</td>
</tr>
<tr>
<td>( A \rightarrow \text{Nothing} )</td>
<td>0.00019</td>
</tr>
<tr>
<td>( b \rightarrow b + B )</td>
<td>0.79</td>
</tr>
<tr>
<td>( B \rightarrow \text{Nothing} )</td>
<td>0.0030</td>
</tr>
<tr>
<td>( A + B \rightarrow A:B )</td>
<td>0.053</td>
</tr>
<tr>
<td>( A:B \rightarrow \text{Nothing} )</td>
<td>0.15</td>
</tr>
<tr>
<td>( b + A \rightarrow b : A )</td>
<td>0.22</td>
</tr>
<tr>
<td>( b : A \rightarrow b + A )</td>
<td>0.31</td>
</tr>
<tr>
<td>( b : A \rightarrow b : A + B )</td>
<td>0.43</td>
</tr>
</tbody>
</table>
Graphical Representation

- Slides by Andrew Phillips
**MAPK Cascade**


Biochemistry: Huang and Ferrell

**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>Range of effective Hill coefficients (mM) predicted for</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. MAPKKK $\rightarrow$ MAPKK</td>
<td>0.0 - 1.0 nM</td>
<td>MAPKK $\rightarrow$ MAPKK $\rightarrow$ MAPK</td>
</tr>
<tr>
<td>2. MAPKK $\rightarrow$ MAPKK</td>
<td>0.0 - 1.0 nM</td>
<td>MAPKK $\rightarrow$ MAPKK $\rightarrow$ MAPK</td>
</tr>
<tr>
<td>3. MAPKK $\rightarrow$ MAPKK</td>
<td>0.0 - 1.0 nM</td>
<td>MAPKK $\rightarrow$ MAPKK $\rightarrow$ MAPK</td>
</tr>
<tr>
<td>4. MAPKK $\rightarrow$ MAPKK</td>
<td>0.0 - 1.0 nM</td>
<td>MAPKK $\rightarrow$ MAPKK $\rightarrow$ MAPK</td>
</tr>
<tr>
<td>5. MAPKK $\rightarrow$ MAPKK</td>
<td>0.0 - 1.0 nM</td>
<td>MAPKK $\rightarrow$ MAPKK $\rightarrow$ MAPK</td>
</tr>
<tr>
<td>6. MAPKK $\rightarrow$ MAPKK</td>
<td>0.0 - 1.0 nM</td>
<td>MAPKK $\rightarrow$ MAPKK $\rightarrow$ MAPK</td>
</tr>
<tr>
<td>7. MAPKK $\rightarrow$ MAPKK</td>
<td>0.0 - 1.0 nM</td>
<td>MAPKK $\rightarrow$ MAPKK $\rightarrow$ MAPK</td>
</tr>
<tr>
<td>8. MAPKK $\rightarrow$ MAPKK</td>
<td>0.0 - 1.0 nM</td>
<td>MAPKK $\rightarrow$ MAPKK $\rightarrow$ MAPK</td>
</tr>
<tr>
<td>9. MAPKK $\rightarrow$ MAPKK</td>
<td>0.0 - 1.0 nM</td>
<td>MAPKK $\rightarrow$ MAPKK $\rightarrow$ MAPK</td>
</tr>
<tr>
<td>10. MAPKK $\rightarrow$ MAPKK</td>
<td>0.0 - 1.0 nM</td>
<td>MAPKK $\rightarrow$ MAPKK $\rightarrow$ MAPK</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.

**Calculations.** Eqs. 1-10 represent the reactions of the MAPK cascade, which are shown schematically in Fig. 1. We have used Goldbeter and Koshland's nomenclature for the rate constants—the letter of denotes association; $i$ denotes dissociation without catalysis, and $i$ denotes product formation (11). KKK denotes MAPKK, KK denotes MAPKK and K denotes MAPK.

1. KKK + E1 $\rightarrow$ KKK+E1 $\rightarrow$ KKK + E1
2. KKK + E2 $\rightarrow$ KKK+E2 $\rightarrow$ KKK + E2
3. KKK + KKK $\rightarrow$ KKK + KKK
4. KKK + KKK $\rightarrow$ KKK + KKK
5. KKK + KKK $\rightarrow$ KKK + KKK
6. KKK + KKK $\rightarrow$ KKK + KKK
7. KKK + KKK $\rightarrow$ KKK + KKK
8. KKK + KKK $\rightarrow$ KKK + KKK
9. KKK + KKK $\rightarrow$ KKK + KKK
10. KKK + KKK $\rightarrow$ KKK + KKK

**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 MAPK 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., Ral-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 MAPKK-P denote singly and doubly phosphorylated MAPK. P$\text{'ase}$ denotes phosphatase.
As 18 Ordinary Differential Equations
Plus 7 conservation equations

\[
\begin{align*}
\frac{d}{dt} [KKK] &= -a_1[KKK][E1] + a_2[KKK-E2] \\
\frac{d}{dt} [KKK-E1] &= a_2[K][KKK][E1] - (a_1 + k_3)[KKK-E1] \\
\frac{d}{dt} [KKK^*-E2] &= a_1[K][KKK][E1] + (a_1 + k_3)[KKK-E1] \\
\frac{d}{dt} [KKK]^* &= -a_1[K][KKK][E1] + k_3[KKK-E1] + (a_1 + k_3)[KKK-E1] \\
\frac{d}{dt} [KKK^*-E2] &= a_1[K][KKK][E1] - (a_1 + k_3)[KKK-E1] \\
\frac{d}{dt} [KK] &= -a_1[K][KKK^*] + a_2[K][KKK] + a_3[KK^*-P][KKP^*] \\
\frac{d}{dt} [KK^*-P][KKP^*] &= a_1[K][KKK^*] - (a_1 + k_3)[KKK^*-E2] \\
\frac{d}{dt} [KKP] &= -a_1[K][KKP][KKP^*] + a_2[K][KKP][KKK^*] + a_3[KK^*-P][KKP^*] \\
\frac{d}{dt} [KKP^*] &= -a_1[K][KKP][KKP^*] + a_2[K][KKP][KKK^*] + a_3[KK^*-P][KKP^*] \\
\frac{d}{dt} [KKP^*-P][KKP^*] &= a_1[K][KKP][KKP^*] - (a_1 + k_3)[KKP^*-E2] \\
\frac{d}{dt} [KKP] &= -a_1[K][KKP][KKP^*] + a_2[K][KKP][KKK^*] + a_3[KK^*-P][KKP^*] \\
\frac{d}{dt} [KKP^*] &= -a_1[K][KKP][KKP^*] + a_2[K][KKP][KKK^*] + a_3[KK^*-P][KKP^*] \\
\frac{d}{dt} [KKP^*-P][KKP^*] &= a_1[K][KKP][KKP^*] - (a_1 + k_3)[KKP^*-E2] \\
\frac{d}{dt} [KKP] &= -a_1[K][KKP][KKP^*] + a_2[K][KKP][KKK^*] + a_3[KK^*-P][KKP^*] \\
\frac{d}{dt} [KKP^*] &= -a_1[K][KKP][KKP^*] + a_2[K][KKP][KKK^*] + a_3[KK^*-P][KKP^*] \\
\frac{d}{dt} [KKP^*-P][KKP^*] &= a_1[K][KKP][KKP^*] - (a_1 + k_3)[KKP^*-E2] \\
\frac{d}{dt} [KKP] &= -a_1[K][KKP][KKP^*] + a_2[K][KKP][KKK^*] + a_3[KK^*-P][KKP^*] \\
\frac{d}{dt} [KKP^*] &= -a_1[K][KKP][KKP^*] + a_2[K][KKP][KKK^*] + a_3[KK^*-P][KKP^*] \\
\frac{d}{dt} [KKP^*-P][KKP^*] &= a_1[K][KKP][KKP^*] - (a_1 + k_3)[KKP^*-E2]
\end{align*}
\]

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).

\[
\begin{align*}
[\text{KKK}_{\text{tot}}] &= [KKK] + [KKK^*] + [KKK\cdot E1] \\
&+ [KKK^* \cdot E2] \\
&+ [KKK^* \cdot K] + [KKK^* \cdot K\cdot P] \\
&+ [KKK\cdot K\cdot P\cdot E] \\
&+ [KKK\cdot K\cdot P\cdot K]\end{align*}
\]

Each molecule is in exactly one state.
Enzymatic Reactions

Reaction View

\[ \begin{align*}
S & \xrightarrow{(c,d,e)} E \\
S & \xrightarrow{P} P
\end{align*} \]

\[ \begin{align*}
E+S & \xrightarrow{c} ES \\
E+S & \xrightarrow{d} ES \\
ES & \xrightarrow{e} P+E
\end{align*} \]

Intermediate complex

Interaction View

\[ \begin{align*}
E & \xrightarrow{\text{bind}} \text{bind} \\
E & \xrightarrow{\text{unbind}} \text{unbind} \\
E & \xrightarrow{\text{react}} \text{react}
\end{align*} \]

\[ \begin{align*}
S() & \triangleq \text{new } u @ d \text{ new } k @ e \\
d & a_c(u,k); (l u_d; S() + ! k_e; P())
\end{align*} \]

Private bindings between one S and one E molecule

\[ \begin{align*}
E() & \triangleq ? a_c(u,k); (? u_d; E() + ? k_e; E())
\end{align*} \]

\[ \begin{align*}
P() & \triangleq \ldots
\end{align*} \]
MAPK Cascade 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
    new u5@d5:Release new k5@r5:React
do !a4(u4,k4); (do !u4;KK_P() or !k4;KK())
or !a5(u5,k5); (do !u5;KK_P() or !k5;KK_PP()))

and KKPse() =
    do ?a4(u4,k4); (do ?u4;KKPse() or ?k4;KKPse())
or ?a6(u6,k6); (do ?u6;KKPse() or ?k6;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())
or ?a10(u10,k10); (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 React = chan()

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

\[ a_i(u_i,k_i): \text{release } (u_i@d_i) \text{ and react } (k_i@r_i) \]
channels passed over bond \( a_i \) channel.

(No behavior attached to channels except interaction rate.)
**MAPK Cascade Simulation in SPiM**

1. **1st stage:**
   - KKK* barely rises

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

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

**Rates and concentrations ARTIFICIAL:**

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

Input is 1xE1.
Output is 90xK-PP (ultrasensitivity).
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

Rates and concentrations as in paper:

- $1 \times E2$ (0.3 nM)
- $1 \times KKPase$ (0.3 nM)
- $120 \times KPase$ (120 nM)
- $3 \times KKK$ (3 nM)
- $1200 \times KK$ (1.2 uM)
- $1200 \times K$ (1.2 uM)

\[ dx = rx = 150, \ ax = 1 \]
\[ K_{mx} = (dx + rx) / ax, \ Km = 300 \text{ nM} \]

$1 \times E1$ injected
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

transcription
translation
interaction
regulation

Lactose Operon

Metabolic space
Protein space
Gene space

Taken from Pedro Mendes

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

- **Positive Regulation**
- **Negative Regulation**
- **Transcription**
- **Coding region**
- **Regulatory region**

**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)

---

*cf. Hybrid Petri Nets [Matsuno, Doi, Nagasaki, Miyano]*
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

- The majority of the genomic sequence in higher organisms (the non-protein-coding DNA) is devoted to the control of developmental programming.
- The majority of the regulatory transactions in higher organisms are conveyed by RNAs, not proteins.
  
  John S. Mattick

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

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 CG\textsubscript{2} and CG\textsubscript{3} 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) Integrating 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.

Taken from Eric H. Davidson
Function of a Regulatory Region

**If (F = 1 or E = 1 or CD = 1) and (Z = 1)**

- $\alpha = 1$
- $\alpha = 0$

**If (P = 1 and CG$_1$ = 1)**

- $\beta = 2$
- $\beta = 0$

**If (CG$_2$ = 1 and CG$_3$ = 1 and CG$_4$ = 1)**

- $\gamma = 2$
- $\gamma = 1$

**$\delta(t) = B(t) \times G(t)$**

**$\epsilon(t) = \beta \times \delta(t)$**

**If (c(t) = 0)**

- $\zeta(t) = \text{Otx}(t)$
- $\xi(t) = \text{r}(t)$
- $\eta(t) = 0$
- $\tau(t) = \xi(t)$
- $\theta(t) = \gamma \times \eta(t)$

**Final step up of system output**

**Repression functions of modules F, E, and DC mediated by Z site**

**Both P and CG$_2$ needed for synergistic link with module B**

**Positive input from modules B and G**

**Synergistic amplification of module B output by CG$_1$-P subsystem**

**Switch determining whether Otx site in module A, or upstream modules (i.e., mainly module B), will control level of activity**

**Repression function inoperative in endoderm but blocks activity elsewhere**

Figure 6. Feature variation during the cell cycle. The temporal variation in nine selected protein features during the cell cycle, with zero time (at the top of the plot) corresponding to the presumed time of cell division (M-G1 transition). The color scales correspond to +/-two standard deviations from the cell cycle average. The concentric feature circles correspond to: isoelectric point, nuclear and extracellular localization predictions, PEST regions, instability index, N-linked glycosylation potential, O-GalNAc glycosylation potential, serine/threonine phosphorylation potential and tyrosine phosphorylation potential. The presumed positions of the four cell cycle phases G1, S, G2 and M are marked. Also depicted are known cell cycle transcriptional activators (marked in blue), positioned at the time where they are reported to function.

Protein Feature Based Identification of Cell Cycle Regulated Proteins in Yeast
Ulrik de Lichtenberg, Thomas S. Jensen, Lars J. Jensen and Søren Brunak

Taken from Brunak
Gene Regulatory Networks

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

Epigenetic Control
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
(The Classical ODE Approach)

\[ \frac{dr}{dt} = f(p) - Vr \]

\[ \frac{dp}{dt} = Lr - Ur \]

n: number of genes
r mRNA concentrations (n-dim vector)
 p protein concentrations (n-dim vector)

I.e.: to model an operating system, write a set of differential equations relating the concentrations in memory of data structures and stack frames over time. (Duh!)

f(p) transcription functions:
(n-dim vector polynomials on p)
A stochastic rate $r$ is always associated with each channel $a_r$ (at channel creation time) and delay $\tau_r$, but is often omitted when unambiguous.
Degradation is extremely important and often deliberate; it changes unbounded growth into (roughly) stable signals.

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

Degradation rate \( \delta \)

A transcription factor is a process (not a message or a channel): it has behavior such as interaction on \( p \) and degradation.

\[
\text{null}(b) \triangleq \tau_\varepsilon; (\text{tr}(b) | \text{null}(b))
\]

Interaction offers on \( b \) (\( \approx \) number of \( \text{tr} \) processes)

Combined effect of production and degradation (without any interaction on \( b \))

interaction site of transcription factor

stochastic choice (race between \( r \) and \( \delta \))

transcription factor

and repeat

degradation

product

null

b

null(b)
Unary Pos Gate

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

- **Input (excitatory)**
- **Output (stimulated or constitutive)**

- **transcription delay with rate** \( \eta \)
- **race between** \( r \) **and** \( \varepsilon \)

- **Parallel, not sequence, to handle self-loops without deadlock**

- **(input, ?) interaction with rate** \( r \)
- **or constitutive transcription to always get things started**

- **unlimited amount of**

**Graphical Representation:**

- **b**
- **Output protein**
- **pos(a, b)**
- **Stimulated**
- **Constitutive**

**Parameters:**

- \( r = 1.0, \varepsilon = 0.01, \eta = 0.1, \delta = 0.001 \)
Unary Neg Gate

\[ \text{input (inhibitory)} \quad \overset{\text{a}}{\underset{\text{neg}}{\rightleftharpoons}} \quad \overset{\text{output (constitutive when not inhibited)}}{\text{b}} \]

- (input, ?) interaction with rate \( r \)
- or constitutive transcription to always get things started
- \( \text{neg}(a,b) \triangleq \ ?a_r; \tau_\eta; \text{neg}(a,b) + \tau_\varepsilon; (\text{tr}(b) \mid \text{neg}(a,b)) \)
- inhibition delay with rate \( \eta \)
- race between \( r \) and \( \varepsilon \)

\[ r=1.0, \varepsilon=0.1, \eta=0.01, \delta=0.001 \]

\[ \text{Neg}(a_r,b) \]

\[ \text{Constitutive} \]

\[ \text{Inhibited} \]

\[ \ast \text{tr}(a_r) \mid \text{neg}(a_r,b) \]
Signal Amplification

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

\[ \text{pos}(a,b) \triangleq \]
\[ \tau_r; (\text{tr}(b) \mid \text{pos}(a,b)) + \tau_\eta; (\text{tr}(b) \mid \text{pos}(a,b)) \]
\[ \text{tr}(p) \triangleq (!p_r; \text{tr}(p)) + \tau_\delta \]

E.g. 1 a that interacts twice before decay can produces 2 b that each interact twice before decay, which produce 4 c...

With little degradation
\[ r=1.0, \eta=0.01, \delta=0.00001 \]

pos(a,b) | pos(b,c)

Even with no a input, constitutive production of b gets amplified to a high c signal

\[ r=1.0, \eta=0.01, \delta=0.001 \]

pos(a,b) | pos(b,c)
**Signal Normalization**

\[ \neg(a,b) \mid \neg(b,c) \]

\[ \neg(a,b) \triangleq \neg r; \tau_h; \neg(a,b) + \tau_\epsilon; (\text{tr}(b) \mid \neg(a,b)) \]

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

A non-zero input level, \(a\), whether weak or strong, is renormalized to a standard level, \(c\).
Self Feedback Circuits

pos(a,a)

\[ \text{pos(a,b)} \triangleq \neg a_r \land (\text{tr}(b) \land \text{pos}(a,b)) + \tau_h \land (\text{tr}(b) \land \text{pos}(a,b)) \]

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

(\text{Can overwhelm degradation, depending on parameters})

neg(a,a)

\[ \text{neg(a,b)} \triangleq \neg a_r \land \tau_h \land \neg a_r \land (\text{tr}(b) \land \neg a_r) + \tau_\delta \land (\text{tr}(b) \land \neg a_r) \]

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

high, to raise the signal

r=1.0, \( \varepsilon = 0.1 \), \( \delta = 0.01 \)

Less degradation

And a bit less

\( \delta = 0.0001 \)
Two-gate Feedback Circuits

For some degradation rates is quite stable:

\[ r = 1.0, \, \varepsilon = 0.1, \, h = 0.01, \, \delta = 0.0005 \]

But with a small change in degradation, it goes wild:

\[ r = 1.0, \, \varepsilon = 0.1, \, h = 0.01, \, \delta = 0.001 \]

5 runs with \( r(a) = 0.1, \, r(b) = 1.0 \) shows that circuit is now biased towards expressing \( b \).
Repressilator

\[
\begin{align*}
\neg(a,b) & | \neg(b,c) | \neg(c,a) \\
\neg(a,b) & \triangleq \neg(a,b) + \tau_{\epsilon}; \ (\text{tr}(b) | \neg(a,b))
\end{align*}
\]

Same circuit, three different degradation models by changing the tr component:

\[
\begin{align*}
\text{tr}(p) & \triangleq \neg p_r & \text{interact once and die otherwise stick around} \\
\text{tr}(p) & \triangleq \neg p_r + \tau_{\delta} & \text{interact once and die otherwise decay} \\
\text{tr}(p) & \triangleq (\neg p_r; \text{tr}(p)) + \tau_{\delta} & \text{interact many times and decay}
\end{align*}
\]

Subtle… at any point one gate is inhibited and the other two can fire constitutively. If one of them fires first, nothing really changes, but if the other one fires first, then the cycle progresses.
The constitutive rate $\epsilon$ (together with the degradation rate) determines oscillation amplitude, while the inhibition rate $\eta$ determines oscillation frequency. We can view the interaction rate $r$ as a measure of the volume (or temperature) of the solution; that is, of how often transcription factors bump into gates. Oscillation frequency and amplitude remain unaffected in a large range of variation of $r$. 
Repressilator in SPiM

val dk = 0.001 (* Decay rate *)
val eta = 0.001 (* Inhibition rate *)
val cst = 0.1 (* Constitutive rate *)

let tr(p:chan()) =
  do !p; tr(p)
  or delay@dk

let neg(a:chan(), b:chan()) =
  do ?a; delay@eta; neg(a,b)
  or delay@cst; (tr(b) | neg(a,b))

(* The circuit *)
val bnd = 1.0 (* Protein binding rate *)
new a@bnd: chan()
new b@bnd: chan()
new c@bnd: chan()

run (neg(c,a) | neg(a,b) | neg(b,c))
System Properties: Fixpoints

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

A sequence of neg gates behaves as expected, with alternating signals, (less “Booleanly” depending on attenuation).

\[ \eta = 100.0 \]

\[ \eta = 1.0 \]

\[ \eta = 0.01 \]

Now add a self-loop at the head. Not a Boolean circuit!

No more alternations, because... each gate is at its fixpoint.
Repressilator ODE Model and Simulation

\[
\begin{align*}
\frac{d[X]}{dt} &= \alpha_0 + \frac{\alpha + \alpha_1 [PY]^n}{K^n + [PY]^n} - k[X], & \frac{d[PY]}{dt} &= \beta([X] - [PX]) \\
\frac{d[Y]}{dt} &= \alpha_0 + \frac{\alpha + \alpha_1 [PZ]^n}{K^n + [PZ]^n} - k[Y], & \frac{d[PY]}{dt} &= \beta([Y] - [PY]) \\
\frac{d[Z]}{dt} &= \alpha_0 + \frac{\alpha + \alpha_1 [PX]^n}{K^n + [PX]^n} - k[Z], & \frac{d[PZ]}{dt} &= \beta([Z] - [PZ])
\end{align*}
\]

Bruce E Shapiro
Cellerator

We can model an inducer like aTc as something that competes for the transcription factor.

IPTG de-represses the lac operon, by binding to the lac repressor (the lacI gene product), preventing it from binding to the operator.
They engineered in E.Coli all genetic circuits with four single-input gates; such as this one:

\[
\begin{array}{cccc}
\text{aTc} & \text{TetR} & \text{LacI} & \text{λcI} \\
\text{PT} & \text{PT} & \text{Pl} & \text{Pl}
\end{array}
\]

\[
\begin{array}{c}
tet \\
lac \\
cI \\
gfp
\end{array}
\]

Then they measured the GFP output (a fluorescent protein) in presence or absence of each of two inhibitors (aTc and IPTG).

Experiment:

\[
\begin{array}{c|c}
\text{aTc} & 0101 \\
\text{IPTG} & 0011 \\
\text{GFP} & 0100
\end{array}
\]

The output of some circuits did not seem to make any sense...

Here “1” means “high brightness” and “0” means “low brightness” on a population of bacteria after some time. (I.e. integrated in space and time.)
Further Building Blocks

**Negp Gate**

\[ \text{negp}(a, (\varepsilon, \eta), p) = a \cdot \tau_{\eta} \cdot \text{negp}(a, (\varepsilon, \eta), p) + \tau_{\varepsilon} \cdot (p() | \text{negp}(a, (\varepsilon, \eta), p)) \]

**rtr(b,r)**

\[ \text{rtr}(b,r) = !b \cdot \text{rtr}(b,r) + !r \cdot 0 + \tau_{\delta} \cdot 0 \]

**rep(r)**

\[ \text{rep}(r) = ?r \cdot \text{rep}(r) \]

**regulatory input**

**rates product**

**product generation**

**repressible factor**

**binding repression degradation**

**interaction delay**

**arbitrary amounts of...**

**repressor**
D038/lac-

Experiment:
- $aTc$: 0101
- $IPTG$: 0011
- $GFP$: 0100

Naïve “Boolean” analysis would suggest $GFP=0.5$ (oscillation) because of self-loop.

$GFP=0$ there is consistent only with (somehow) the head loop setting $TetR=LacI=0$. But in that case, $aTc$ should have no effect (it can only subtract from those signals) but instead it sets $GFP=1$.

Hence we need to understand better the “dynamics” of this network.
Simulation results for D038/lac-

Experiment:
\[
\begin{array}{c|c}
\text{aTc} & 0101 \\
\text{IPTG} & 0011 \\
\text{GFP} & 0100 \\
\end{array}
\]

\(r=1.0, \varepsilon=0.1, h=1.0, \delta=0.001\)

The fixpoint effect can explain this (all signals set very low).

Then, aTc can destabilize the fixpoint, explaining GFP high (oscillating)

\(r = 1.0, \varepsilon = 0.1, \eta = 0.25 \) (\(P^T\)), \( \eta = 1.0 \) (\(P^{\lambda_2}, P^{\lambda}\)), \( \delta = 0.001 \)

D038/lac-

\[
\begin{align*}
\text{aTc} & \rightarrow \text{TetR} \\
\text{IPTG} & \rightarrow \text{LacI} \\
\text{lcI} & \rightarrow \text{GFP} \\
\end{align*}
\]

\[
\begin{align*}
P^T & \rightarrow \text{tet} \\
P^T & \rightarrow \text{lac} \\
P^{\lambda_2} & \rightarrow \text{cl} \\
P^{\lambda} & \rightarrow \text{gfp} \\
\end{align*}
\]
One theory: aTc prevents the self-inhibition of tet, so that a very large quantity of TetR is produced. That then overloads the overall degradation machinery of the cell, affecting the rest of the circuit.

How can aTc affect the result??

Even so, how can GFP be high here?

Even the fixpoint explanation fails here, unless we assume that the lac gate is operating in its instability region.
Simulation results for D016/lac-

Experiment:
\[
\begin{array}{c|c|c|c}
    aTc & 0101 \\
    IPTG & 0011 \\
    GFP  & 1000 \\
\end{array}
\]

The fixpoint effect, in instability region, explains this: GFP high because wildly oscillating.

Overloading of degradation machinery, induced by aTc, can reinstate the fixpoint regime.

\[\delta = 0.005\]
\[aTc = 0, IPTG = 0\]

\[\delta = 0.001\]
\[aTc = 1, IPTG = 0\]

\[\delta = 0.00001\]
\[aTc = 1, IPTG = 1\]

\[\delta = 0.001\]
\[aTc = 0, IPTG = 1\]

\[\delta = 0.005\]
\[aTc = 0, IPTG = 0\]

\[\delta = 0.001\]
\[aTc = 1, IPTG = 0\]

\[\delta = 0.00001\]
\[aTc = 1, IPTG = 1\]

\[\delta = 0.001\]
\[aTc = 0, IPTG = 1\]

\[\delta = 0.005\]
\[aTc = 0, IPTG = 0\]
What was the point?

- Deliberately pick a controversial/unsettled example to test the methodology.
- Show that we can easily “play with the model” and run simulations.
- Get a feeling for the kind of subtle effects that may play a role.
- Get a feeling for kind of analysis that is required to understand the behavior of these systems.
- In the end, we are never “understanding” anything; we are just building theories/models that support of contradict experiments (and that suggest further experiments).
Transport Networks

- Gene Machine
  - Makes proteins, where/when/how much
  - Signals conditions and events
  - Directs membrane construction and protein embedding
  - Holds genome(s), confines regulators

- Protein Machine
  - Hosts reactions
  - Amino acids

- Membrane Machine
  - Hosts receptors, actuator, hosts reactions
  - Phospholipids

- Confinement Storage
- Bulk Transport

- Transport Networks
  - Implements fusion, fission
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]
Membranes are Oriented 2D Surfaces

Lipid Bilayer
Self-assembling
Largely impermeable
Asymmetrical (in real cells)
With embedded proteins
A 2D fluid inside a 3D fluid!
Membrane Fusion

Positive curvature to Negative curvature transition in 3D

Cell membrane

Virus membrane

1 2 3

Proposed sequence of events in pH sensitive hemagglutinin membrane fusion

4 5 6

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

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

Aggressive fusion (virus)

Cooperative fusion (vesicle)
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)
Membrane Algorithms

Protein Production and Secretion


LDL-Cholesterol Degradation


Viral Replication

• **LDL-Cholesterol Degradation**
  - A cast of many thousands (molecules) just to get one molecule from A to B.
  - Membranes are key to the algorithm, we want to model *them*, not their individual millions of molecules.

• **Some very fancy chemistry**
  - But its “purpose” is to reliably implement a specific sequence of discrete steps.

Taken from MCB p.730
Receptor-Mediate Degradation Pathway

As a state transition diagram

- **Target particle (e.g. LDL Cholesterol)**
  - Clathrin
  - Cell
  - Clathrin coat

- **Ligand**
  - Receptor

- **Bind**
  - Endo

- **Depoly**
  - Merge
  - low pH

- **Sort**
  - Exo
  - Merge
  - Lysosome
  - Enzymes
  - Several hundred round-trips in lifespan of receptor

- **Unbind**
  - high pH

- **Degrade**
Membrane Orientation

Membranes are closed non-intersecting curves, with an orientation\(^{(1)}\).

Each membrane has two faces. A cytosolic (~inner) face and an exoplasmic (~outer) face. Nested membranes alternate orientation. (E.g. cytosolic faces always face each other, by definition, or by fusion/fission dynamics)

This alternation is illustrated by using two tones: blue (cytosol\(^{(2)}\)) and white (exosol\(^{(3)}\)). Bitonal diagrams.

Double membranes (e.g. the nuclear membrane) gives us blue-in-blue components.

---

1. A membrane is built from a phospholipid bilayer that is asymmetrical. Moreover, all real membranes are heavily sprinkled with proteins: "each type of integral membrane protein has a single specific orientation with respect to the cytosolic and exoplasmic faces of a cellular membrane, and all molecules of any particular integral membrane protein share this orientation. This absolute asymmetry in protein orientation confers different properties on the two membrane faces." MCB p162.

2. Short for Cytoplasmic Solution.

3. Short for Exoplasmic Region (I am making this one up).
Contiguous Membranes have Opposite Orientation

True “by construction”: look at the basic biological operations that increase the number of membranes in a system:
Bitonal Structure

**Bitonality**
Blue and white areas alternate.

**Bitonal Invariant (~ Orientation Invariant)**
Bitonality and subsystem coloring is preserved by reactions. I.e., blue and white fluids *never mix and never flip color*.

**Bitonal Duality**
Reactions come in complementary-tone versions.

The cell maintains a strong compartment-based separation between *inside fluids* and *outside fluids* even when incorporating foreign material.

---

Evolutionary explanations of bitonal structure

- Mitochondria acquisition
- Mitochondria to Chloroplasts
- Pre-Eukarya to Eukarya
Membrane Reactions

Membrane System

What reactions “make sense” biologically?

Reactions that “make sense” from a local, molecular viewpoint

Switch

(Symmetric by 90° rotation.)
Global Membrane Reactions

Reactions that “make sense” from a descriptive, global viewpoint

Same Local View!
Mito/Mate by 3 Endo/Exo
Ex: Autophagic Process

Lysosome and target don’t just merge.

Biologically, Mito/Mate clearly happens. However, weird sequences of Endo/Exo are also common.
Some global reactions are ruled out by bitonality, and by locality:

Violate bitonality.

Non implementable by “local” membrane operations.

Not observed (except gradual Open during “digestion” or “lysis”).

Happen to be the Ambient Calculus operations :-(

Non-Reactions
The Membrane Machine “Instruction Set”

- **Mate**
  - Mito
  - Arbitrary subsystem

- **Zero case**
  - Drip
  - Bud
  - One case

- **Endo**
  - special cases
  - Fusion
  - Fission
  - Fusion
  - Fission

- **Exo**
  - special cases
  - Mito: special cases
  - Fusion
  - Fission

- **Pino**
  - special cases
  - One case

- **Phago**
  - special cases
  - Arbitrary subsystem
... in 3D
Notations for the Membrane Machine

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

- **P-Systems**
    [http://psystems.disco.unimib.it/](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...
Brane Calculi
Computation “on” the membrane
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**

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

(fill in as needed)

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

Two commutative monoids instead of one of normal process calculi

\[
\sigma(P) \quad \sigma|\tau(P) \quad \text{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.}
\]

\[ a.\sigma|\tau = (a.\sigma)|\tau \]
## Congruence $\equiv$ and Reaction

<table>
<thead>
<tr>
<th>System</th>
<th>Brane</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P \circ Q \equiv Q \circ P$</td>
<td>$\sigma</td>
</tr>
<tr>
<td>$P \circ (Q \circ R) \equiv (P \circ Q) \circ R$</td>
<td>$\sigma</td>
</tr>
<tr>
<td>$P \circ \emptyset \equiv P$</td>
<td>$\sigma</td>
</tr>
</tbody>
</table>

### Fluidity

#### Plentitude

$!P \equiv P \circ !P$ etc.

$!\sigma \equiv \sigma | !\sigma$ etc.

#### Units

$0 \oplus \emptyset \equiv \emptyset$ Froth/Fizz

$1.\sigma \equiv \sigma$ Inaction

### Congruence

#### Reaction is up to congruence

$P \equiv P' \land P' \rightarrow Q' \land Q' \equiv Q \Rightarrow P \rightarrow Q$

#### Reactions in solution

$P \rightarrow Q \Rightarrow P \circ R \rightarrow Q \circ R$

$P \rightarrow Q \Rightarrow \sigma(P) \rightarrow \sigma(Q)$

This is the whole semantics, except for the effects of individual actions.
“Determinization”

Arbitrary subsystem

Endo

Exo

Zero case

Pino

One case

Phago

Arbitrary subsystem

Mito

Mate

Zero case

Drip

One case

Bud
Brane Reactions

\[
a ::= \ldots | \psi_n | \psi_n^\perp(\rho) | \psi_n | \psi_n^\perp | \odot(\rho)
\]

phago \(\psi\), exo \(\psi\), pino \(\odot\)

Old "spontaneous" endo splits into phagocytosis (phago, often still pronounced endo) and pinocytosis (pino).
Brane Reactions (Cartoons)

A Turing-Complete language [Busi Gorrieri]
\[ \forall_n \sigma \mid \sigma' \langle P \rangle \circ \forall_n' (\rho). \tau \mid \tau' \langle Q \rangle \rightarrow \tau \mid \tau' \langle \rho \langle \sigma \mid \sigma' \langle P \rangle \rangle \rangle \circ Q \]

\[ \forall_n' \tau \mid \tau' \langle \forall_n . \sigma \mid \sigma' \langle P \rangle \rangle \circ Q \rightarrow P \circ \sigma \mid \sigma' \mid \tau \mid \tau' \langle Q \rangle \]

\[ \odot (\rho) . \sigma \mid \sigma' \langle P \rangle \rightarrow \sigma \mid \sigma' \langle \rho \odot \odot \odot P \rangle \]

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

\[ p_1 \circ p_1 (p_2) \Rightarrow q_1 (q_2) . \alpha \mid \sigma \langle p_2 \circ P \rangle \rightarrow q_1 \circ \alpha \mid \sigma \langle q_2 \circ P \rangle \]

(multiset rewriting, inside and outside membranes)
N.B.: in Phago (and Pino), one could perhaps require \( r \) to be, conservatively, a piece of \( t \), by a non-linear rewrite:

\[
\text{CPhago} \quad \varrho_n.\sigma|\sigma'(P) \circ \varrho_n^\perp(\rho).\tau'|\rho(Q) \rightarrow \tau|\tau'(\rho(\sigma|\sigma'(P))\circ Q)
\]
Derivable Reactions (Cartoons)

A Decidable-Termination language
[Busi Gorrieri]
Abbreviations: Mate

\[
\text{Mate } \text{mate}_n.\sigma = \otimes_n.\otimes_{n'.}\sigma \\
\text{mate}^{\perp}_n.\tau = \otimes_{n}(\otimes_{n'.}\otimes_{n''}).\otimes_{n''}.\tau
\]
Abbreviations: Bud

\[ \text{Bud} \quad \text{bud}_n^\sigma = \varphi_n^\sigma \]
\[ \text{bud}_n^{\perp}(\rho).\tau = \varphi(\varphi_n^{\perp}(\rho).\varphi_n^{\perp}).\varphi_n^{\perp}.\tau \]

A budding version of old "spontaneous" mito, to avoid arbitrary splits. Follows the pattern of inverse-mate.
Abbreviations: Drip

\[ \text{drip}_n(\rho).\sigma = \circ(\circ(\rho).\bigtriangledown_n).\bigtriangledown^\perp_n.\sigma \]

A zero-expelled-membranes version of old "spontaneous" mito, to avoid arbitrary splits. Follows the pattern of inverse-mate.
Ex: Viral Infection

- **virus**
  - \(\ominus \ominus (\text{nucap})\)
  - \(\ominus (\text{mate})\)
  - \(\ominus (\text{mate})\)

- **cell**
  - \(\ominus \ominus (\text{mate})\)
  - \(\ominus (\text{mate})\)
  - \(\ominus (\text{mate})\)
  - \(\ominus (\text{mate})\)
  - \(\ominus (\text{mate})\)

- **membrane**
  - \(\ominus \ominus (\text{nucap})\)
  - \(\ominus (\text{mate})\)
  - \(\ominus (\text{mate})\)

- **vesicle**
  - \(\ominus \ominus (\text{nucap})\)
  - \(\ominus (\text{mate})\)
  - \(\ominus (\text{mate})\)

- **endosome**
  - \(\ominus \ominus (\text{nucap})\)
  - \(\ominus (\text{mate})\)
  - \(\ominus (\text{mate})\)

- **Phago**
- **Mate**
- **Exo**
Ex: Viral Progeny

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

Then:
Brane-Molecule Reactions (Cartoons)

With *molecule multisets* \( p, q \):
We now add *molecules* to the model:

**Systems**
\[ P, Q ::= ... | m \]
\[ p, q ::= m_1 \circ \ldots \circ m_k \]

**Actions**
\[ a ::= ... | p_1(p_2) \Rightarrow q_1(q_2) \]

This single operation can essentially account for the whole Protein Machine, including its interactions with membranes. Except that, one must add some form of protein complexation, either as in BioSPi by adding restriction, or as in \( \kappa \)-calculus by adding complex molecules.
\[
\text{B&R} \quad p_1 \circ (p_2) \Rightarrow q_1(q_2).\alpha|\sigma(p_2 \circ P) \Rightarrow q_1 \circ \alpha|\sigma(q_2 \circ P)
\]

(multiset rewriting, inside and outside membranes)

Simple bindings and releases - “\(\diamond(\diamond)\)” is omitted:

- \(m(\diamond) \Rightarrow \text{bind out} \quad \Rightarrow m(\diamond) \quad \text{release out}\)
- \(\diamond(m) \Rightarrow \text{bind in} \quad \Rightarrow \diamond(m) \quad \text{release in}\)
Ex: Molecular Pumps and Channels

Proton Pump
ATP charges up the vacuole with $H^+$. Several other pumps work off that charge.

Ion Channel

Proton Antiporter

A plant vacuole membrane has all those things on it.
ProtonPump = ! ATP(\diamond) \Rightarrow ADP \circ P_i (H^+ \circ H^+)
IonChannel = ! Cl^- (H^+) \Rightarrow \diamond (H^+ \circ Cl^-)
ProtonAntiporter = ! Na^+ (H^+) \Rightarrow H^+(Na^+)

PlantVacuole = 
ProtonPump \mid IonChannel \mid ProtonAntiporter (\diamond \diamond)

Hence this reaction notation, \Rightarrow, is “like” chemical reaction notation, \rightarrow, but talking about both sides on a membrane at once.

(N.B. no built-in conservation of mass in either case.)
Special Cases of B&R

**Chemical reaction catalysis** (inside a compartment)

\[ p \rightarrow q \triangleq ! p(\diamond) \Rightarrow q(\diamond) \sqcup \]  
\[ p \leftrightarrow q \triangleq p \rightarrow q \circ q \rightarrow p \]

E.g. peptide bond between two aminoacids \( R^1 R^2 \):
\[ R^1-COOH \circ H_2N-R^2 \rightarrow R^1-CO-HN-R^2 \circ H_2O \]

---

**Compartment conditions** (on the membrane of a compartment)

\[ p \rightarrow q \triangleq ! \diamond (p) \Rightarrow \diamond (q) \]
\[ p \rightarrow q | \sigma (p) \]

Condition affecting \( P \)

E.g. a condition-driven reaction:
\[ p \rightarrow q | \sigma (p) \rightarrow p \rightarrow q | \sigma (q) \]
Ex: Virus Replication

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

\[ ER \triangleq \text{!vRNA}(\diamond) \Rightarrow \text{vRNA}(\diamond). \text{drip}(\text{nucap budding receptor}) (\text{Nucleus}) \]

when triggered by vRNA

exo to cell membrane

virus membrane

(See paper for the other two vRNA pathways)
“On Brane” vs. “In Brane”

- 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.

Awkward encoding. And all kinds of things can go wrong in the intermediate state.
Adding Frills to the Framework

- **So far, purely combinatorial:**
  - No name binding, channel creation, communication...
  - Closer to combinatorial flavor of protein interactions
  - Goes a long way: do not try to extend needlessly.

- **But one can easily add all that, and more:**
  - **CCS-style communication**
    - Diffusion of molecules on cellular membrane
  - **BioAmbients-style communication**
    - Diffusion of molecules across cellular membrane
  - **BioAmbients-like mobility**
    - Non-bitonal
  - **π-style restriction**

- **We have a framework where we can plug&play a rich set of interactions, while supporting compartments.**
Towards the Million-Line Model
From Chemical Reactions to ODE's

r₁: A+B →ₖ₁ C+C
r₂: A+C →ₖ₂ D
r₃: C →ₖ₃ E+F
r₄: F →ₖ₄ B

Write the coefficients by columns

Read the concentration changes from the rows

Read the rate laws from the columns

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

\[ \begin{align*}
\frac{d[A]}{dt} &= -v₁ - v₂ \\
\frac{d[B]}{dt} &= -v₁ + v₄ \\
\frac{d[C]}{dt} &= 2 \cdot v₁ - v₂ - v₃ \\
\frac{d[D]}{dt} &= v₂ \\
\frac{d[E]}{dt} &= v₃ \\
\frac{d[F]}{dt} &= v₃ - v₄ \\
\end{align*} \]

E.g. \[ \frac{d[A]}{dt} = -k₁[A]⋅[B] - k₂[A]⋅[C] \]

<table>
<thead>
<tr>
<th>N</th>
<th>r₁</th>
<th>r₂</th>
<th>r₃</th>
<th>r₄</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-1</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>B</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>-1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td>D</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>E</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ v = \begin{align*} v₁ &= k₁[A]⋅[B] \\
v₂ &= k₂[A]⋅[C] \\
v₃ &= k₃[C] \\
v₄ &= k₄[F] \end{align*} \]

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

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

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.

Write the coefficients by columns

<table>
<thead>
<tr>
<th>N</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></td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>-1</td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>2</td>
<td>-1</td>
<td>-1</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td></td>
<td></td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>F</td>
<td></td>
<td></td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

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.

Read the process interactions from the rows

(Rate laws are implicit in stochastic semantics)
• **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 stochiometric matrix (and hence without ODE’s).
Complexes: The ODE Way

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 ($[ABC] + [A_p BC] = \text{constant}$, etc.).

**System description is exponential in the number of basic components.**

**Stoichiometric Matrix**

| N    | v_1 | v_2 | v_3 | v_4 | v_5 | v_6 | v_7 | v_8 | v_9 | v_{10} | v_{11} | v_{12} | v_{13} | v_{14} | v_{15} | v_{16} | v_{17} | v_{18} | v_{19} | v_{20} | v_{21} | v_{22} | v_{23} | v_{24} |
|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| ABC  |     |     |     |     |     |     |     |     |     |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| A_pBC|     |     |     |     |     |     |     |     |     |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| ABpC |     |     |     |     |     |     |     |     |     |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| ABCp |     |     |     |     |     |     |     |     |     |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| ApBC |     |     |     |     |     |     |     |     |     |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| ABpCp|     |     |     |     |     |     |     |     |     |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| ApBCp|     |     |     |     |     |     |     |     |     |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| ABpCp|     |     |     |     |     |     |     |     |     |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |
| ApBCp|     |     |     |     |     |     |     |     |     |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |        |

$2^n \times 2n(2^n-1)$
Complexes: The Reactive System Way

\[ A \rightleftharpoons A_p \]
\[ B \rightleftharpoons B_p \]
\[ C \rightleftharpoons C_p \]

\[ 2n \quad \text{(domain reactions)} \]

\[ A = ?kn;A_p \quad A_p = ?ph;A \]
\[ B = ?kn;B_p \quad B_p = ?ph;B \]
\[ C = ?kn;C_p \quad C_p = ?ph;C \]

\[ 2n \quad \text{(processes)} \]

\[ n \quad A \mid B \mid C \]

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, but its description does not.)
Model Validation
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** [Lecaa, 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 π-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 Wicklicky].
Model Validation: Modelchecking

- **Temporal**
  - Software verification of biomolecular systems (NA pump) [Ciobanu]
  - Analysis of mammalian cell cycle (after Kohn) in CTL. [Chabrier-Rivier Chiaverini Danos Fages Schachter]
    - 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.
    - \( \text{Eventually(Always (PRPP = 1.7 \times PRPP1))} \)
    - \( \text{implies} \)
    - \( \text{steady\_state()} \)
    - \( \text{and Eventually(Always(IMP < 2 \times IMP1))} \)
    - \( \text{and Eventually(Always(hx\_pool < 10\times hx\_pool1))} \)

- **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 (modelchecking).

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 modelchecking).
- 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