Abstract Machines of Systems Biology

Luca Cardelli

Microsoft Research
Cambridge UK

2005-06-12..16 Dobbiaco

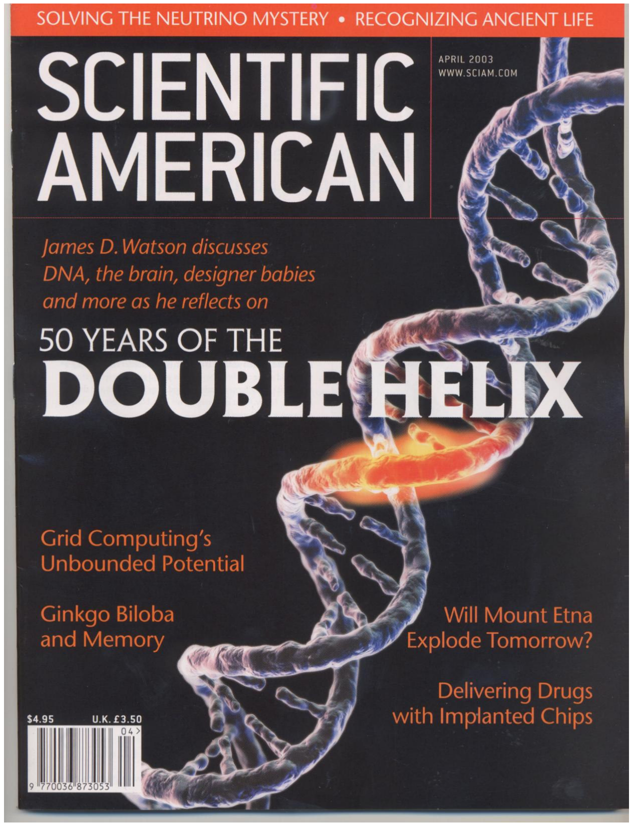
www.luca.demon.co.uk/BioComputing.htm

50 Years of <u>Molecular Cell Biology</u>

- 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


SOLVING THE NEUTRINO MYSTERY • RECOGNIZING ANCIENT LIFE
SCIENTIFIC AMERICAN
APRIL 2003
WWW.SCIAM.COM
James D. Watson discusses DNA, the brain, designer babies and more as he reflects on
50 YEARS OF THE
DOUBLE HELIX
Grid Computing's Unbounded Potential
Ginkgo Biloba and Memory
Will Mount Etna Explode Tomorrow?
Delivering Drugs with Implanted Chips
$4.95
U.K. £3.50

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.

Storing *Processes*

- Today we represent, store, search, and analyze:
  - Gene sequence data
  - Protein structure data
  - Metabolic network data
  - Signaling pathway data
  - …

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

- 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.

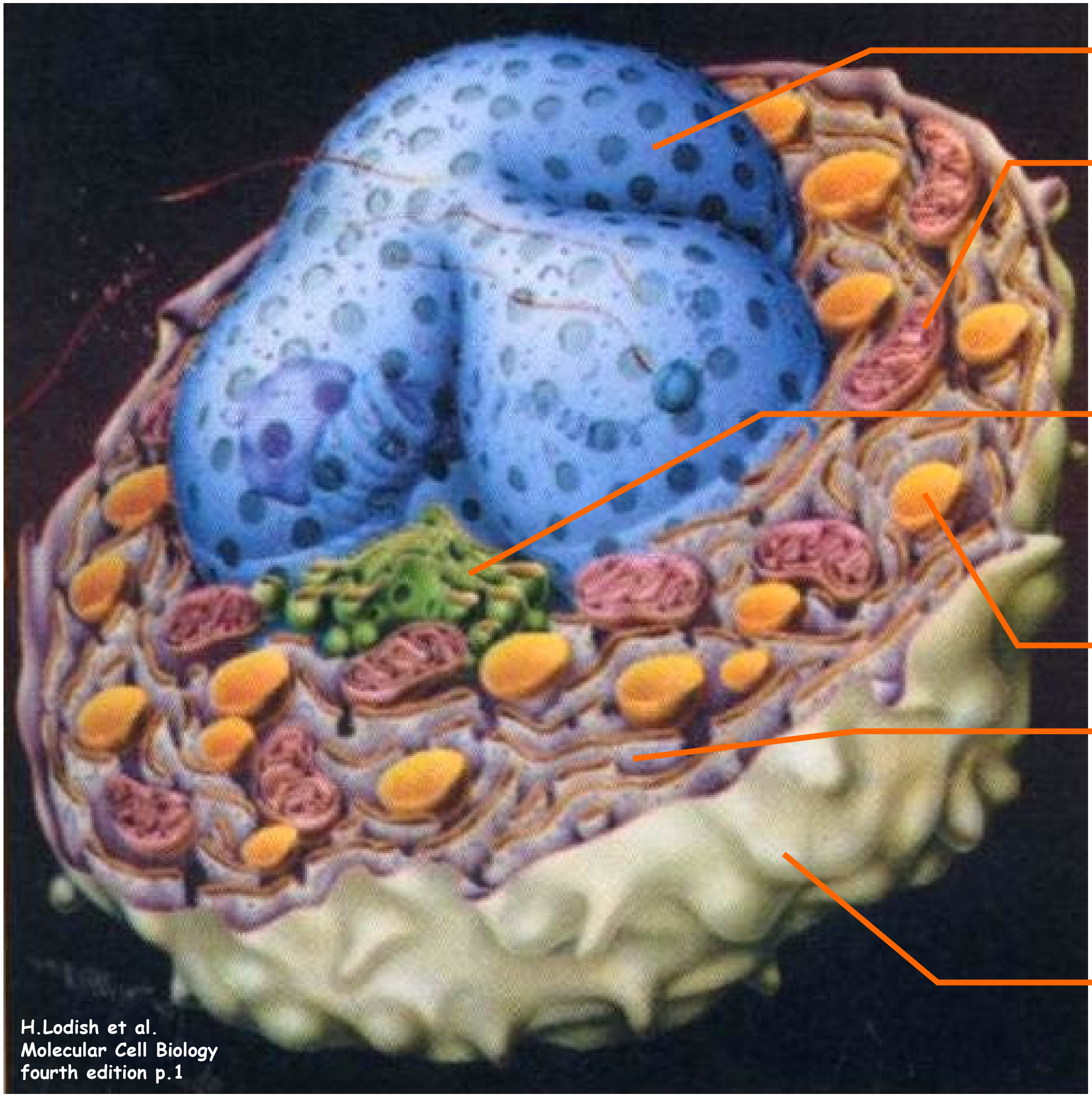
- In computing…
  - There are well-established scalable representations of dynamic reactive processes.
  - They look more or less like little, mathematically based, programming languages.

Structural Architecture

Eukaryotic Cell

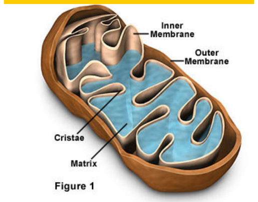
(10~100 trillion in human body)

Membranes everywhere

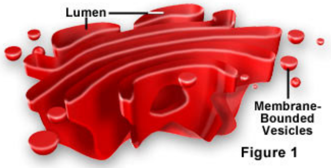

H.Lodish et al.
Molecular Cell Biology
fourth edition p.1

Nuclear membrane

Mitochondria

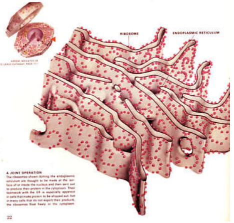


Golgi



Vesicles

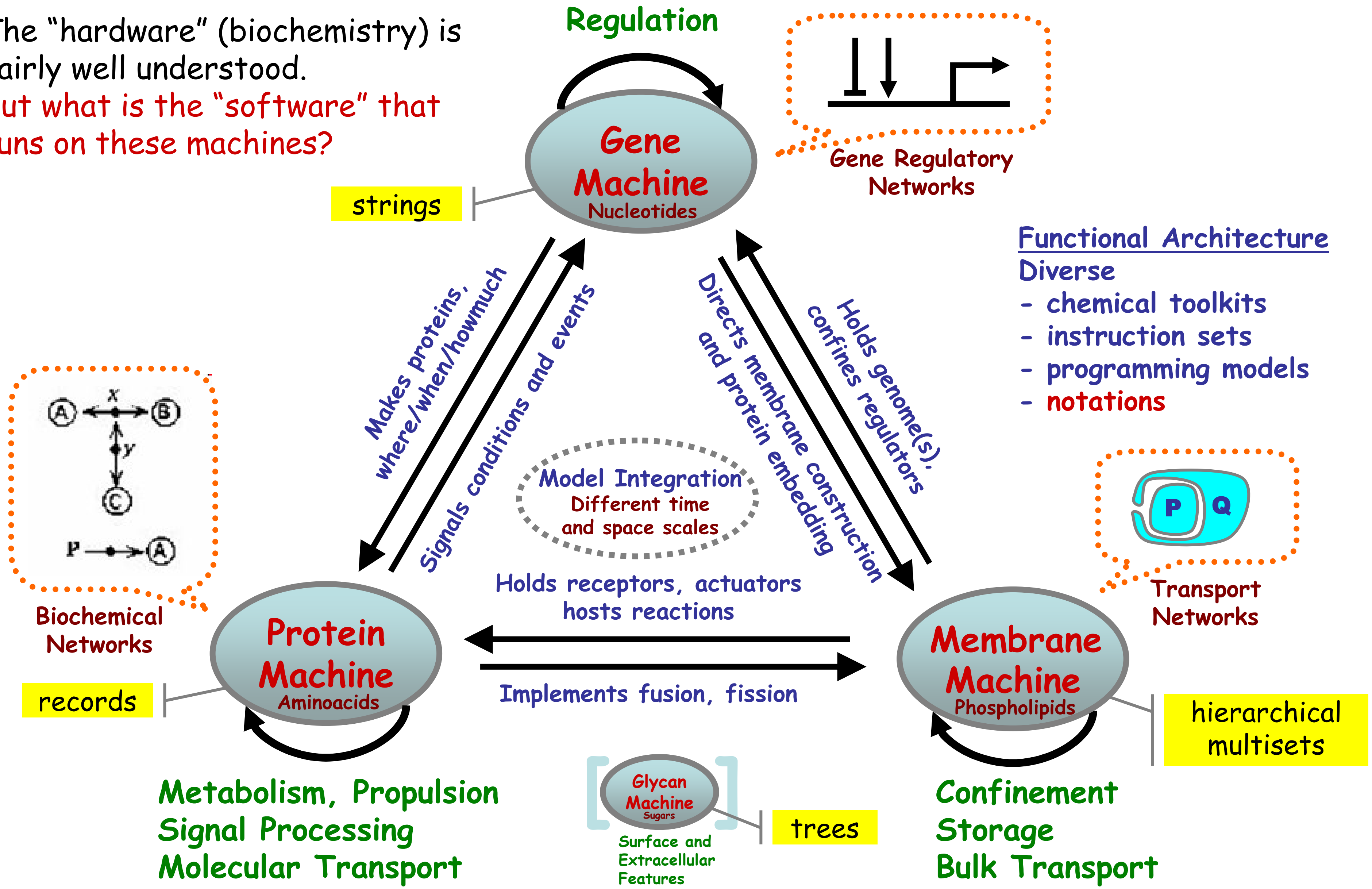
E.R.



Plasma membrane (<10% of all membranes)

Abstract Machines of Systems Biology

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



Regulation

Gene Machine
Nucleotides

strings

Gene Regulatory Networks

Makes proteins, where/when/howmuch

Signals conditions and events

Directs membrane construction and protein embedding

Holds genome(s), confines regulators

Model Integration
Different time and space scales

Biochemical Networks

Protein Machine
Aminoacids

records

Holds receptors, actuators hosts reactions

Implements fusion, fission

Transport Networks

Membrane Machine
Phospholipids

hierarchical multisets

Metabolism, Propulsion
Signal Processing
Molecular Transport

Glycan Machine
Sugars

Surface and Extracellular Features

trees

Confinement
Storage
Bulk Transport

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

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]

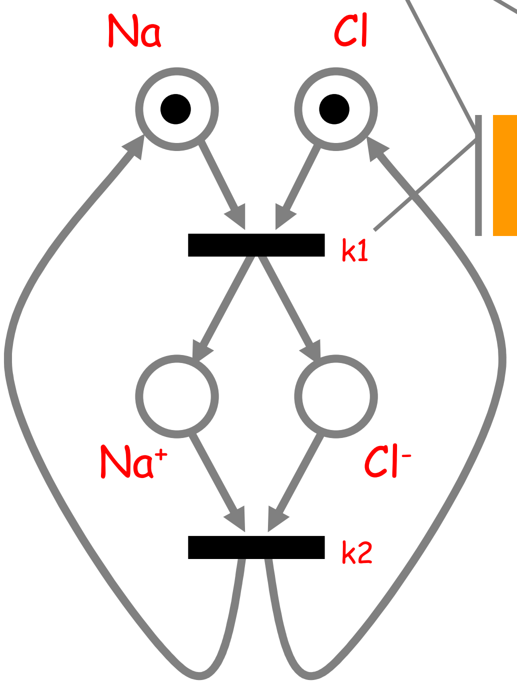
- Many biological systems exhibit features of reactive systems:
  - Deep layering of abstractions
  - Complex composition of simple components
  - Discrete transitions between states
  - Digital coding and processing of information
  - Reactive information-driven behavior
  - High degree of concurrency and nondeterminism
  - “Emergent behavior” not obvious from part list

Chemistry vs. π-calculus

A process calculus (chemistry, or SBML)

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

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



1 line per reaction

Reaction oriented

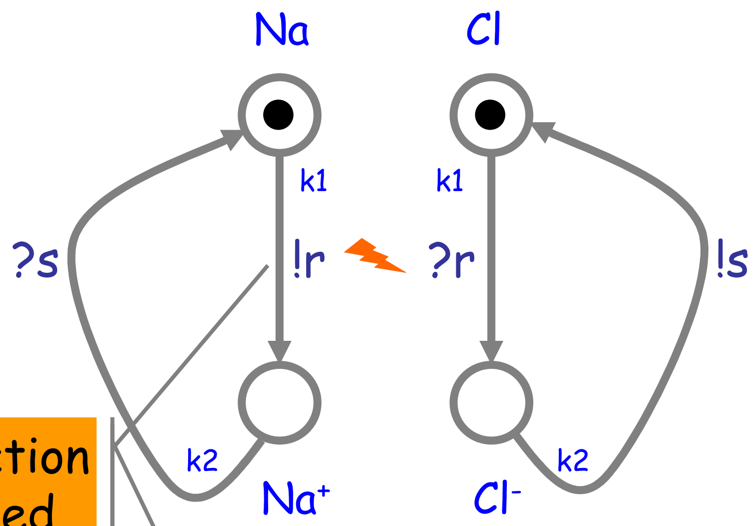
This Petri-Net-like graphical representation degenerates into spaghetti diagrams: precise and dynamic, but not scalable, structured, or maintainable.

The same “model”

Maps to a CTMC

Maps to a CTMC

A compositional graphical representation, and the corresponding calculus.



Interaction oriented

1 line per component

$$Na = !r_{k1};\ \overbrace{?s_{k2};\ Na}^{Na^{+}}$$

$$Cl = \ ?r_{k1};\ \underbrace{!s_{k2};\ Cl}_{Cl^{-}}$$

A different process calculus (π)

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
  - Modelchecking
    - Standard, Quantitative, Probabilistic

Basic Modeling Guidelines

- Regev-Shapiro: “Molecules as Processes”:

Molecule	Process
Interaction capability	Channel
Interaction	Communication
Modification (of chemical components)	State change (state-transition systems)

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

- They chose π-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 (π) 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)

π-calculus (a Process Algebra)

- Processes P,Q,... - components of a system
- Channels a,b,... - interactions between components

0 the process that does nothing

!a(b); P the process that outputs b on channel a (and then does P)

?a(x); P the process that inputs b on channel a (and then does P{x})

P | Q the process made of subprocesses P and Q running concurrently

P + Q the process that behaves like either P or Q nondeterministically

*P the process that behaves like unboundedly many copies of P

=> recursive processes

=> unbounded number and species of processes

new x; P the process that creates a new channel x (and then does P{x})

=> private interactions

=> unbounded number and species of interactions

π-calculus (a Process Algebra)

- Dynamics

$$(\underline{!a(b)};\ P) + P' \quad | \quad (\underline{?a(x)};\ Q\{x\}) + Q' \quad \to \quad P \quad | \quad Q\{b\}$$

$$\begin{array}{lllll} \text{Ex.} & & !a(b);\ ?b & | & ?a(x);\ (!x + ?b) \\ & \to & ?b & | & !b + ?b \\ & \to & 0 & | & 0 \\ & = & 0 & & \end{array}$$

- “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

$$\begin{array}{lll} \pi & ::= & x(y) \quad \text{receive } y \text{ along } x \\ & & \overline{x}\langle y\rangle \quad \text{send } y \text{ along } x \end{array}$$

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

Structural congruence

Renaming of bound variables

$$\begin{array}{rcll} x(y).P & = & x(z).(\{z/y\}\,P) & \text{if } z \notin FN(P) \\ (\mathsf{new}\,y).P & = & (\mathsf{new}\,z).(\{z/y\}\,P) & \text{if } z \notin FN(P) \end{array}$$

Structural congruence laws

$P\vert Q$	$\equiv$	$Q\vert P$		commutativity of parallel composition
$(P\vert Q)\vert R$	$\equiv$	$P\vert(Q\vert R)$		associativity of parallel composition
$P+Q$	$\equiv$	$Q+P$		commutativity of summation
$(P+Q)+R$	$\equiv$	$P+(Q+R)$		associativity of summation
$(\mathsf{new}\,x)0$	$\equiv$	0		restriction of inert processes
$(\mathsf{new}\,x)(\mathsf{new}\,y)P$	$\equiv$	$(\mathsf{new}\,y)(\mathsf{new}\,x)P$		polyadic restriction
$((\mathsf{new}\,x)P)\vert Q)$	$\equiv$	$(\mathsf{new}\,x)(P\vert Q)$	if $x \notin FN(Q)$	scope extrusion
$!P$	$\equiv$	$P\vert !P$		replication

Reaction rules

$$(\cdots + \overline{x}\langle z\rangle.Q)|(\cdots + x(y).P) \to Q|P\,\{z/y\}$$ communication (COMM)

$$\frac{P \to P'}{P|Q \to P'|Q}$$ reaction under parallel composition (PAR)

$$\frac{P \to P'}{(\mathsf{new}\,x)P \to (\mathsf{new}\,x)P'}$$ reaction under restriction (RES)

$$\frac{Q \equiv P \;\; P \to P' \;\; P' \equiv Q'}{Q \to Q'}$$ structural congruence (STRUCT)

Syntax

Chemical Mixing

Reactions

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).

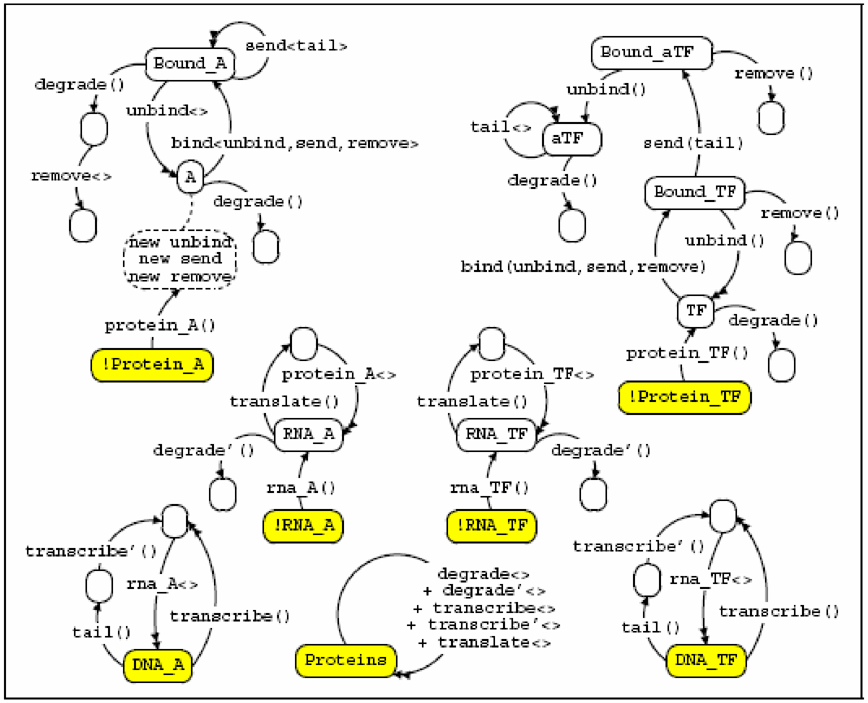


Figure 2. Regulating Gene Expression by Positive Feedback [9]

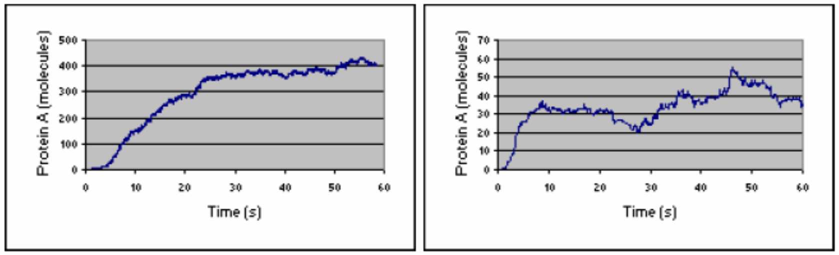


Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of TF

A.Phillips, L.Cardelli. BioConcur'04.

Stochastic π-calculus

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

a.P 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

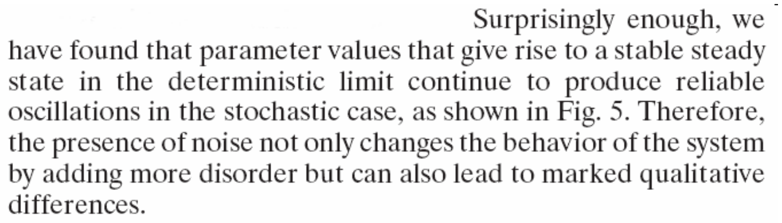
- Continuous Time Markov Chains
  - Finite State Machines, with state transition times exponentially distributed (memoryless)
  - Well studied class of stochastic processes
  - Efficient analysis algorithms for stationary and transient analysis

- 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.



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.

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

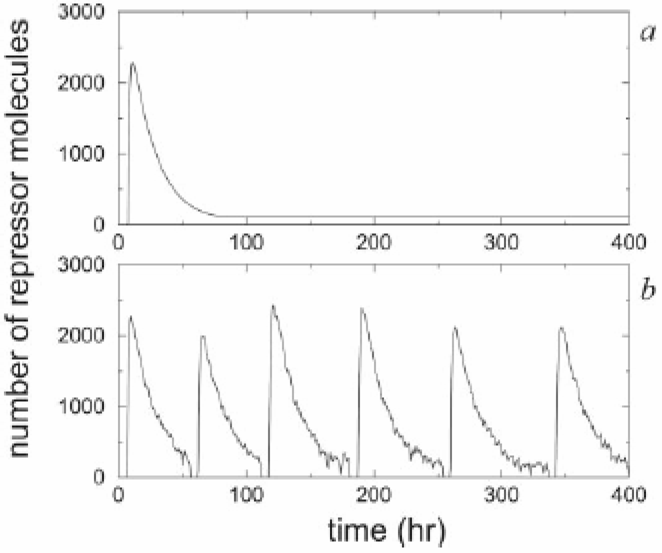


Fig. 5. Time evolution of R for the deterministic Eq. [1] (a) and stochastic (b) versions of the model. The values of the parameters are as in the caption of Fig. 1, except that now we set $\delta_R = 0.05\ \mathrm{h}^{-1}$. For these parameter values, $\tau < 0$, so that the fixed point is stable.

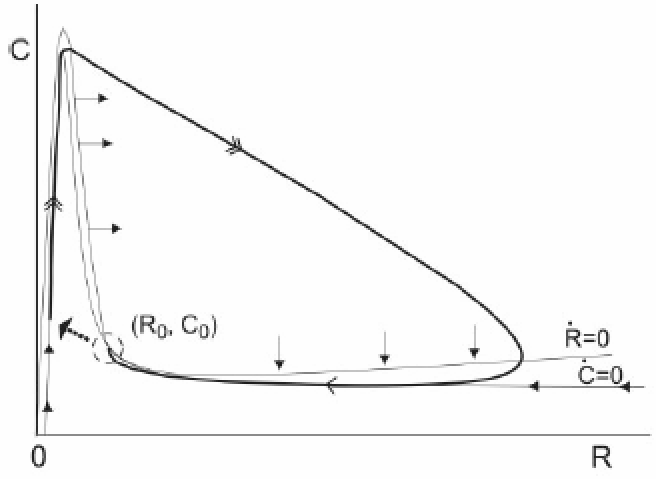
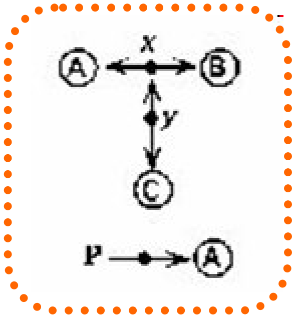


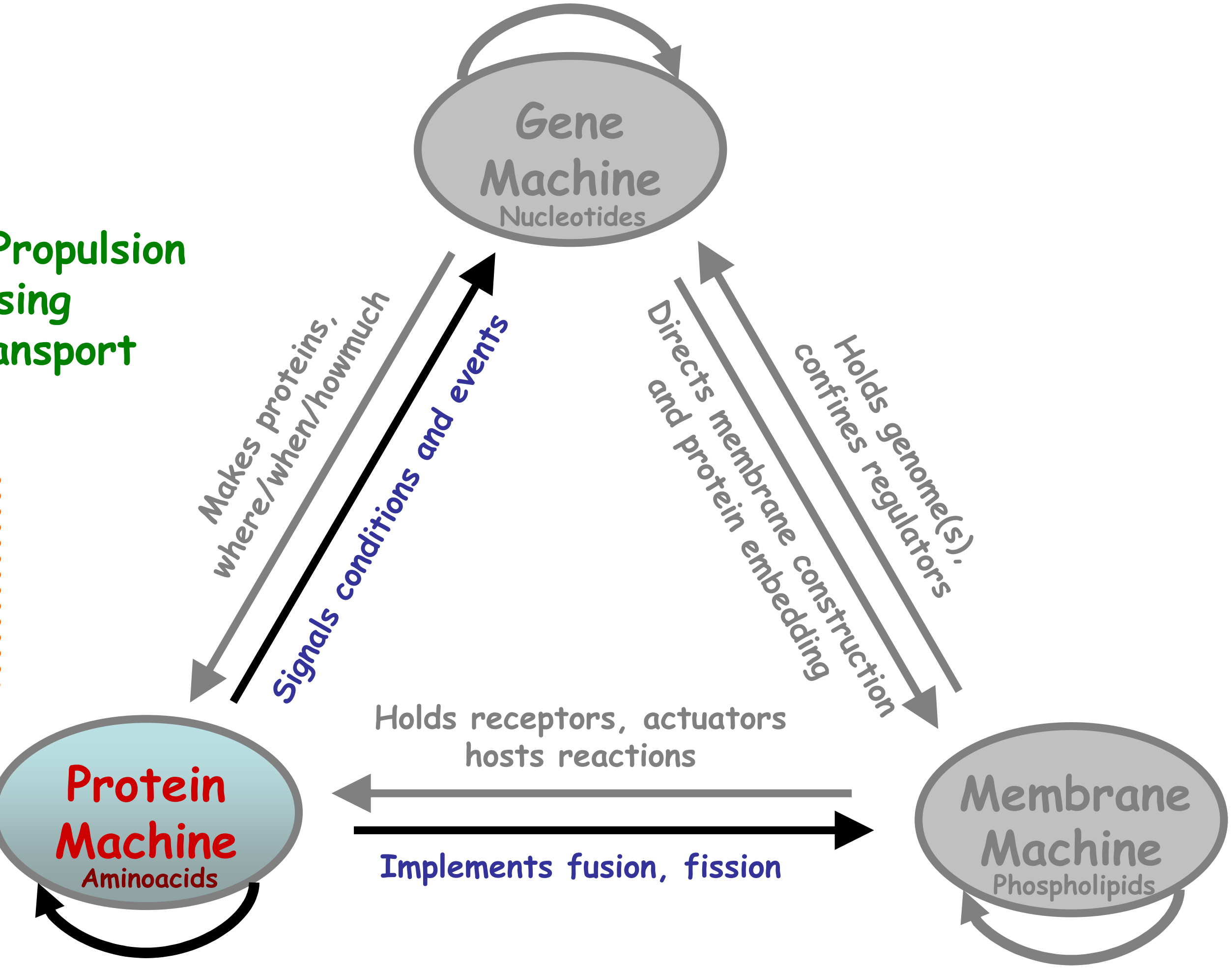
Fig. 6. Phase portrait as in Fig. 4 but for a situation in which the system falls into the stable fixed point (R_0, C_0). The dotted arrow to the left of the fixed point illustrates a perturbation that would initiate a single sweep of the (former) oscillatory trajectory.

Protein Networks

Metabolism, Propulsion
Signal Processing
Molecular Transport



Biochemical Networks



Gene Machine
Nucleotides

Makes proteins, where/when/howmuch

Signals conditions and events

Directs membrane construction and protein embedding

Holds genome(s), confines regulators

Protein Machine
Aminoacids

Holds receptors, actuators hosts reactions

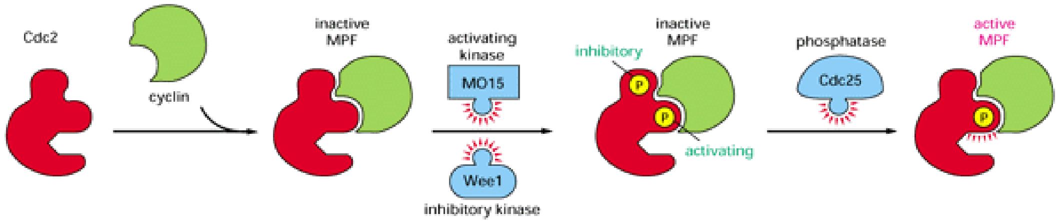
Implements fusion, fission

Membrane Machine
Phospholipids

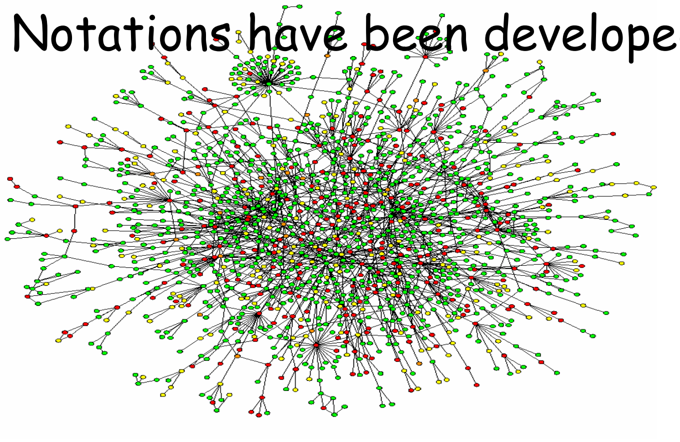
1. The Protein Machine

Very close to the atoms.

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



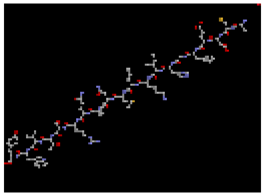
- Mapping out such networks:
  - In principle, it’s “just” a very large set of chemical equations.
  - Notations have been developed to summarize and abstract.



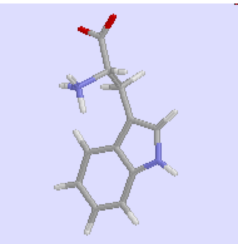
An actual molecular interaction network.
(Nodes are distinct protein kinds,
arcs mean that two kinds of proteins interact.)

Protein Structure

Primary

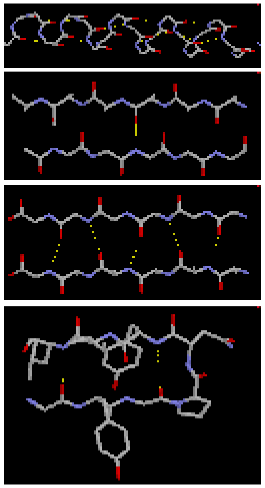


The 20 Aminoacids



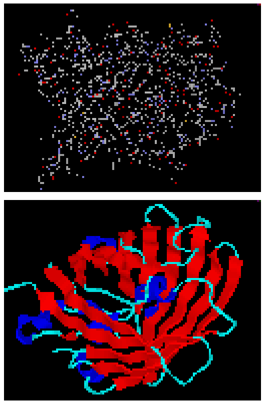
Tryptophan

Secondary



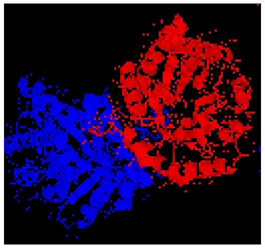
Alpha Helix, Beta Sheet

Tertiary



Green Fluorescent Protein

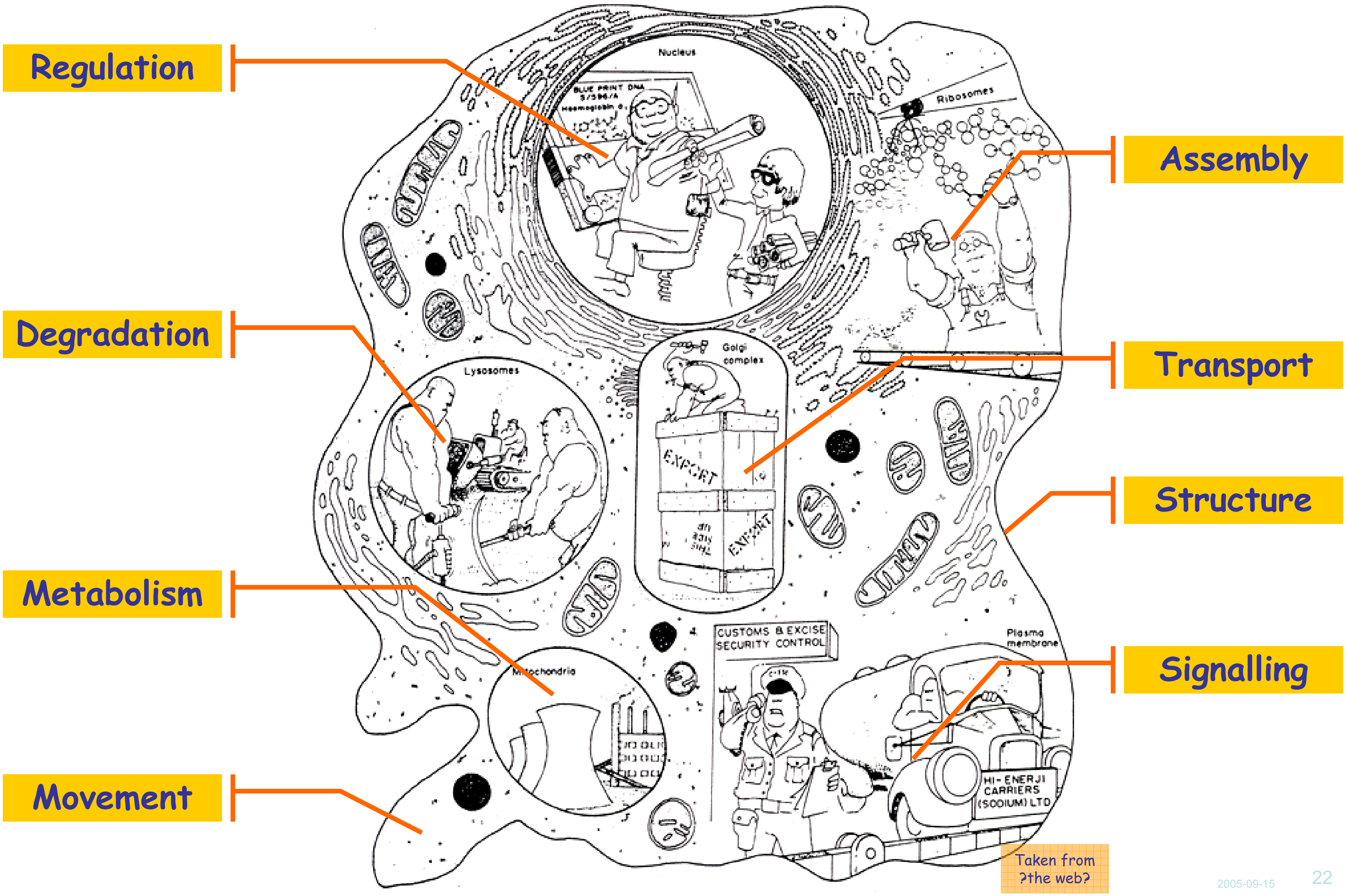
Quaternary



Triose Phosphate Isomerase

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

Protein Function



Regulation

Degradation

Metabolism

Movement

Assembly

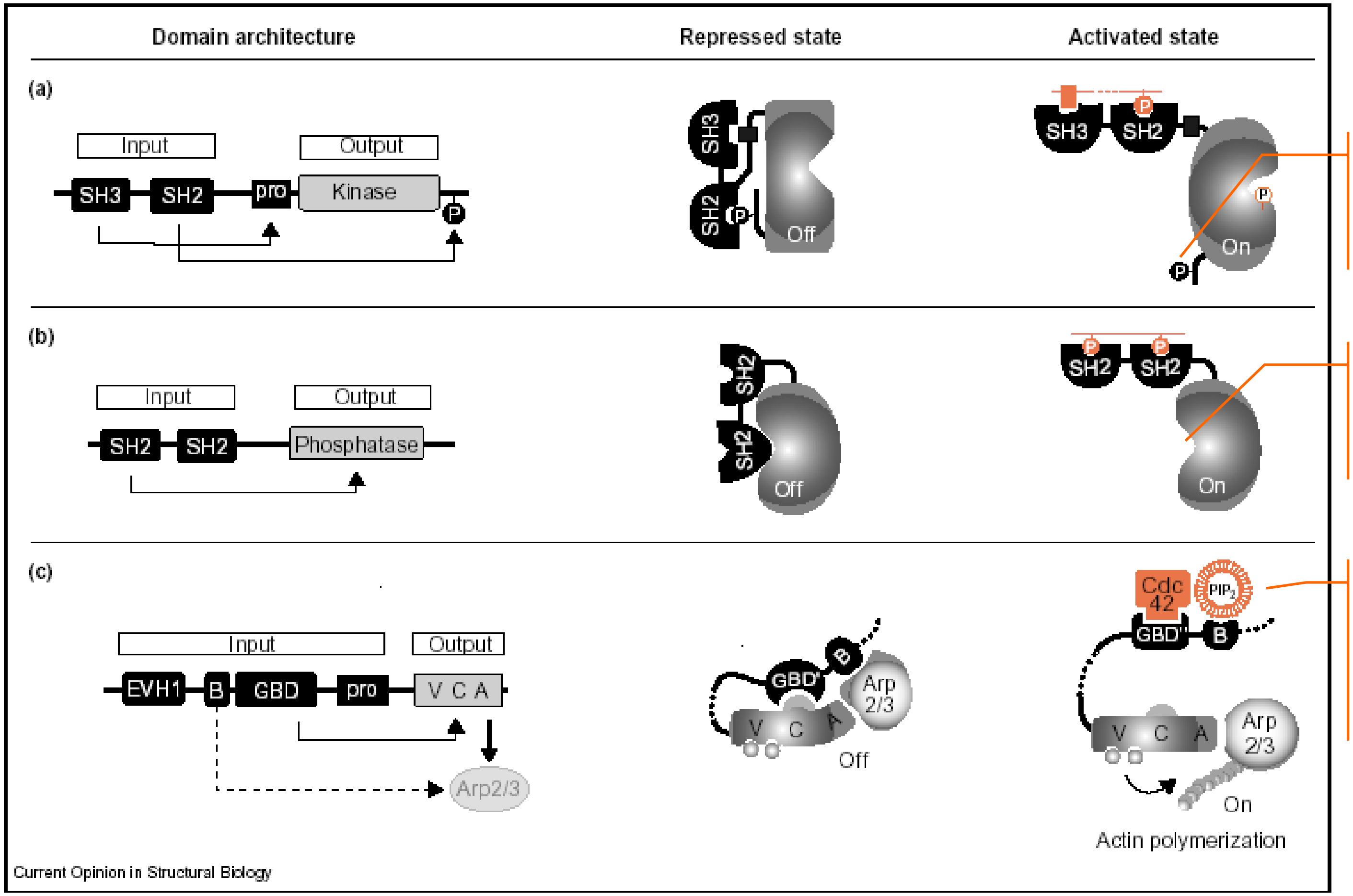
Transport

Structure

Signalling

Taken from ?the web?

Some Allosteric Switches



Current Opinion in Structural Biology

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 VASP homology 1 (EVH1) domain, a B motif, a GBD, a proline-rich segment (pro) and an output region (VCA) that alone binds the Arp2/3 complex and stimulates its actin nucleation activity. The B and GBD motifs are required to repress activity and, by current models, are thought to participate in intracomplex interactions (only the structure of the GBD intramolecular complex for WASP is known). GTP-bound Cdc42 and PIP_2 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

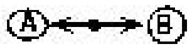
at equal concentrations of the individual input stimuli, activation is much higher if both stimuli are present

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

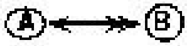
Humans have the same number of modular protein *domains* (building blocks) as worms, but twice the number of multi-domain proteins.

Taken from Wendell Lim

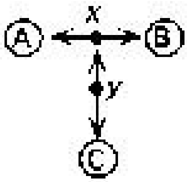
MIM: Molecular Interaction Maps (Kohn)



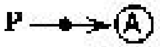
The double-arrowed line indicates that proteins **A** and **B** can bind to each other. The "node" placed on the line represents the **A:B** complex.



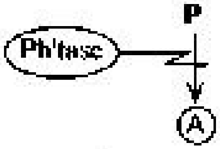
Asymmetric binding where protein **A** donates a peptide that binds to a receptor site or pocket on protein **B**.



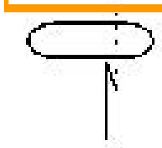
Representation of multimolecular complexes: *x* is **A:B**; *y* is (**A:B**):**C**. This notation is extensible to any number of components in a complex.



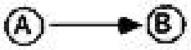
Covalent modification of protein **A**. The single-arrowed line indicates that **A** can exist in a phosphorylated state. The node represents the phosphorylated species.



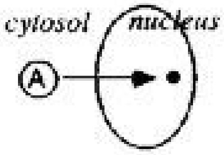
Cleavage of a covalent bond: dephosphorylation of **A** by a phosphatase.



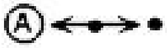
Proteolytic cleavage at a specific site within a protein.



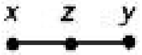
Stoichiometric conversion of **A** into **B**.



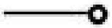
Transport of **A** from cytosol to nucleus. The node represents **A** after it has been transported into the nucleus.



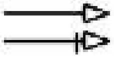
Formation of a homodimer. Filled circle on the right represents another copy of **A**. The node on the line represents the homodimer **A:A**.



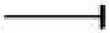
z is the combination of states defined by *x* and *y*.



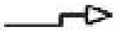
Enzymatic stimulation of a reaction.



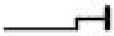
General symbol for stimulation.
A bar behind the arrowhead signifies necessity.



General symbol for inhibition.



Shorthand symbol for transcriptional activation.



Shorthand symbol for transcriptional inhibition.

Degradation products

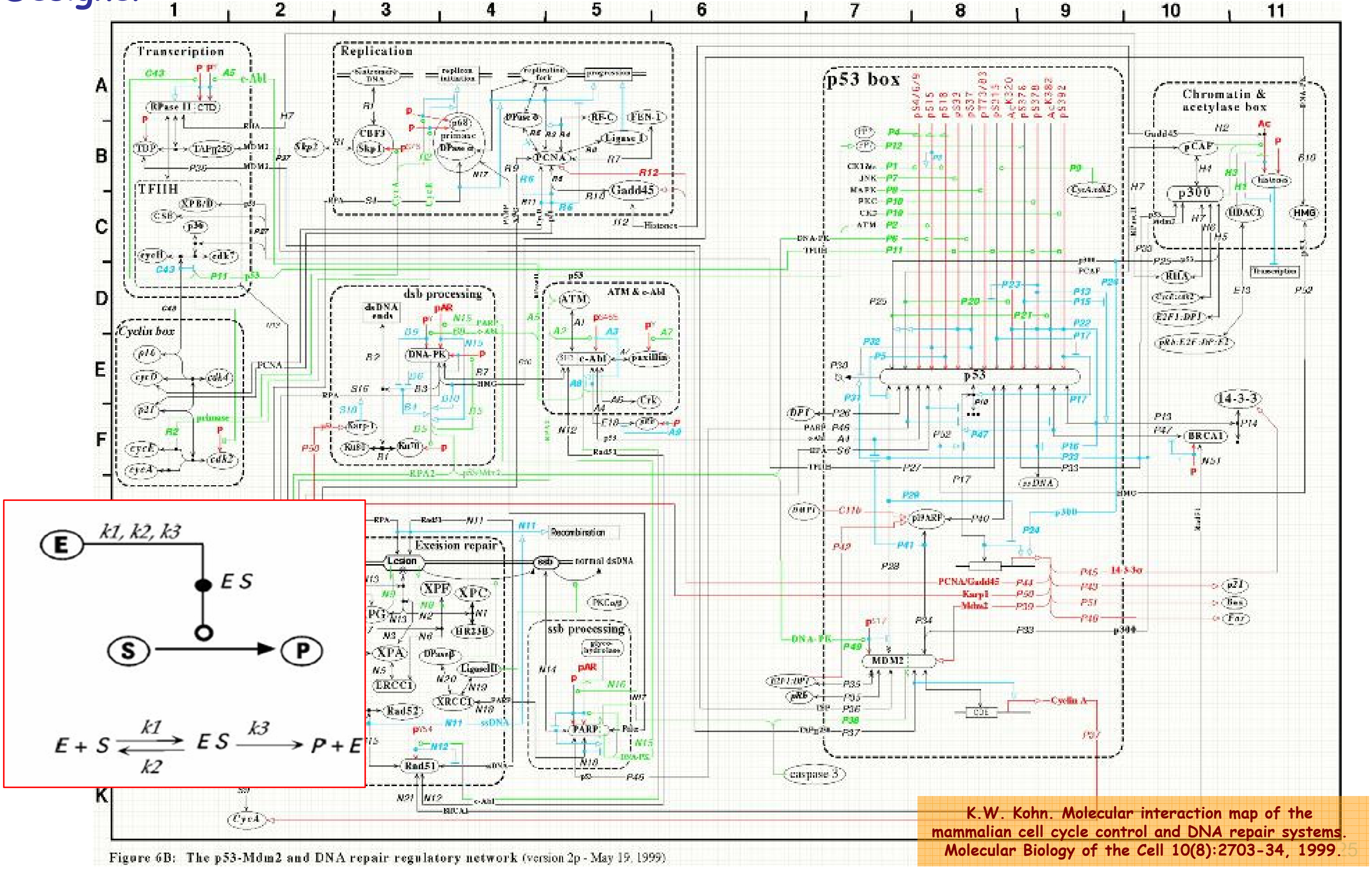
Taken from
Kurt W. Kohn

Molecular Interaction Maps

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

JDesigner

The p53-Mdm2 and DNA Repair Regulatory Network



$$E + S \underset{k2}{\overset{k1}{\rightleftharpoons}} ES \xrightarrow{k3} P + E$$

Figure 6B: The p53-Mdm2 and DNA repair regulatory network (version 2p - May 19, 1999)

K.W. Kohn. Molecular interaction map of the mammalian cell cycle control and DNA repair systems. Molecular Biology of the Cell 10(8):2703-34, 1999.

Kohn Diagrams

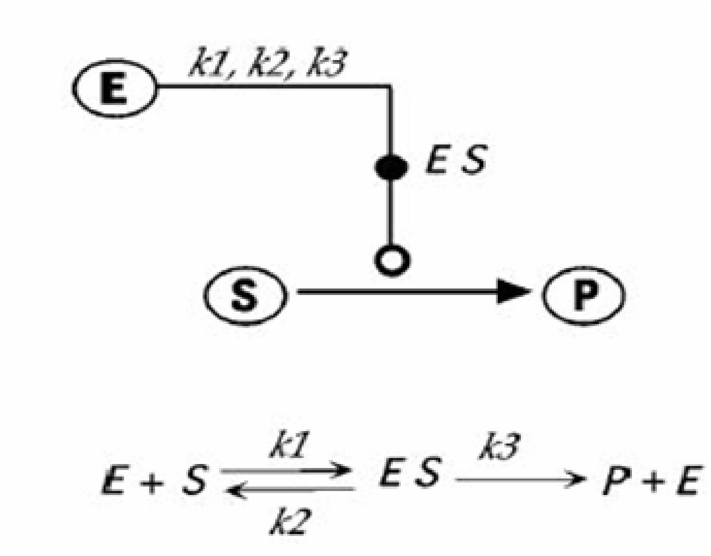


FIG. 3. Simple one-way enzymatic reaction. (If there is an energy source, such as ATP hydrolysis, it can be omitted when ATP concentration is not an important factor.) In explicit formulations, the reaction identifiers or rate constant designations can be placed on the enzyme reaction line, and the node ES can identify the enzyme-substrate species.

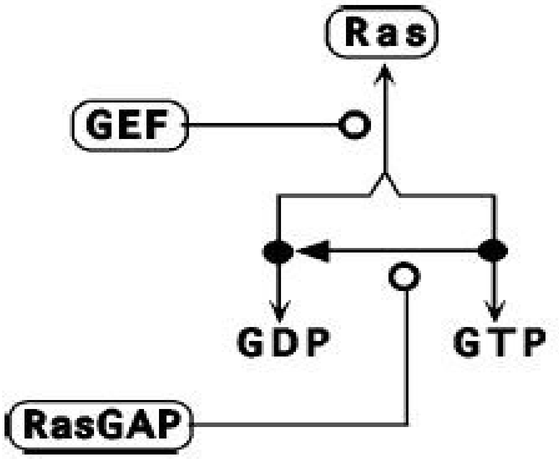


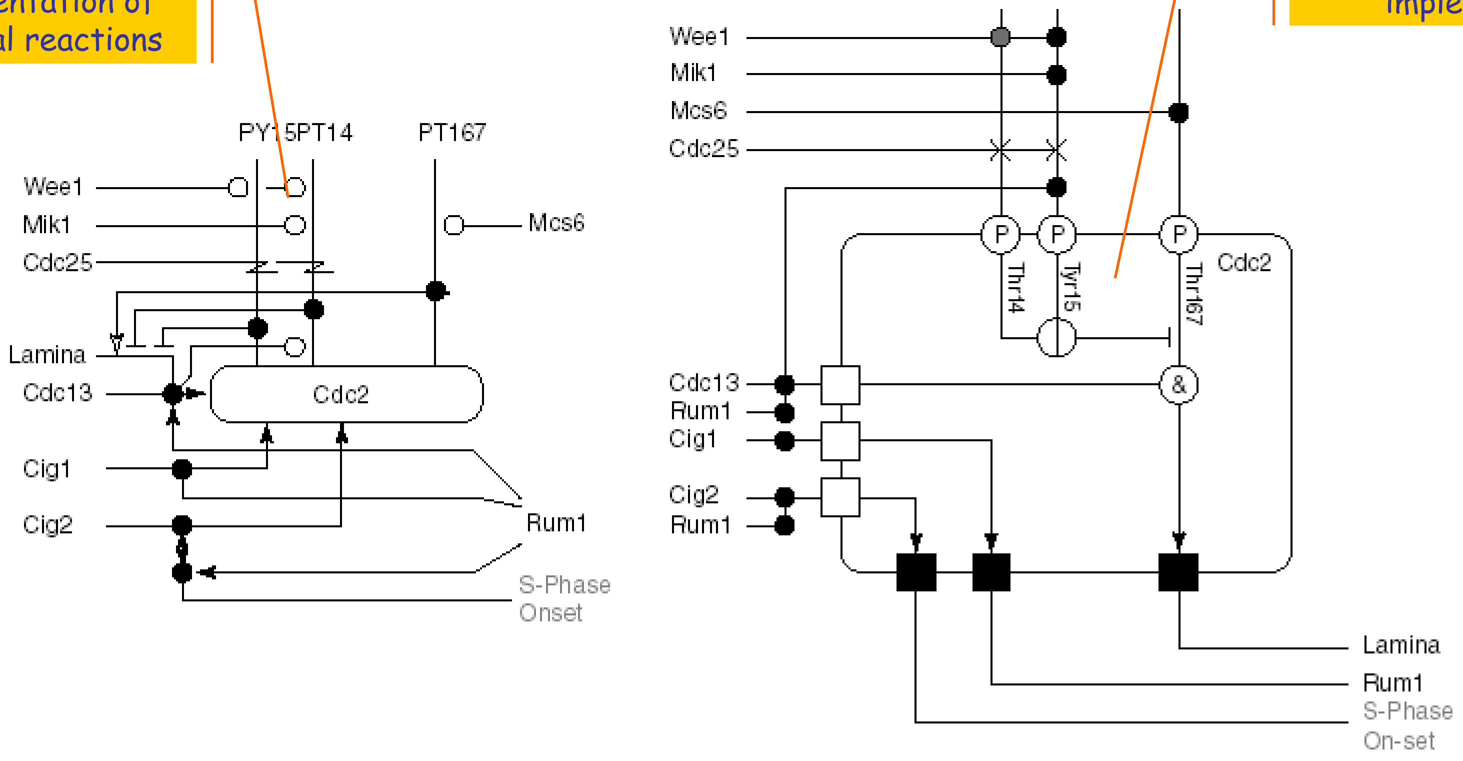
FIG. 4. Interconversions between the GTP- and GDP-bound states of Ras. (1) GDP and GTP compete with each other for binding to a site on Ras (this binding is only slowly reversible). (2) GEF (guanine nucleotide exchange factor) facilitates the binding or dissociation of GDP or GTP (the concentration of 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.)

Taken from
Kurt W. Kohn

Kitano Diagrams

From direct graphical representation of chemical reactions

To more abstract representation of the logic such reactions implement



(a) Graphical representation of fission yeast Cdc2 in Kohn diagram

(b) Proposed improvements of graphical representation of fission yeast Cdc2

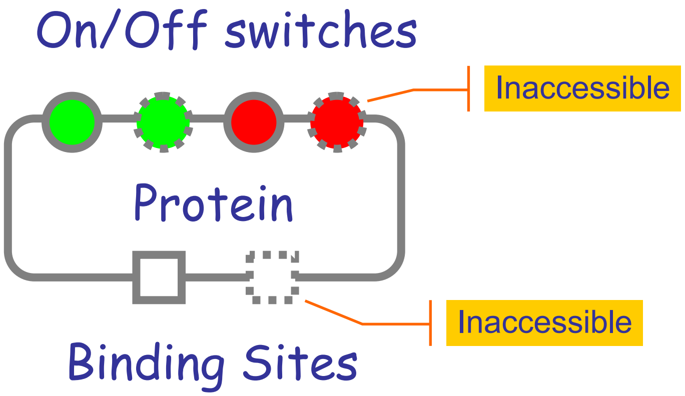
BioSilico

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 Try15 of its Cdc2. The complex of Cdc2 and Cig1 interacts with Rum1. Cdc2-Cdc13 complex and Cdc-Cig2 complex form heterotrimers involving Rum1.

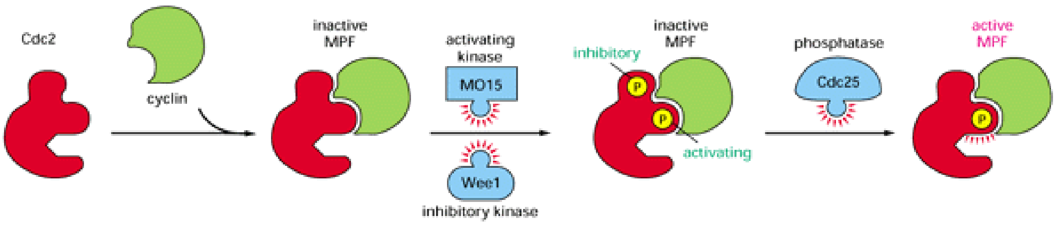
Taken from Hiroaki Kitano

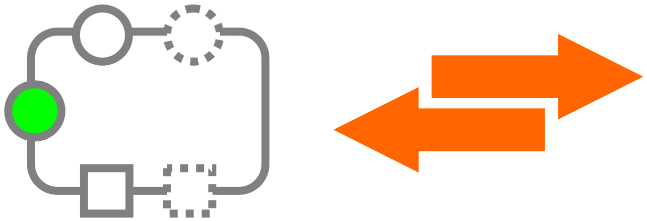
The Protein Machine "Instruction Set"

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



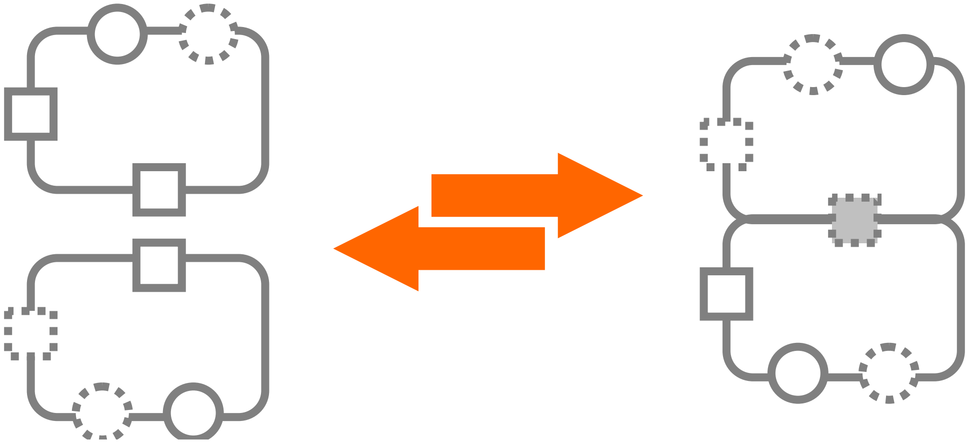
Each protein has a structure of binary switches and binding sites. But not all may be always *accessible*.





Switching of accessible switches.

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



Binding on accessible sites.

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

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

François & Hakim

Design of genetic networks with specified functions by evolution in silico
P. François, V. Hakim, Proc. Natl. Acad. Sci. USA, (101)2, 580-585, 2004.

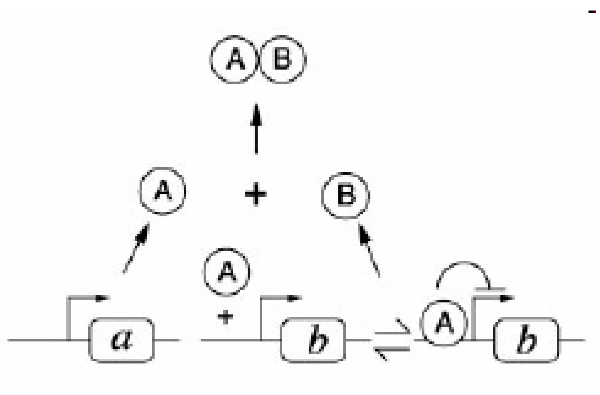
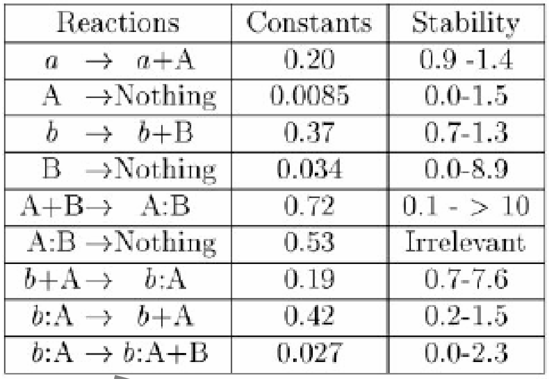


Fig 3A



Reactions	Constants	Stability
$a \rightarrow a$+A	0.20	0.9 -1.4
A →Nothing	0.0085	0.0-1.5
$b \rightarrow b$+B	0.37	0.7-1.3
B →Nothing	0.034	0.0-8.9
A+B→ A:B	0.72	0.1 - > 10
A:B →Nothing	0.53	Irrelevant
b+A→ b:A	0.19	0.7-7.6
b:A → b+A	0.42	0.2-1.5
b:A → b:A+B	0.027	0.0-2.3

Reaction oriented

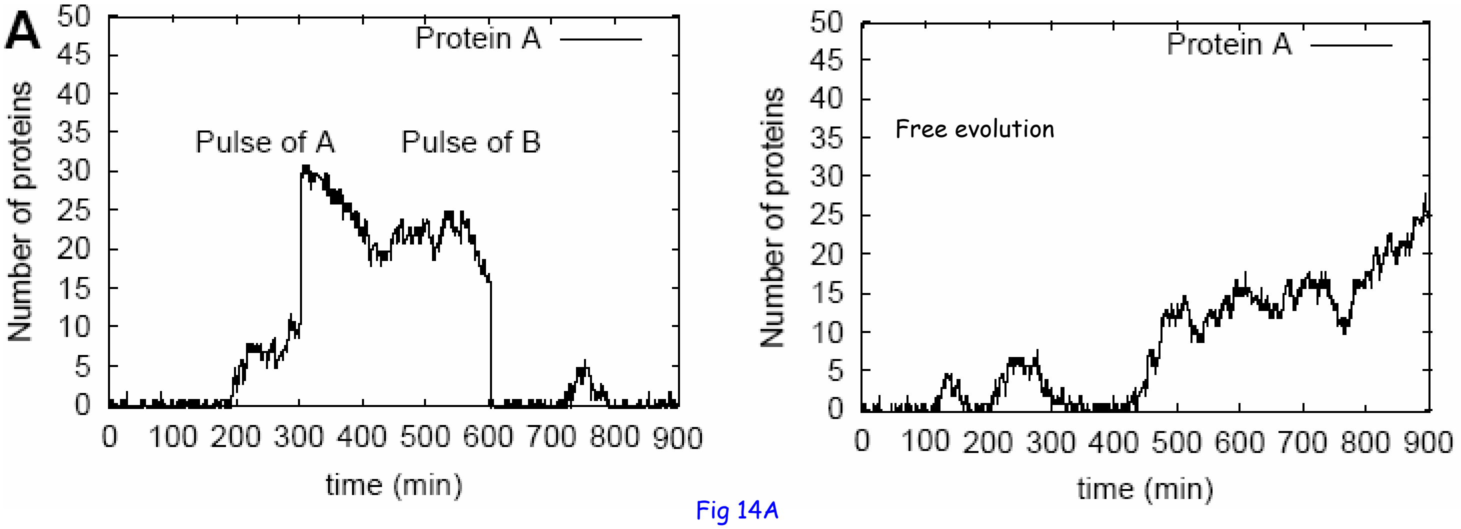
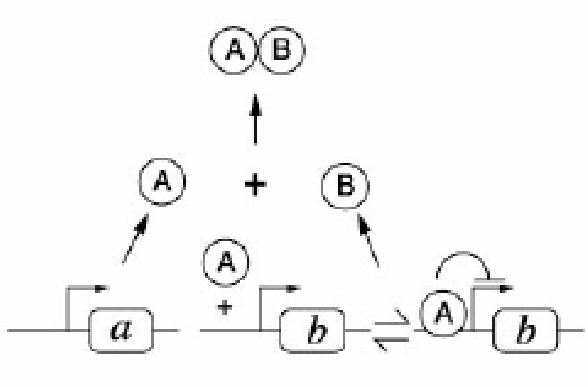


Fig 14A

François & Hakim 3A

rates

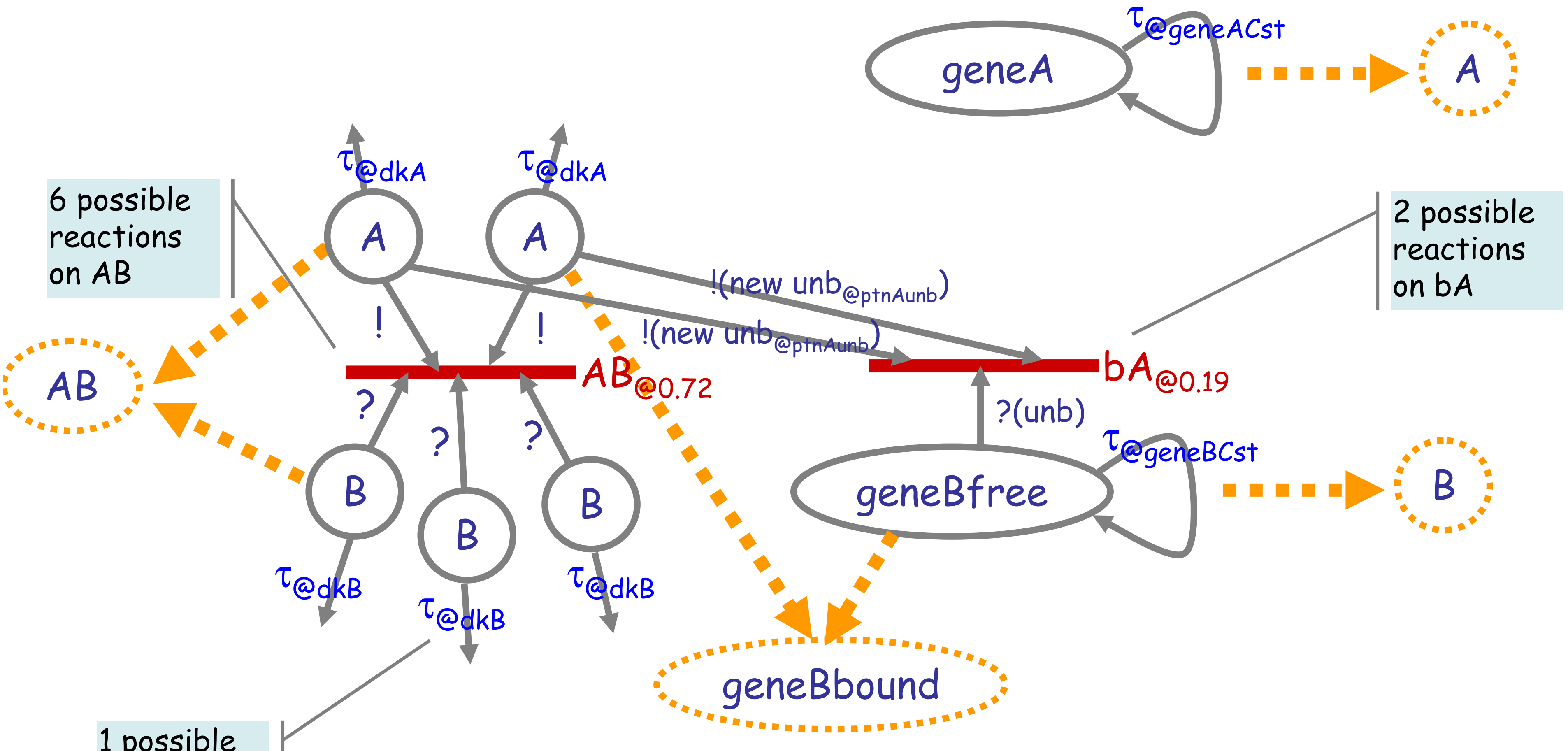
```
dkA ≜ 0.0085
dkB ≜ 0.034
dkAB ≜ 0.53
pntAunb ≜ 0.42
geneACst ≜ 0.20
geneBCst ≜ 0.37
geneBInh ≜ 0.027
new bA @ 0.19
new AB @ 0.72
```



Interaction oriented

ptnB() ≜	
$\tau_{@dkB}$	degrade
+ ?AB; cpxAB()	complex with prot A
ptnA() ≜	
$\tau_{@dkA}$	degrade
+ !AB	complex with prot B
+ new unb$_{@pntAunb}$	bind to gene b
!bA(unb); ?unb; ptnA()	
cpxAB() ≜ $\tau_{@dkAB}$	degrade
geneA() ≜ $\tau_{@geneACst}$: (ptnA() | geneA())	constit. make prot A
geneBfree() ≜	
$\tau_{@geneBCst}$: (ptnB() | geneBfree())	constit. make prot B
+ ?bA(unb); geneBbound(unb)	bind to prot A (inhibit)
geneBbound(unb) ≜	
$\tau_{@geneBInh}$: (ptnB() | geneBbound(unb))	inhib. make prot B
+ !unb; geneBfree()	unbind from prot A
geneA() | geneBfree()	1 gene a and 1 gene b

One State of the Simulation



6 possible reactions on AB

2 possible reactions on bA

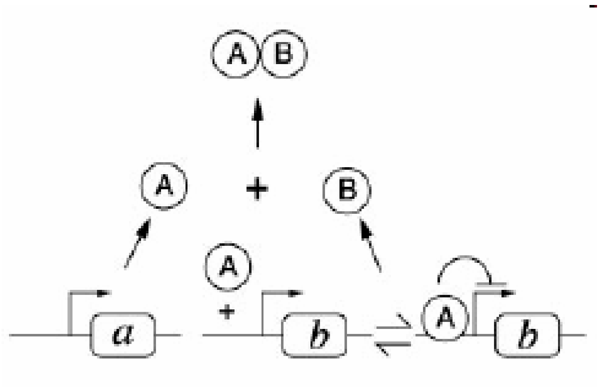
1 possible reactions on each τ

Use Gillespie to pick a reaction and move on

François & Hakim 3A in SPiM

Interaction oriented

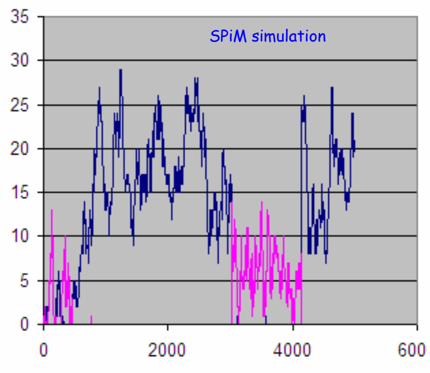
```
(* Francois and Hakim circuit 3A *)

val pntAunb = 0.42
val geneACst = 0.20
val geneBCst = 0.37
val geneBInh = 0.027
val dkA = 0.0085
val dkB = 0.034
val dkAB = 0.53
new bA @ 0.19
new AB @ 0.72
```



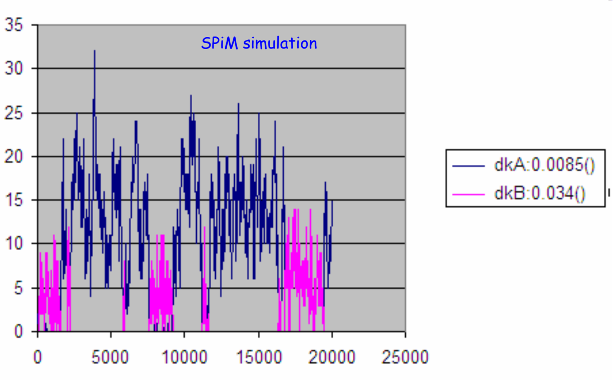
```
let ptnA() =
     do delay@dkA               degrade
     or !AB                     complex with prot B
     or (new unb@pntAunb        bind to gene b
         !bA(unb);(?unb; ptnA()))

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

let cpxAB() = delay@dkAB        degrade

let geneA() =
    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 !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

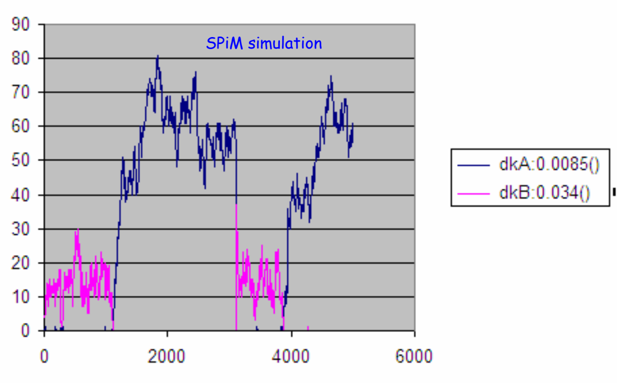


Spontaneous switch at ~500
(as discussed in Supporting Text)
30xB injected at ~3000
30xA injected at ~4000



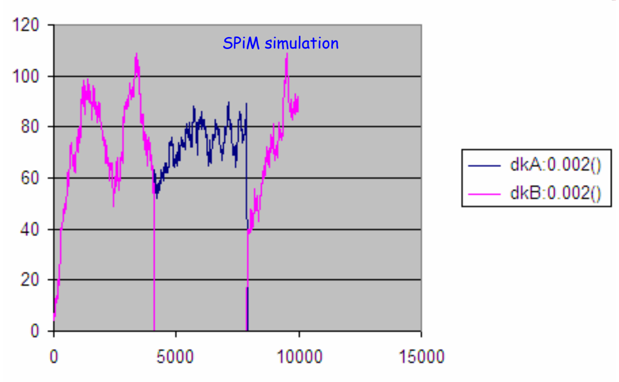
Free evolution

3 copies of each gene.

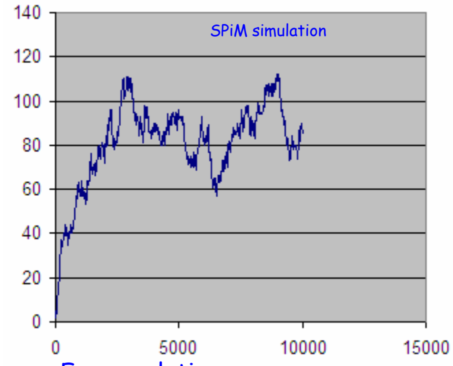


Spontaneous switch at ~1100
100xB injected at ~3000
30xA injected at ~4000

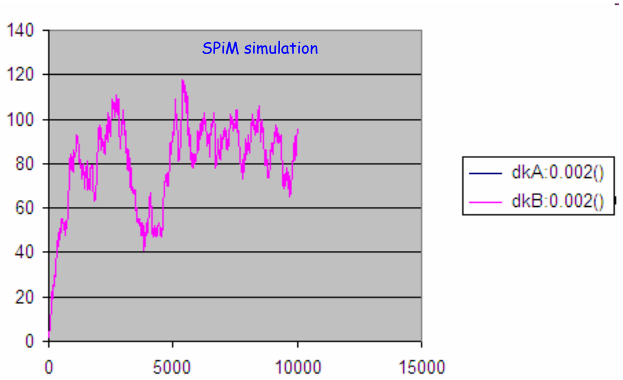
Modified for stability: dkA = 0.02, dkB = 0.02



120xA injected at ~4000
120xB injected at ~8000

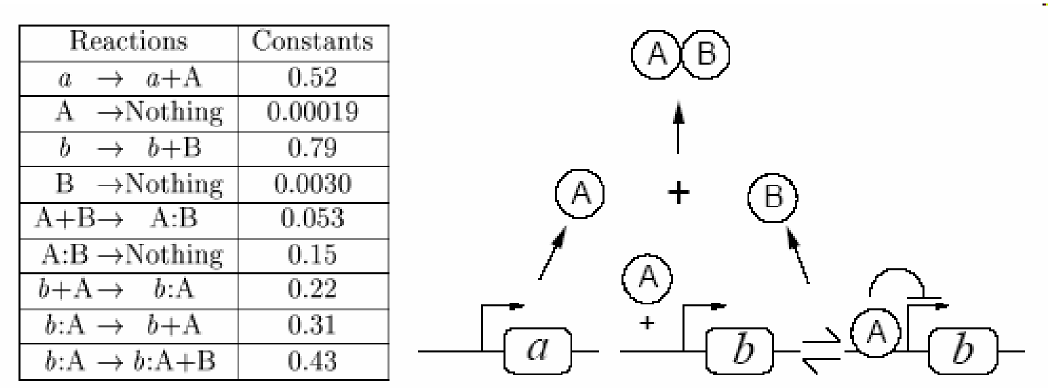


Free evolution



François & Hakim Fig3Ast8

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



Reactions	Constants
$a \rightarrow a$+A	0.52
A $\rightarrow$Nothing	0.00019
$b \rightarrow b$+B	0.79
B $\rightarrow$Nothing	0.0030
A+B$\rightarrow$ A:B	0.053
A:B $\rightarrow$Nothing	0.15
b+A$\rightarrow$ b:A	0.22
b:A $\rightarrow$ b+A	0.31
b:A $\rightarrow$ b:A+B	0.43

Fig 8

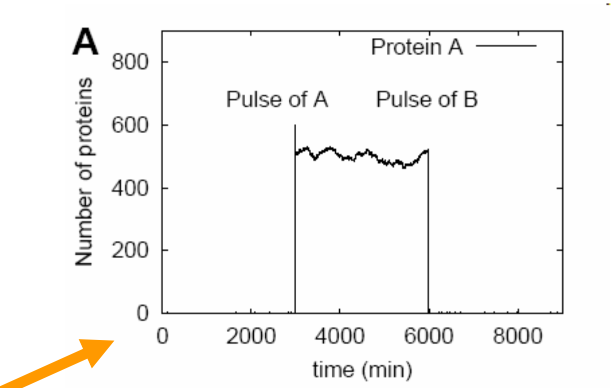
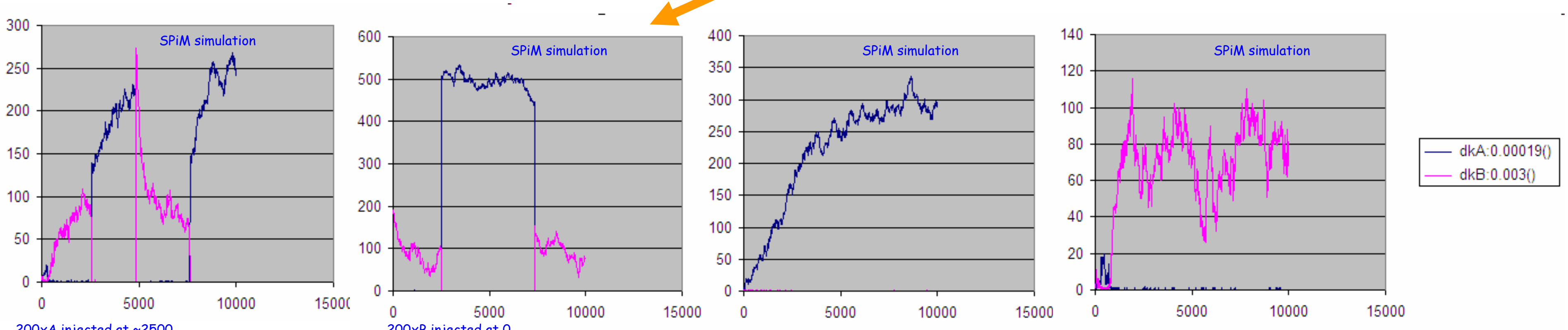


Fig 13A



200xA injected at ~2500
500xB injected at ~5000
200xA injected at ~7500

200xB injected at 0
600xA injected at ~2500
600xB injected at ~7500

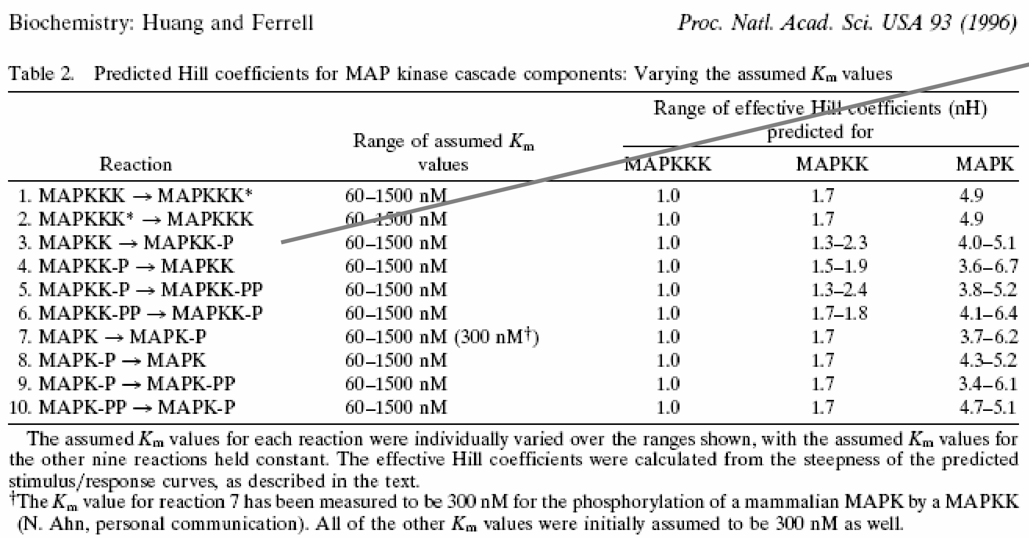
Free evolution

Graphical Representation

- <Slides by Andrew Phillips>

MAPK Cascade

Ultrasensitivity in the mitogen-activated protein **cascade**, Chi-Ying F. Huang and James E. Ferrell, Jr., 1996, *Proc. Natl. Acad. Sci. USA*, 93, 10078-10083.



Biochemistry: Huang and Ferrell — *Proc. Natl. Acad. Sci. USA 93 (1996)*

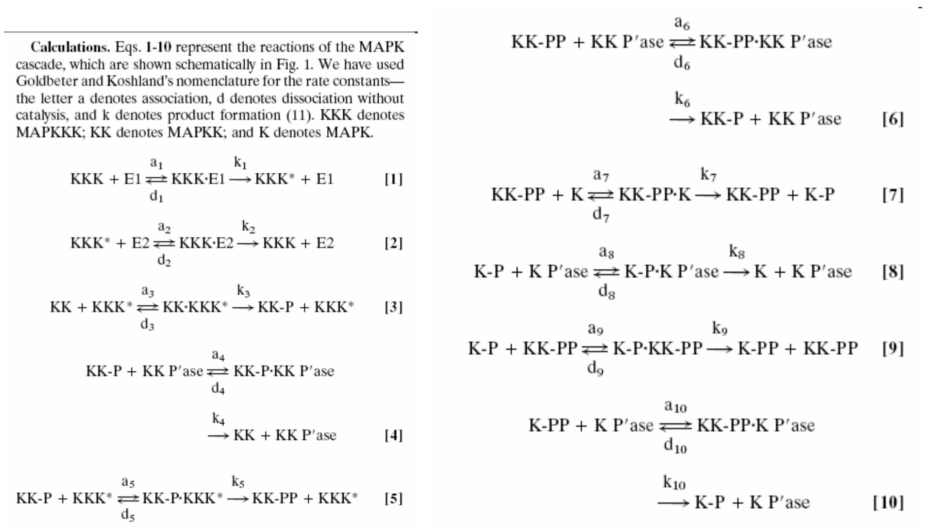
Table 2. Predicted Hill coefficients for MAP kinase cascade components: Varying the assumed K_m values

Reaction	Range of assumed K_m values	Range of effective Hill coefficients (nH) predicted for		
		MAPKKK	MAPKK	MAPK
1. MAPKKK → MAPKKK*	60–1500 nM	1.0	1.7	4.9
2. MAPKKK* → MAPKKK	60–1500 nM	1.0	1.7	4.9
3. MAPKK → MAPKK-P	60–1500 nM	1.0	1.3–2.3	4.0–5.1
4. MAPKK-P → MAPKK	60–1500 nM	1.0	1.5–1.9	3.6–6.7
5. MAPKK-P → MAPKK-PP	60–1500 nM	1.0	1.3–2.4	3.8–5.2
6. MAPKK-PP → MAPKK-P	60–1500 nM	1.0	1.7–1.8	4.1–6.4
7. MAPK → MAPK-P	60–1500 nM (300 nM†)	1.0	1.7	3.7–6.2
8. MAPK-P → MAPK	60–1500 nM	1.0	1.7	4.3–5.2
9. MAPK-P → MAPK-PP	60–1500 nM	1.0	1.7	3.4–6.1
10. MAPK-PP → MAPK-P	60–1500 nM	1.0	1.7	4.7–5.1

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.

10 chemical reactions



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 a denotes association, d denotes dissociation without catalysis, and k denotes product formation (11). KKK denotes MAPKKK; KK denotes MAPKK; and K denotes MAPK.

$$\mathrm{KKK + E1 \underset{d_1}{\overset{a_1}{\rightleftharpoons}} KKK{\cdot}E1 \xrightarrow{k_1} KKK^* + E1} \quad [1]$$

$$\mathrm{KKK^* + E2 \underset{d_2}{\overset{a_2}{\rightleftharpoons}} KKK{\cdot}E2 \xrightarrow{k_2} KKK + E2} \quad [2]$$

$$\mathrm{KK + KKK^* \underset{d_3}{\overset{a_3}{\rightleftharpoons}} KK{\cdot}KKK^* \xrightarrow{k_3} KK\text{-}P + KKK^*} \quad [3]$$

$$\mathrm{KK\text{-}P + KK\ P'ase \underset{d_4}{\overset{a_4}{\rightleftharpoons}} KK\text{-}P{\cdot}KK\ P'ase \xrightarrow{k_4} KK + KK\ P'ase} \quad [4]$$

$$\mathrm{KK\text{-}P + KKK^* \underset{d_5}{\overset{a_5}{\rightleftharpoons}} KK\text{-}P{\cdot}KKK^* \xrightarrow{k_5} KK\text{-}PP + KKK^*} \quad [5]$$

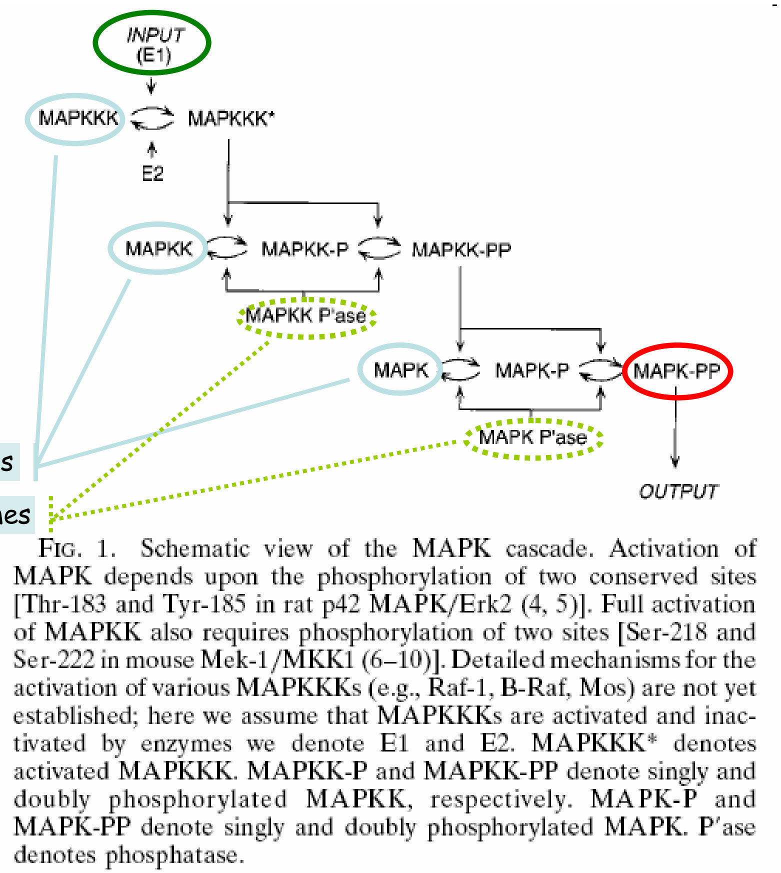
$$\mathrm{KK\text{-}PP + KK\ P'ase \underset{d_6}{\overset{a_6}{\rightleftharpoons}} KK\text{-}PP{\cdot}KK\ P'ase \xrightarrow{k_6} KK\text{-}P + KK\ P'ase} \quad [6]$$

$$\mathrm{KK\text{-}PP + K \underset{d_7}{\overset{a_7}{\rightleftharpoons}} KK\text{-}PP{\cdot}K \xrightarrow{k_7} KK\text{-}PP + K\text{-}P} \quad [7]$$

$$\mathrm{K\text{-}P + K\ P'ase \underset{d_8}{\overset{a_8}{\rightleftharpoons}} K\text{-}P{\cdot}K\ P'ase \xrightarrow{k_8} K + K\ P'ase} \quad [8]$$

$$\mathrm{K\text{-}P + KK\text{-}PP \underset{d_9}{\overset{a_9}{\rightleftharpoons}} K\text{-}P{\cdot}KK\text{-}PP \xrightarrow{k_9} K\text{-}PP + KK\text{-}PP} \quad [9]$$

$$\mathrm{K\text{-}PP + K\ P'ase \underset{d_{10}}{\overset{a_{10}}{\rightleftharpoons}} KK\text{-}PP{\cdot}K\ P'ase \xrightarrow{k_{10}} K\text{-}P + K\ P'ase} \quad [10]$$

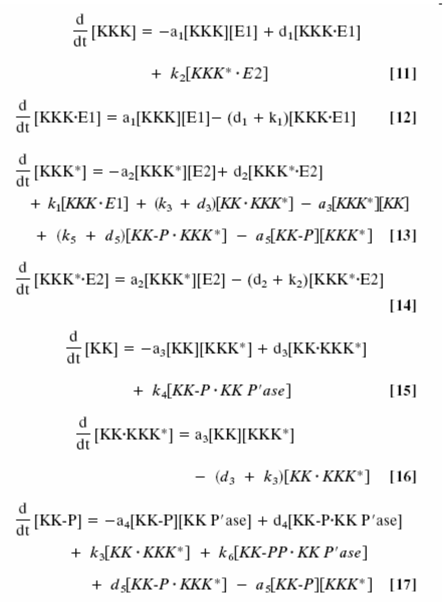


Reservoirs

Back Enzymes

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

As 18 Ordinary Differential Equations Plus 7 conservation equations



$$\frac{d}{dt}[\text{KKK}] = -a_1[\text{KKK}][\text{E1}] + d_1[\text{KKK}\cdot\text{E1}] + k_2[KKK^* \cdot E2] \qquad [11]$$

$$\frac{d}{dt}[\text{KKK}\cdot\text{E1}] = a_1[\text{KKK}][\text{E1}] - (d_1 + k_1)[\text{KKK}\cdot\text{E1}] \qquad [12]$$

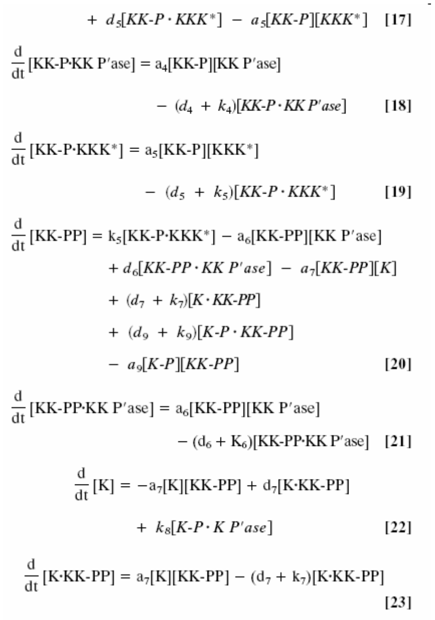
$$\frac{d}{dt}[\text{KKK}^*] = -a_2[\text{KKK}^*][\text{E2}] + d_2[\text{KKK}^*\cdot\text{E2}] + k_1[KKK \cdot E1] + (k_3 + d_3)[KK \cdot KKK^*] - a_3[KKK^*][KK] + (k_5 + d_5)[\text{KK-P} \cdot KKK^*] - a_5[\text{KK-P}][KKK^*] \qquad [13]$$

$$\frac{d}{dt}[\text{KKK}^*\cdot\text{E2}] = a_2[\text{KKK}^*][\text{E2}] - (d_2 + k_2)[\text{KKK}^*\cdot\text{E2}] \qquad [14]$$

$$\frac{d}{dt}[\text{KK}] = -a_3[\text{KK}][\text{KKK}^*] + d_3[\text{KK}\cdot\text{KKK}^*] + k_4[\text{KK-P} \cdot \text{KK P}'\text{ase}] \qquad [15]$$

$$\frac{d}{dt}[\text{KK}\cdot\text{KKK}^*] = a_3[\text{KK}][\text{KKK}^*] - (d_3 + k_3)[KK \cdot KKK^*] \qquad [16]$$

$$\frac{d}{dt}[\text{KK-P}] = -a_4[\text{KK-P}][\text{KK P}'\text{ase}] + d_4[\text{KK-P}\cdot\text{KK P}'\text{ase}] + k_3[KK \cdot KKK^*] + k_6[\text{KK-PP} \cdot \text{KK P}'\text{ase}] + d_5[\text{KK-P} \cdot KKK^*] - a_5[\text{KK-P}][KKK^*] \qquad [17]$$



$$+ d_5[\text{KK-P} \cdot KKK^*] - a_5[\text{KK-P}][KKK^*] \qquad [17]$$

$$\frac{d}{dt}[\text{KK-P}\cdot\text{KK P}'\text{ase}] = a_4[\text{KK-P}][\text{KK P}'\text{ase}] - (d_4 + k_4)[\text{KK-P} \cdot \text{KK P}'\text{ase}] \qquad [18]$$

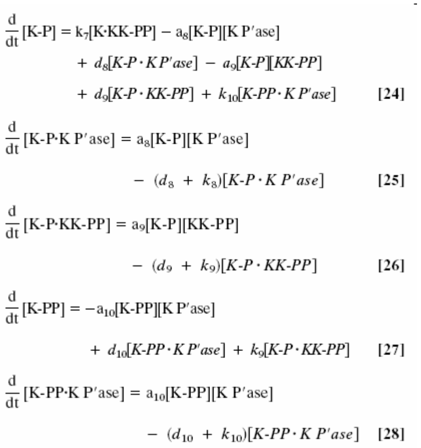
$$\frac{d}{dt}[\text{KK-P}\cdot\text{KKK}^*] = a_5[\text{KK-P}][\text{KKK}^*] - (d_5 + k_5)[\text{KK-P} \cdot KKK^*] \qquad [19]$$

$$\frac{d}{dt}[\text{KK-PP}] = k_5[\text{KK-P}\cdot\text{KKK}^*] - a_6[\text{KK-PP}][\text{KK P}'\text{ase}] + d_6[\text{KK-PP} \cdot \text{KK P}'\text{ase}] - a_7[\text{KK-PP}][K] + (d_7 + k_7)[K \cdot \text{KK-PP}] + (d_9 + k_9)[\text{K-P} \cdot \text{KK-PP}] - a_9[\text{K-P}][\text{KK-PP}] \qquad [20]$$

$$\frac{d}{dt}[\text{KK-PP}\cdot\text{KK P}'\text{ase}] = a_6[\text{KK-PP}][\text{KK P}'\text{ase}] - (d_6 + K_6)[\text{KK-PP}\cdot\text{KK P}'\text{ase}] \qquad [21]$$

$$\frac{d}{dt}[\text{K}] = -a_7[\text{K}][\text{KK-PP}] + d_7[\text{K}\cdot\text{KK-PP}] + k_8[\text{K-P} \cdot \text{K P}'\text{ase}] \qquad [22]$$

$$\frac{d}{dt}[\text{K}\cdot\text{KK-PP}] = a_7[\text{K}][\text{KK-PP}] - (d_7 + k_7)[\text{K}\cdot\text{KK-PP}] \qquad [23]$$



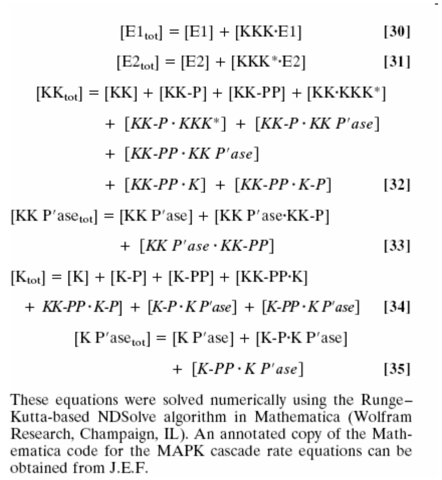
$$\frac{d}{dt}[\text{K-P}] = k_7[\text{K}\cdot\text{KK-PP}] - a_8[\text{K-P}][\text{K P}'\text{ase}] + d_8[\text{K-P} \cdot \text{K P}'\text{ase}] - a_9[\text{K-P}][\text{KK-PP}] + d_9[\text{K-P} \cdot \text{KK-PP}] + k_{10}[\text{K-PP} \cdot \text{K P}'\text{ase}] \qquad [24]$$

$$\frac{d}{dt}[\text{K-P}\cdot\text{K P}'\text{ase}] = a_8[\text{K-P}][\text{K P}'\text{ase}] - (d_8 + k_8)[\text{K-P} \cdot \text{K P}'\text{ase}] \qquad [25]$$

$$\frac{d}{dt}[\text{K-P}\cdot\text{KK-PP}] = a_9[\text{K-P}][\text{KK-PP}] - (d_9 + k_9)[\text{K-P} \cdot \text{KK-PP}] \qquad [26]$$

$$\frac{d}{dt}[\text{K-PP}] = -a_{10}[\text{K-PP}][\text{K P}'\text{ase}] + d_{10}[\text{K-PP} \cdot \text{K P}'\text{ase}] + k_9[\text{K-P} \cdot \text{KK-PP}] \qquad [27]$$

$$\frac{d}{dt}[\text{K-PP}\cdot\text{K P}'\text{ase}] = a_{10}[\text{K-PP}][\text{K P}'\text{ase}] - (d_{10} + k_{10})[\text{K-PP} \cdot \text{K P}'\text{ase}] \qquad [28]$$



$$[\text{E1}_{\text{tot}}] = [\text{E1}] + [\text{KKK}\cdot\text{E1}] \qquad [30]$$

$$[\text{E2}_{\text{tot}}] = [\text{E2}] + [\text{KKK}^*\cdot\text{E2}] \qquad [31]$$

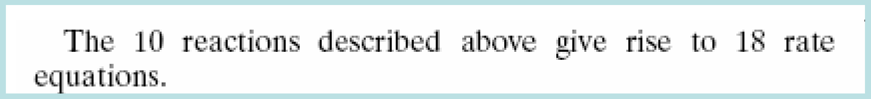
$$[\text{KK}_{\text{tot}}] = [\text{KK}] + [\text{KK-P}] + [\text{KK-PP}] + [\text{KK}\cdot\text{KKK}^*] + [\text{KK-P} \cdot KKK^*] + [\text{KK-P} \cdot \text{KK P}'\text{ase}] + [\text{KK-PP} \cdot \text{KK P}'\text{ase}] + [\text{KK-PP} \cdot K] + [\text{KK-PP} \cdot \text{K-P}] \qquad [32]$$

$$[\text{KK P}'\text{ase}_{\text{tot}}] = [\text{KK P}'\text{ase}] + [\text{KK P}'\text{ase}\cdot\text{KK-P}] + [\text{KK P}'\text{ase} \cdot \text{KK-PP}] \qquad [33]$$

$$[\text{K}_{\text{tot}}] = [\text{K}] + [\text{K-P}] + [\text{K-PP}] + [\text{KK-PP}\cdot\text{K}] + \text{KK-PP} \cdot \text{K-P}] + [\text{K-P} \cdot \text{K P}'\text{ase}] + [\text{K-PP} \cdot \text{K P}'\text{ase}] \qquad [34]$$

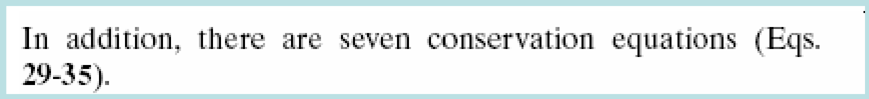
$$[\text{K P}'\text{ase}_{\text{tot}}] = [\text{K P}'\text{ase}] + [\text{K-P}\cdot\text{K P}'\text{ase}] + [\text{K-PP} \cdot \text{K P}'\text{ase}] \qquad [35]$$

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.

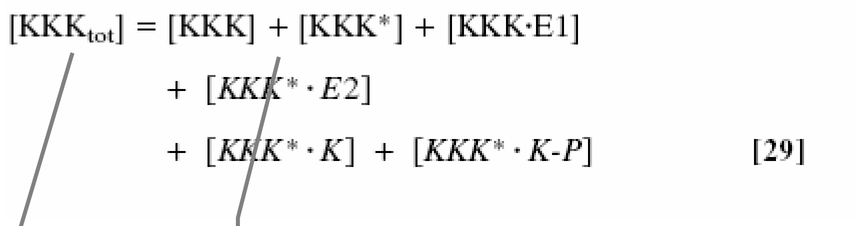


The 10 reactions described above give rise to 18 rate equations.

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).

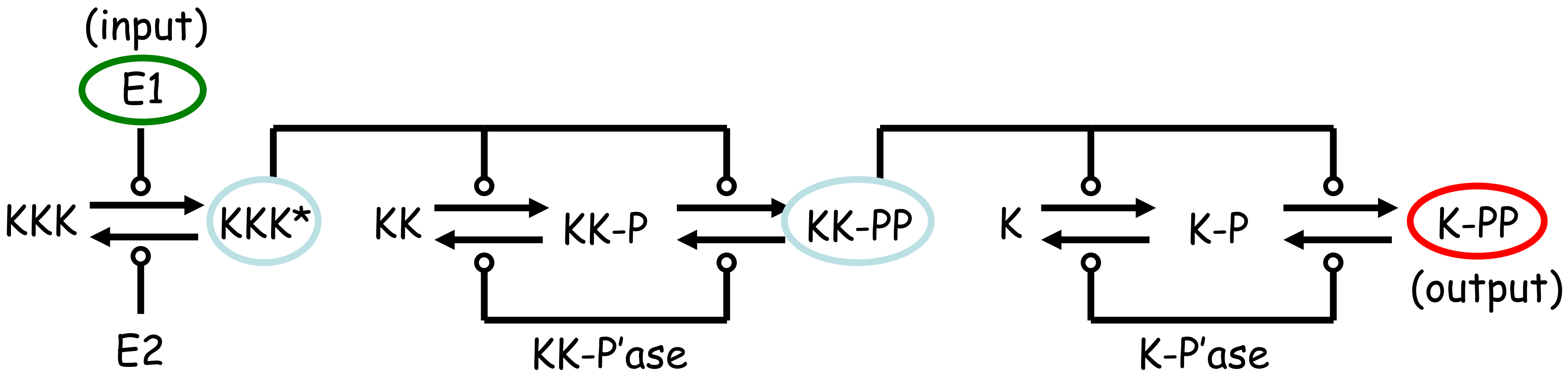


$$[\text{KKK}_{\text{tot}}] = [\text{KKK}] + [\text{KKK}^*] + [\text{KKK}\cdot\text{E1}] + [KKK^* \cdot E2] + [KKK^* \cdot K] + [KKK^* \cdot \text{K-P}] \qquad [29]$$

in exactly one state

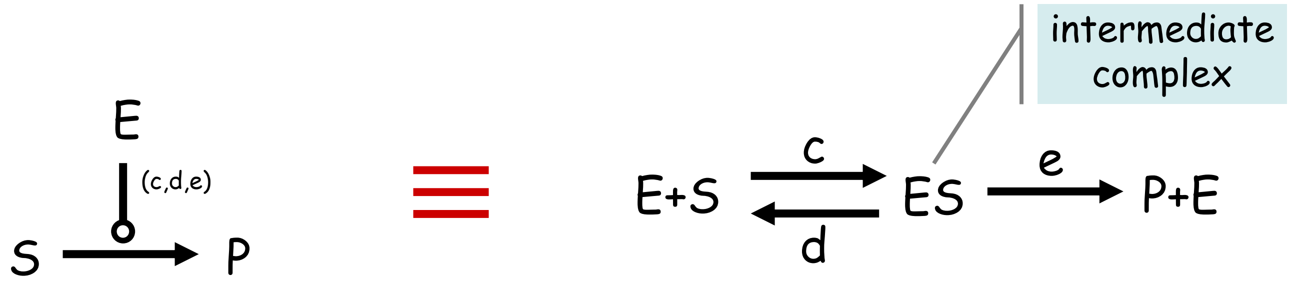
Each molecule

The Circuit



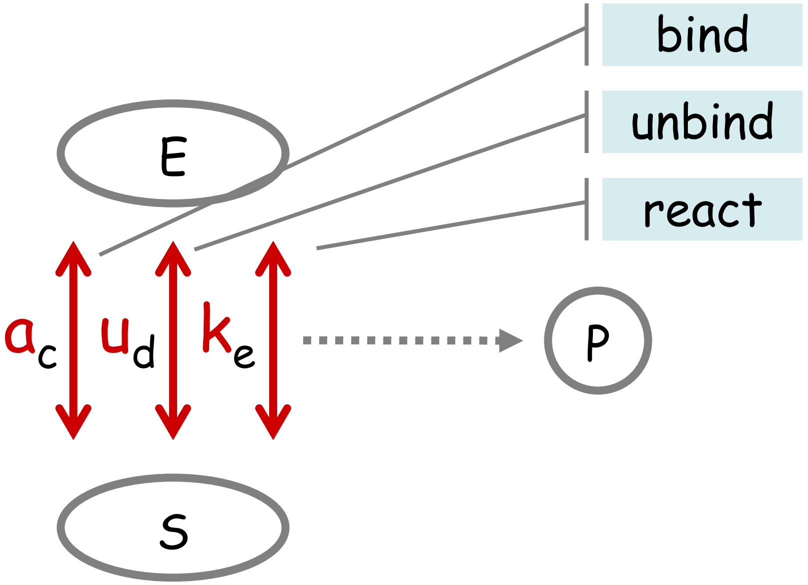
Enzymatic Reactions

Reaction View



intermediate complex

Interaction View



private bindings between one S and one E molecule

$S() \triangleq$ new u@d new k@e
$!a_c(u,k); (!u_d; S() + !k_e; P())$

bind unbind react

$E() \triangleq ?a_c(u,k); (?u_d; E() + ?k_e; E())$

$P() \triangleq$ …

MAPK Cascade in SPiM

```
let KKK() =
   (new u1@d1:Release new k1@r1:React
    !a1(u1,k1); (do !u1;KKK() or !k1;KKKst()))          [1]substrate

and KKKst() =
   (new u2@d2:Release new k2@r2:React
    do !a2(u2,k2); (do !u2;KKKst() or !k2;KKK())        [2]substrate
    or ?a3(u3,k3); (do ?u3;KKKst() or ?k3;KKKst())      [3]kinase
    or ?a5(u5,k5); (do ?u5;KKKst() or ?k5;KKKst()))     [5]kinase
```

KKK:E1 complex

```
let E1() =
   ?a1(u1,k1); (do ?u1;E1() or ?k1;E1())                [1]enzyme
```

E1:KKK complex

```
let E2() =
   ?a2(u2,k2); (do ?u2;E2() or ?k2;E2())                [2]enzyme

let KK() =
   (new u3@d3:Release new k3@r3:React
    !a3(u3,k3); (do !u3;KK() or !k3;KK_P()))            [3]substrate

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())          [4]substrate
    or !a5(u5,k5); (do !u5;KK_P() or !k5;KK_PP()))      [5]substrate
```

```
and KK_PP() =
   (new u6@d6:Release new k6@r6:React
    do !a6(u6,k6); (do !u6;KK_PP() or !k6;KK_P())       [6]substrate
    or ?a7(u7,k7); (do ?u7;KK_PP() or ?k7;KK_PP())      [7]kinase
    or ?a9(u9,k9);                                 ))   [9]kinase

and KKPse() =
   do ?a4(u4,k4); (do ?u4;KKPse() or ?k4;KKPse())       [4]phtase
   or ?a6(u6,k6); (do ?u6;KKPse() or ?k6;KKPse())       [6]phtase

let K() =
   (new u7@
    !a7(u7,k7)                                          [7]substrate

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())            [8]substrate
    or !a9(u9,k9); (do !u9;K_P() or !k9;K_PP()))        [9]substrate

and K_PP() =
   (new u10@d10:Release new k10@r10:React
    !a10(u10,k10); (do !u10;K_PP() or !k10;K_P()))      [10]substrate

and KPse() =
   do ?a8(u8,k8); (do ?u8;KPse() or ?k8;KPse())         [8]phtase
   or ?a10(u10,k10); (do ?u10;KPse() or ?k10;KPse())    [10]phtase
```

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

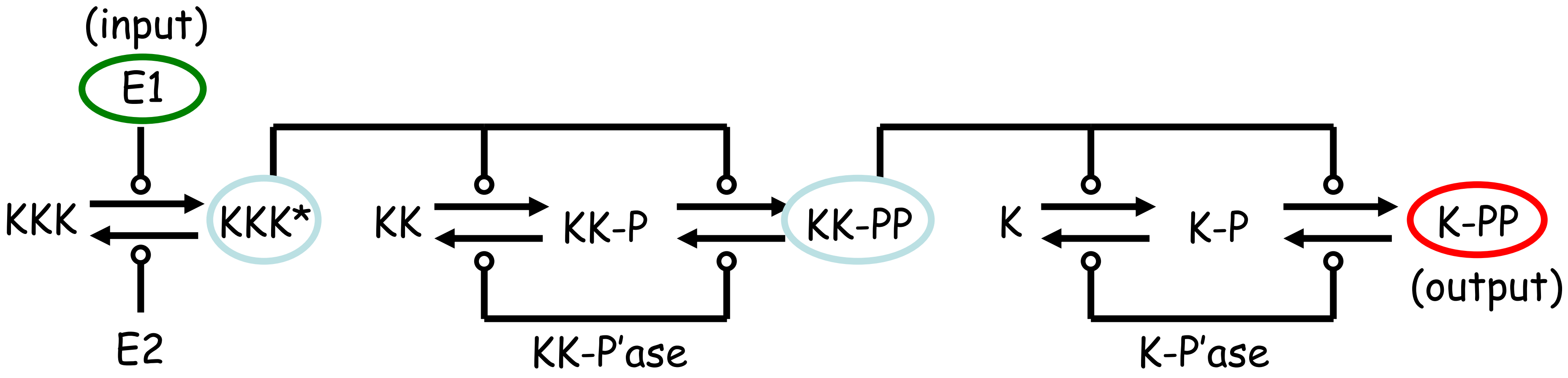
No need for conservation equations: implicit in “choice” operator in the calculus.

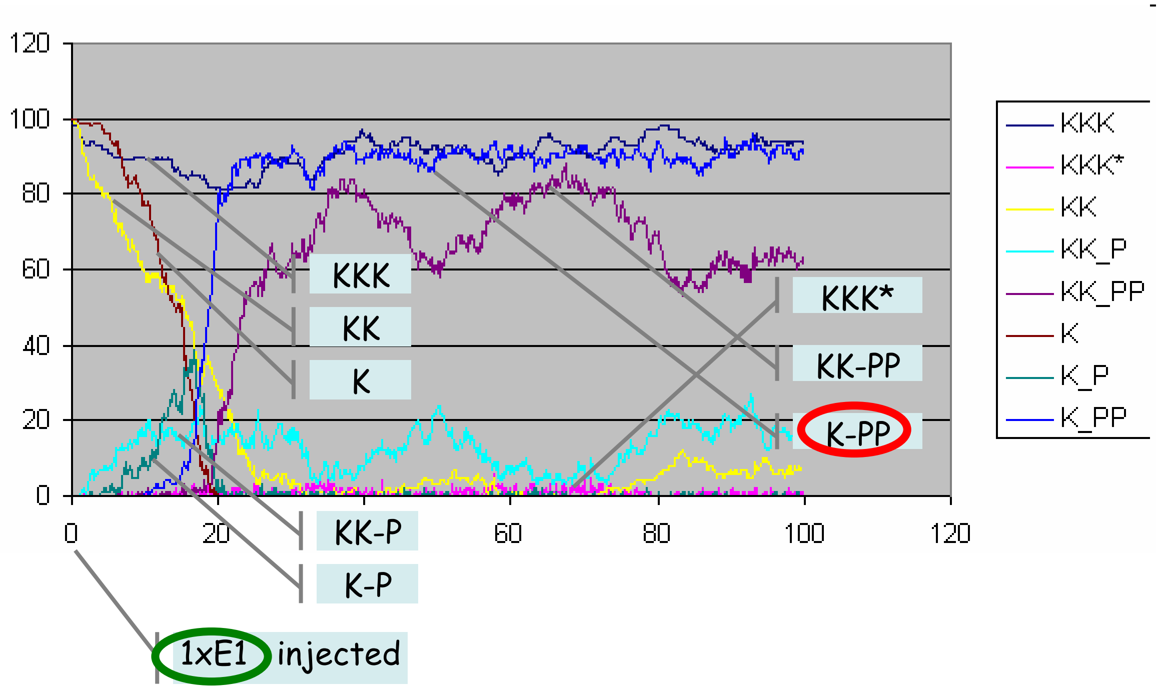
… global channels

```
type Release = chan()
type React = chan()
type Bond = chan(Release,React)

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

…

run 100 of KKK()  run 100 of KK()   run 100 of K()
run 1 of E2()  run 1 of KKPse()  run 1 of KPse()
run 1 of E1()
```

$a_i(u_i,k_i)$: release (u_i@d_i) 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





1st stage:

KKK* barely rises

2nd stage:

KK-PP rises, but is not stable

3rd stage:

K-PP flips up to max

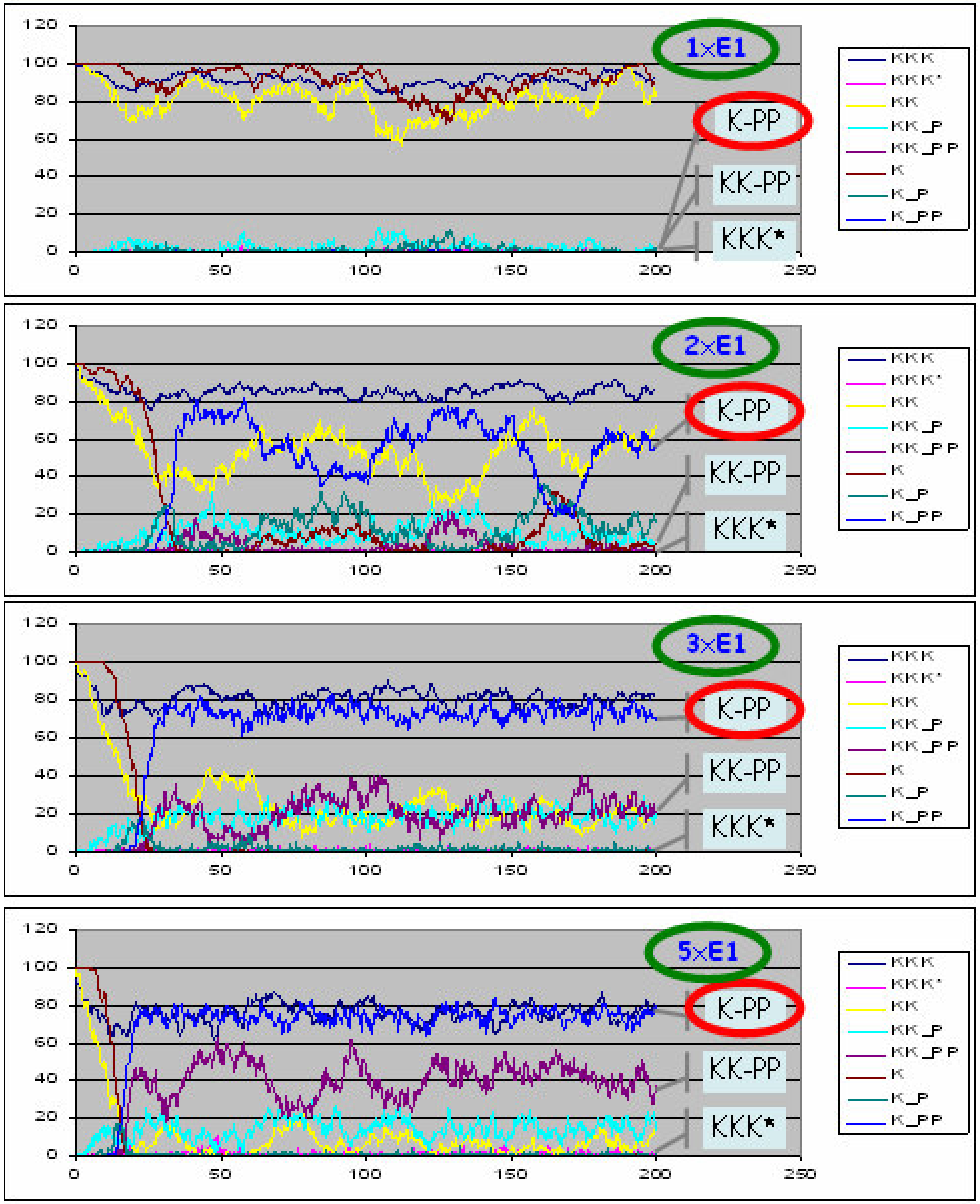
even anticipating 2nd stage

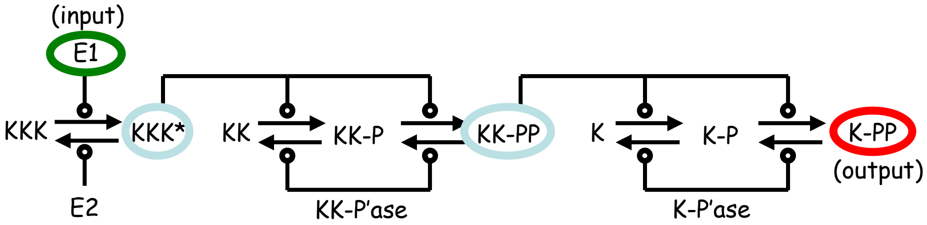
Rates and concentrations ARTIFICIAL:

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

Input is 1x**E1**.
Output is 90x**K-PP** (ultrasensitivity).

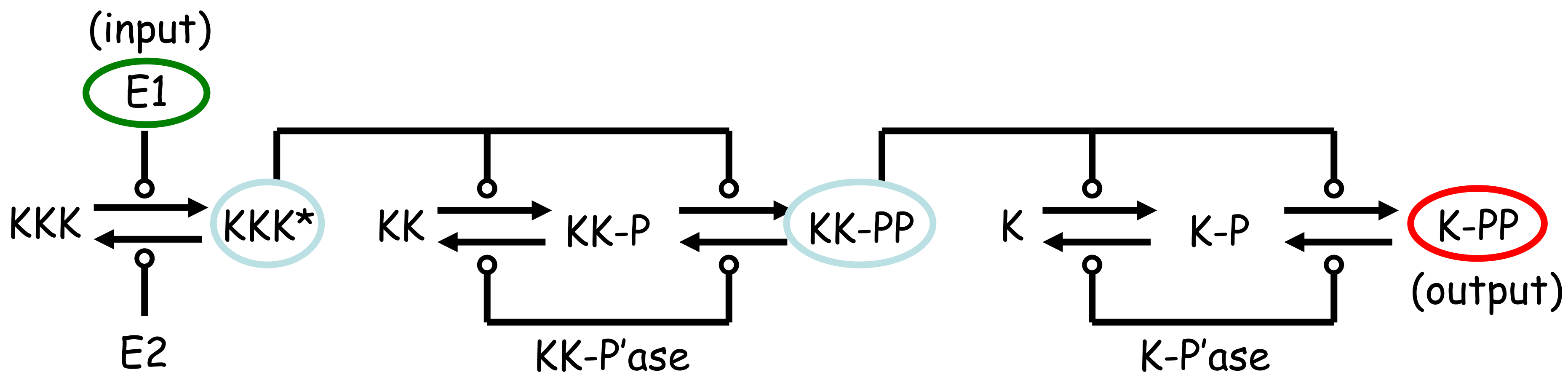
MAPK Cascade Simulation in SPiM



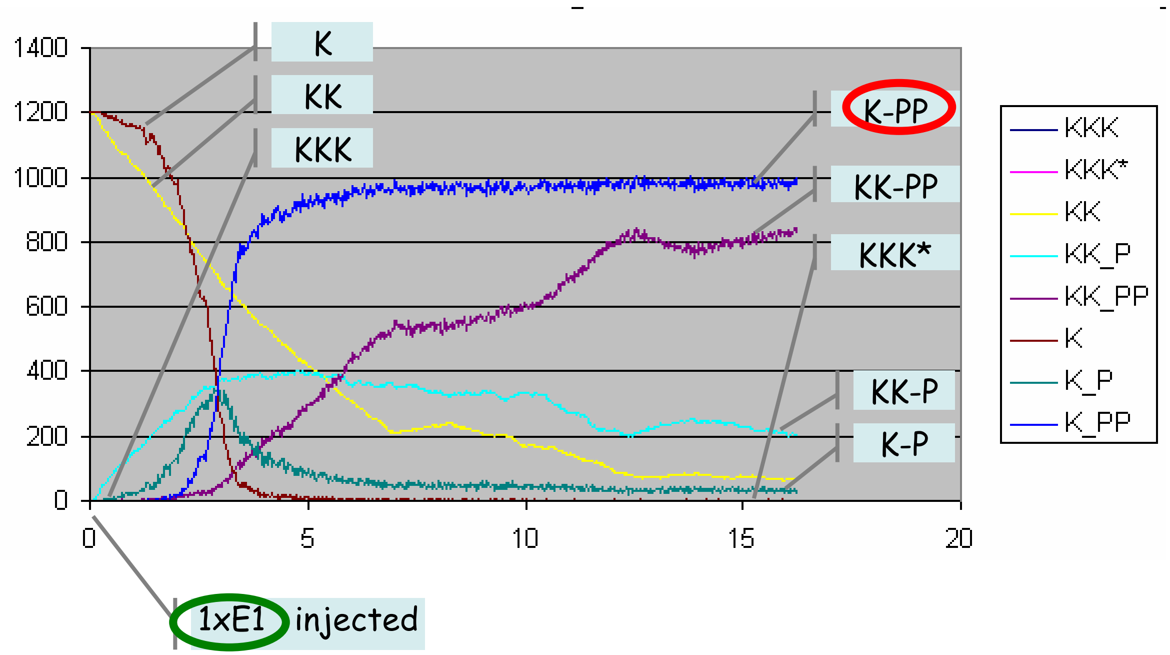


All coefficients 1.0 !!!
100xKKK, 100xKK, 100xK,
13xE2, 13xKKPse, 13xKPse.
nxE1 as indicated
(1xE1 is not sufficient to produce an output)

MAPK Cascade Simulation in SPiM



(input)
E1
KKK
KKK*
E2
KK
KK-P
KK-PP
KK-P'ase
K
K-P
K-PP
(output)
K-P'ase



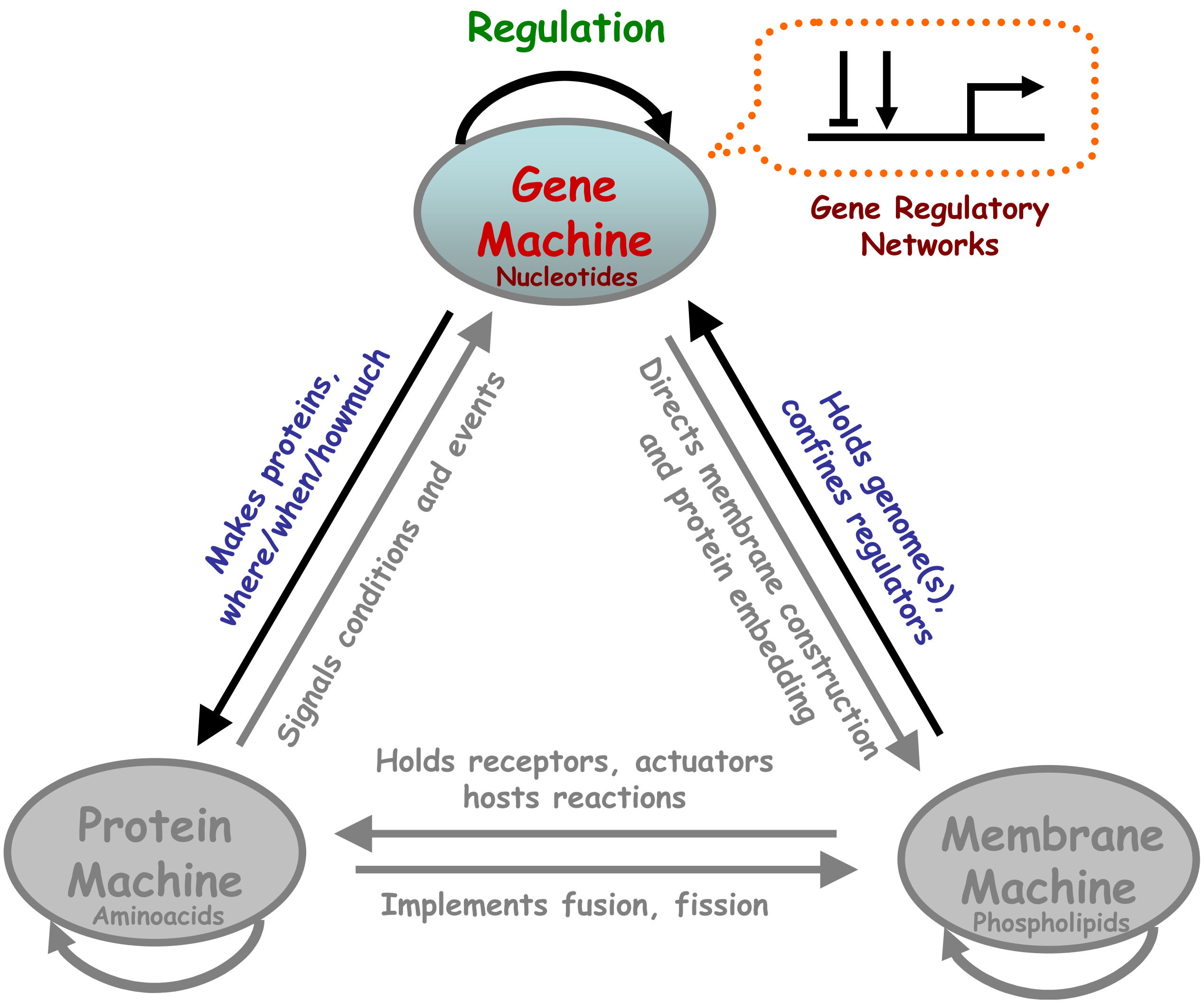
Rates and concentrations
AS IN PAPER:

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

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

1xE1

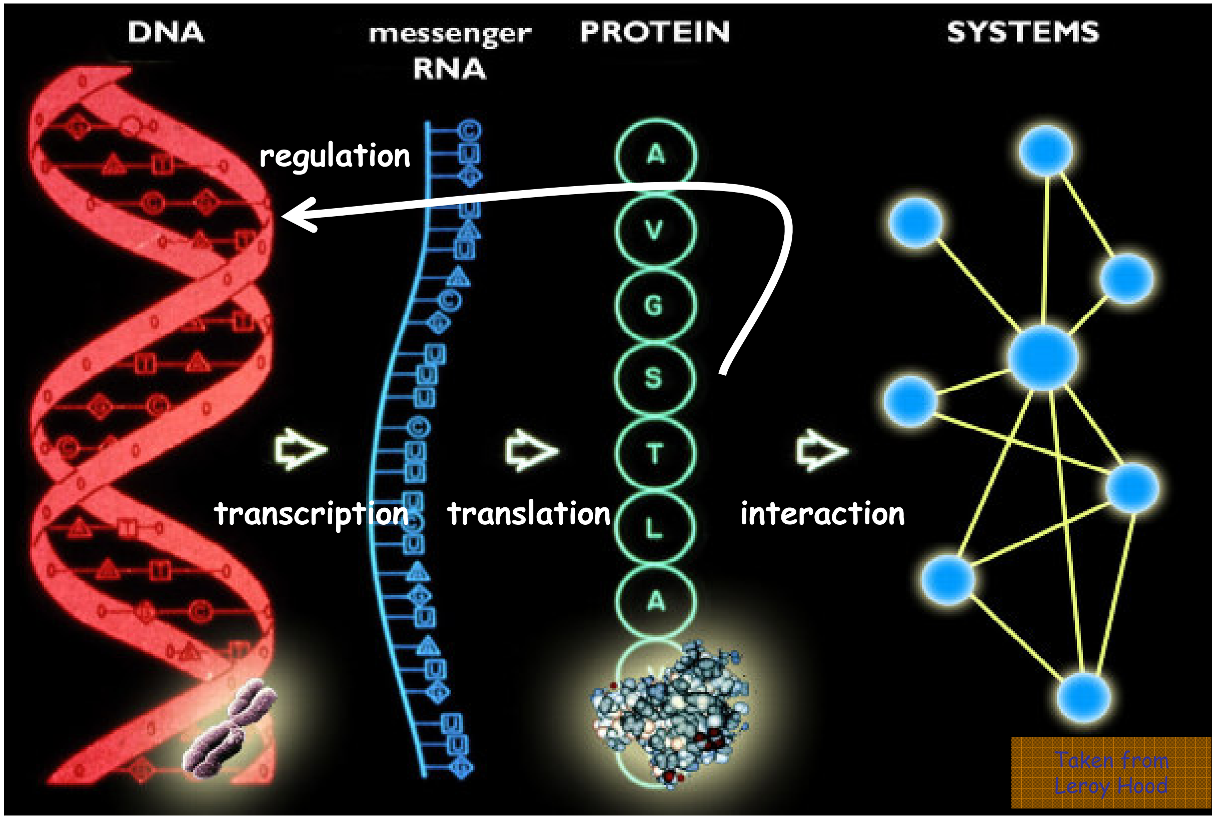
Gene Networks



2. The Gene Machine

Pretty far from the atoms.

The "Central Dogma" of Molecular Biology



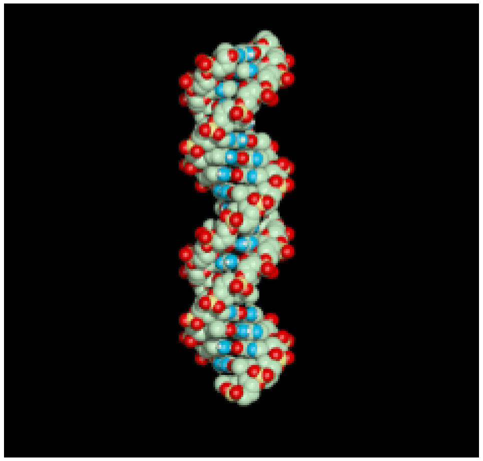
Taken from Leroy Hood

4-letter digital code

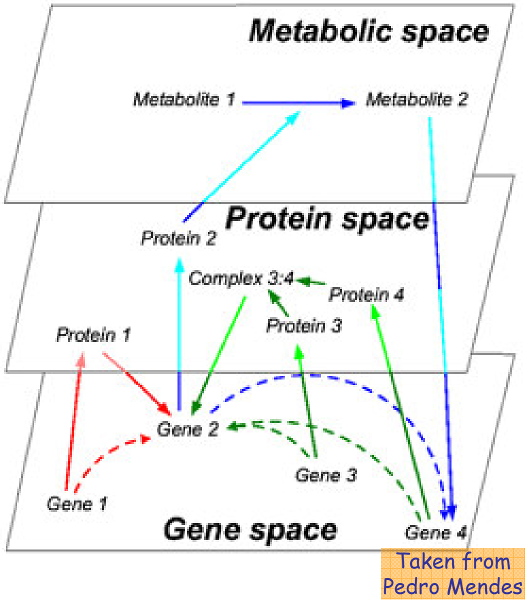
4-letter digital code

20-letter digital code

50.000(?) shapes

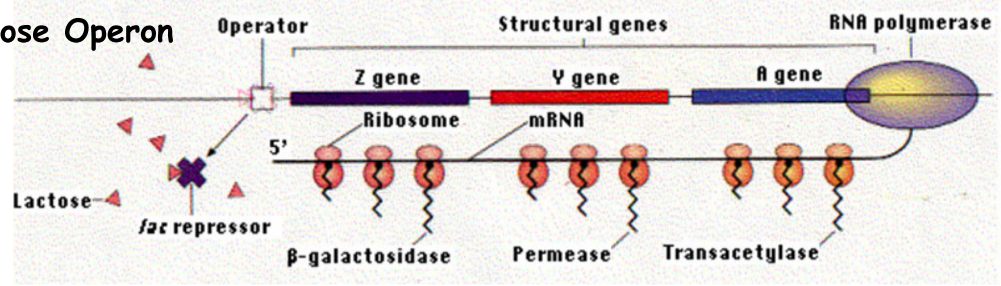


DNA Tutorial



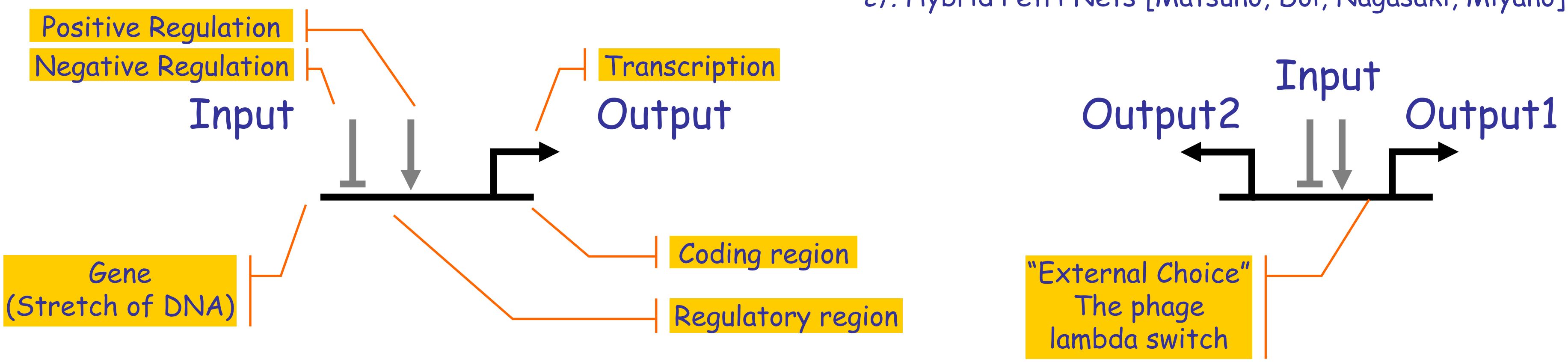
Taken from Pedro Mendes

Lactose Operon



The Gene Machine “Instruction Set”

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



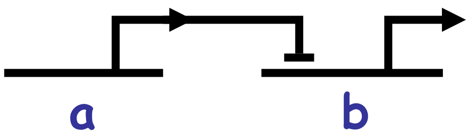
Regulation of a gene (positive and negative) influences transcription. The regulatory region has precise DNA sequences, but not meant for coding proteins: meant for binding regulators.

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

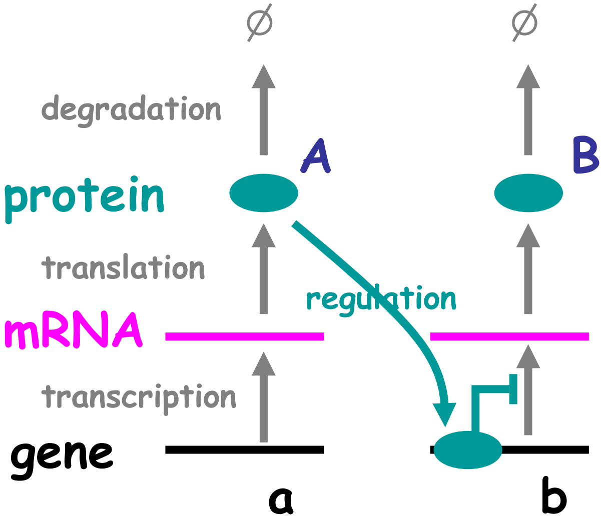
Human (and mammalian) Genome Size
3Gbp (Giga base pairs) 750MB @ 4bp/Byte (CD)
Non-repetitive: 1Gbp 250MB
In genes: 320Mbp 80MB
Coding: 160Mbp 40MB
Protein-coding genes: 30,000-40,000

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



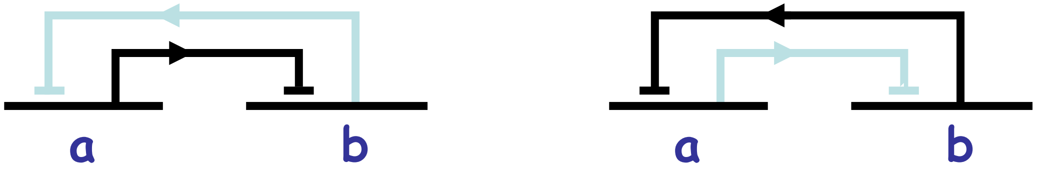
Is a shorthand for:



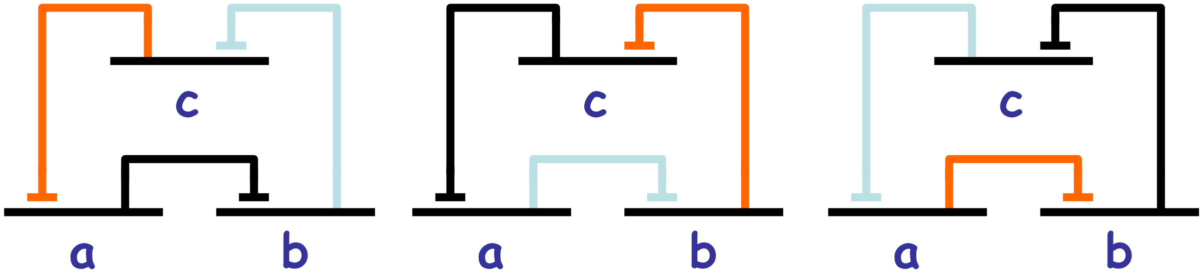
Under the assumptions [Kim & Tidor]

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



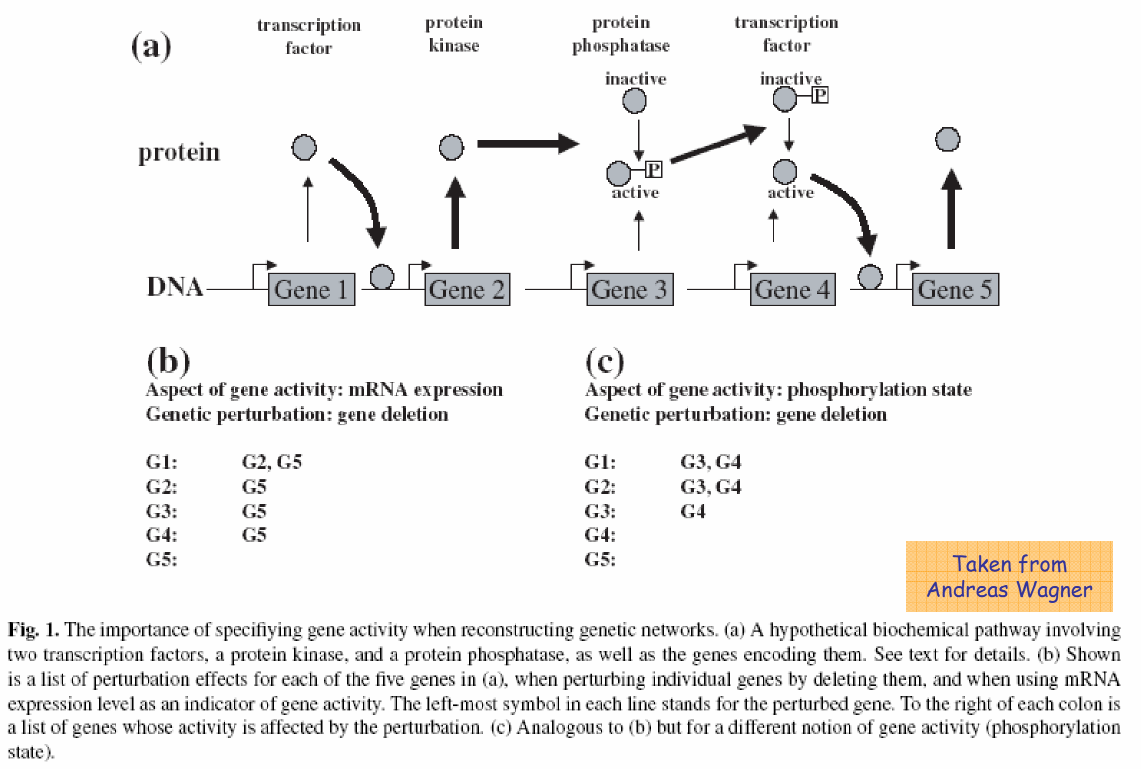
Expressed

Repressed

Expressing

Indirect Gene Effects

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



(b)
Aspect of gene activity: mRNA expression
Genetic perturbation: gene deletion

G1:	G2, G5
G2:	G5
G3:	G5
G4:	G5
G5:	

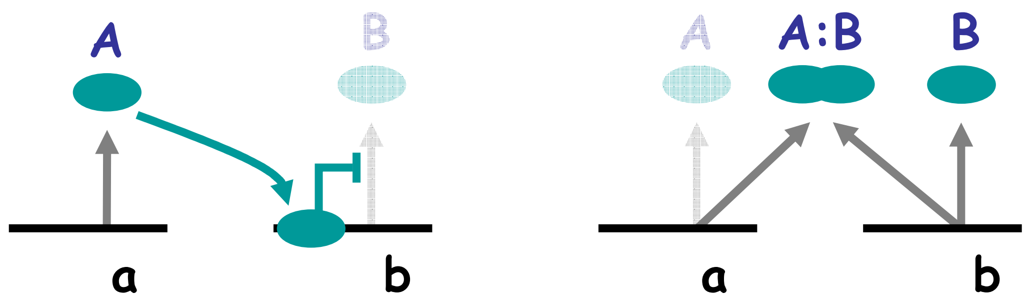
(c)
Aspect of gene activity: phosphorylation state
Genetic perturbation: gene deletion

G1:	G3, G4
G2:	G3, G4
G3:	G4
G4:	
G5:	

Taken from Andreas Wagner

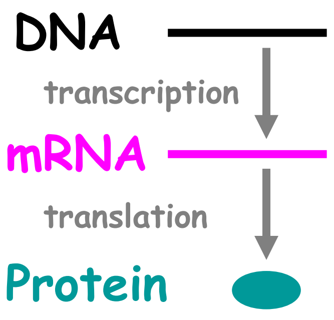
Fig. 1. The importance of specifiying gene activity when reconstructing genetic networks. (a) A hypothetical biochemical pathway involving two transcription factors, a protein kinase, and a protein phosphatase, as well as the genes encoding them. See text for details. (b) Shown is a list of perturbation effects for each of the five genes in (a), when perturbing individual genes by deleting them, and when using mRNA expression level as an indicator of gene activity. The left-most symbol in each line stands for the perturbed gene. To the right of each colon is a list of genes whose activity is affected by the perturbation. (c) Analogous to (b) but for a different notion of gene activity (phosphorylation state).

One of many bistable switches that cannot be described by pure gene regulatory networks [Francois & Hakim].



Structure of the Coding Region

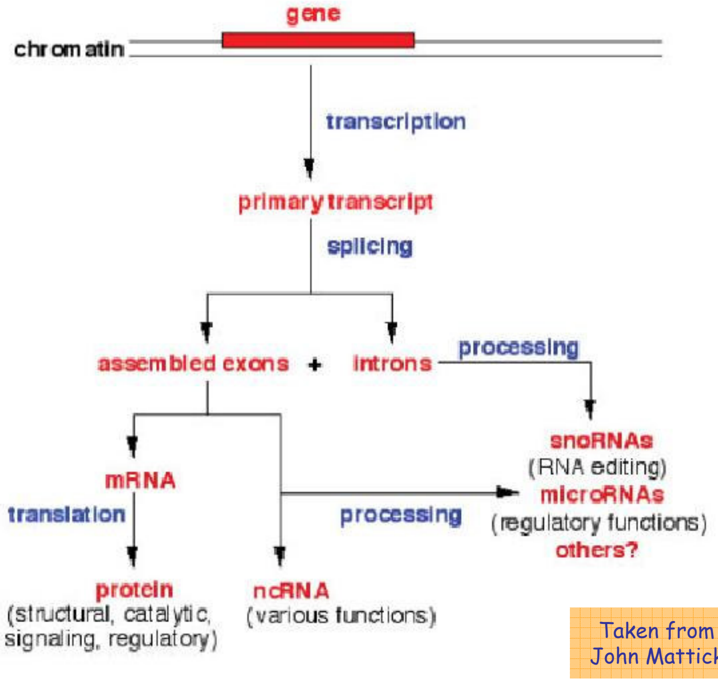
The Central Dogma



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

Challenging the Dogma (in higher organisms)



Taken from John 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

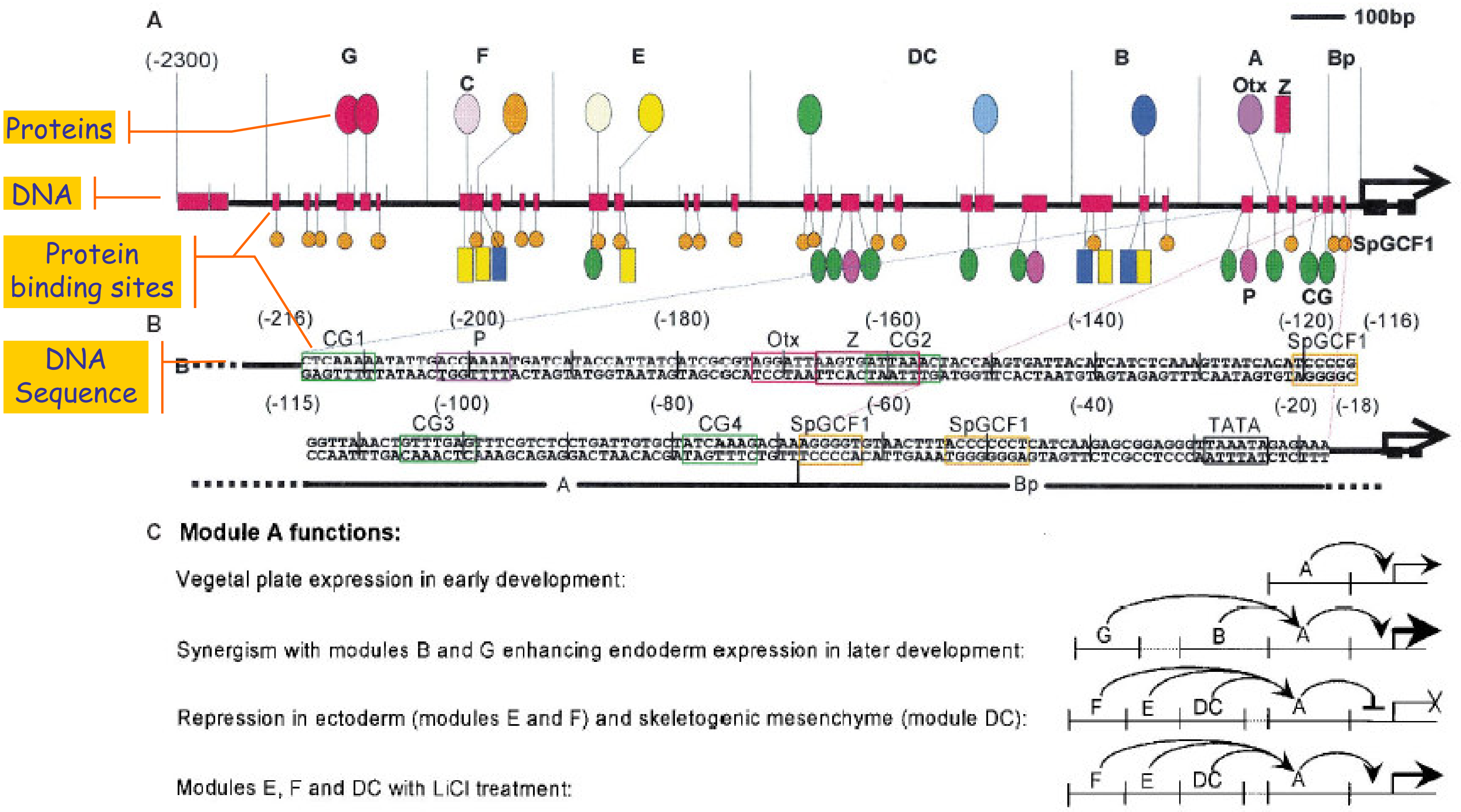

Proteins
DNA
Protein binding sites
DNA Sequence
A
100bp
(-2300)
G
F
C
E
DC
B
A
Otx
Z
Bp
SpGCF1
P
CG
B
(-216)
(-200)
(-180)
(-160)
(-140)
(-120)
(-116)
CG1
P
Otx
Z
CG2
SpGCF1
CTCAAAAATATTGACCAAAATGATCATACCATTATCATCGCGTAGGATTAAGTGATTAAACTACCAAGTGATTACATCATCTCAAAGTTATCACATCCCCG
GAGTTTTTATAACTGGTTTTACTAGTATGGTAATAGTAGCGCATCCTAATTCACTAATTTGATGGTTCACTAATGTAGTAGAGTTTCAATAGTGTAGGGGC
(-115)
(-100)
(-80)
(-60)
(-40)
(-20)
(-18)
CG3
CG4
SpGCF1
SpGCF1
TATA
GGTTAAAACTGTTTGAGTTTCGTCTCCTGATTGTGCTATCAAAGACAAAGGGGTGTAACTTTACCCCCCTCATCAAGAGCGGAGGGTTAAATAGAGAAA
CCAATTTTGACAAACTCAAAGCAGAGGACTAACACGATAGTTTCTGTTTCCCCACATTGAAATGGGGGGAGTAGTTCTCGCCTCCCAATTTATCTCTTT
A
Bp
C Module A functions:
Vegetal plate expression in early development:
A
Synergism with modules B and G enhancing endoderm expression in later development:
G
B
A
Repression in ectoderm (modules E and F) and skeletogenic mesenchyme (module DC):
F
E
DC
A
Modules E, F and DC with LiCl treatment:
F
E
DC
A

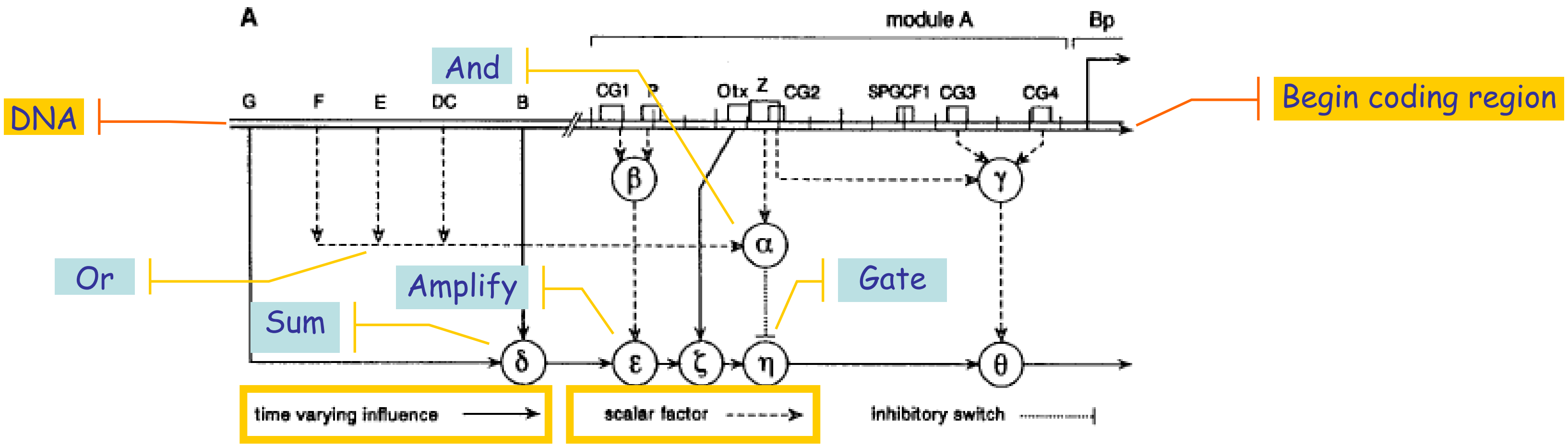
Fig. 1. *Endo16* cis-regulatory system and interactive roles of module A. **(A)** Diversity of protein binding sites and organization into modular subregions [modified from (*7*)]. Specific DNA binding sites are indicated as red blocks; modular subregions are denoted by letters G to A (Bp, basal promoter). 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_3 and CG_4 sites as well as Bp has no endoderm-specific activity and services other upstream cis-regulatory systems promiscuously; similarly, the *Endo16* cis-regulatory system functions specifically with heterologous promoters substituted for Bp (*5*, *8*, *19*). Boxed sequences indicate conserved core elements of the target sites (*7*, *12*, *14*), not the complete target site sequences. **(C)** Integrative and interactive functions of module A (*5*, *8*). Module A communicates the output of all upstream modules to the basal transcription apparatus. It also initiates endoderm expression, increases the output of modules B and G, and is required for functions of the upstream modules F, E, and DC. These functions are repression of expression in nonendodermal domains and enhancement of expression in response to LiCl.

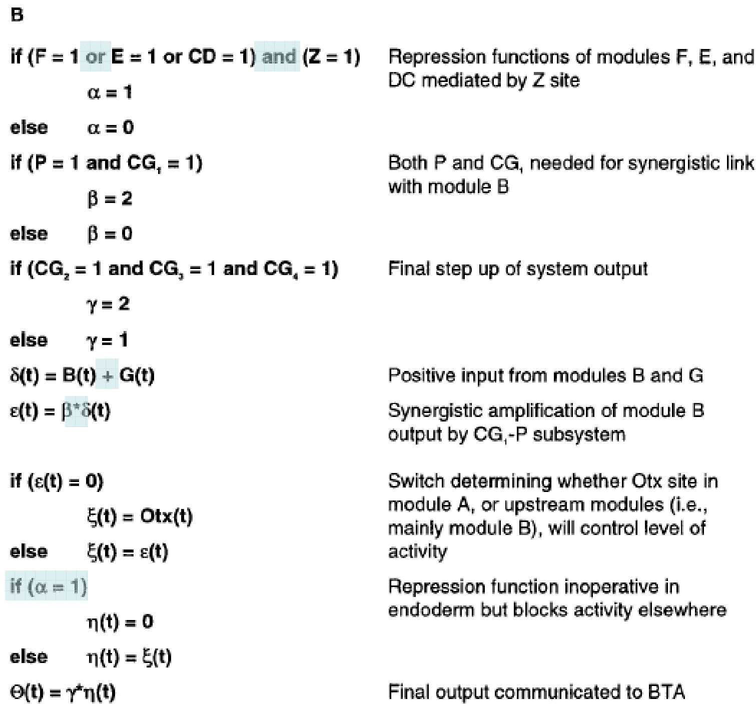
2300bp!

> average protein

Taken from Eric H Davidson

Function of a Regulatory Region





B

if (F = 1 or E = 1 or CD = 1) and (Z = 1)
$\alpha = 1$
else $\alpha = 0$
	Repression functions of modules F, E, and DC mediated by Z site

if (P = 1 and CG_1 = 1)
$\beta = 2$
else $\beta = 0$
	Both P and CG_1 needed for synergistic link with module B

if (CG_2 = 1 and CG_3 = 1 and CG_4 = 1)
$\gamma = 2$
else $\gamma = 1$
	Final step up of system output

$\delta(t) = B(t) + G(t)$	Positive input from modules B and G
$\varepsilon(t) = \beta * \delta(t)$	Synergistic amplification of module B output by CG_1-P subsystem

if ($\varepsilon(t) = 0$)
$\xi(t) = Otx(t)$
else $\xi(t) = \varepsilon(t)$
	Switch determining whether Otx site in module A, or upstream modules (i.e., mainly module B), will control level of activity

if ($\alpha = 1$)
$\eta(t) = 0$
else $\eta(t) = \xi(t)$
	Repression function inoperative in endoderm but blocks activity elsewhere

$\Theta(t) = \gamma * \eta(t)$	Final output communicated to BTA

C-H.Yuh, H.Bolouri, E.H.Davidson. Genomic Cis-Regulatory Logic: Experimental and Computational Analysis of a Sea Urchin Gene. Science 279:1896-1902, 1998

The Raw Data

S.cerevisiae yeast cell cycle

Known cell cycle transcriptional activators

Cyclins

Average value of a given feature over all cell-cycle-proteins whose genes are maximally expressed at a given time in the cycle.

Measured protein features

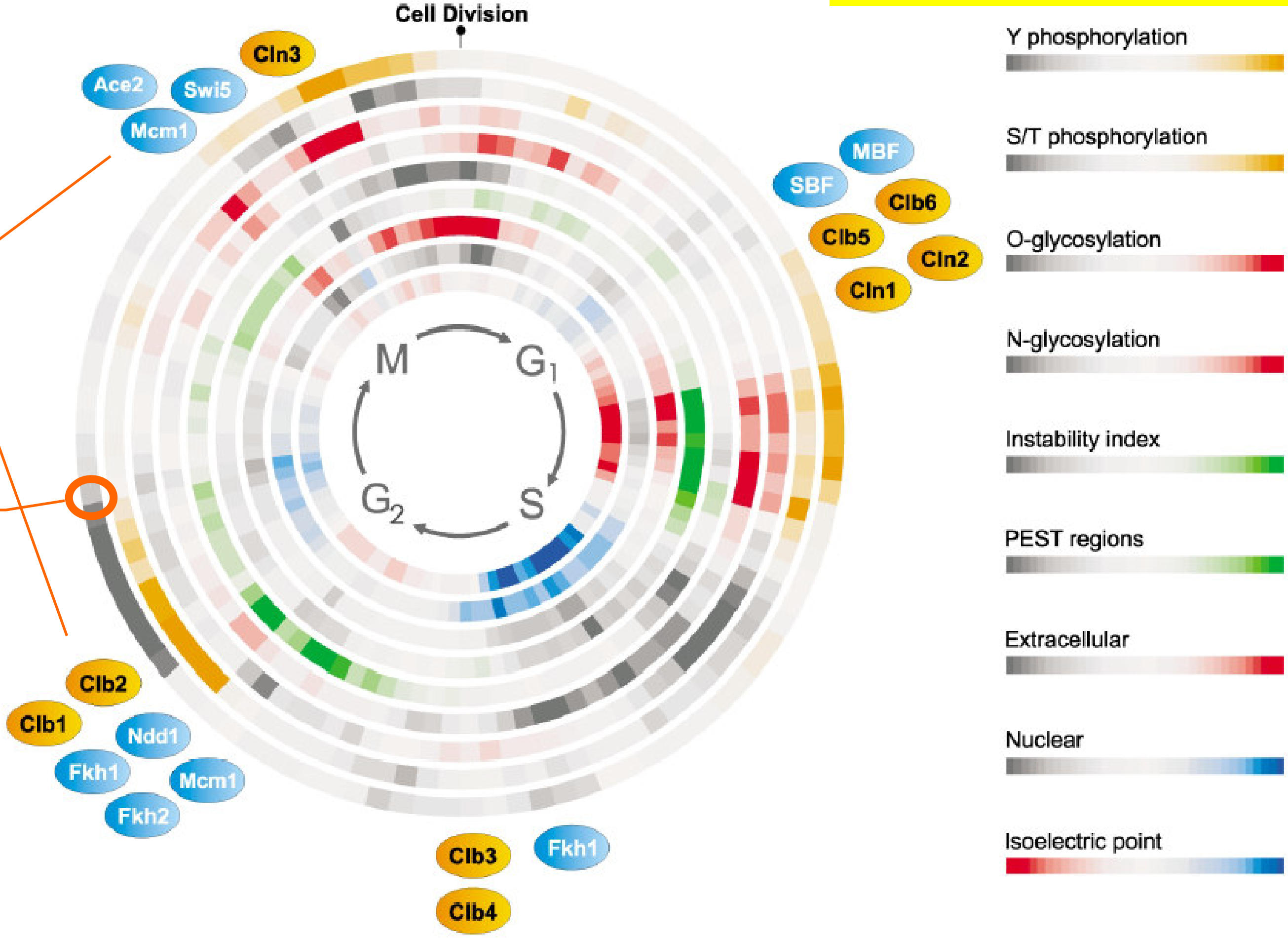


Figure 6. Feature variation during the cell cycle. The temporal variation in nine selected protein features during the cell cycle, with zero time (at the top of the plot) corresponding to the presumed time of cell division (M=G1 transition). The color scales correspond to +/-two standard deviations from 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.

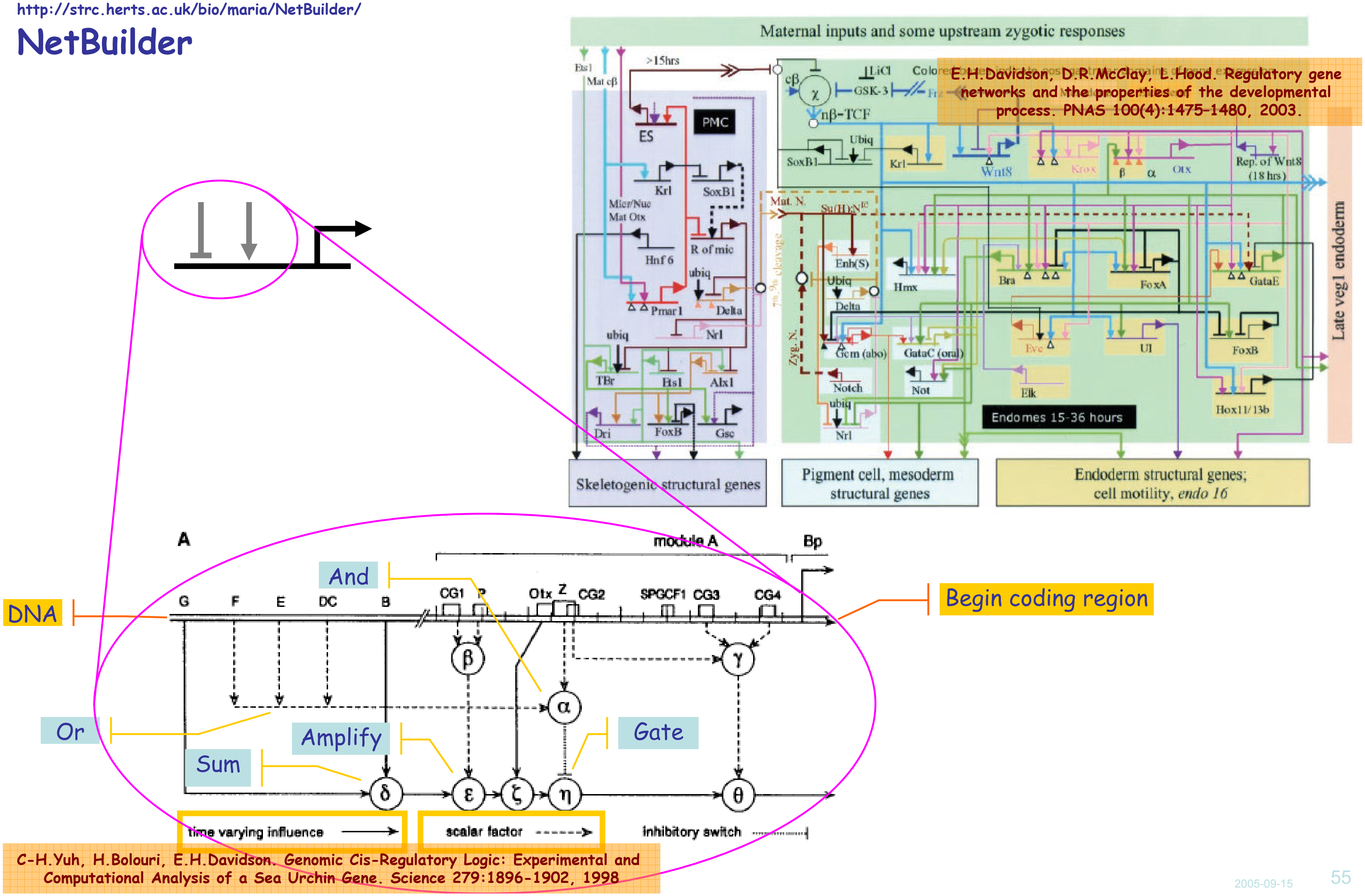
Protein Feature Based Identification of Cell Cycle Regulated Proteins in Yeast
Ulrik de Lichtenberg, Thomas S. Jensen, Lars J. Jensen and Søren Brunak

Taken from Brunak

Gene Regulatory Networks

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

NetBuilder



E.H.Davidson, D.R.McClay, L.Hood. Regulatory gene networks and the properties of the developmental process. PNAS 100(4):1475-1480, 2003.

C-H.Yuh, H.Bolouri, E.H.Davidson. Genomic Cis-Regulatory Logic: Experimental and Computational Analysis of a Sea Urchin Gene. Science 279:1896-1902, 1998

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.

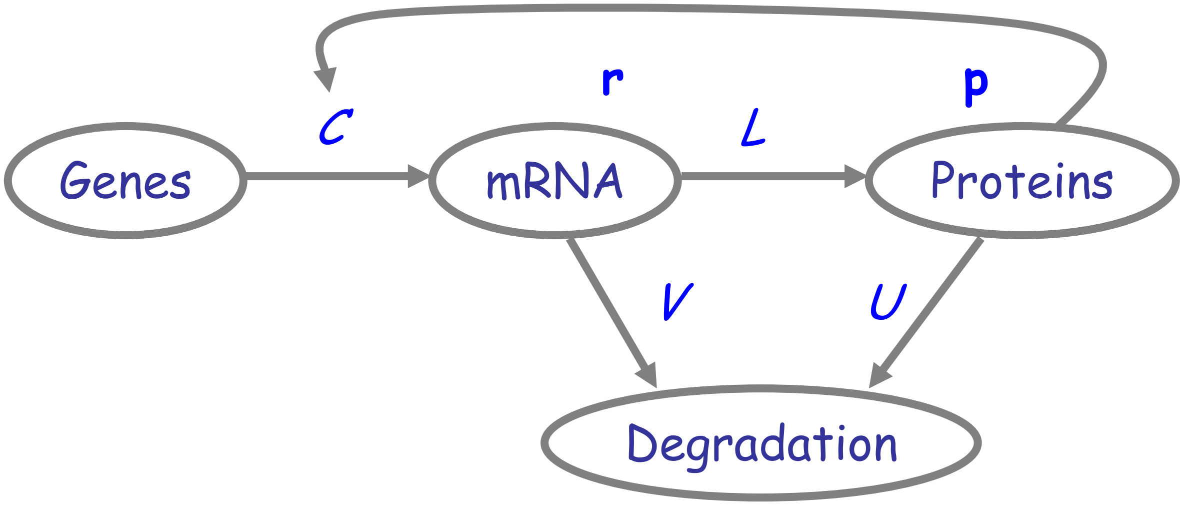
Notations for the Gene Machine

- 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)

[Chen, He, Church]



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!)

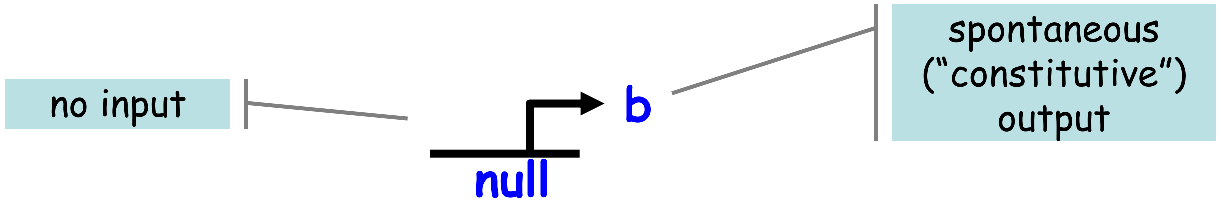
$$\frac{d\mathbf{r}}{dt} = f(\mathbf{p}) - V\mathbf{r}$$

$$\frac{d\mathbf{p}}{dt} = L\mathbf{r} - U\mathbf{r}$$

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

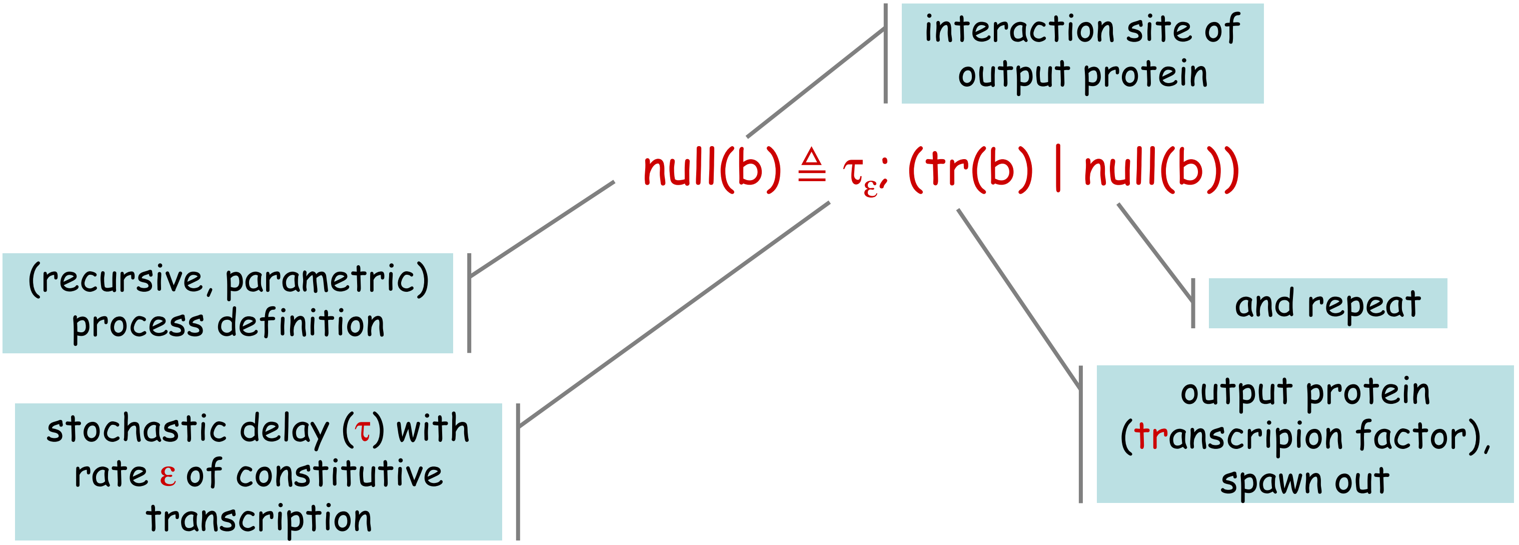
f(**p**) transcription functions:
(n-dim vector polynomials on **p**)

Nullary Gate



no input

spontaneous ("constitutive") output



$$\mathrm{null}(b) \triangleq \tau_{\varepsilon};\ (\mathrm{tr}(b) \mid \mathrm{null}(b))$$

interaction site of output protein

(recursive, parametric) process definition

stochastic delay (τ) with rate ε of constitutive transcription

and repeat

output protein (transcription factor), spawn out

A stochastic rate r is always associated with each channel a_r (at channel creation time) and delay τ_r, but is often omitted when unambiguous.

Production and Degradation

Degradation is extremely important and often deliberate; it changes unbounded growth into (roughly) stable signals.

transcripton factor

and repeat

degradation

$$tr(p) \triangleq (!p_r;\ tr(p)) + \tau_\delta$$

degradation rate δ

interaction site of transcription factor

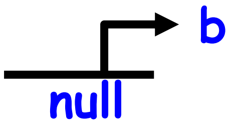
stochastic choice (race between r and δ)

(output, !) interaction with rate r (input, ?, is on the target gene)

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

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

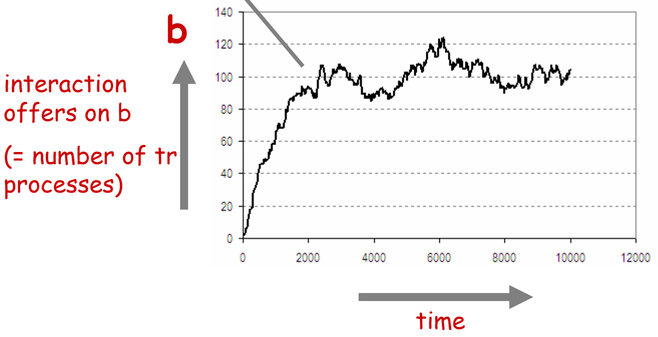
product



$$null(b) \triangleq \tau_\varepsilon;\ (tr(b)\ |\ null(b))$$

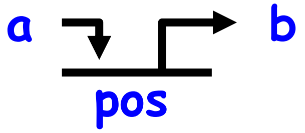
null(b)

ε=0.1, δ=0.001



Unary Pos Gate

input (excitatory)



output (stimulated or constitutive)

$$\mathrm{pos}(a,b) \triangleq$$
$$?a_r;\ \tau_\eta;\ (\mathrm{tr}(b) \mid \mathrm{pos}(a,b))\ +$$
$$\tau_\varepsilon;\ (\mathrm{tr}(b) \mid \mathrm{pos}(a,b))$$

(input, ?) interaction with rate r

transcripton delay with rate η

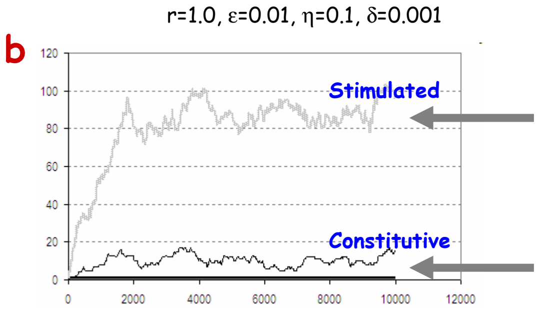
or constitutive transcription to always get things started

race between r and ε

output protein

parallel, not sequence, to handle self-loops without deadlock

unlimited amount of

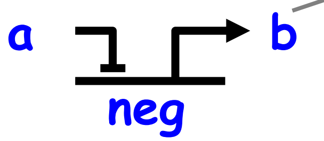


$*\mathrm{tr}(a_r) \mid \mathrm{pos}(a_r,b)$

$\mathrm{pos}(a,b)$

Unary Neg Gate

input (inhibitory)



output (constitutive when not inhibited)

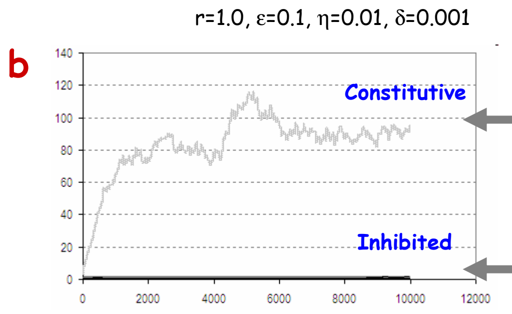
(input, ?) interaction with rate r

or constitutive transcription to always get things started

$$neg(a,b) \triangleq$$
$$?a_r;\ \tau_\eta;\ neg(a,b)\ +$$
$$\tau_\varepsilon;\ (tr(b)\ |\ neg(a,b))$$

inhibition delay with rate η

race between r and ε

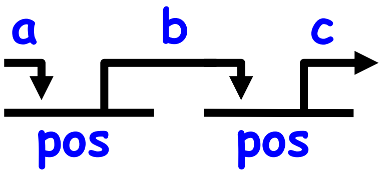


$neg(a_r,b)$

$*tr(a_r)\ |\ neg(a_r,b)$

Signal Amplification

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

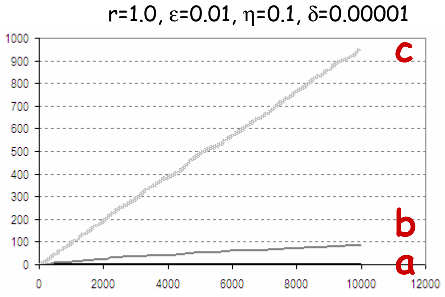


$\mathrm{pos}(a,b) \triangleq$
$\quad ?a_r;\ \tau_\eta;\ (\mathrm{tr}(b) \mid \mathrm{pos}(a,b)) +$
$\quad \tau_\varepsilon;\ (\mathrm{tr}(b) \mid \mathrm{pos}(a,b))$

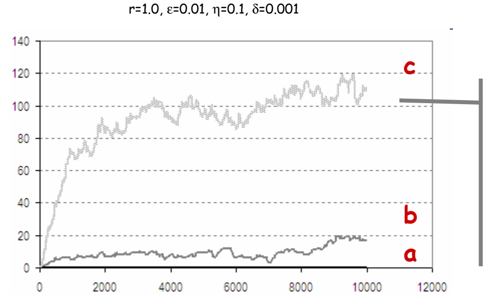
$\mathrm{tr}(p) \triangleq (!p_r;\ \mathrm{tr}(p)) + \tau_\delta$

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



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

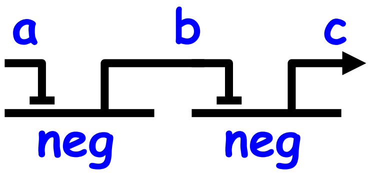


pos(a,b) | pos(b,c)

even with no a input, consitutive production of b gets amplified to a high c signal

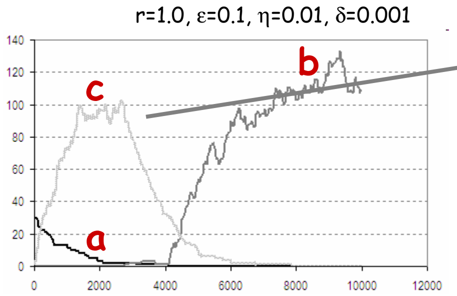
Signal Normalization

neg(a,b) |
neg(b,c)



$\mathrm{neg}(a,b) \triangleq$
$\quad ?a_r;\ \tau_h;\ \mathrm{neg}(a,b)\ +$
$\quad \tau_\varepsilon;\ (\mathrm{tr}(b)\ |\ \mathrm{neg}(a,b))$

$\mathrm{tr}(p) \triangleq (!p_r;\ \mathrm{tr}(p)) + \tau_\delta$

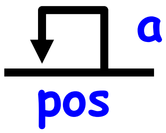


^{30*}tr(a) | neg(a,b) | neg(b,c)

a non-zero input level, a, whether weak or strong, is renormalized to a standard level, c.

Self Feedback Circuits

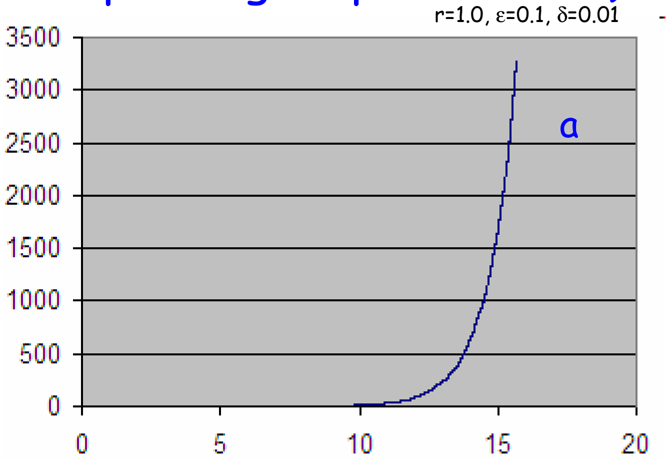
pos(a,a)



$\text{pos}(a,b) \triangleq$
$\quad ?a_r;\ (\text{tr}(b) \mid \text{pos}(a,b))\ +$
$\quad \tau_\varepsilon;\ (\text{tr}(b) \mid \text{pos}(a,b))$

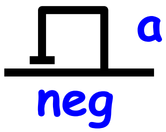
$\text{tr}(p) \triangleq (!p_r;\ \text{tr}(p)) + \tau_\delta$

(Can overwhelm degradation, depending on parameters)



pos(a,a)

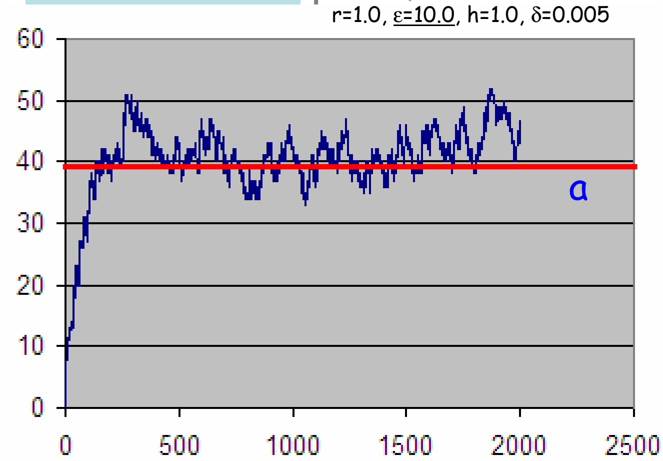
neg(a,a)



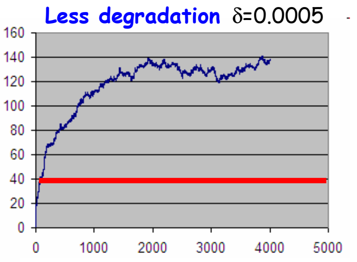
$\text{neg}(a,b) \triangleq$
$\quad ?a_r;\ \tau_h;\ \text{neg}(a,b)\ +$
$\quad \tau_\varepsilon;\ (\text{tr}(b) \mid \text{neg}(a,b))$

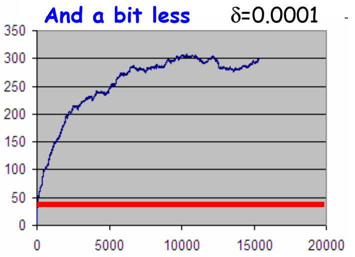
$\text{tr}(p) \triangleq (!p_r;\ \text{tr}(p)) + \tau_\delta$

high, to raise the signal



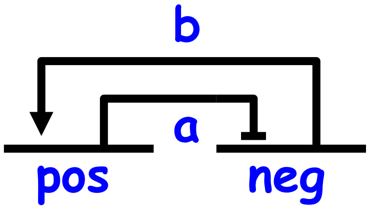
neg(a,a)





Two-gate Feedback Circuits

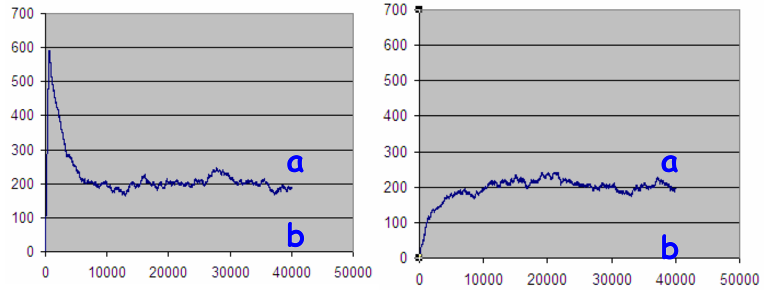
pos(b,a) | neg(a,b)



Monostable:

For some degradation rates is quite stable:

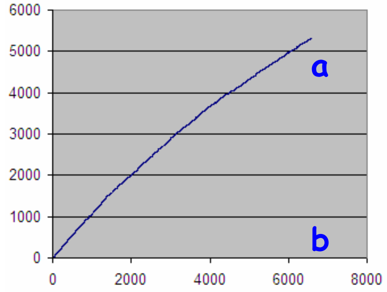
r=1.0, ε=0.1, h=0.01, δ=0.0005



pos(b,a) | neg(a,b)

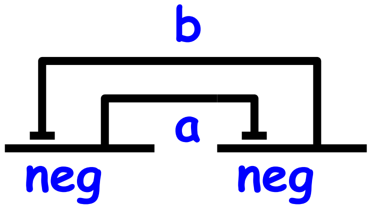
But with a small change in degradation, it goes wild:

r=1.0, ε=0.1, h=0.01, δ=0.0001



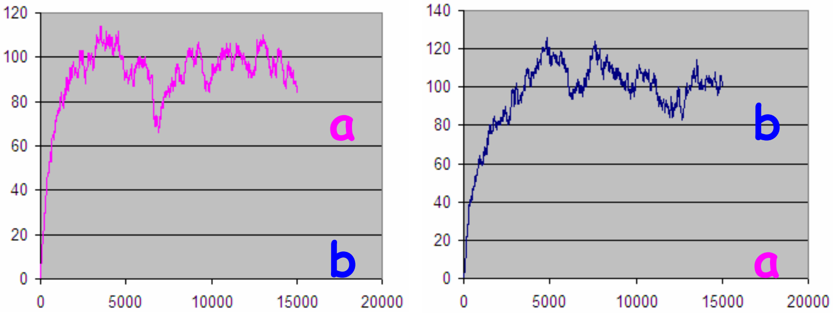
pos(b,a) | neg(a,b)

neg(b,a) | neg(a,b)



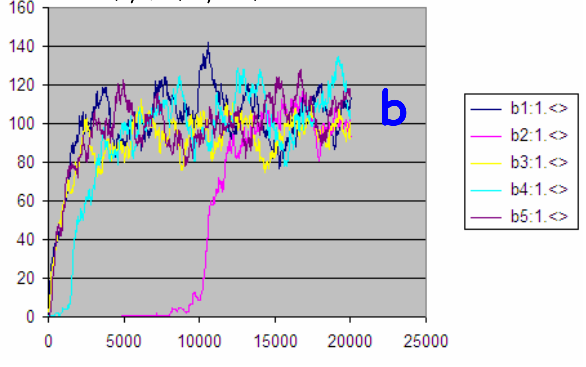
Bistable:

r=1.0, ε=0.1, h=0.01, δ=0.001



neg(b,a) | neg(a,b)

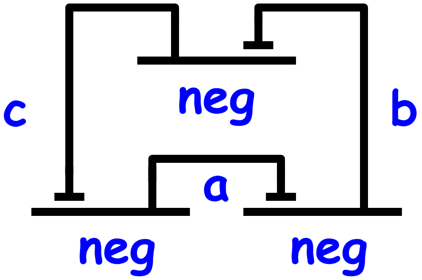
ε=0.1, h=0.01, δ=0.001



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

Repressilator

neg(a,b) |
neg(b,c) |
neg(c,a)



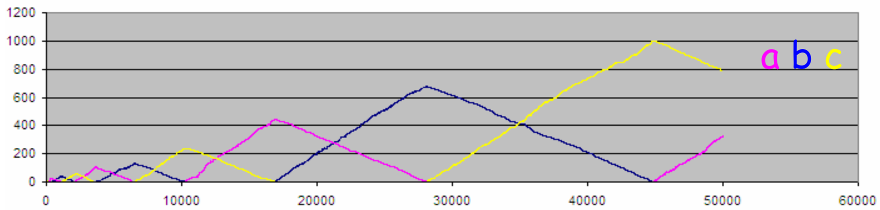
$\mathrm{neg}(a,b) \triangleq$
$\quad ?a_r;\ \tau_h;\ \mathrm{neg}(a,b)\ +$
$\quad \tau_\varepsilon;\ (\mathrm{tr}(b)\ |\ \mathrm{neg}(a,b))$

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

$\mathrm{tr}(p) \triangleq\ !p_r$

interact once and die
otherwise stick around

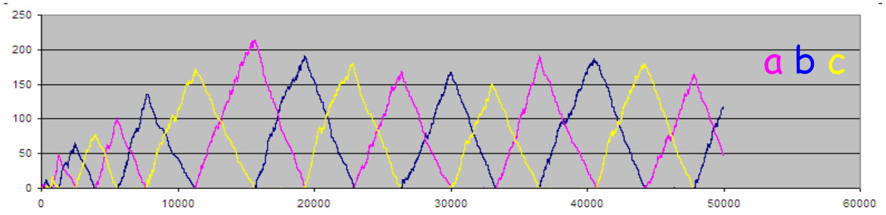
r=1.0, ε=0.1, h=0.04



$\mathrm{tr}(p) \triangleq\ !p_r + \tau_\delta$

interact once and die
otherwise decay

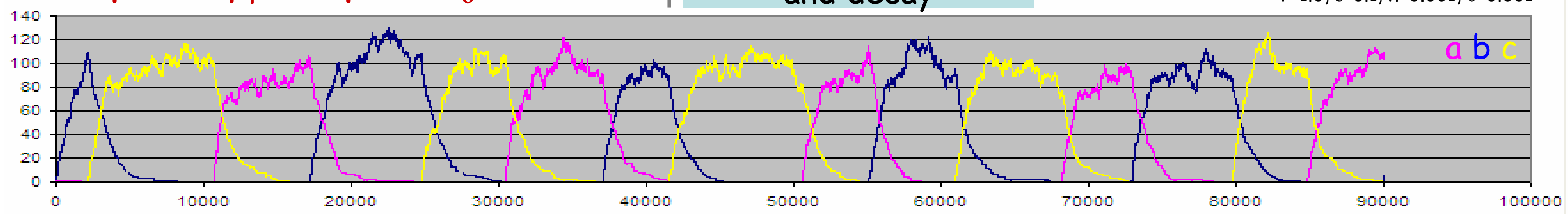
r=1.0, ε=0.1, h=0.04, δ=0.0001



$\mathrm{tr}(p) \triangleq (!p_r;\ \mathrm{tr}(p)) + \tau_\delta$

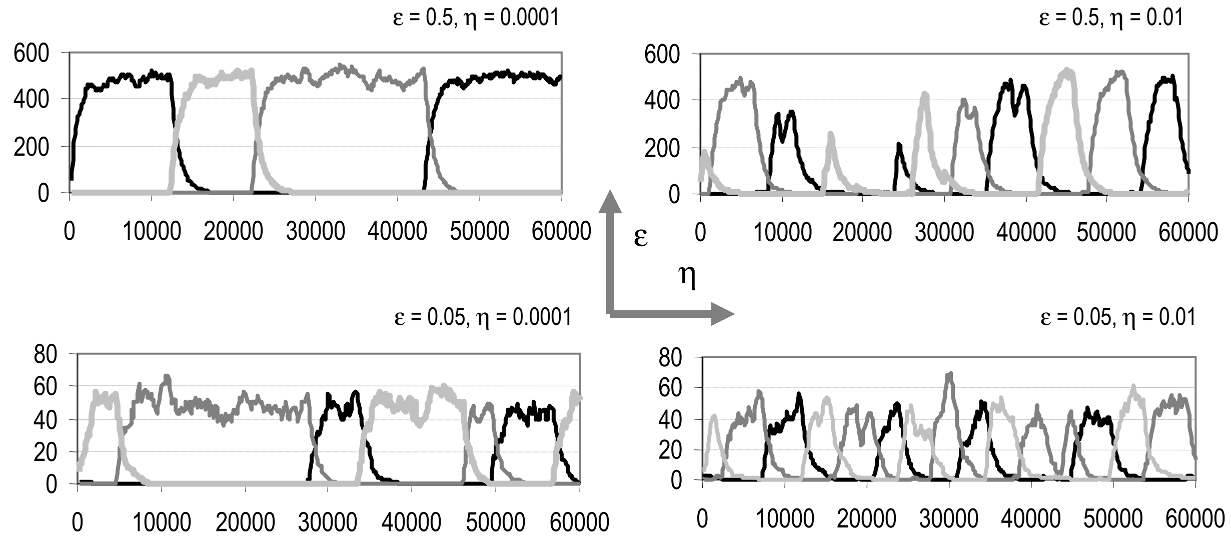
interact many times
and decay

r=1.0, ε=0.1, h=0.001, δ=0.001

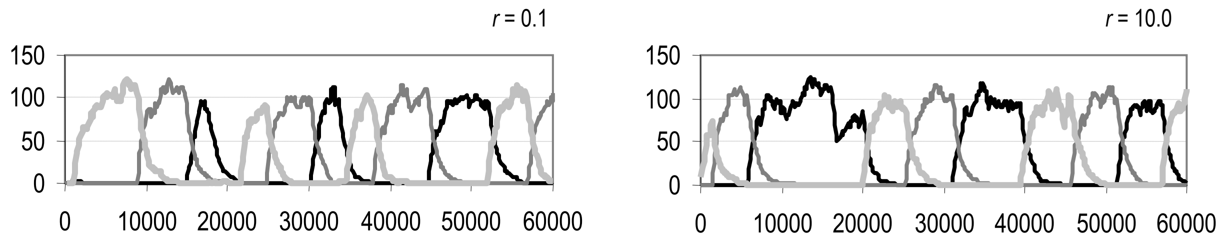


Subtle… at any point one gate is inhibited and the other two can fire constitutively. If one of them fires first, nothing really changes, but if the other one fires first, then the cycle progresses.

System Properties: Oscillation Parameters



The constitutive rate ε (together with the degradation rate) determines oscillation amplitude, while the inhibition rate η determines oscillation frequency.

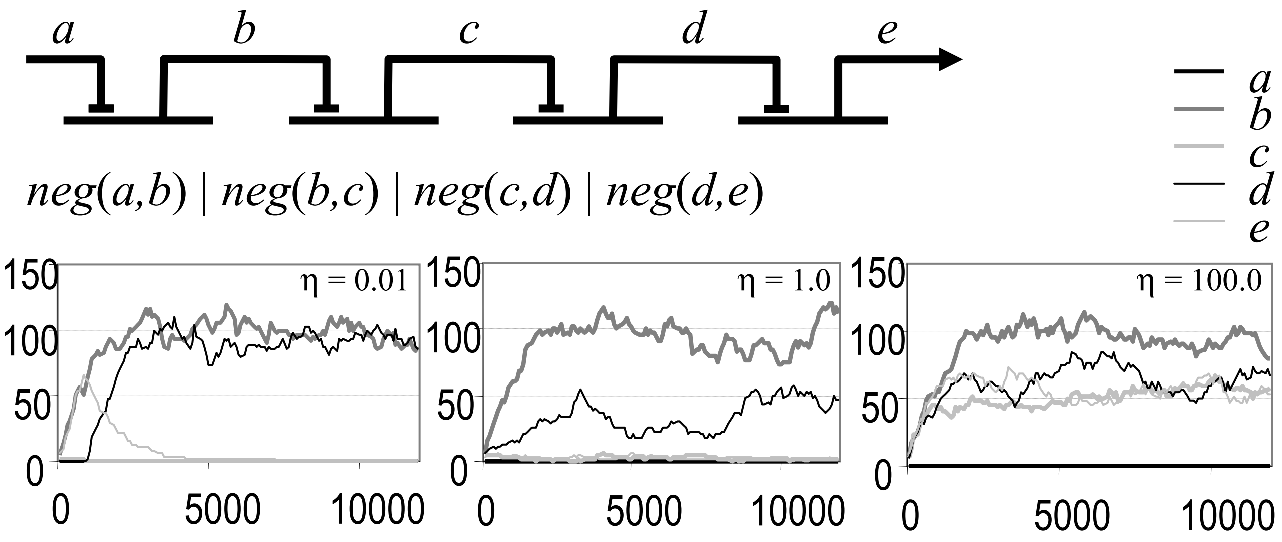


We can view the interaction rate *r* as a measure of the volume (or temperature) of the solution; that is, of how often transcription factors bump into gates. Oscillation frequency and amplitude remain unaffected in a large range of variation of *r*.

Repressilator in SPiM

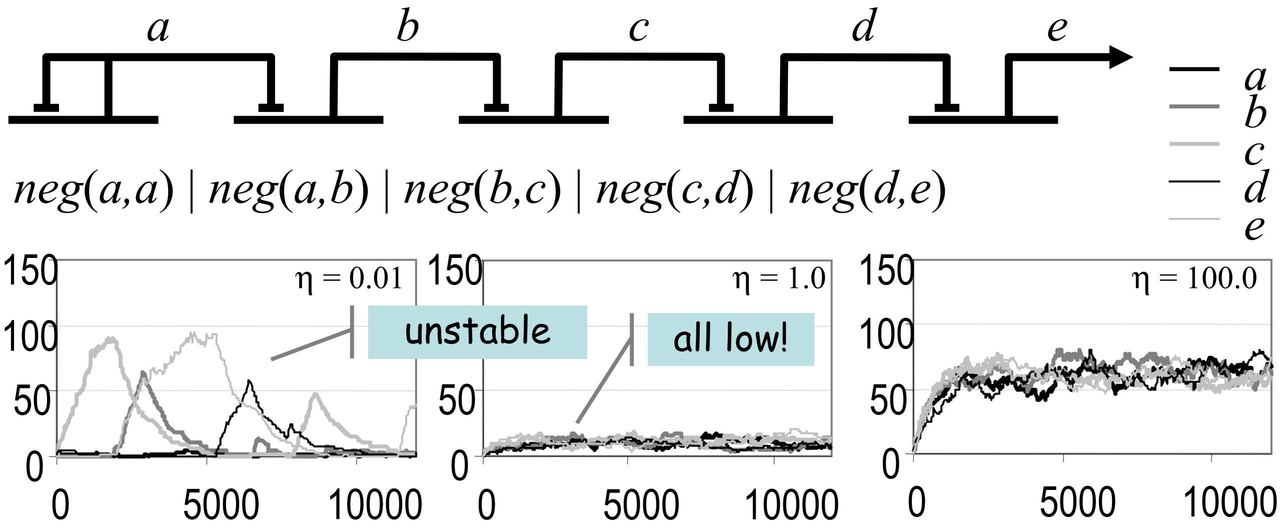
```
val dk = 0.001     (* Decay rate *)
val eta = 0.001    (* Inhibition rate *)
val cst = 0.1      (* Constitutive rate *)

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



neg(*a*,*b*) | *neg*(*b*,*c*) | *neg*(*c*,*d*) | *neg*(*d*,*e*)

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



neg(*a*,*a*) | *neg*(*a*,*b*) | *neg*(*b*,*c*) | *neg*(*c*,*d*) | *neg*(*d*,*e*)

unstable

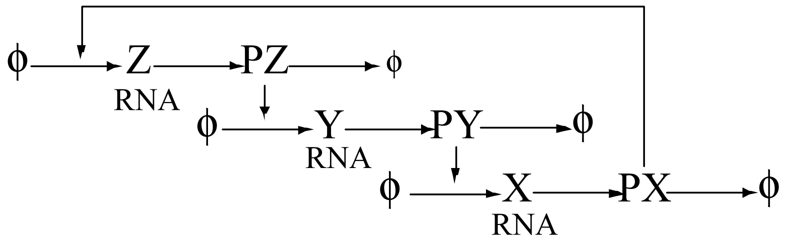
all low!

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

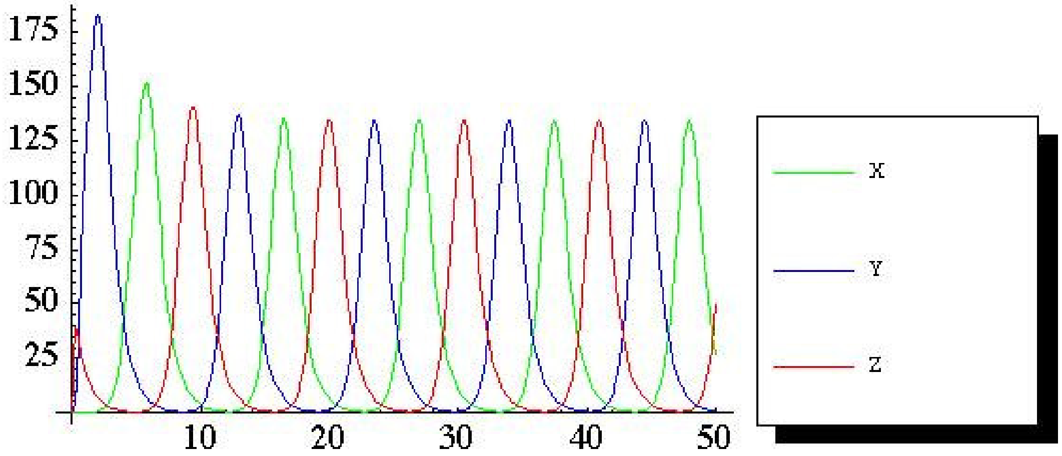
Bruce E Shapiro
Cellerator



$$\frac{d[\mathrm{X}]}{dt}=\alpha_0+\frac{\alpha+\alpha_1[\mathrm{PY}]^n}{K^n+[\mathrm{PY}]^n}-k[\mathrm{X}],\quad \frac{d[\mathrm{PX}]}{dt}=\beta\{[\mathrm{X}]-[\mathrm{PX}]\}$$

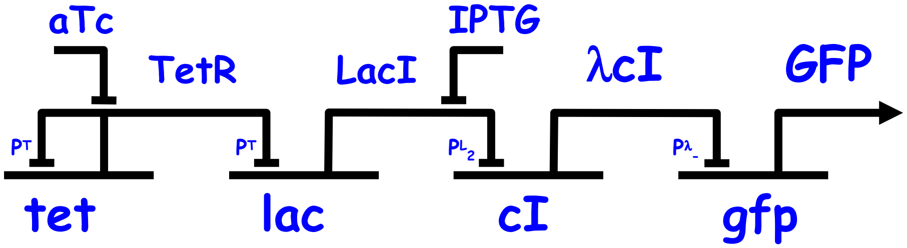
$$\frac{d[\mathrm{Y}]}{dt}=\alpha_0+\frac{\alpha+\alpha_1[\mathrm{PZ}]^n}{K^n+[\mathrm{PZ}]^n}-k[\mathrm{Y}],\quad \frac{d[\mathrm{PY}]}{dt}=\beta\{[\mathrm{Y}]-[\mathrm{PY}]\}$$

$$\frac{d[\mathrm{Z}]}{dt}=\alpha_0+\frac{\alpha+\alpha_1[\mathrm{PX}]^n}{K^n+[\mathrm{PX}]^n}-k[\mathrm{Z}],\quad \frac{d[\mathrm{PZ}]}{dt}=\beta\{[\mathrm{Z}]-[\mathrm{PZ}]\}$$



Guet et al.: D038/lac$^-$

Combinatorial Synthesis of Genetic Networks, **Guet, Elowitz, Hsing, Leibler, 1996,** *Science*, May 2002, 1466-1470.

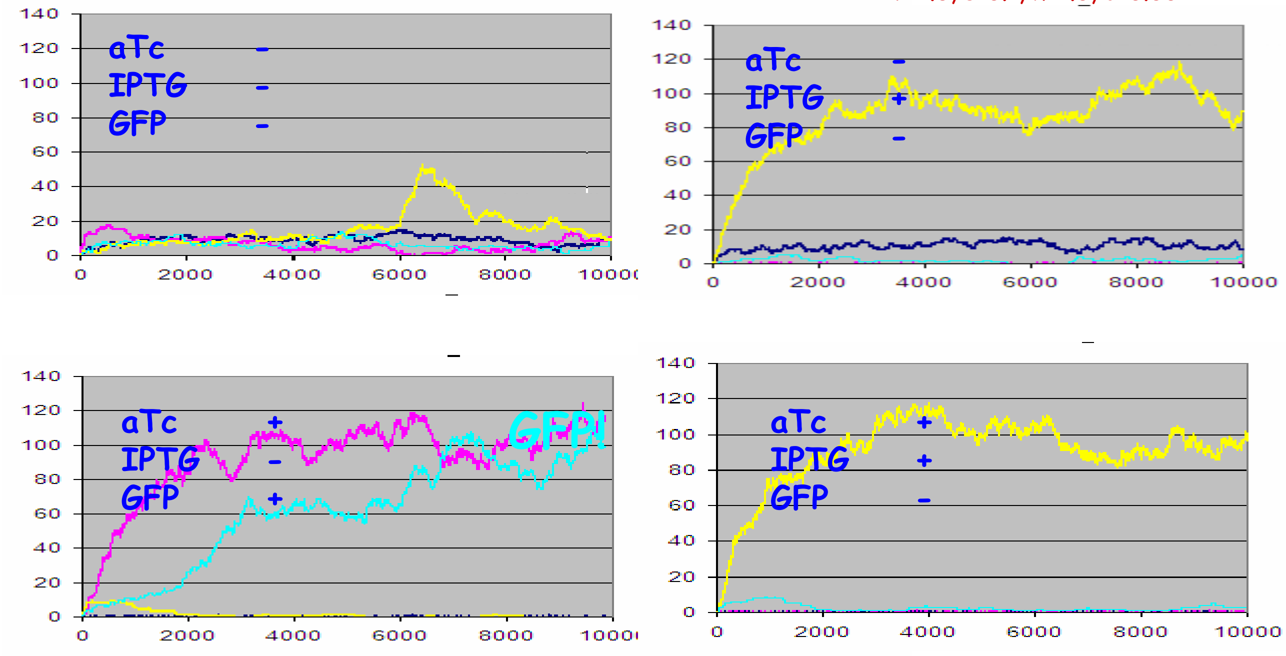


experiment:

aTc	**-**	**+**	**-**	**+**
IPTG	**-**	**-**	**+**	**+**
GFP	**-**	**+**	**-**	**-**
(LacI	-	+	-	-)

neg(TetR,TetR) | neg(TetR,LacI) | neg(LacI,λcI) | neg(λcI,GFP)

r=1.0, ε=0.1, h=1.0, δ=0.001



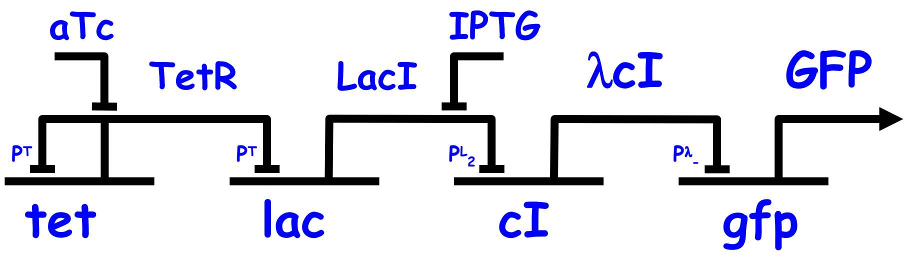
We can model an inducer like aTc as something that competes for the transcription factor.

IPTG de-represses the **lac operon**, by binding to the **lac** repressor (the **lac** I gene product), preventing it from binding to the operator.

Guet et al.

Combinatorial Synthesis of Genetic Networks, **Guet, Elowitz, Hsing, Leibler, 1996,** *Science*, May 2002, 1466-1470.

They engineered in E.Coli all genetic circuits with four single-input gates; such as this one:



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

Experiment:

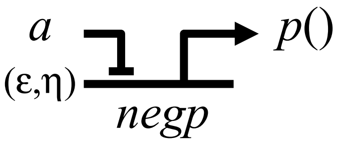
aTc	0101
IPTG	0011
GFP	0100

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



regulatory input

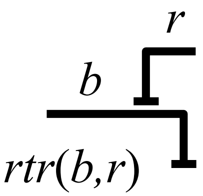
rates

product

$$negp(a,(\varepsilon,\eta),p) =$$

product generation

$$?a.\ \tau_{\eta}.\ negp(a,(\varepsilon,\eta),p)\ +$$
$$\tau_{\varepsilon}.\ (p()\ |\ negp(a,(\varepsilon,\eta),p))$$



repressible factor

$$rtr(b,r) =$$

interaction | $!b.\ rtr(b,r)\ +$ | binding

interaction | $!r.\ 0\ +$ | repression

delay | $\tau_{\delta}.\ 0$ | degradation

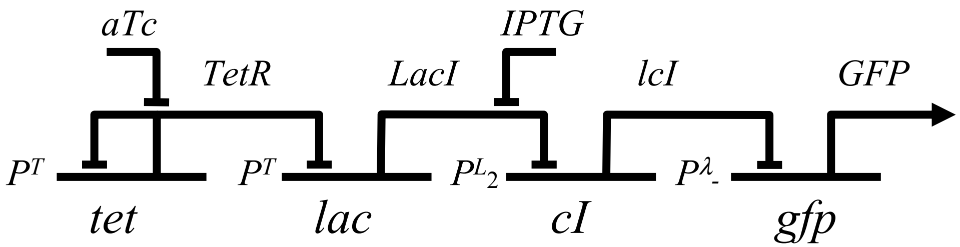
repressor

$$rep(r) = ?r.\ rep(r)$$

arbitray amounts of..

D038/lac$^-$

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.

channels $\boxed{TetR{:}\mathbf{r_1},\ LacI{:}\mathbf{r_2},\ lcI{:}\mathbf{r_3},\ GFP{:}\mathbf{r_4},\ aTc{:}\mathbf{r_5},\ IPTG{:}\mathbf{r_6}}$

molecules

$$\boxed{P^T = (\varepsilon_1, \eta_1) \quad P^L{}_2 = (\varepsilon_2, \eta_2) \quad P^\lambda{}_- = (\varepsilon_3, \eta_3)}$$

promoters

$$tet = negp(TetR,\ P^T,\ rtr(TetR, aTc))$$
$$lac = negp(TetR,\ P^T,\ rtr(LacI, IPTG))$$
$$cI = negp(LacI,\ P^L{}_2,\ tr(lcI))$$
$$gfp = negp(lcI,\ P^\lambda{}_-,\ tr(GFP))$$

genes

repressors (when present)

$$D038lac^- = \boxed{tet \mid lac \mid cI \mid gfp} \ \mid\mid\ \boxed{rep(aTc) \mid rep(IPTG)}$$

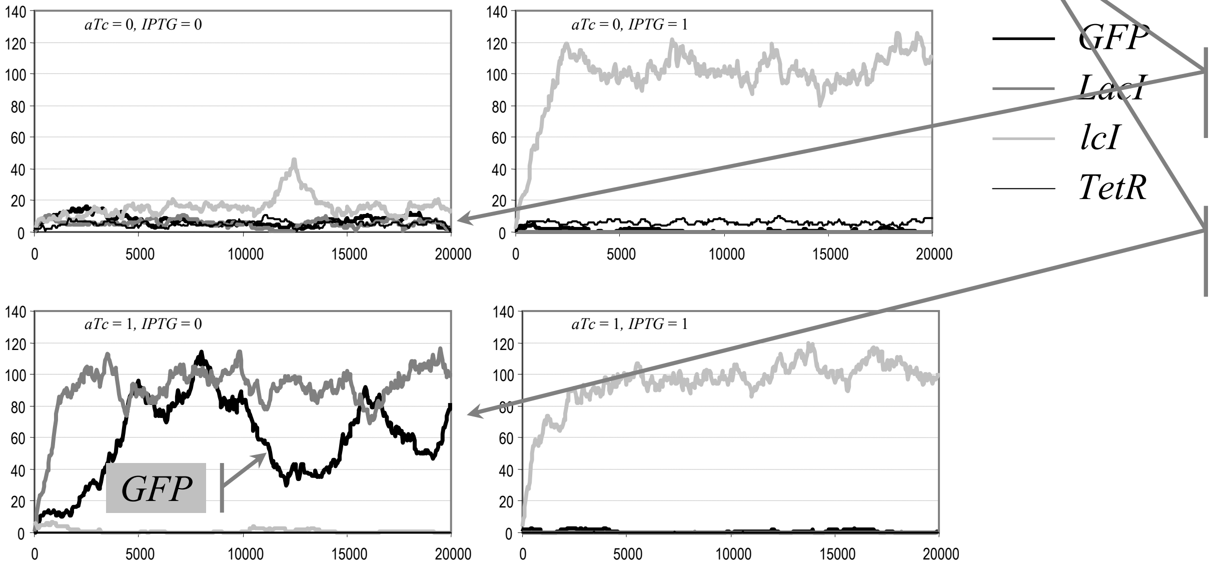
Hence we need to understand better the "dynamics" of this network.

Simulation results for D038/lac$^-$

Experiment:

aTc	0101
IPTG	0011
GFP	0100

r=1.0, ε=0.1, h=1.0, δ=0.001

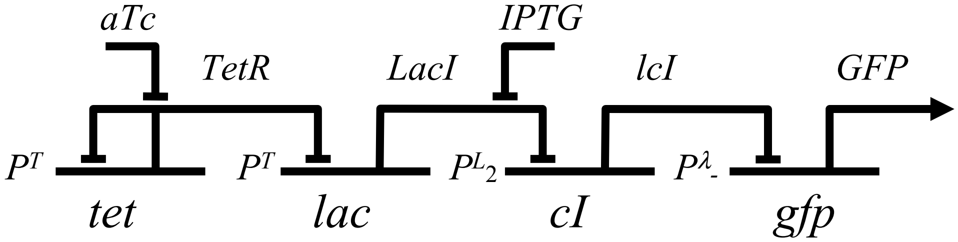


The fixpoint effect can explain this (all signals set very low).

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

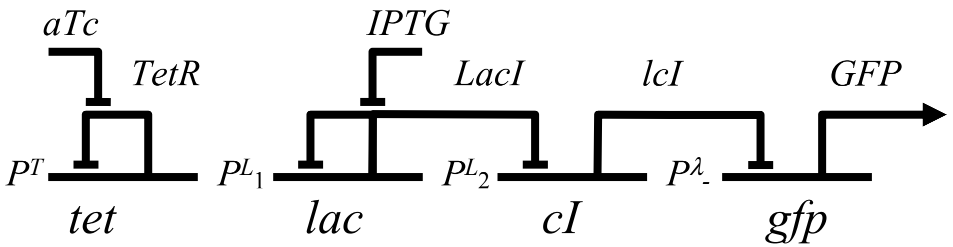
$r = 1.0$, $\varepsilon = 0.1$, $\eta = 0.25$ (P^T), $\eta = 1.0$ ($P^L{}_2$, $P^\lambda{}_-$), $\delta = 0.001$

D038/lac$^-$



D016/lac$^-$

D016/lac$^-$



Experiment:	
aTc	0101
IPTG	0011
GFP	1000

How can aTc affect the result??

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.

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.

__channels__ $\boxed{TetR{:}\mathbf{r_1},\ LacI{:}\mathbf{r_2},\ lcI{:}\mathbf{r_3},\ GFP{:}\mathbf{r_4},\ aTc{:}\mathbf{r_5},\ IPTG{:}\mathbf{r_6}}$

$$\boxed{P^T = [\varepsilon_1, \eta_1] \quad P^L{}_2 = [\varepsilon_2, \eta_2] \quad P^\lambda{}_- = [\varepsilon_3, \eta_3] \quad P^L{}_1 = [\varepsilon_4, \eta_4]}$$

promoters

$tet = negp[TetR,\ P^T,\ rtr[TetR,aTc]]$
$lac = negp[LacI,\ P^L{}_1,\ rtr[LacI,IPTG]]$
$cI = negp[LacI,\ P^L{}_2,\ tr[lcI]]$
$gfp = negp[lcI,\ P^\lambda{}_-,\ tr[GFP]]$

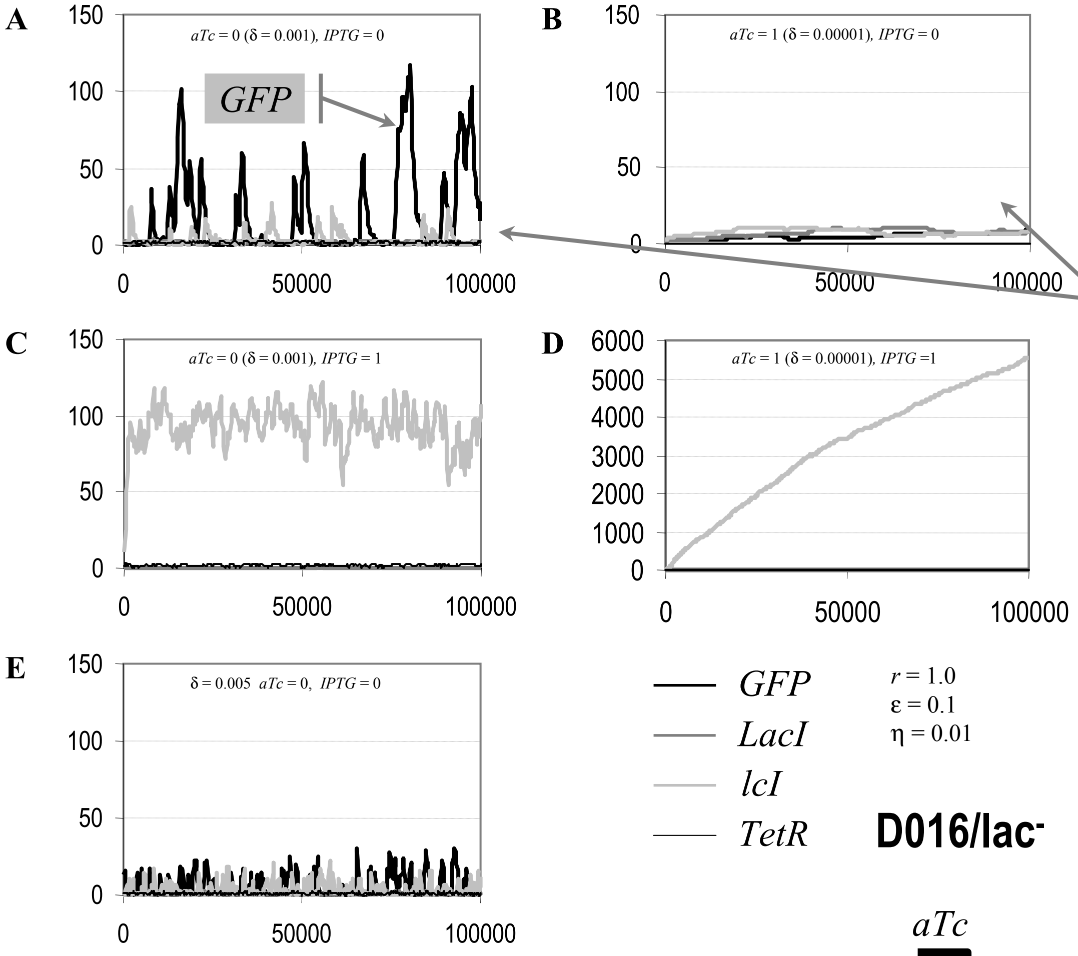
genes

repressors

$$D016lac^- = \boxed{tet \mid lac \mid cI \mid gfp} \mid \boxed{rep[aTc] \mid rep[IPTG]}$$

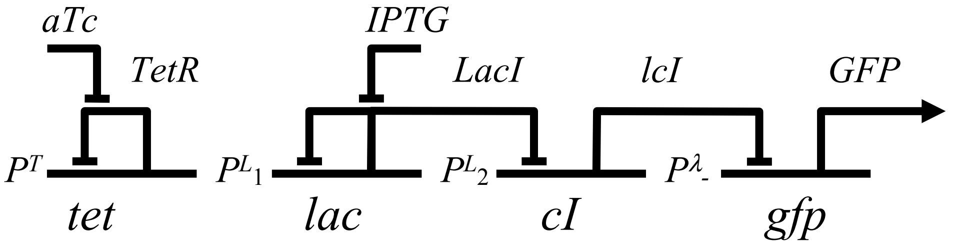
Simulation results for D016/lac$^-$

Experiment:	
aTc	0101
IPTG	0011
GFP	1000



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

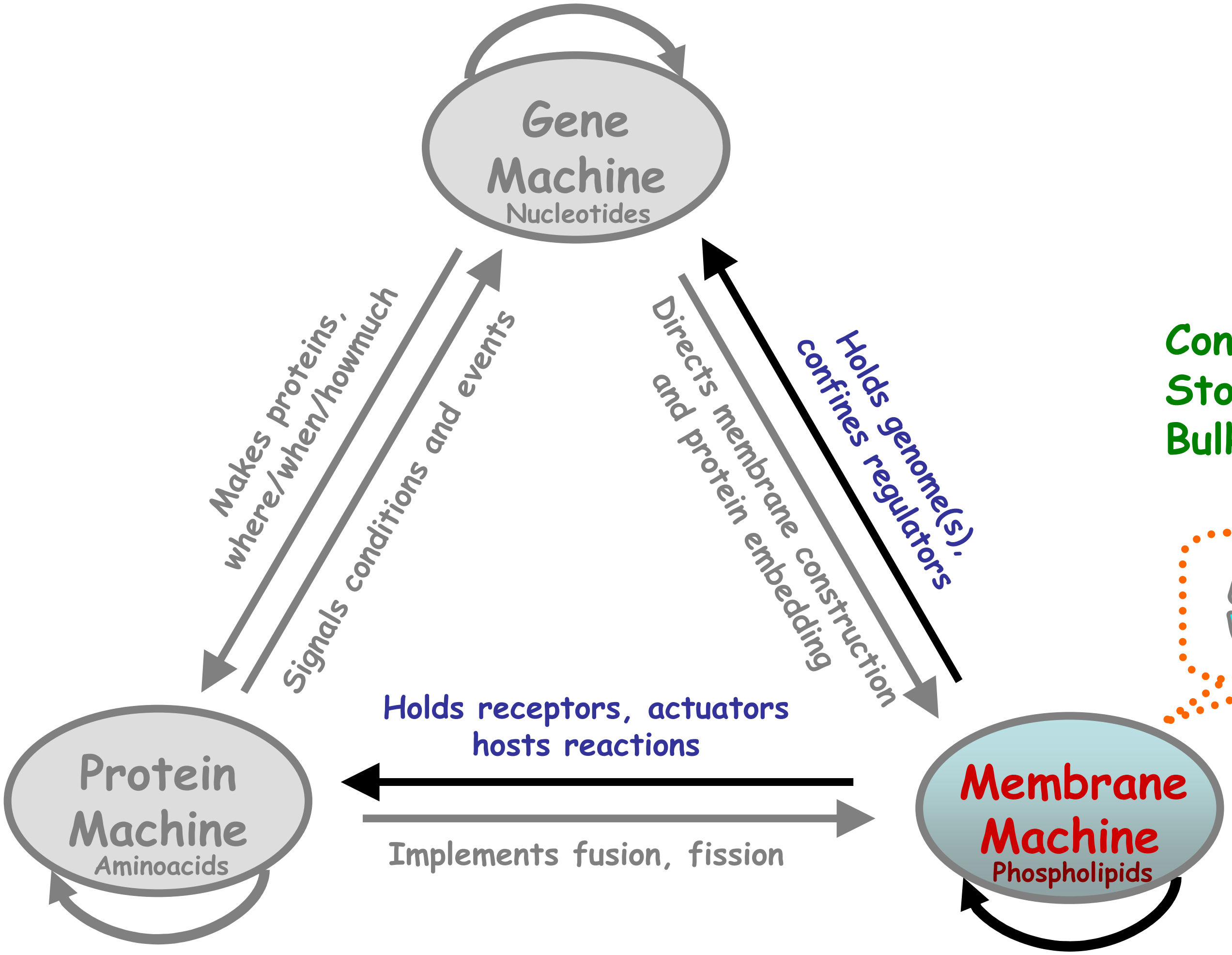
Overloading of degradation machinery, induced by aTc, can reinstate the fixpoint regime.



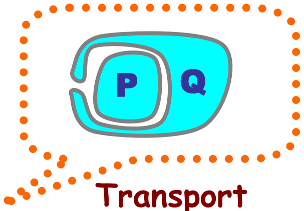
What was the point?

- 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
Nucleotides
Makes proteins, where/when/howmuch
Signals conditions and events
Directs membrane construction and protein embedding
Holds genome(s), confines regulators
Protein
Machine
Aminoacids
Holds receptors, actuators hosts reactions
Implements fusion, fission
Membrane
Machine
Phospholipids

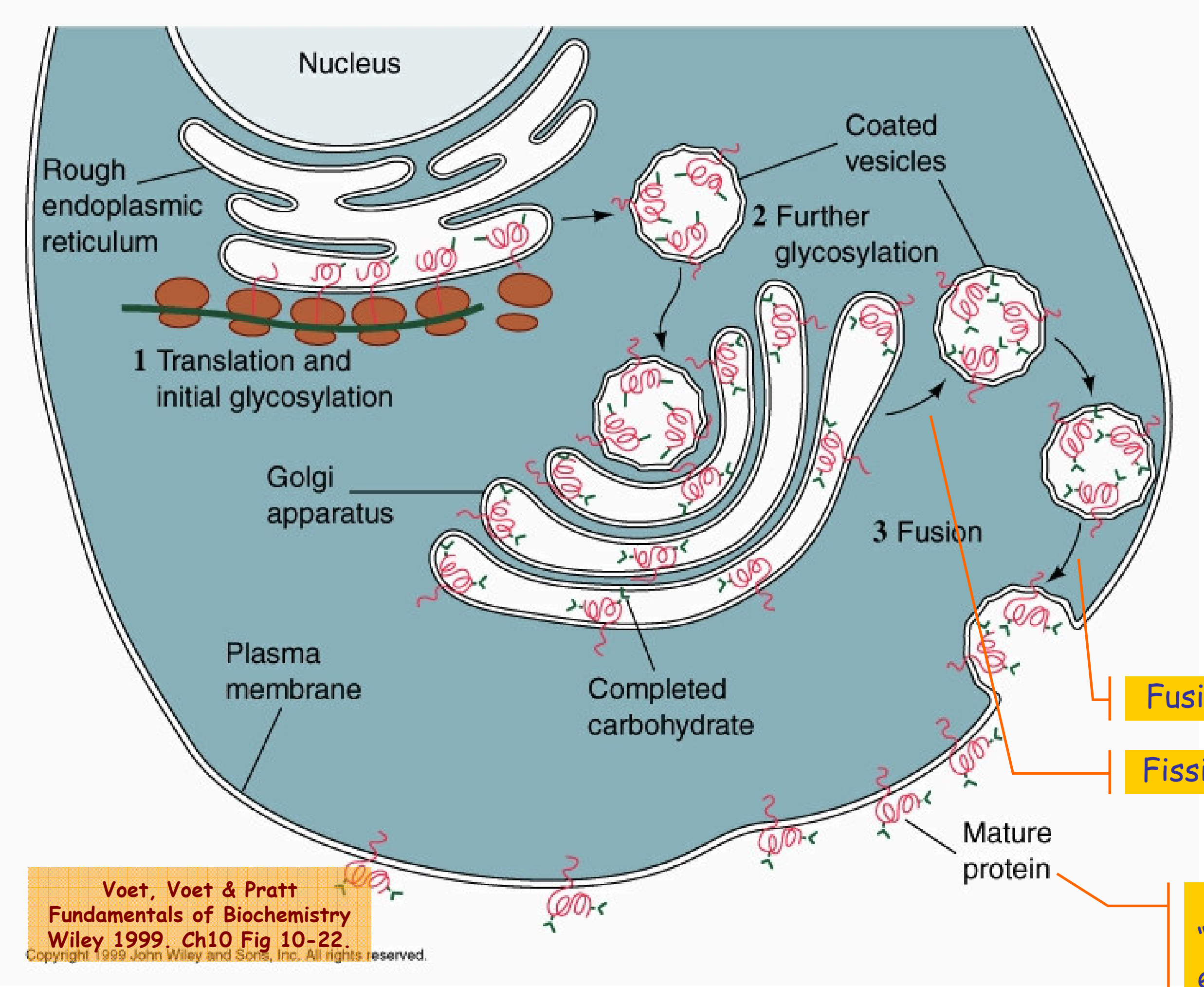
Confinement
Storage
Bulk Transport



Transport
Networks

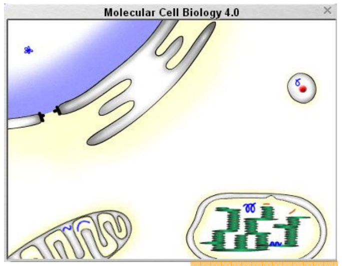
3. The Membrane Machine

Very far from the atoms.



Voet, Voet & Pratt
Fundamentals of Biochemistry
Wiley 1999. Ch10 Fig 10-22.

Molecular transport and transformation through dynamic compartment fusion and fission.

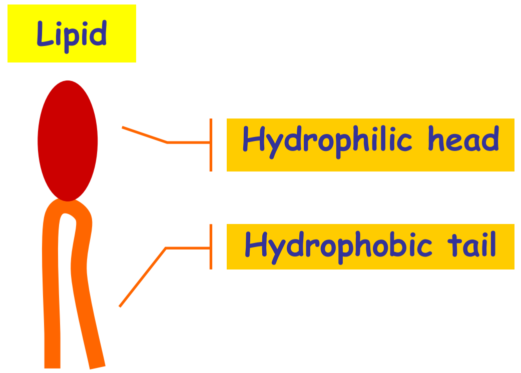


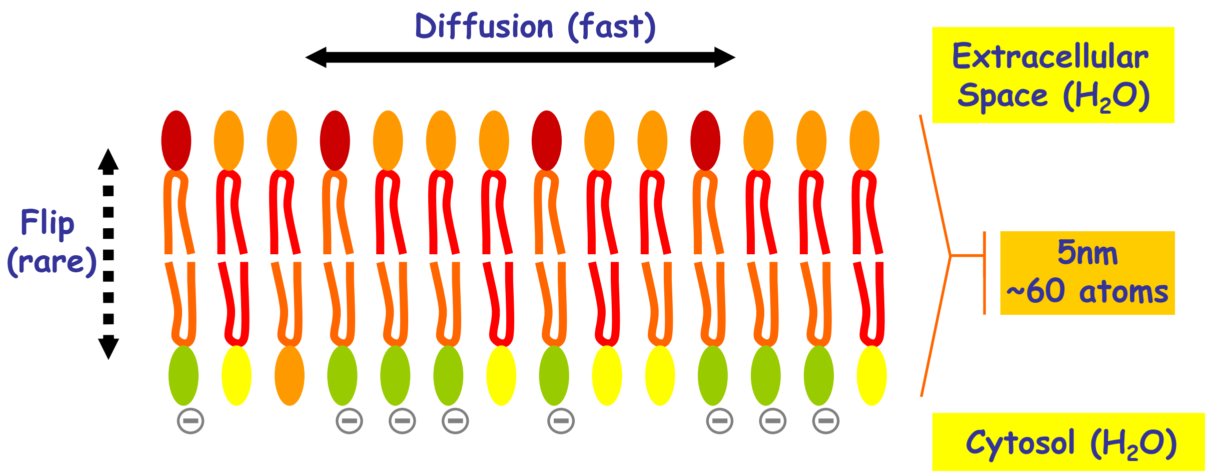
Taken from MCB CD

Fusion
Fission
} **The Instruction Set**

Well, what is all that for?
"Given the complicated pathways that have evolved to synthesize them, it seems likely that these [modified proteins] have important functions, but for the most part these functions are not known" [MBC p.609]

Membranes are Oriented 2D Surfaces





Lipid Bilayer

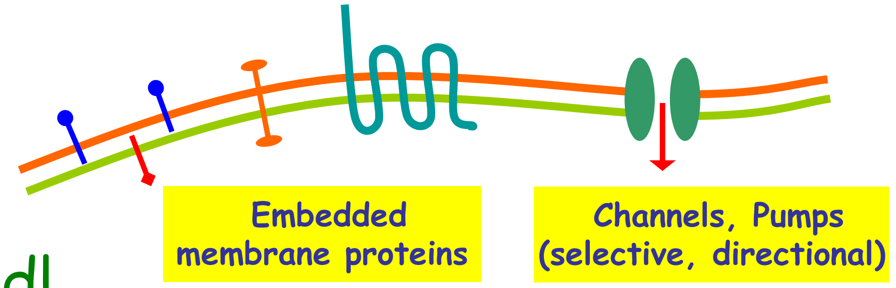
Self-assembling

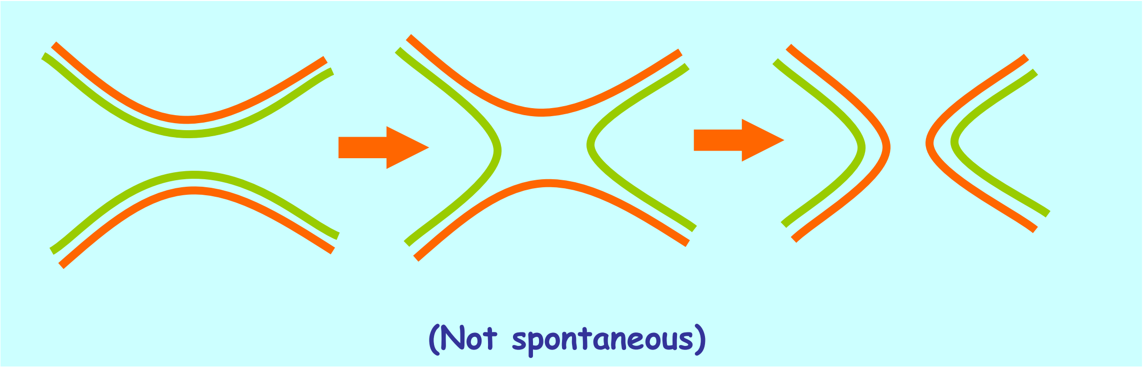
Largely impermeable

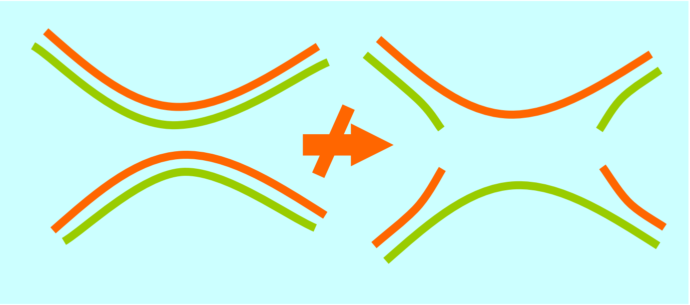
Asymmetrical (in real cells)

With embedded proteins

A 2D fluid inside a 3D fluid!

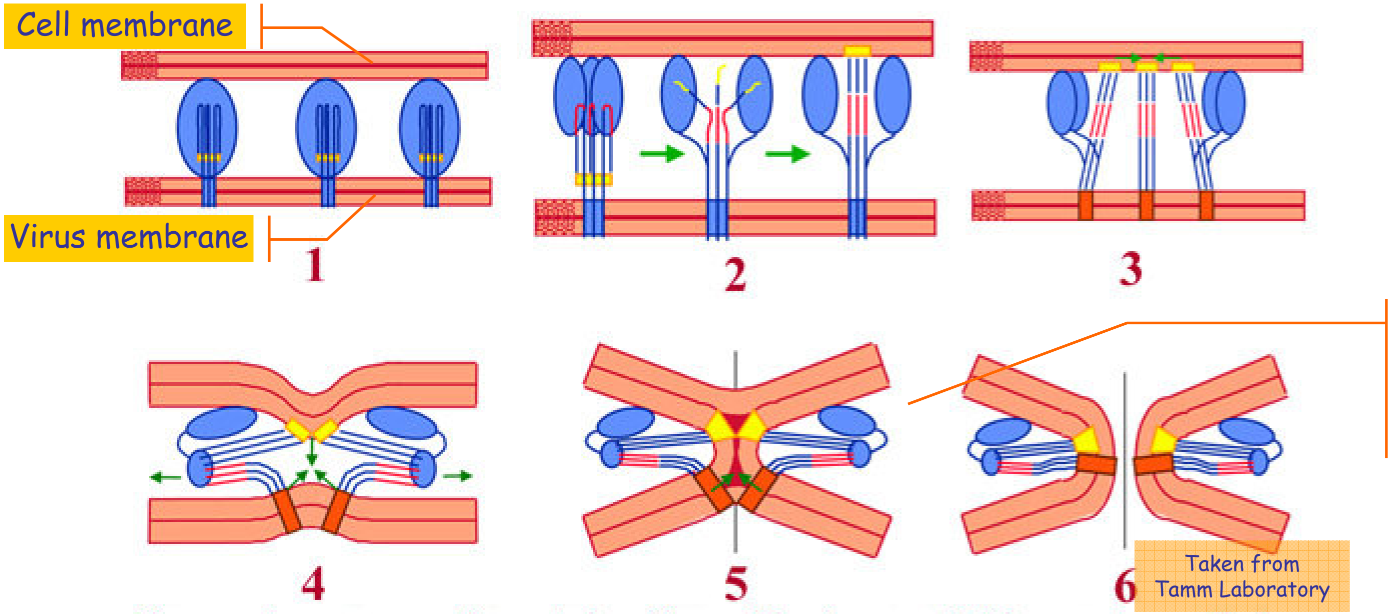

Embedded membrane proteins
Channels, Pumps (selective, directional)





Membrane Fusion

Positive curvature to Negative curvature transition in 3D

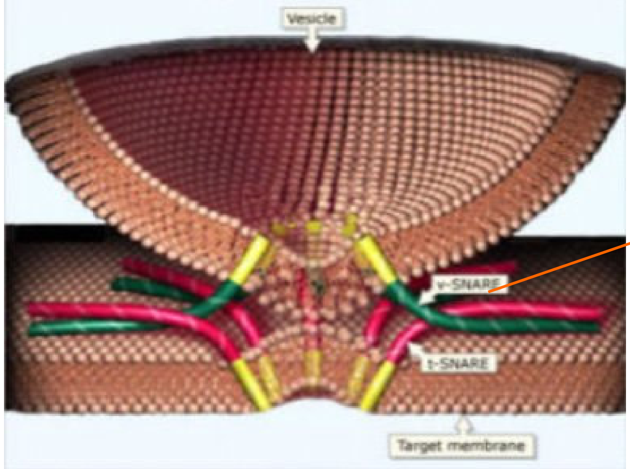


Taken from Tamm Laboratory

Proposed sequence of events in pH sensitive hemagglutinin membrane fusion

Aggressive fusion
(virus)

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

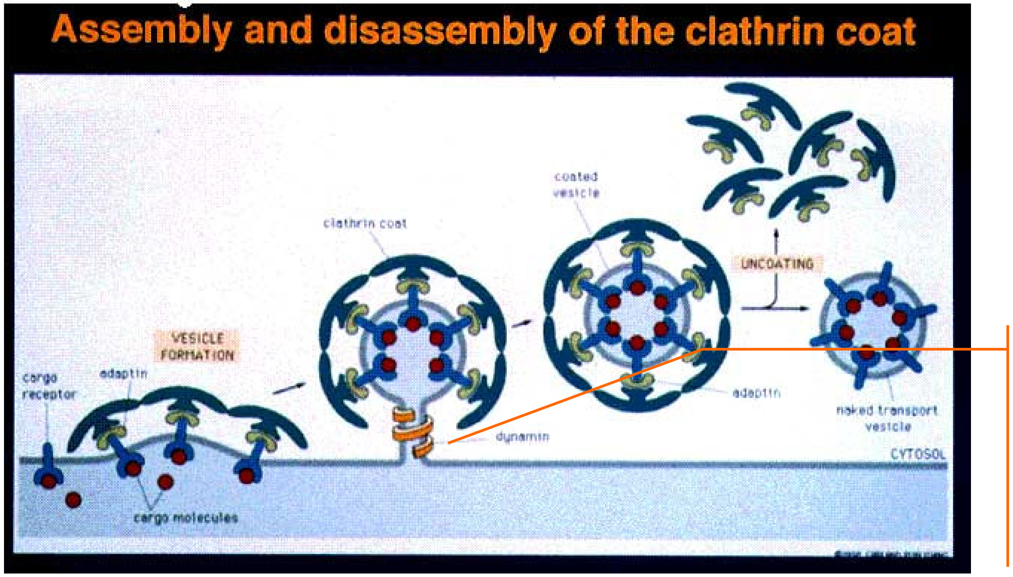


Cooperative fusion
(vesicle)

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

Membrane Fission

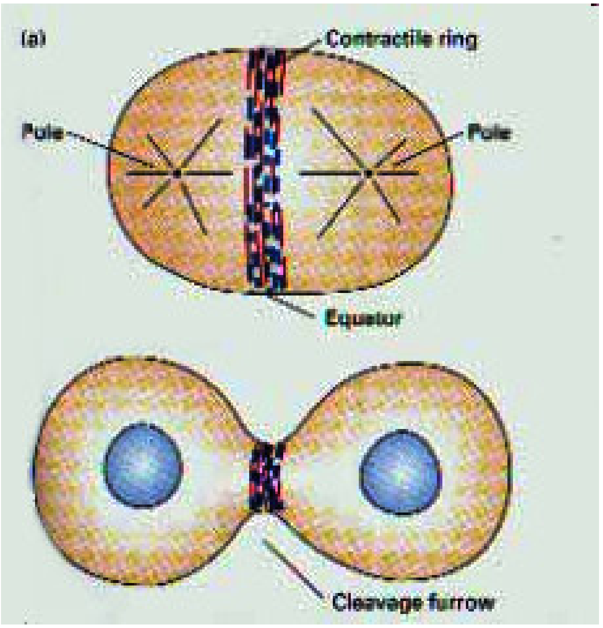
Negative curvature to Positive curvature transition in 3D



Vesicle Formation

Movie by Allison Bruce

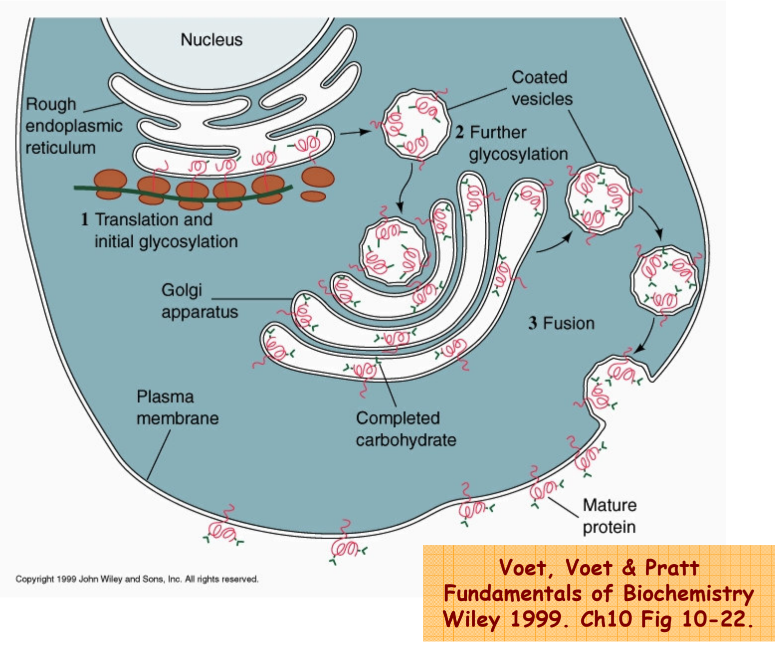
"Nonetheless, the actual process whereby a segment of phospholipid bilayer is 'pinched off' to form a pit and eventually a new vesicle is still not understood" [MCB p.746]



Cytokinesis (Mitosis)

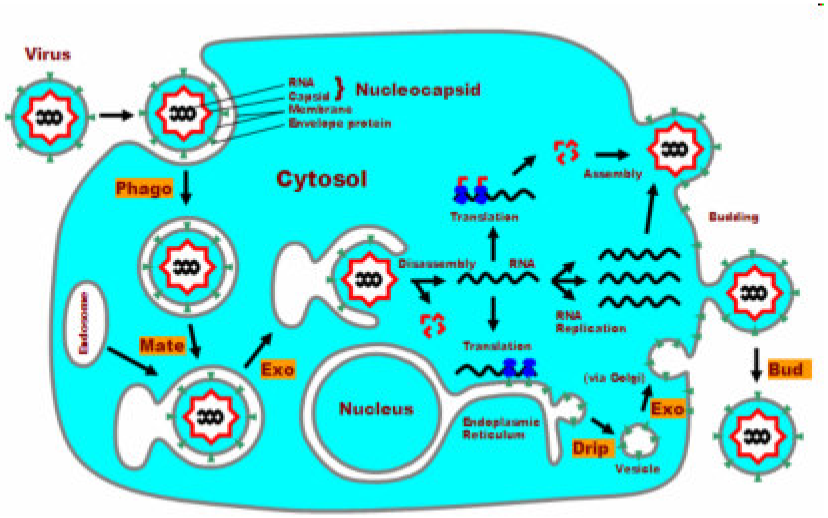
Membrane Algorithms

Protein Production and Secretion


Nucleus
Rough endoplasmic reticulum
1 Translation and initial glycosylation
Golgi apparatus
Plasma membrane
Completed carbohydrate
Coated vesicles
2 Further glycosylation
3 Fusion
Mature protein
Copyright 1999 John Wiley and Sons, Inc. All rights reserved.

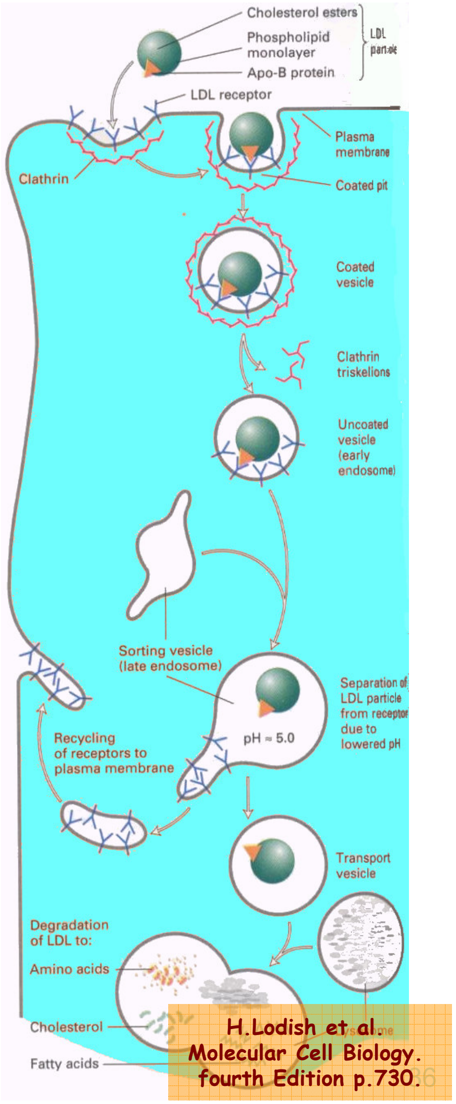
Voet, Voet & Pratt
Fundamentals of Biochemistry
Wiley 1999. Ch10 Fig 10-22.

Viral Replication



Adapted from: B.Alberts et al.
Molecular Biology of the Cell
third edition p.279.

LDL-Cholesterol Degradation



H.Lodish et al.
Molecular Cell Biology.
fourth Edition p.730.

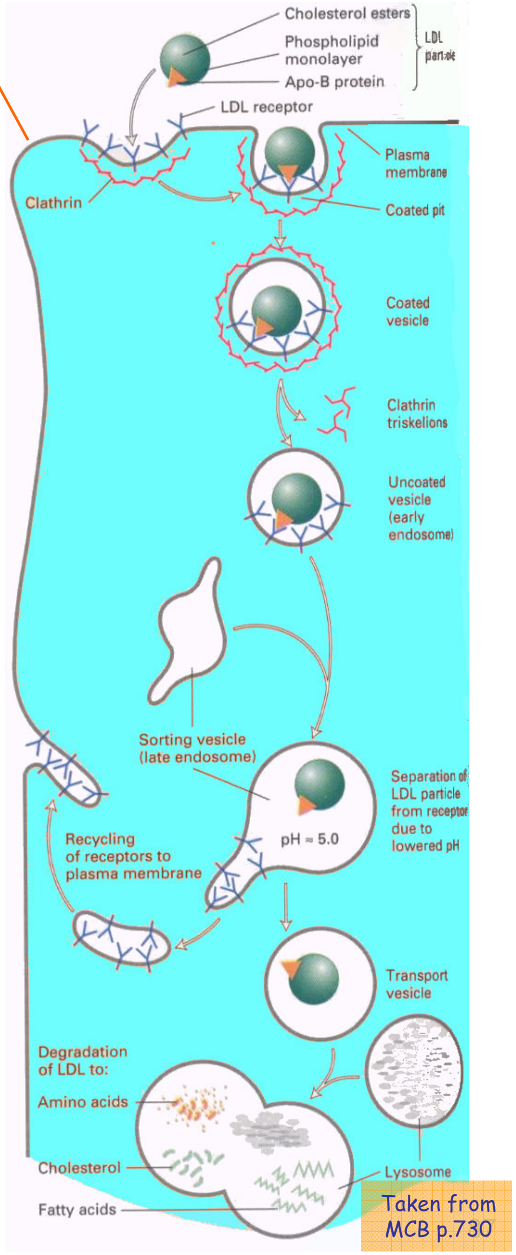
Receptor-Mediate Degradation Pathway

As a “shapshot” diagram

- 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.

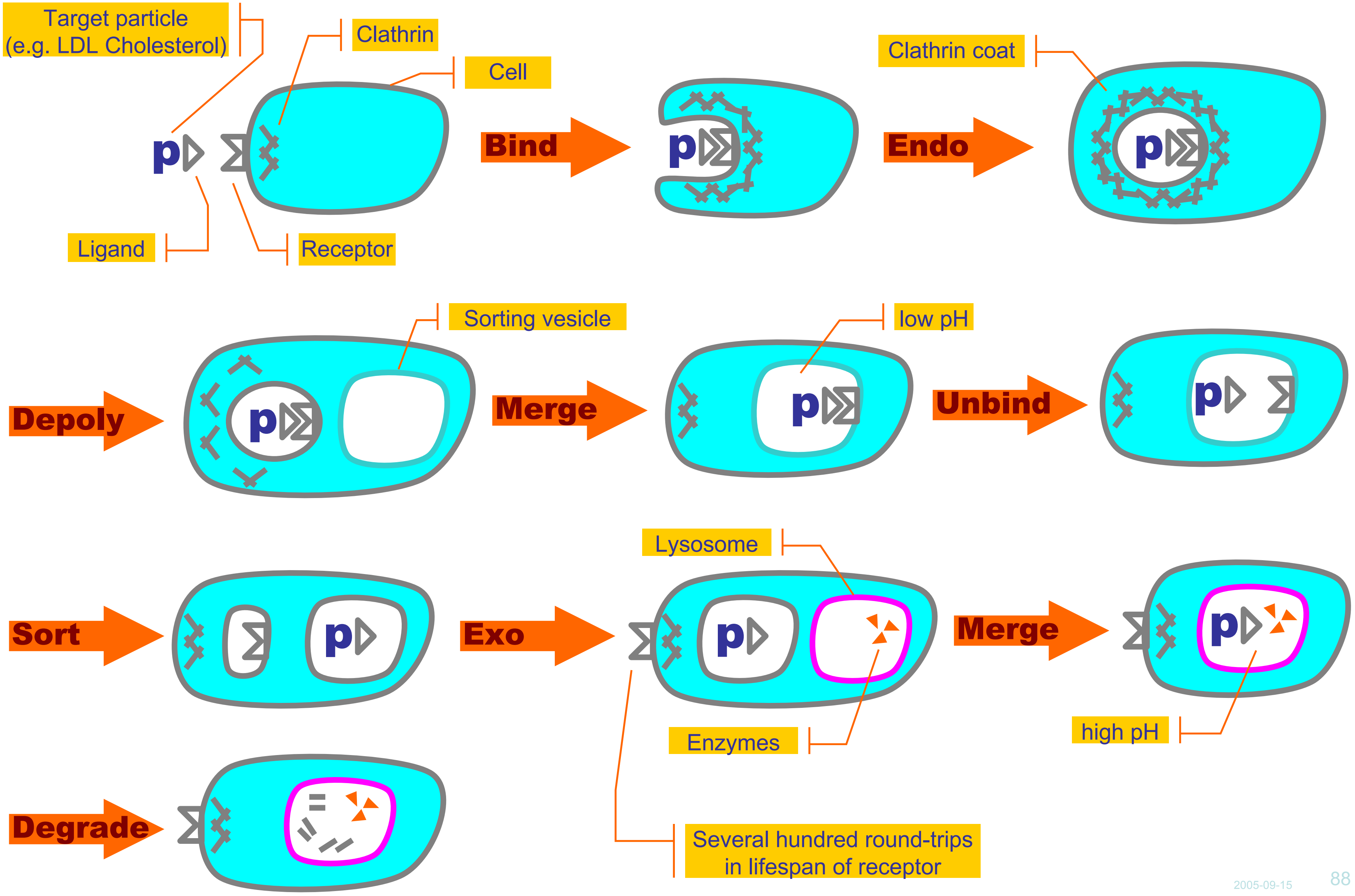
Lipid bilayer



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

Sorting vesicle

low pH

Depoly

Merge

Unbind

Lysosome

Sort

Exo

Merge

Enzymes

high pH

Degrade

Several hundred round-trips
in lifespan of receptor

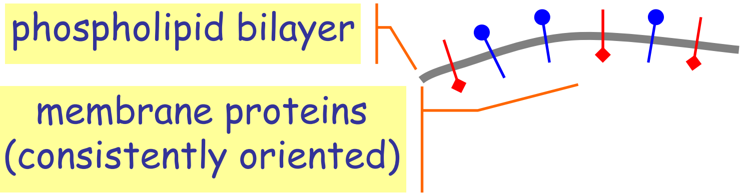
Membrane Orientation

Membranes are closed non-intersecting curves, with an orientation[1].

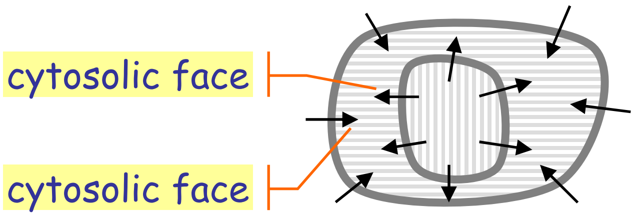
Each membrane has two faces. A cytosolic (~*inner*) face and an exoplasmic (~*outer*) face. Nested membranes alternate orientation. (E.g. cytosolic faces always face each other, by definition, or by fusion/fission dynamics)

This alternation is illustrated by using two tones: blue (cytosol[2]) and white (exosol[3]). Bitonal diagrams.

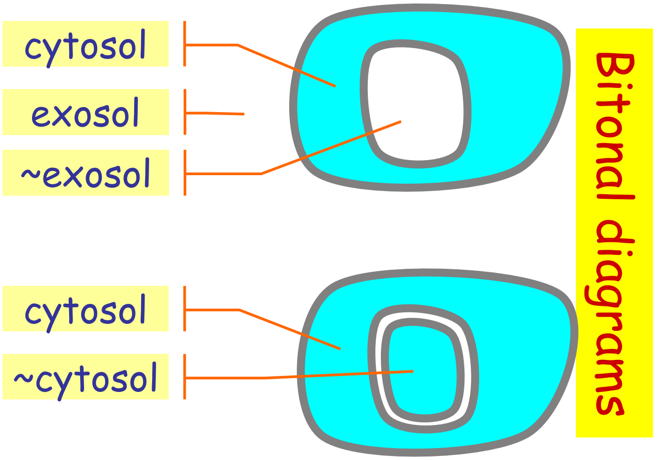
Double membranes (e.g. the nuclear membrane) gives us blue-in-blue components.


phospholipid bilayer

membrane proteins (consistently oriented)


cytosolic face

cytosolic face


cytosol

exosol

~exosol

cytosol

~cytosol

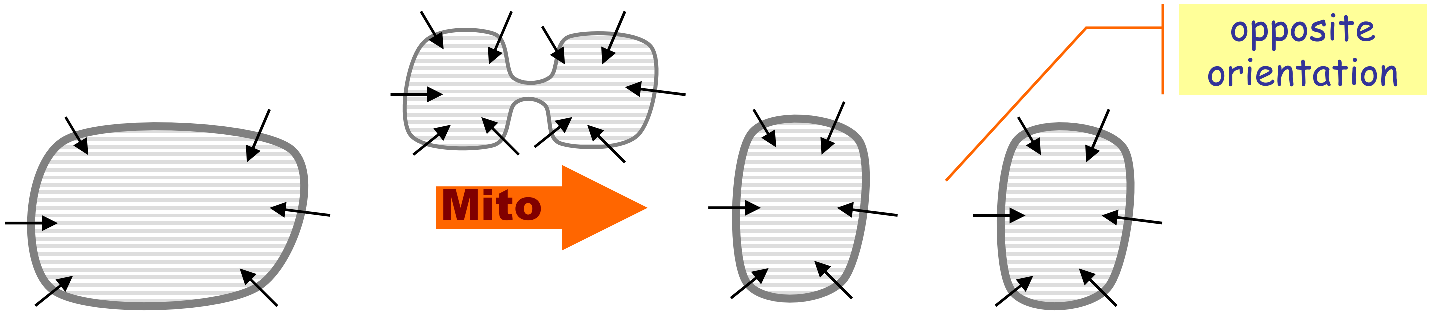
Bitonal diagrams

[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.

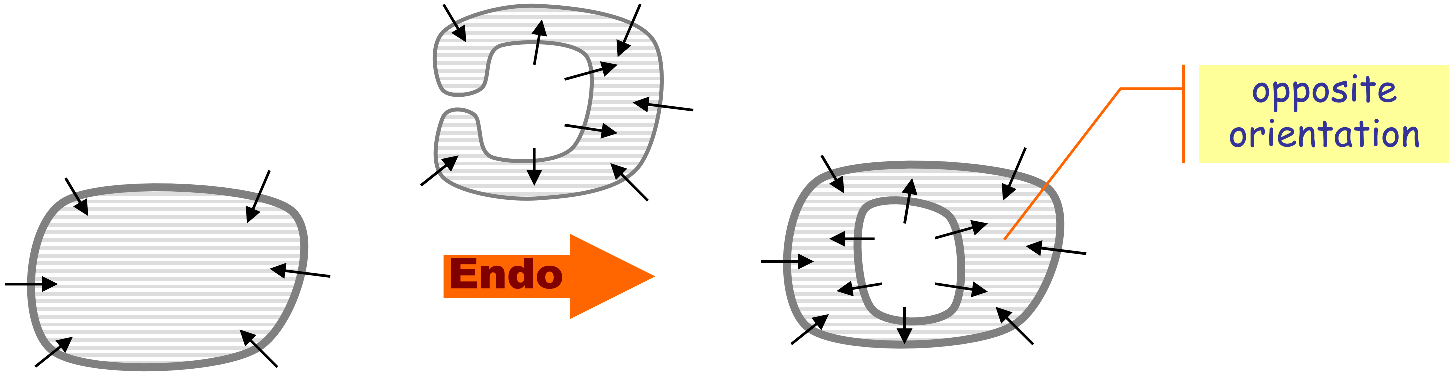
[2] Short for Cytoplasmic Solution. [3] Short for Exoplasmic Region (I am making this one up).

Contiguous Membranes have Opposite Orientation

True "by construction": look at the basic biological operations that increase the number of membranes in a system:



opposite orientation



opposite orientation

Bitonal Structure

Bitonality

Blue and white areas alternate.

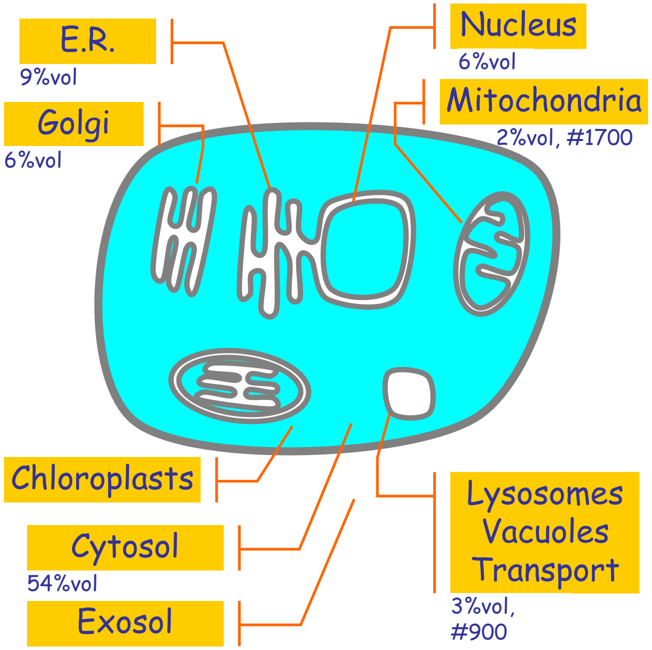
Bitonal Invariant (~ Orientation Invariant)

Bitonality and subsystem coloring is preserved by reactions. I.e., blue and white fluids *never mix and never flip color.*

Bitonal Duality

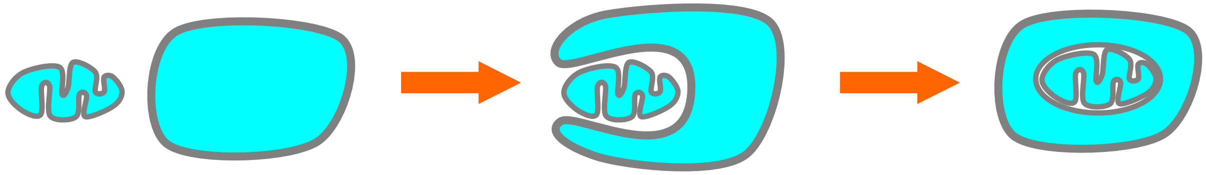
Reactions come in complementary-tone versions.

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

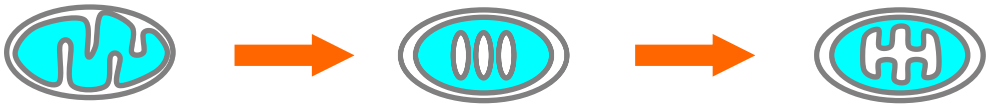


Evolutionary explanations of bitonal structure

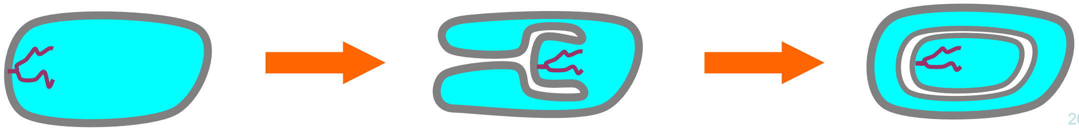
Mitochondria acquisition



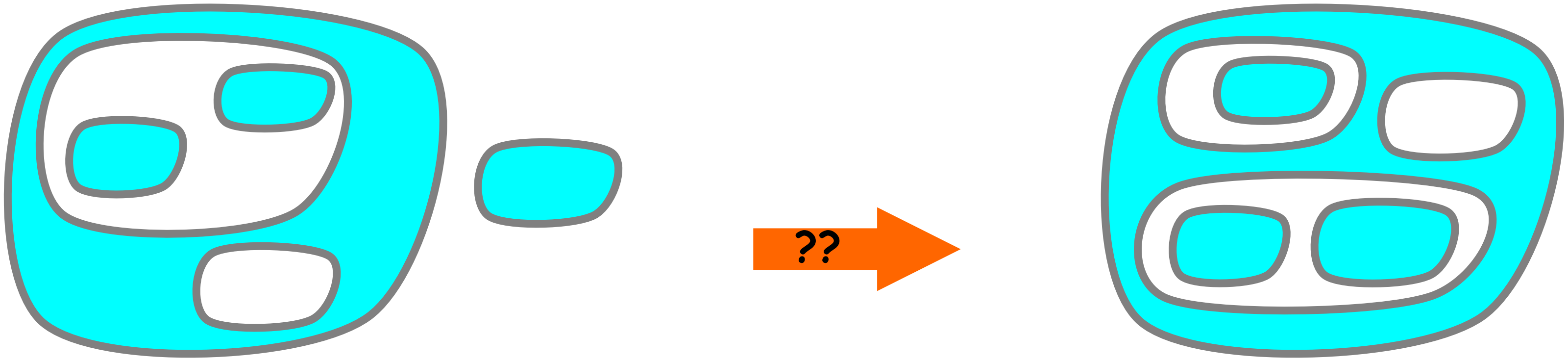
Mitochondria to Chloroplasts



Pre-Eukarya to Eukarya



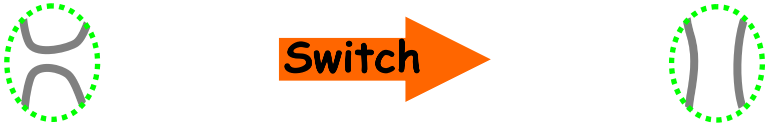
Membrane Reactions



Membrane System

What reactions "make sense" biologically?

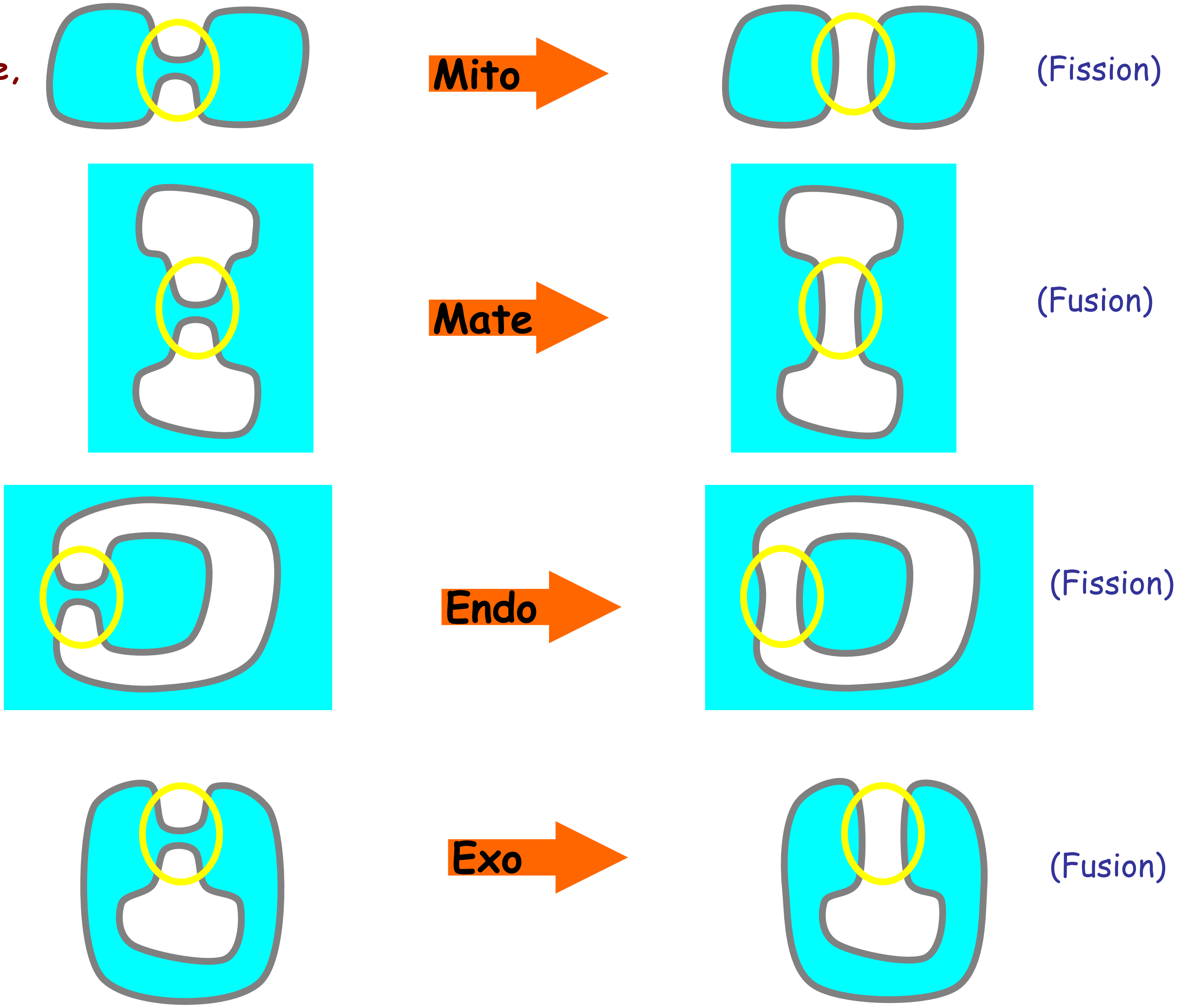
Reactions that "make sense" from a local, molecular viewpoint



(Symmetric by 90° rotation.)

Global Membrane Reactions

Reactions that "make sense" from a descriptive, global viewpoint

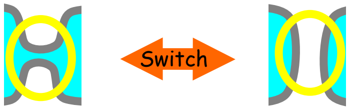


Mito (Fission)

Mate (Fusion)

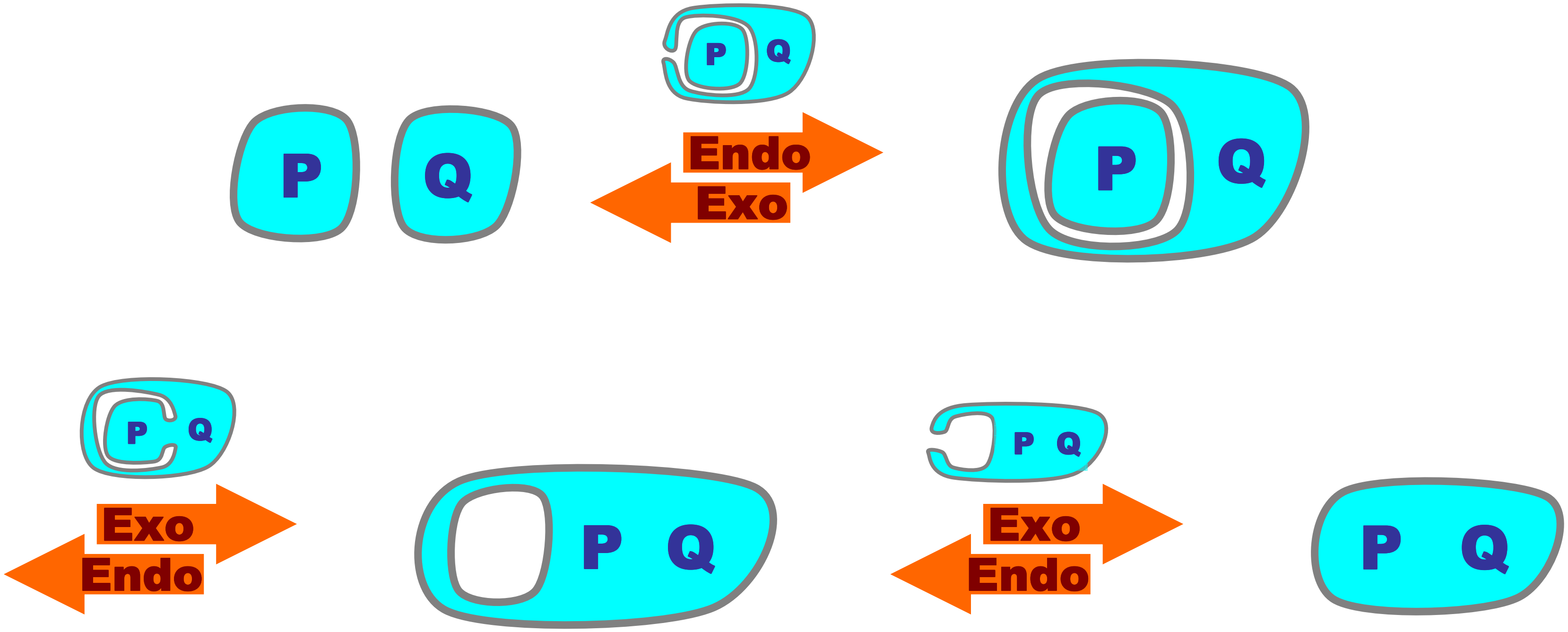
Endo (Fission)

Exo (Fusion)



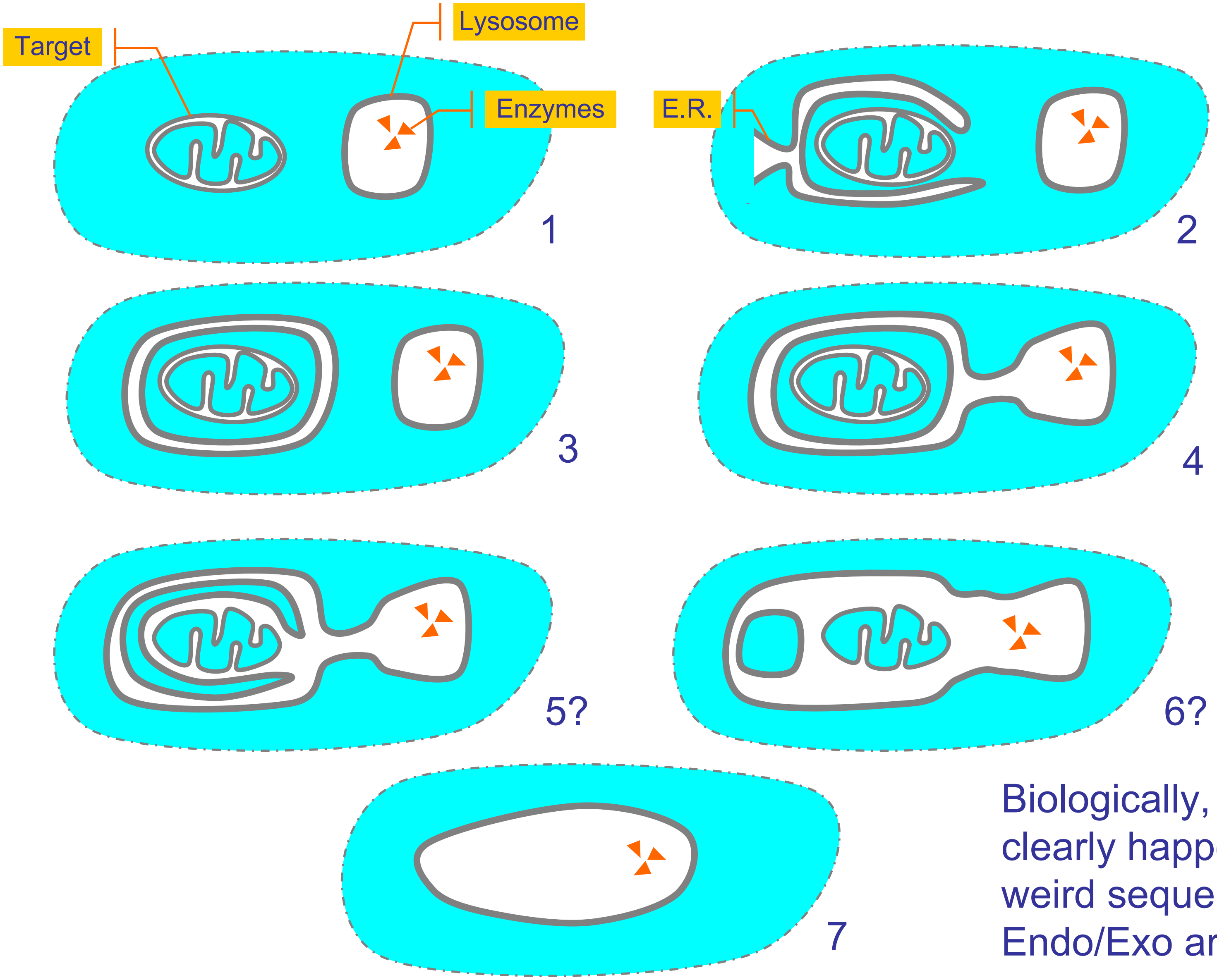
Same
Local
View!

Mito/Mate by 3 Endo/Exo



Ex: Autophagic Process

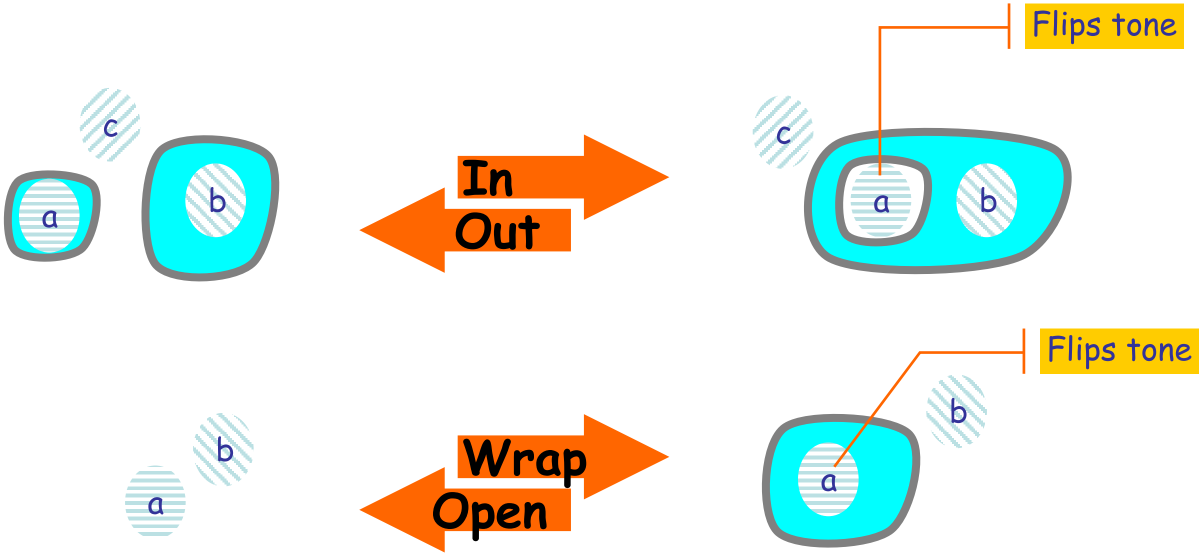
Lysosome and target don’t just merge.



Biologically, Mito/Mate clearly happens. However, weird sequences of Endo/Exo are also common.

Non-Reactions

Some global reactions are *ruled out* by bitonality,
and by locality:



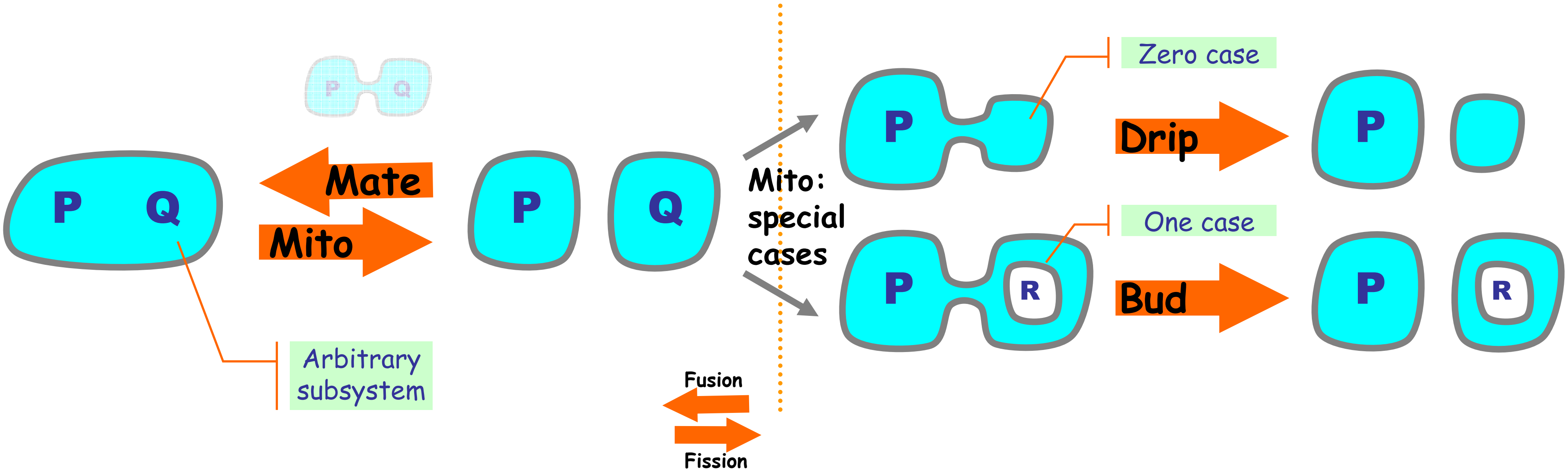
Violate bitonality.

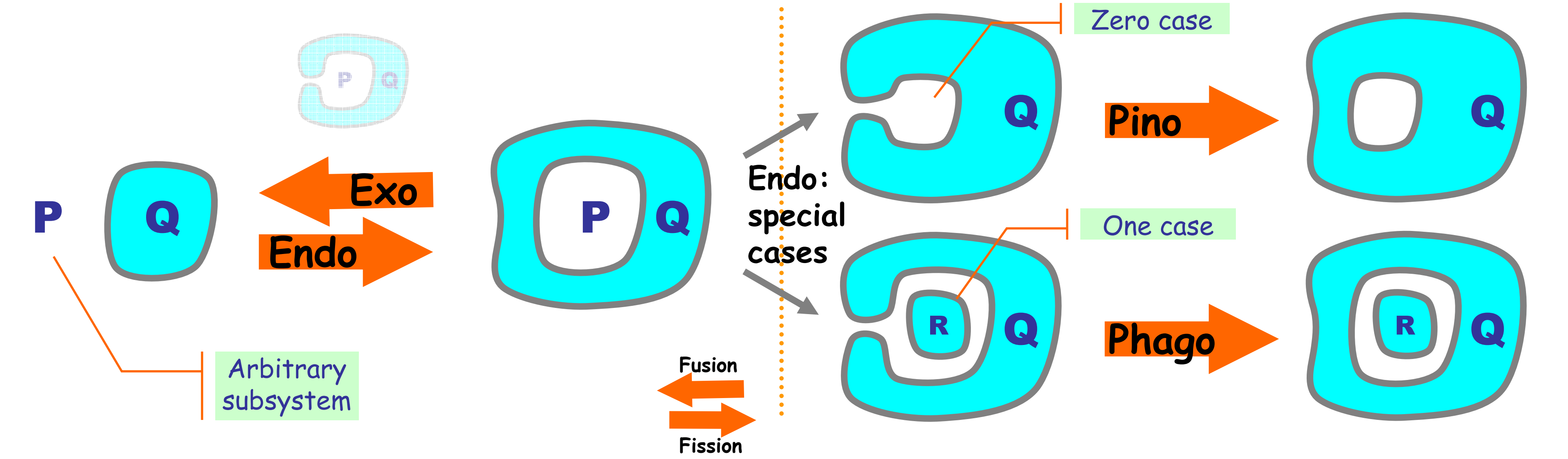
Non implementable by “local” membrane operations.

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

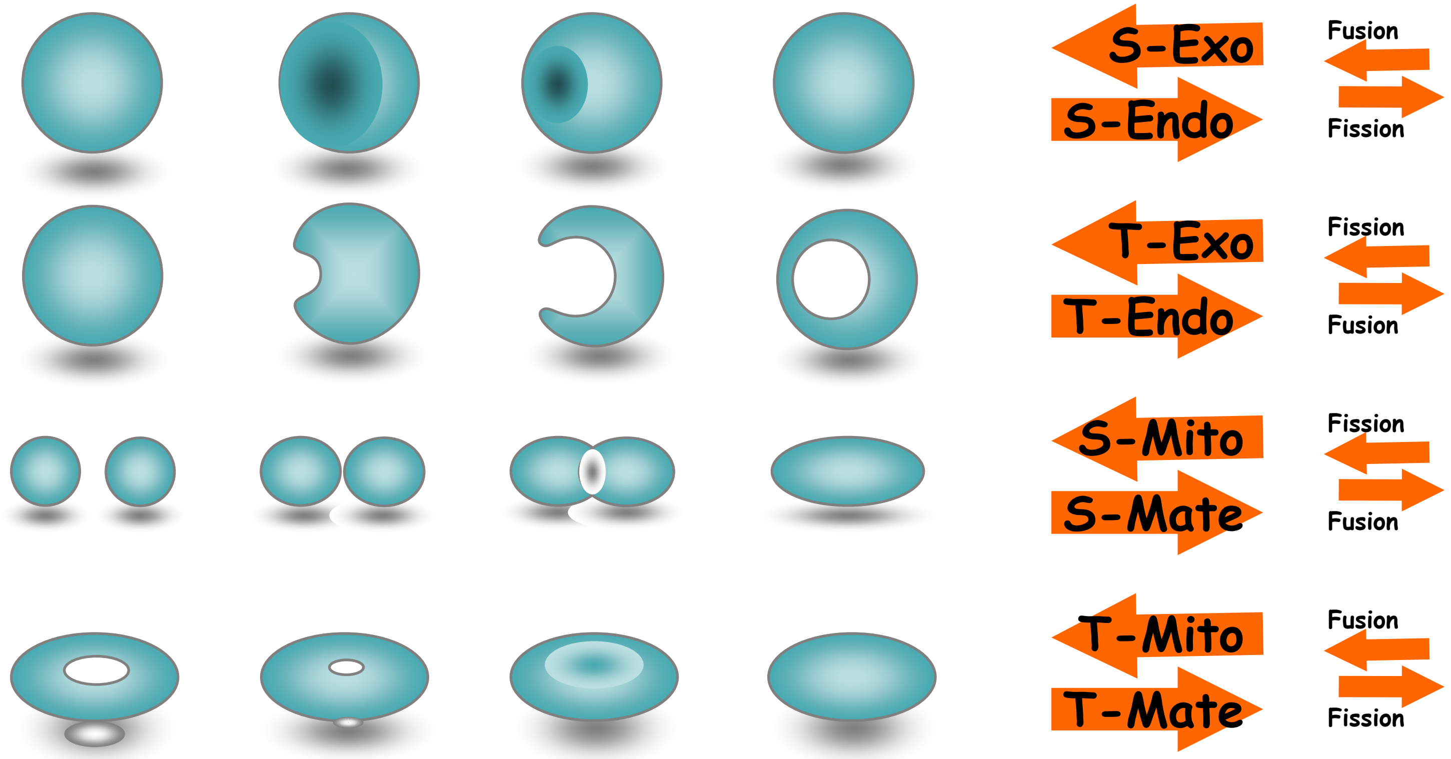
Happen to be the Ambient Calculus operations :-(

The Membrane Machine "Instruction Set"





… in 3D



S-Exo

S-Endo

Fusion

Fission

T-Exo

T-Endo

Fission

Fusion

S-Mito

S-Mate

Fission

Fusion

T-Mito

T-Mate

Fusion

Fission

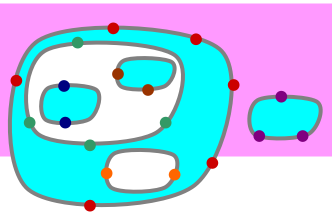
Notations for the Membrane Machine

- "Snapshot" diagrams
  - In biology literature.
- P-Systems
  - G.Paun uses ideas from the theory of grammars and formal languages to model "Membrane Computing" (book 2002). 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	$P,Q ::= \diamond \mid P \circ Q \mid !P \mid \sigma(\!|P|\!)$	nests of membranes
branes	$\sigma,\tau ::= 0 \mid \sigma\|\tau \mid !\sigma \mid a.\sigma$	combinations of actions
actions	$a ::= 1 \mid \ldots$	(fill in as needed)

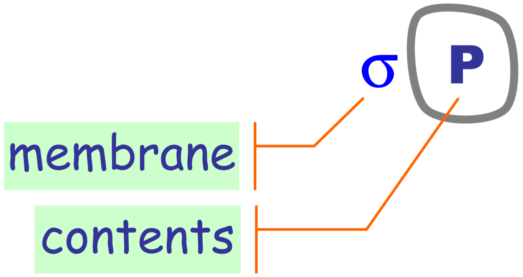
1D fluids (σ) inside a 2D fluid (P)

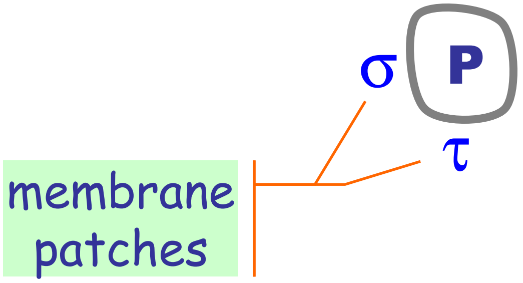
TWO commutative monoids instead of ONE of normal process calculi

$\sigma(\!|P|\!)$

$\sigma|\tau(\!|P|\!)$

$a.\sigma|\tau = (a.\sigma)|\tau$





N.B. Restriction (νn) could be added to both systems and branes. It usually would originate in branes, but would extrude to whole systems.

Congruence $\equiv$ and Reaction $\rightarrow$

	System	Brane
Fluidity	$P \circ Q \equiv Q \circ P$ $P \circ (Q \circ R) \equiv (P \circ Q) \circ R$ $P \circ \diamond \equiv P$	$\sigma \mid \tau \equiv \tau \mid \sigma$ $\sigma \mid (\tau \mid \rho) \equiv (\sigma \mid \tau) \mid \rho$ $\sigma \mid 0 \equiv \sigma$
Plentitude	$!P \equiv P \circ !P$ etc.	$!\sigma \equiv \sigma \mid !\sigma$ etc.
Units	$0\llparenthesis \diamond \rrparenthesis \equiv \diamond$ Froth/Fizz	$1.\sigma \equiv \sigma$ Inaction
Congruence	$P \equiv Q \Rightarrow P \circ R \equiv Q \circ R$ $P \equiv Q \Rightarrow !P \equiv !Q$ $P \equiv Q \wedge \sigma \equiv \tau \Rightarrow \sigma\llparenthesis P \rrparenthesis \equiv \tau\llparenthesis Q \rrparenthesis$	$\sigma \equiv \tau \Rightarrow \sigma \mid \rho \equiv \tau \mid \rho$ $\sigma \equiv \tau \Rightarrow !\sigma \equiv !\tau$ $\sigma \equiv \tau \Rightarrow a.\sigma \equiv a.\tau$

Reaction is up to congruence

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

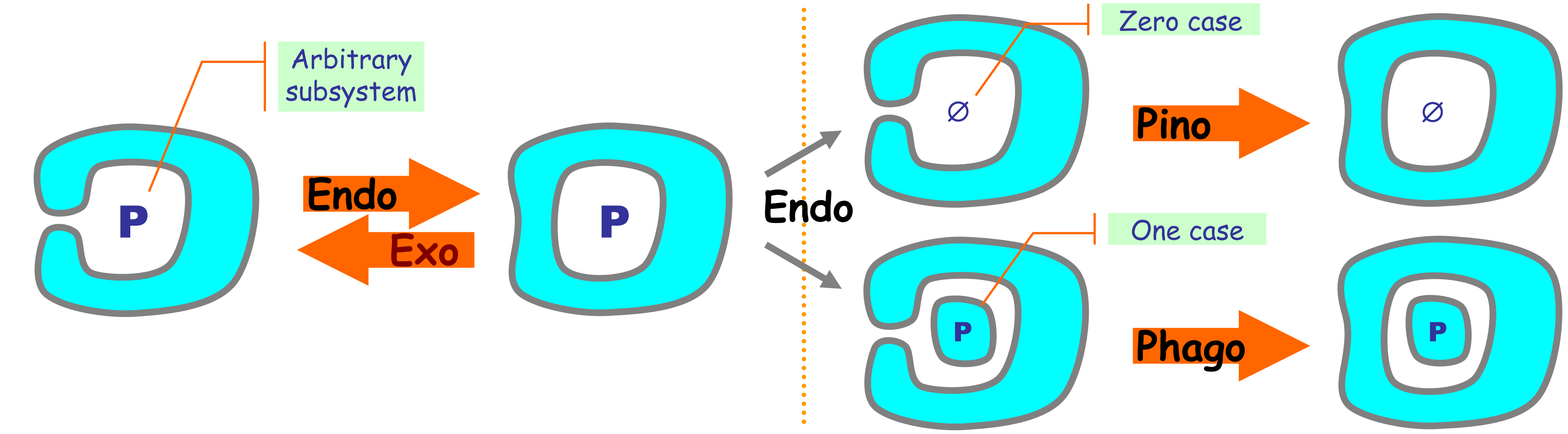
Reactions in solution

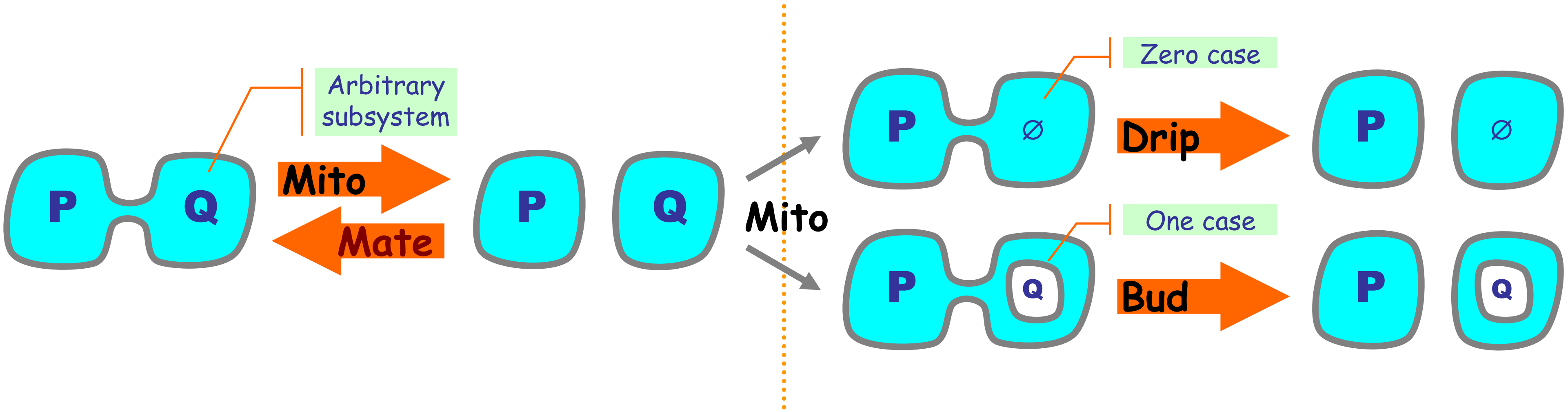
$$P \rightarrow Q \Rightarrow P \circ R \rightarrow Q \circ R$$

$$P \rightarrow Q \Rightarrow \sigma\llparenthesis P \rrparenthesis \rightarrow \sigma\llparenthesis Q \rrparenthesis$$

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

“Determinization”

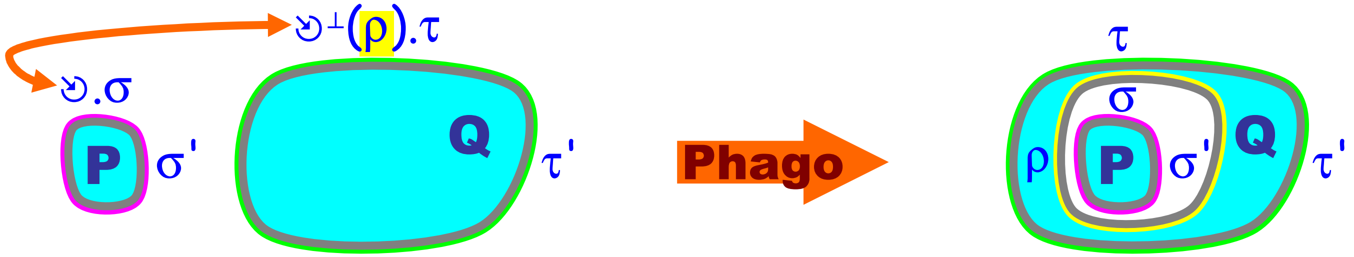


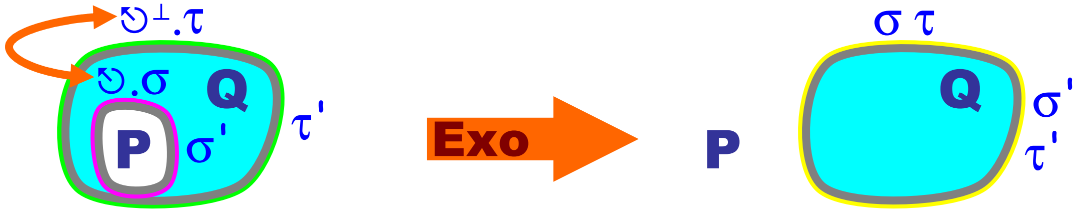


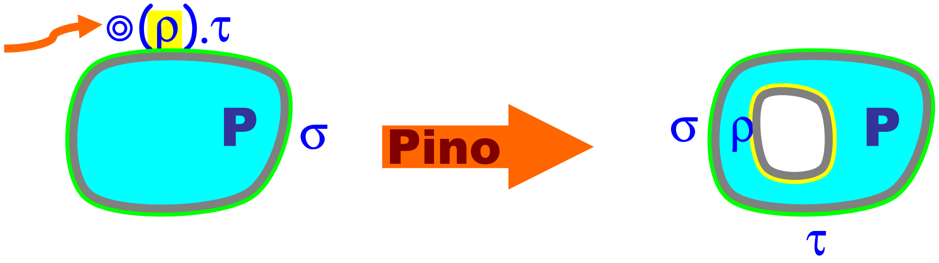
Brane Reactions

actions a ::= … | $⟳_n$ | $⟳^{\perp}_n(\rho)$ | $⟲_n$ | $⟲^{\perp}_n$ | $◎(\rho)$ phago ⟳, exo ⟲, pino ◎

coordination tags
sometimes omitted



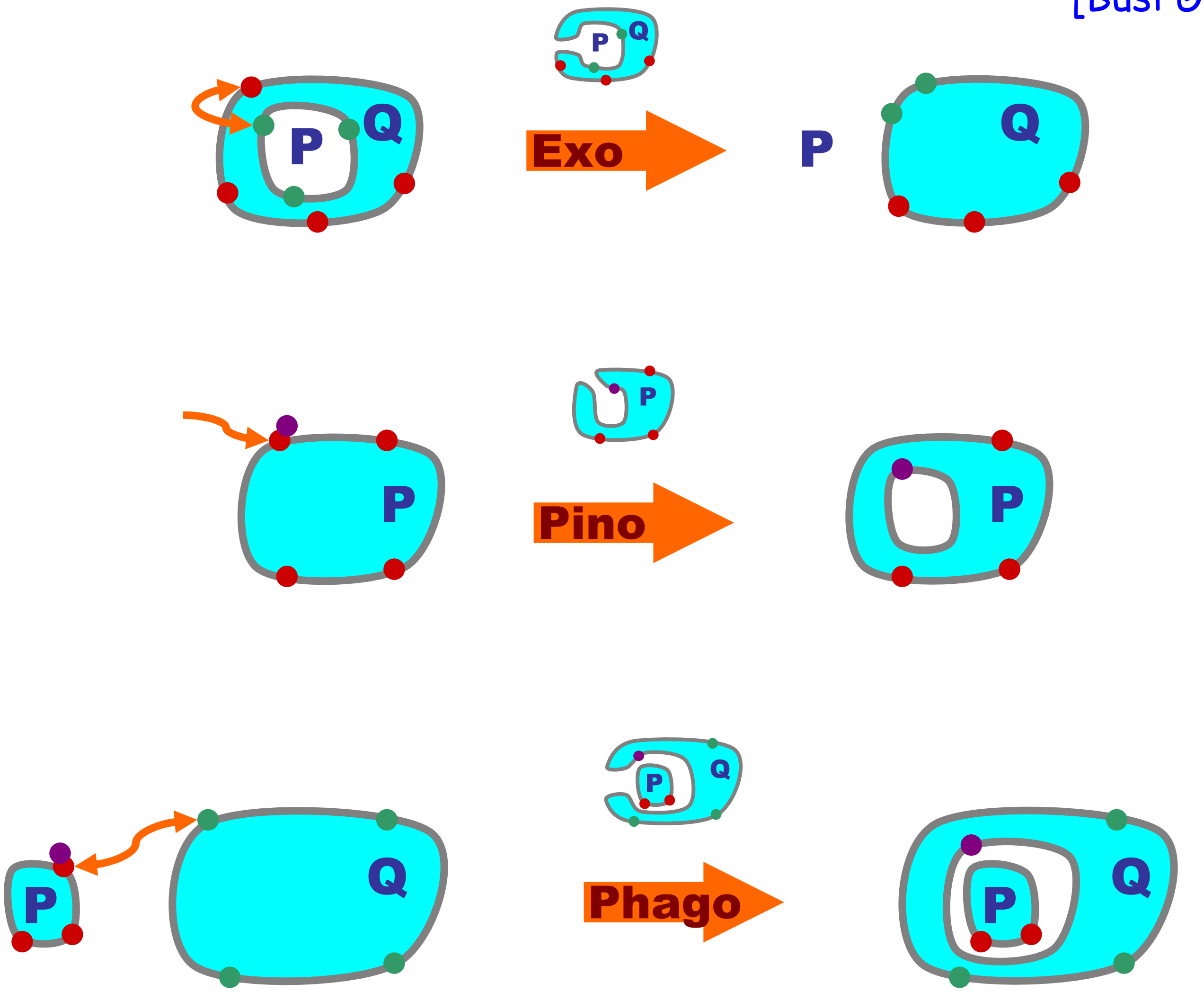




Old "spontaneous" **endo** splits into phagocytosis (**phago**, often still pronounced endo) and pinocytosis (**pino**).

Brane Reactions (Cartoons)

A Turing-Complete language
[Busi Gorrieri]



...

Phago $\circlearrowleft_n.\sigma|\sigma'\llparenthesis P\rrparenthesis \circ \circlearrowleft^{\perp}_n(\rho).\tau|\tau'\llparenthesis Q\rrparenthesis \rightarrow \tau|\tau'\llparenthesis \rho\llparenthesis \sigma|\sigma'\llparenthesis P\rrparenthesis\rrparenthesis \circ Q\rrparenthesis$

Exo $\circlearrowright^{\perp}_n.\tau|\tau'\llparenthesis \circlearrowright_n.\sigma|\sigma'\llparenthesis P\rrparenthesis \circ Q\rrparenthesis \rightarrow P \circ \sigma|\sigma'|\tau|\tau'\llparenthesis Q\rrparenthesis$

Pino $\circledcirc(\rho).\sigma|\sigma'\llparenthesis P\rrparenthesis \rightarrow \sigma|\sigma'\llparenthesis \rho\llparenthesis \diamond\rrparenthesis \circ P\rrparenthesis$

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

B&R $p_1 \circ p_1(p_2) \rightrightarrows q_1(q_2).\alpha|\sigma\llparenthesis p_2 \circ P\rrparenthesis \rightarrow q_1 \circ \alpha|\sigma\llparenthesis q_2 \circ P\rrparenthesis$

(multiset rewriting, inside and outside membranes)

...

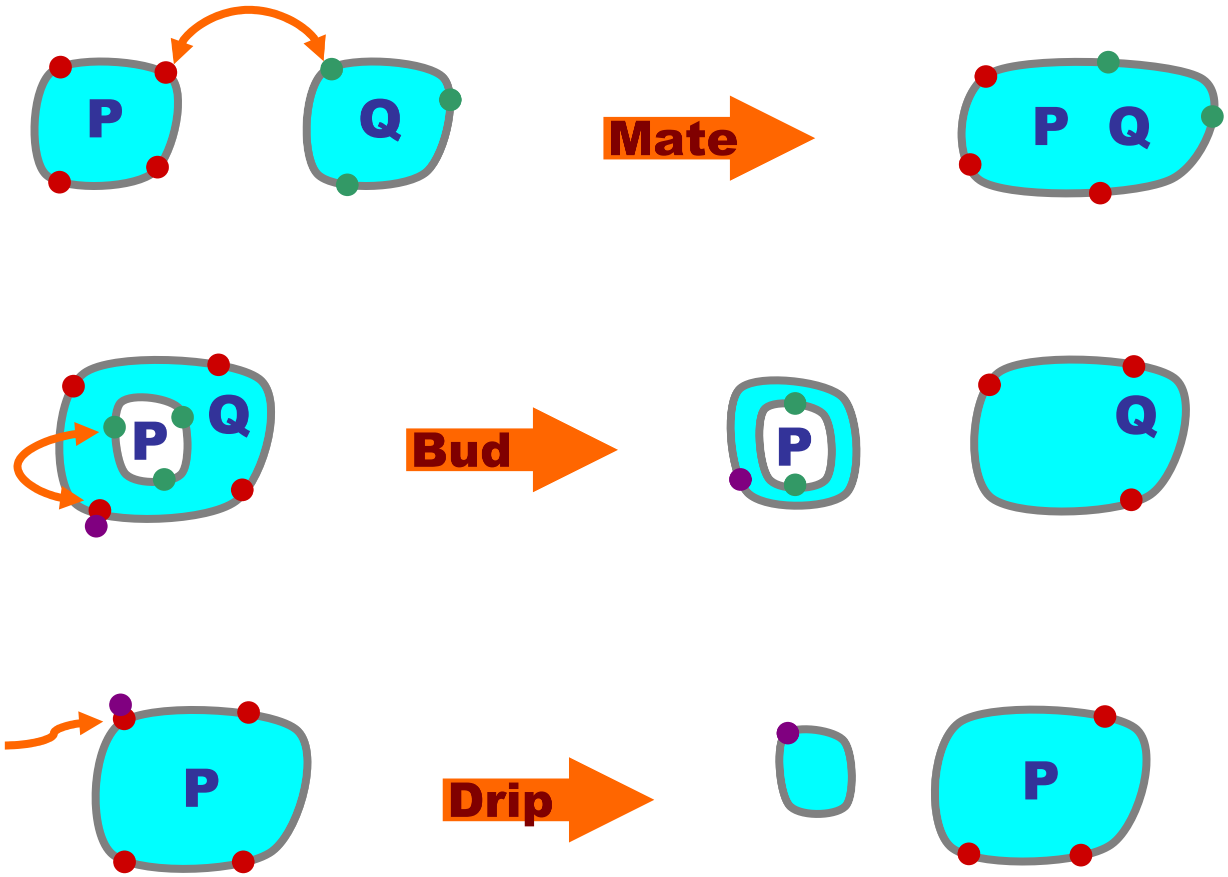
N.B.: in Phago (and Pino), one could perhaps require r to be, conservatively, a piece of t, by a non-linear rewrite:

CPhago

$$\begin{tikzcd}[column sep=small] \circlearrowleft_{n}.\sigma|\sigma'\llparenthesis P\rrparenthesis \circ \arrow[r, leftrightarrow, bend right] & \circlearrowleft^{\perp}_{n}(\rho).\tau|\tau'|\rho\llparenthesis Q\rrparenthesis \longrightarrow \tau|\tau'\llparenthesis \rho\llparenthesis \sigma|\sigma'\llparenthesis P\rrparenthesis\rrparenthesis \circ Q\rrparenthesis \end{tikzcd}$$

Derivable Reactions (Cartoons)

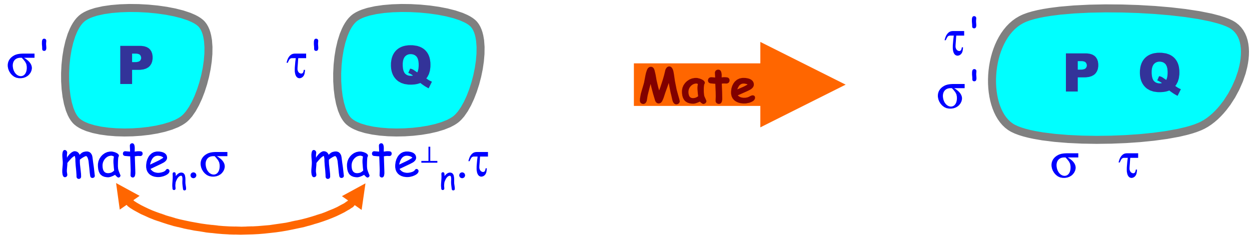
A Decidable-Termination language
[Busi Gorrieri]

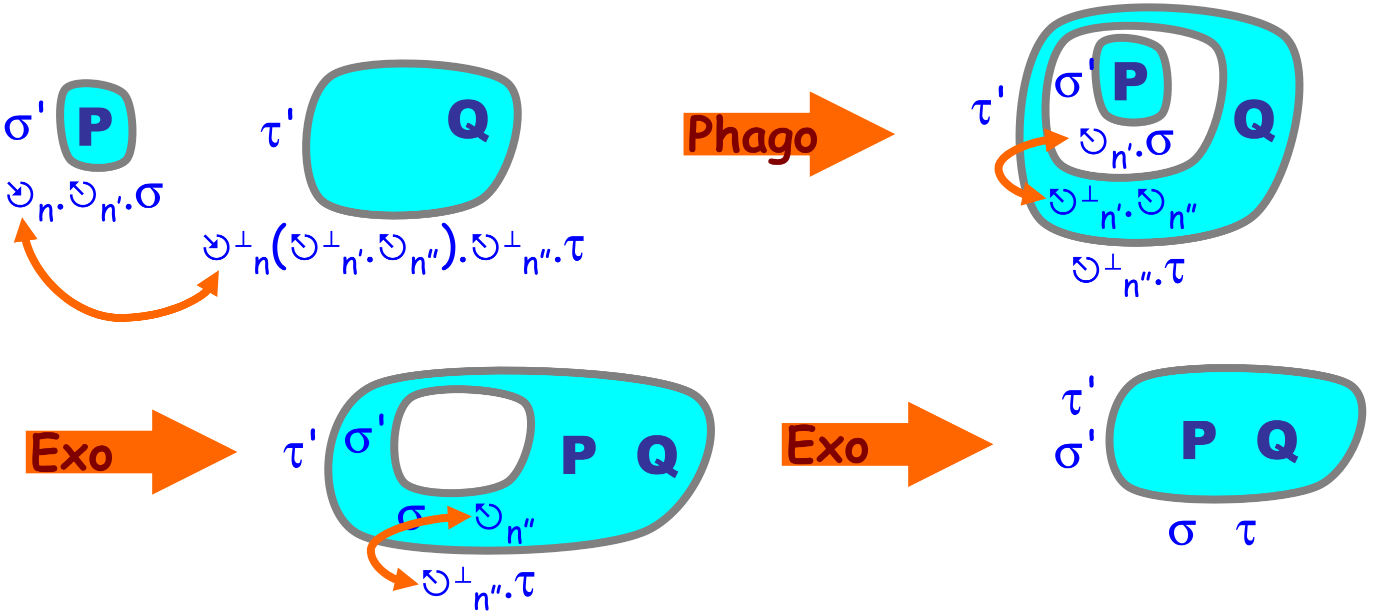


Abbreviations: Mate

Mate $\text{mate}_n.\sigma \;=\; \circlearrowright_n.\circlearrowleft_{n'}.\sigma$

$\text{mate}^\perp_n.\tau \;=\; \circlearrowright^\perp_n(\circlearrowleft^\perp_{n'}.\circlearrowleft_{n''}).\circlearrowleft^\perp_{n''}.\tau$




σ'
P
$\circlearrowright_n.\circlearrowleft_{n'}.\sigma$
τ'
Q
$\circlearrowright^\perp_n(\circlearrowleft^\perp_{n'}.\circlearrowleft_{n''}).\circlearrowleft^\perp_{n''}.\tau$
Phago
τ'
σ'
P
$\circlearrowleft_{n'}.\sigma$
Q
$\circlearrowleft^\perp_{n'}.\circlearrowleft_{n''}$
$\circlearrowleft^\perp_{n''}.\tau$
Exo
τ'
σ'
P Q
σ
$\circlearrowleft_{n''}$
$\circlearrowleft^\perp_{n''}.\tau$
Exo
τ'
σ'
P Q
σ τ

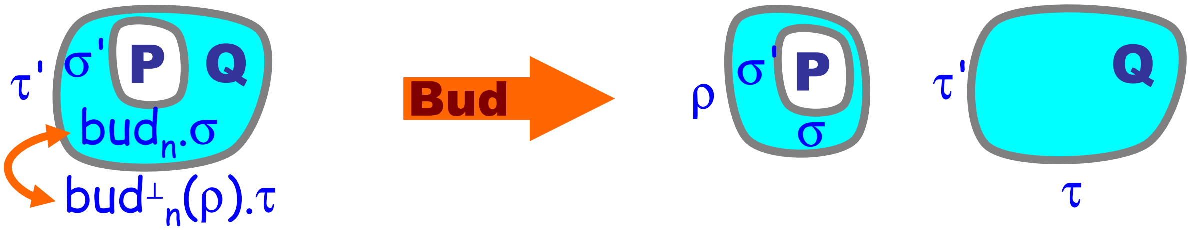
Abbreviations: Bud

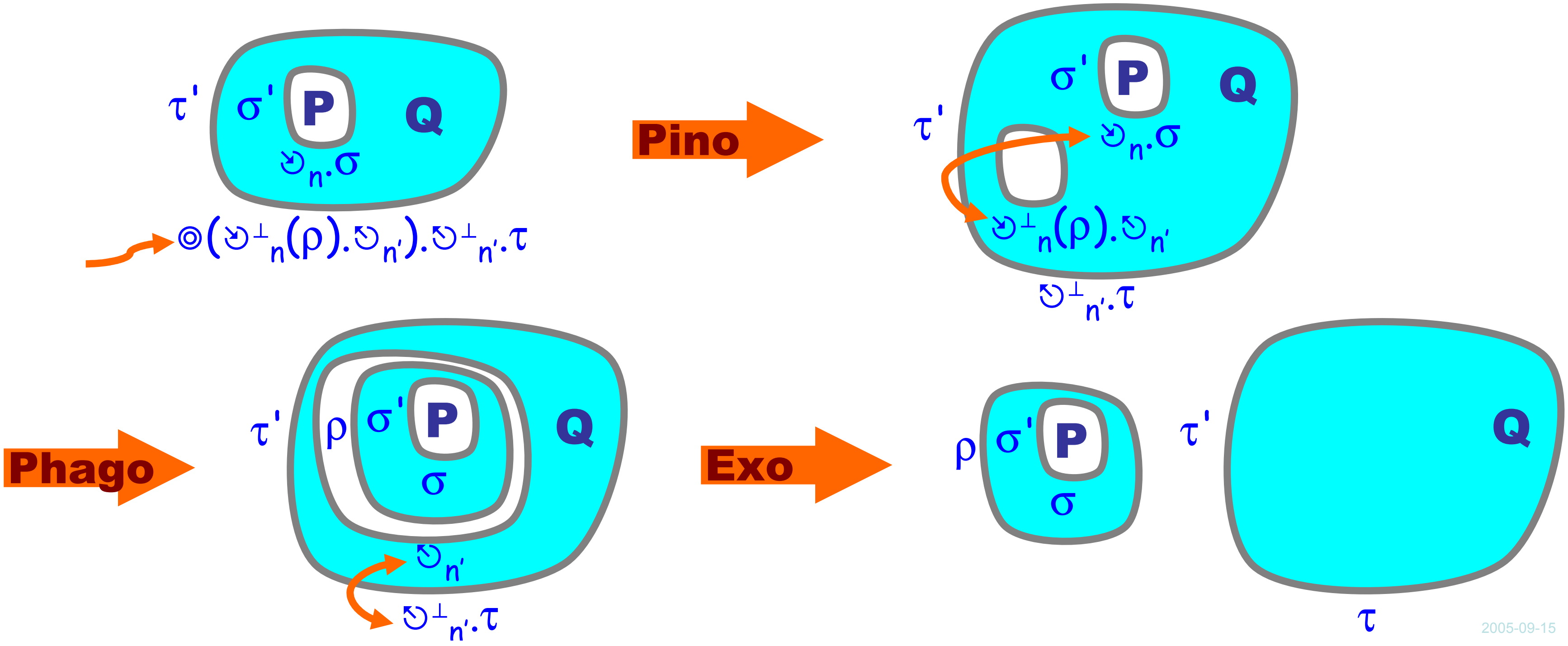
Bud

$$\text{bud}_n.\sigma = \circlearrowright_n.\sigma$$

$$\text{bud}^{\perp}_n(\rho).\tau = \circledcirc(\circlearrowright^{\perp}_n(\rho).\circlearrowleft_{n'}).\circlearrowleft^{\perp}_{n'}.\tau$$

A budding version of old "spontaneous" mito, t avoid arbitrary splits. Follows the pattern of inverse-mate.

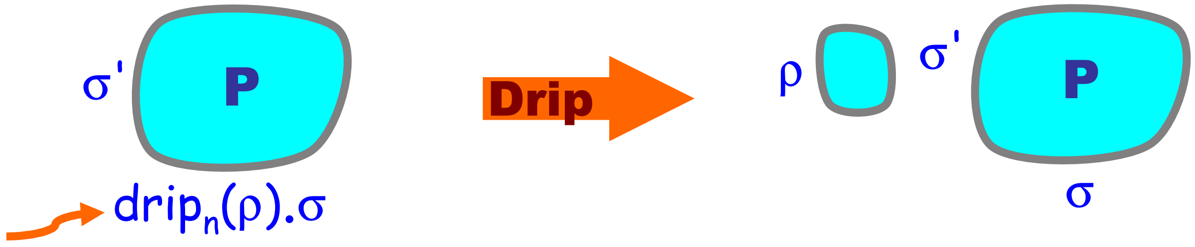


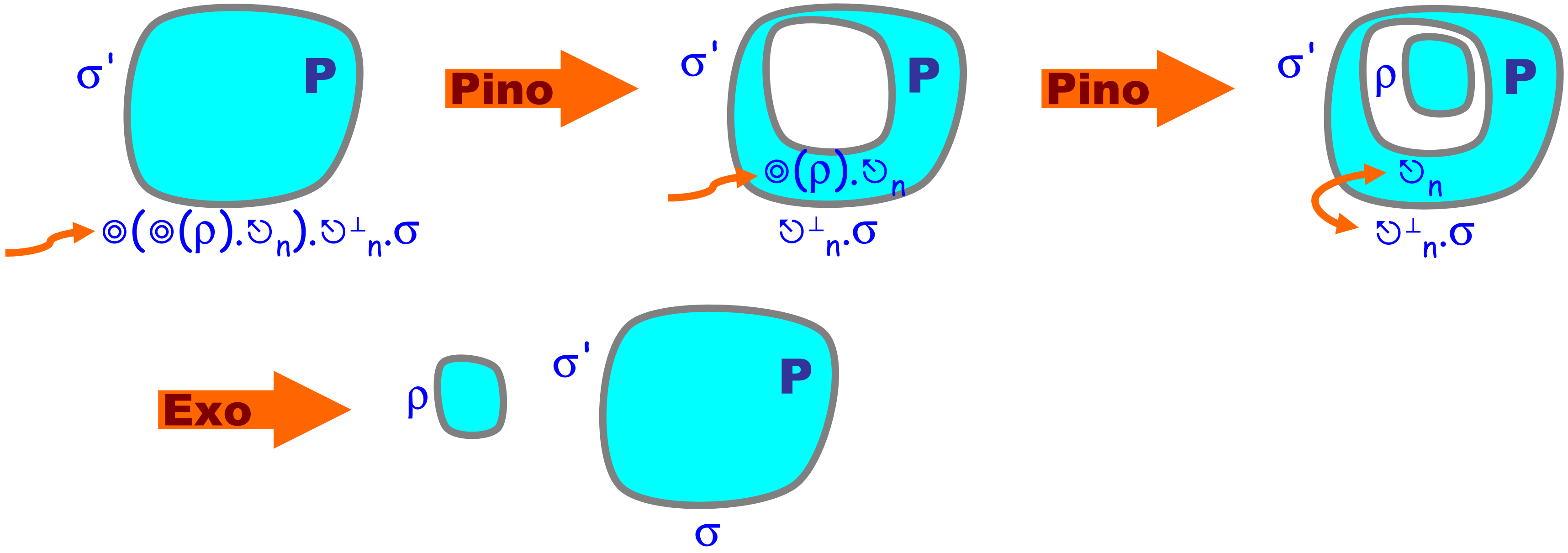

τ'
σ'
P
Q
$\circlearrowright_n.\sigma$
$\circledcirc(\circlearrowright^{\perp}_n(\rho).\circlearrowleft_{n'}).\circlearrowleft^{\perp}_{n'}.\tau$
Pino
τ'
σ'
P
Q
$\circlearrowright_n.\sigma$
$\circlearrowright^{\perp}_n(\rho).\circlearrowleft_{n'}$
$\circlearrowleft^{\perp}_{n'}.\tau$
Phago
τ'
ρ
σ'
P
σ
Q
$\circlearrowleft_{n'}$
$\circlearrowleft^{\perp}_{n'}.\tau$
Exo
ρ
σ'
P
σ
τ'
Q
τ

Abbreviations: Drip

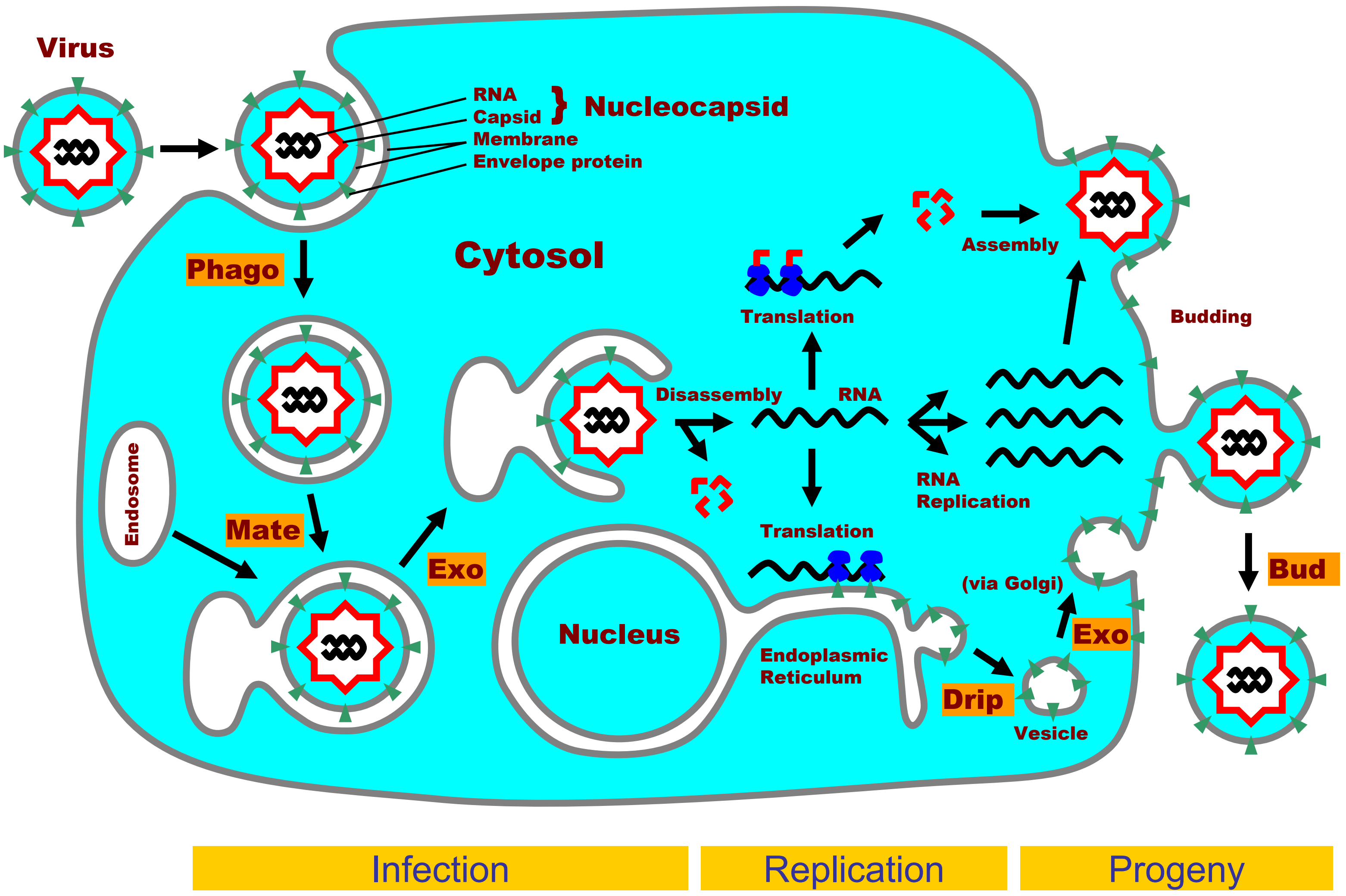
Drip $\text{drip}_n(\rho).\sigma \;=\; \circledcirc(\circledcirc(\rho).\circlearrowleft_n).\circlearrowleft^{\perp}_n.\sigma$

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



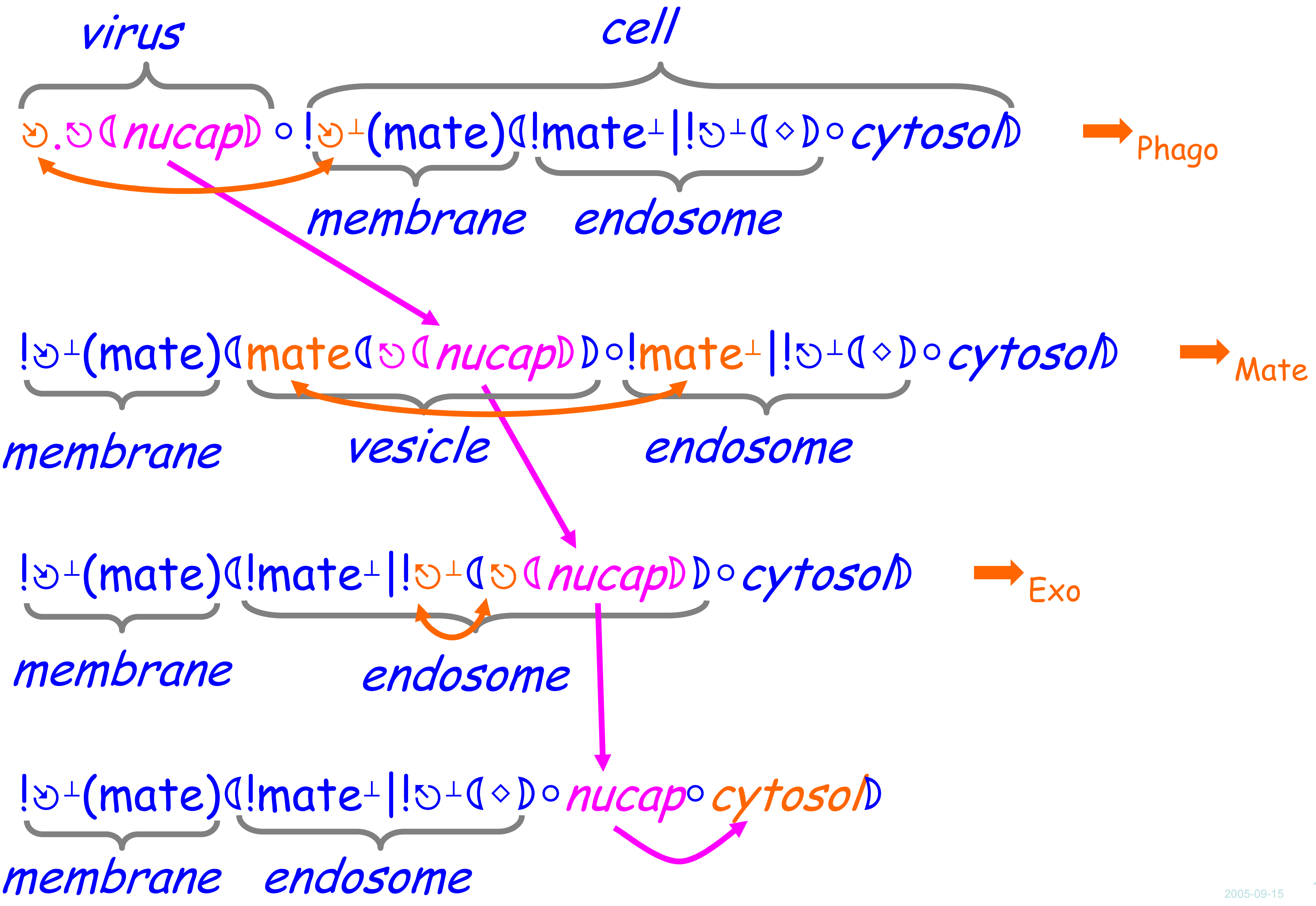

σ'
P
$\circledcirc(\circledcirc(\rho).\circlearrowleft_n).\circlearrowleft^{\perp}_n.\sigma$
Pino
σ'
P
$\circledcirc(\rho).\circlearrowleft_n$
$\circlearrowleft^{\perp}_n.\sigma$
Pino
σ'
ρ
P
$\circlearrowleft_n$
$\circlearrowleft^{\perp}_n.\sigma$
Exo
ρ
σ'
P
σ

Viral Reproduction



[MBC p.279]
annotated

Ex: Viral Infection



virus *cell*

↘.⟲⦇*nucap*⦈ ∘ !↘⊥(mate)⦇!mate⊥|!⟲⊥⦇◇⦈∘*cytosol*⦈ ➡ Phago

membrane *endosome*

!↘⊥(mate)⦇mate⦇⟲⦇*nucap*⦈⦈∘!mate⊥|!⟲⊥⦇◇⦈∘*cytosol*⦈ ➡ Mate

membrane *vesicle* *endosome*

!↘⊥(mate)⦇!mate⊥|!⟲⊥⦇⟲⦇*nucap*⦈⦈∘*cytosol*⦈ ➡ Exo

membrane *endosome*

!↘⊥(mate)⦇!mate⊥|!⟲⊥⦇◇⦈∘*nucap*∘*cytosol*⦈

membrane *endosome*

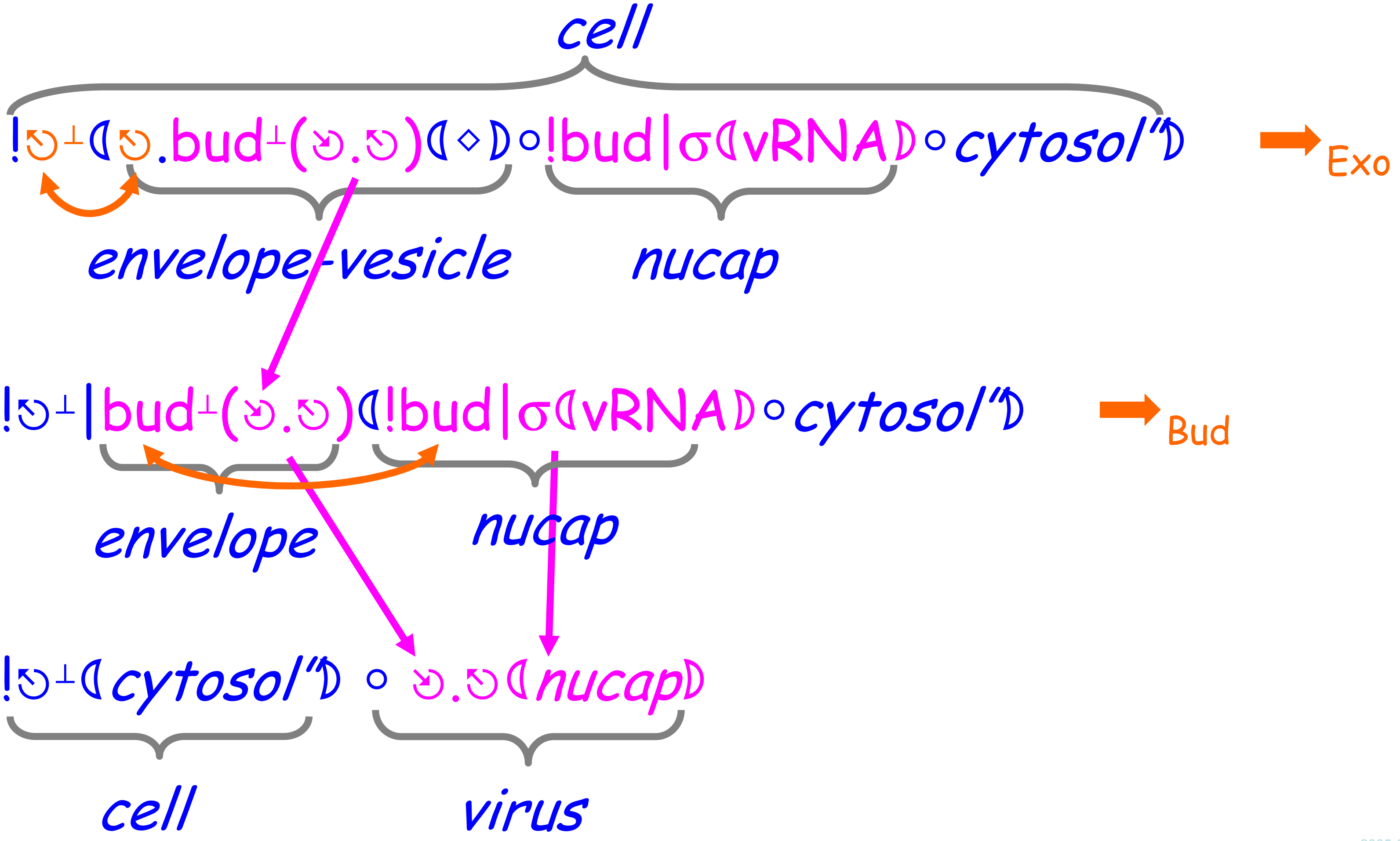
Ex: Viral Progeny

Assume:

nucap ∘ *cytosol* ➡➡ *nucap*^{n} ∘ *envelope-vesicle*^{m} ∘ *cytosol'*

by available cellular machinery

Then:



cell

!⟲⊥⦇⟲.bud⊥(⟳.⟲)⦇◇⦈∘!bud|σ⦇vRNA⦈∘*cytosol''*⦈ ➡ Exo

envelope-vesicle *nucap*

!⟲⊥|bud⊥(⟳.⟲)⦇!bud|σ⦇vRNA⦈∘*cytosol''*⦈ ➡ Bud

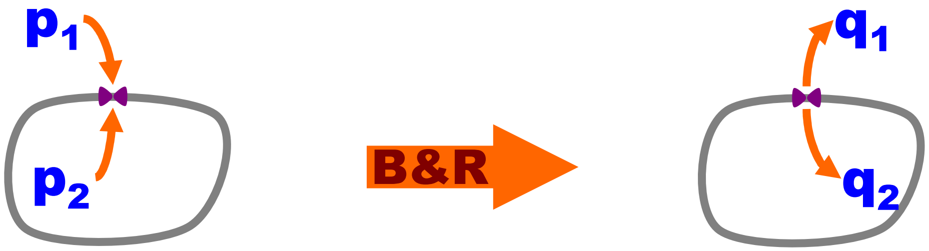
envelope *nucap*

!⟲⊥⦇*cytosol''*⦈ ∘ ⟳.⟲⦇*nucap*⦈

cell *virus*

Brane-Molecule Reactions (Cartoons)

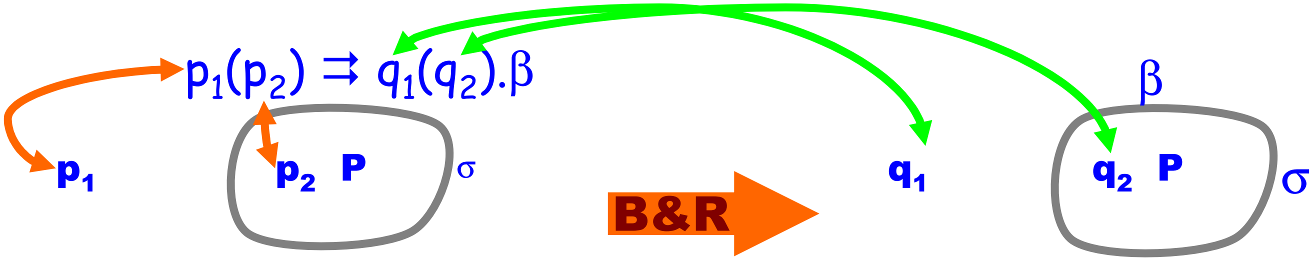
With *molecule multisets* **p**,**q**:



Molecules

We now add *molecules* to the model:

systems	$P,Q ::= \ldots \mid m$	$m \in M$ molecules
	$p,q ::= m_1 \circ \ldots \circ m_k$	molecule multisets
actions	$a ::= \ldots \mid p_1(p_2) \rightrightarrows q_1(q_2)$	bind&release



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 κ-calculus by adding complex molecules.

...

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

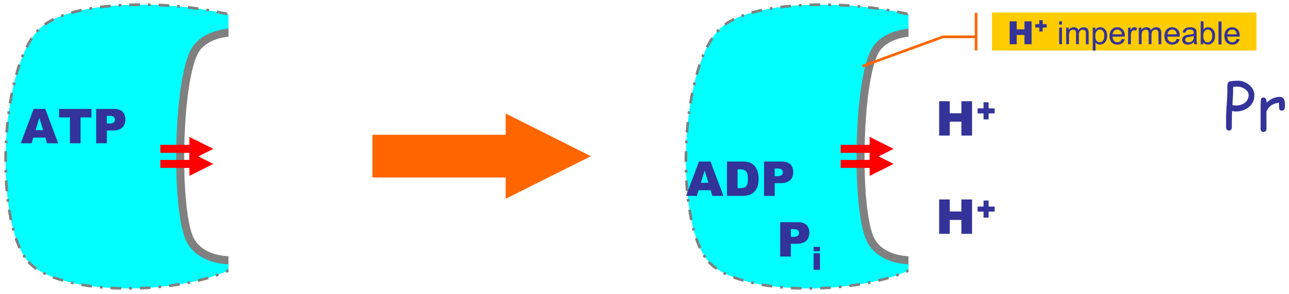
(multiset rewriting, inside and outside membranes)

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

$m(\diamond)\rightrightarrows$	bind out	$\rightrightarrows m(\diamond)$	release out
$\diamond(m)\rightrightarrows$	bind in	$\rightrightarrows\diamond(m)$	release in

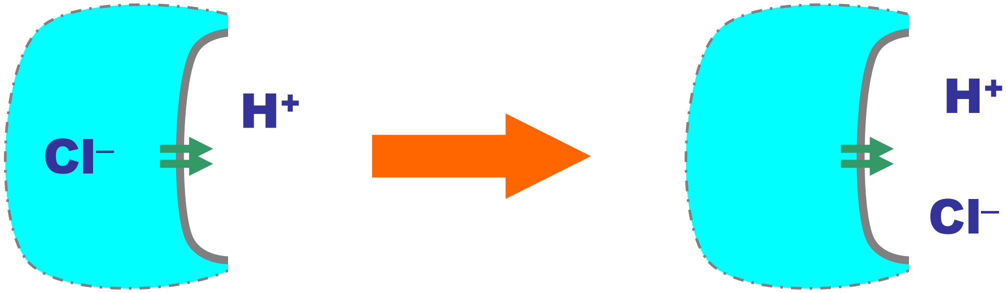
Ex: Molecular Pumps and Channels

E.g. plant vacuole (white).

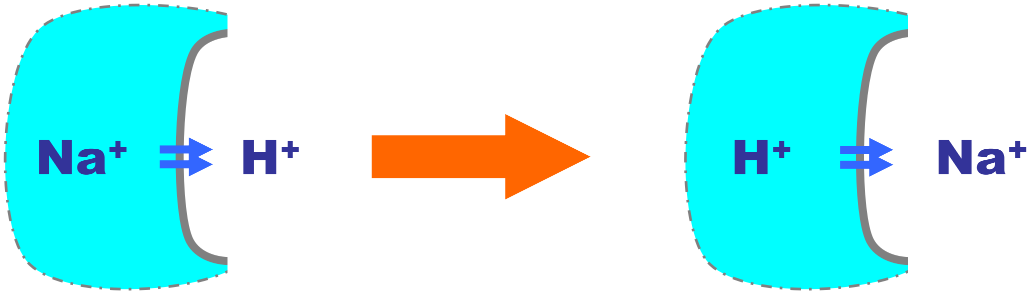


Proton Pump

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.

...

$\mathit{ProtonPump} = !\ ATP(\diamond) \rightrightarrows ADP \circ P_i(H^+ \circ H^+)$

$\mathit{IonChannel} = \ !\ Cl^-(H^+) \rightrightarrows \diamond(H^+ \circ Cl^-)$

$\mathit{ProtonAntiporter} = !\ Na^+(H^+) \rightrightarrows H^+(Na^+)$

$\mathit{PlantVacuole} =$

$\quad \mathit{ProtonPump} \mid \mathit{IonChannel} \mid \mathit{ProtonAntiporter}\ \llparenthesis \diamond \rrparenthesis$

Hence this reaction notation, ⇉, is "like" chemical reaction notation, →, 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)

$$p \longrightarrow q \quad \triangleq \quad !\, p(\diamond) \rightrightarrows q(\diamond) \llparenthesis\, \rrparenthesis$$

$$p \leftrightharpoons q \quad \triangleq \quad p \longrightarrow q \circ q \longrightarrow p$$

E.g. peptide bond between two aminoacids $R^1 R^2$:

$$R^1\text{-}COOH \circ H_2N\text{-}R^2 \longrightarrow R^1\text{-}CO\text{-}HN\text{-}R^2 \circ H_2O$$

Compartment conditions (on the membrane of a compartment)

$$p \underset{\cdot}{\rightarrow} q \quad \triangleq \quad !\, \diamond(p) \rightrightarrows \diamond(q)$$

$$p \underset{\cdot}{\rightarrow} q \,|\, \sigma \llparenthesis P \rrparenthesis$$

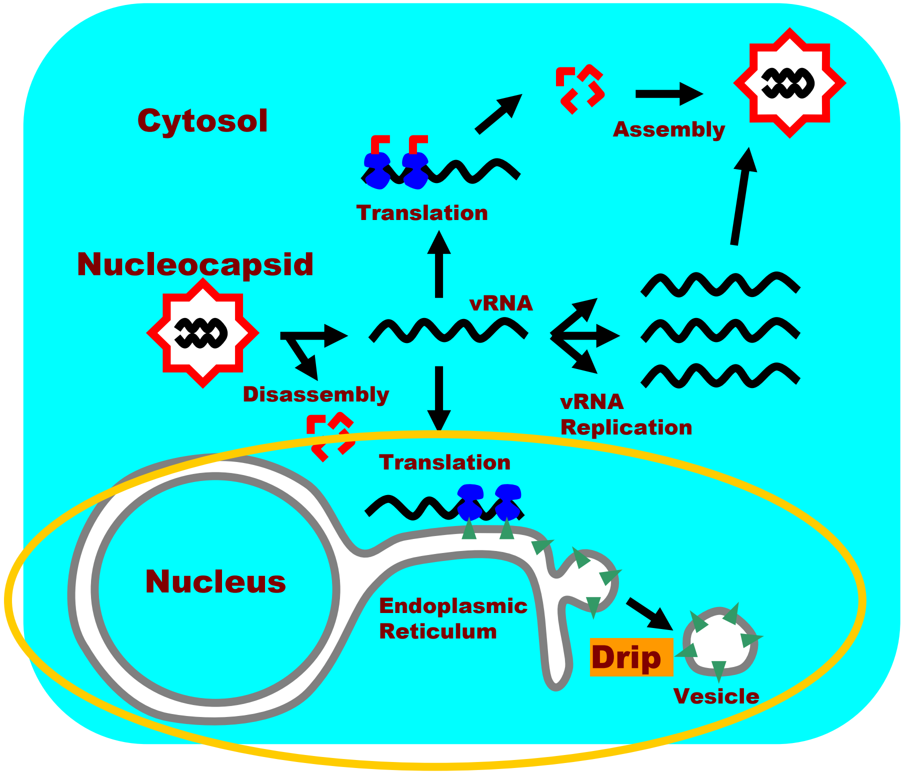
Condition affecting P

E.g. a condition-driven reaction:

$$p \underset{\cdot}{\rightarrow} q \,|\, \sigma \llparenthesis p \rrparenthesis \quad ➡ \quad p \underset{\cdot}{\rightarrow} q \,|\, \sigma \llparenthesis q \rrparenthesis$$

Ex: Virus Replication

nucap ∘ *cytosol* ➡➡ *nucap*^{n} ∘ *envelope-vesicle*^{m} ∘ *cytosol'*



ER ≜ !vRNA(◇)⇉vRNA(◇). drip(⟲.bud⊥(⟲.⟲))⦇*Nucleus*⦈

when triggered by vRNA

exo to cell membrane

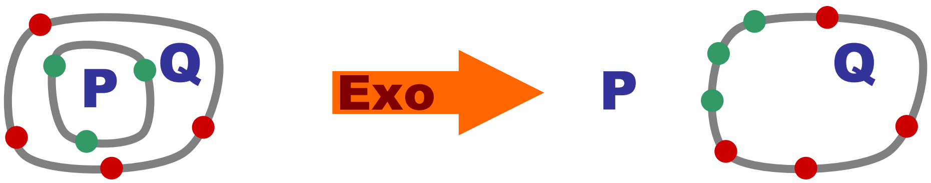
nucap budding receptor

virus membrane

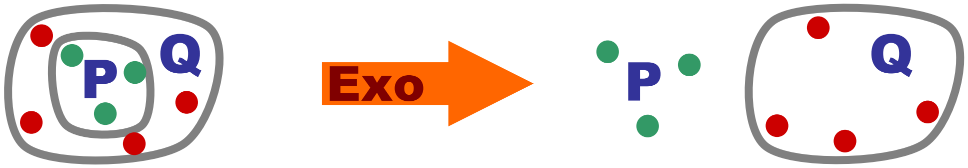
envelope-vesicle

(See paper for the other two vRNA pathways)

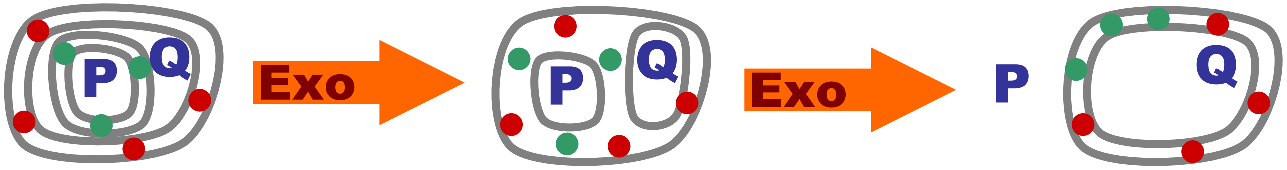
“On Brane” vs. “In Brane”



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.

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

r_1: $A+B \xrightarrow{k_1} C+C$

r_2: $A+C \xrightarrow{k_2} D$

r_3: $C \xrightarrow{k_3} E+F$

r_4: $F \xrightarrow{k_4} B$

Write the coefficients by columns

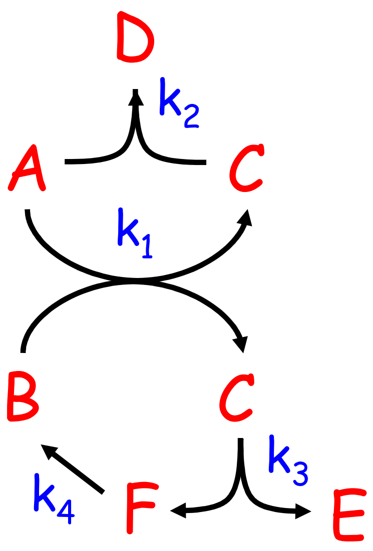
reactions

species

N	r_1	r_2	r_3	r_4
A	-1	-1		
B	-1			1
C	2	-1	-1	
D		1		
E			1	
F			1	-1

x

Stoichiometric Matrix



Concentration changes

Stoichiometric matrix

Rate laws

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

$d[A]/dt = -v_1 - v_2$

$d[B]/dt = -v_1 + v_4$

$d[C]/dt = 2\cdot v_1 - v_2 - v_3$

$d[D]/dt = v_2$

$d[E]/dt = v_3$

$d[F]/dt = v_3 - v_4$

Read the concentration changes from the rows

E.g. $d[A]/dt =$
$-k_1\cdot[A]\cdot[B] - k_2\cdot[A]\cdot[C]$

Read the rate laws from the columns

$v_i(\mathbf{x}, e_i, k_i)$

	v
v_1	$k_1\cdot[A]\cdot[B]$
v_2	$k_2\cdot[A]\cdot[C]$
v_3	$k_3\cdot[C]$
v_4	$k_4\cdot[F]$

x: chemical species

[-]: concentrations

v: rate laws

k: kinetic parameters

N: stoichiometric matrix

e: catalysts (if any)

From Chemical Reactions to Processes

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: $F \rightarrow_{k_4} B$

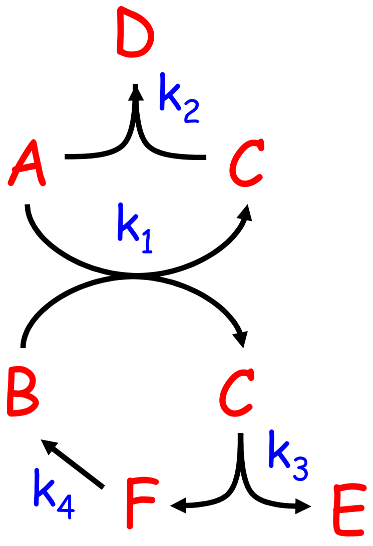
Write the coefficients by columns

interactions

processes

N	r_1	r_2	r_3	r_4
A	-1	-1		
B	-1			1
C	2	-1	-1	
D		1		
E			1	
F			1	-1

Stoichiometric Matrix



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

$A = ?v_1k_1.(C|C) + ?v_2k_2.D + ?a$

$B = !v_1k_1 + ?b$

$C = !v_2k_2 + \tau k_3(E|F) + ?c$

$D = 0 + ?d$

$E = 0 + ?e$

$F = \tau k_3.B + ?f$

Add a barb for counting and plotting

Read the process interactions from the rows

(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 stochiometric matrix (and hence without ODE's).

Complexes: The ODE Way

n
domains

A, B, C

2n
domain
reactions

$A \rightleftharpoons A_p$

$B \rightleftharpoons B_p$

$C \rightleftharpoons C_p$

1
complex

ABC

2^n
species

ABC
A_pBC
AB_pC
ABC_p
A_pB_pC
A_pBC_p
AB_pC_p
$A_pB_pC_p$

$2n(2^{n-1})$
reactions
(twice number of edges in n-dim hypercube)

$ABC \rightleftharpoons A_pBC$
$ABC \rightleftharpoons AB_pC$
$ABC \rightleftharpoons ABC_p$
$A_pBC \rightleftharpoons A_pB_pC$
$A_pBC \rightleftharpoons A_pBC_p$
$AB_pC \rightleftharpoons A_pB_pC$
$AB_pC \rightleftharpoons AB_pC_p$
$ABC_p \rightleftharpoons A_pBC_p$
$ABC_p \rightleftharpoons AB_pC_p$
$A_pB_pC \rightleftharpoons A_pB_pC_p$
$A_pBC_p \rightleftharpoons A_pB_pC_p$
$AB_pC_p \rightleftharpoons A_pB_pC_p$

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

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																								
ApBC																								
ABpC																								
ABCp																								
ApBpC																								
ApBCp																								
ABpCp																								
ApBpCp																								

$2^n \times 2n(2^{n-1})$

Complexes: The Reactive System Way

2n

domain reactions

$A \leftrightharpoons A_p$

$B \leftrightharpoons B_p$

$C \leftrightharpoons C_p$

2n

processes

$A = ?kn;A_p \quad A_p = ?ph;A$

$B = ?kn;B_p \quad B_p = ?ph;B$

$C = ?kn;C_p \quad C_p = ?ph;C$

n

$A \mid B \mid C$

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.)

When the local domain reactions are not independent, we can use lateral communication so that each component is aware of the relevant others.

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.

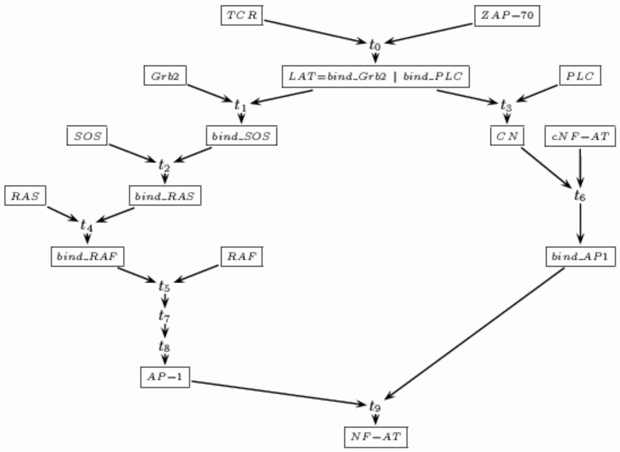


Fig. 2. A computation of *Sys*. For readability, the processes, enclosed in boxes, have no address. Causality (both on transitions and processes) is represented by the (Hasse diagram resulting from the) arrows; their absence makes it explicit concurrent activities.

- 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].

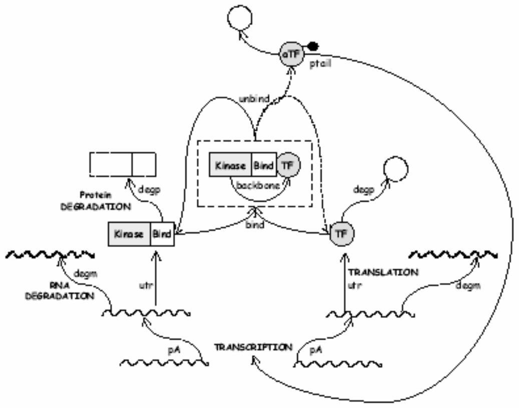


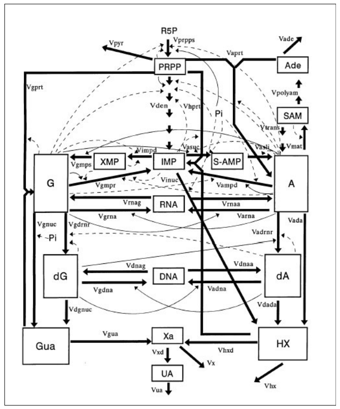
Fig. 1. Graphical presentation of Transcriptional Regulation by Positive Feedback [25].

- 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.



```
Eventually(Always (PRPP = 1.7 * PRPP1)
           implies
           steady_state()
           and Eventually(Always(IMP < 2 * IMP1))
           and Eventually(Always(hx_pool < 10*hx_pool1)))
```

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

What Reactive Systems Do For Us

We can write things down precisely

- We can modularly describe high structural and combinatorial complexity ("do programming").

We can calculate and analyze

- Directly support simulation.
- Support analysis (e.g. control flow, causality, nondeterminism).
- Support state exploration (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).

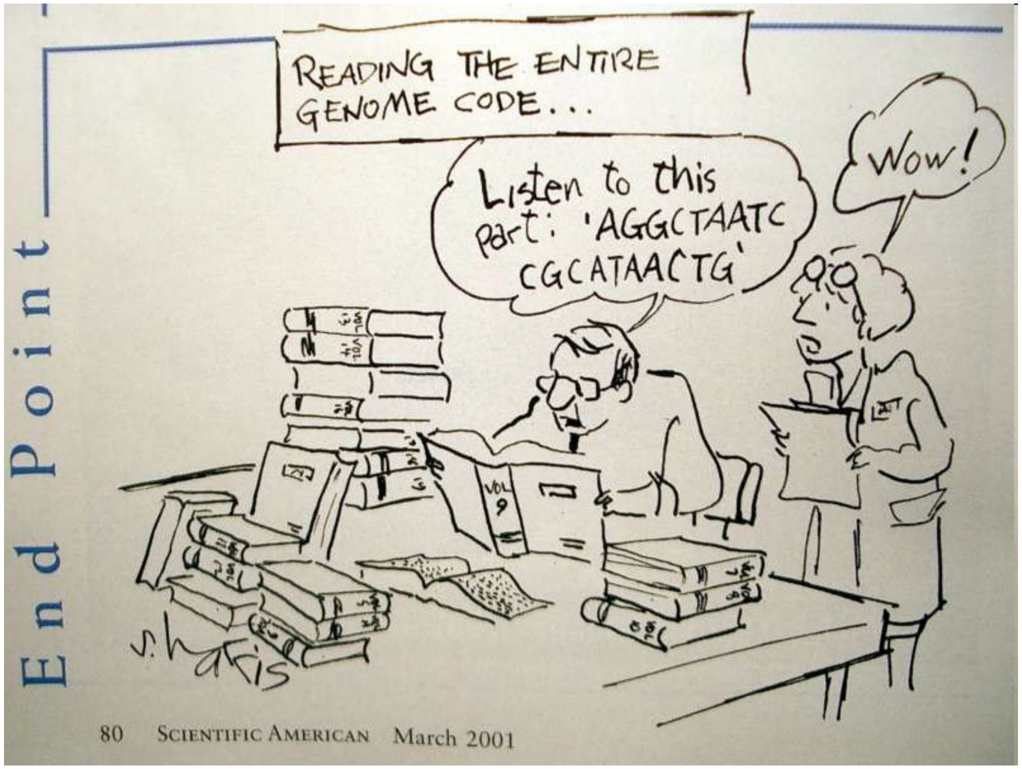
Disclaimers

- 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:

⇒ *Proc. Computational Methods in Systems Biology* [2003-2005]

Conclusions


End Point
READING THE ENTIRE GENOME CODE...
Listen to this part: 'AGGCTAATC CGCATAACTG'
Wow!
VOL 9
S. Harris
80 SCIENTIFIC AMERICAN March 2001

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

[MCB] Molecular Cell Biology, Freeman.

[MBC] Molecular Biology of the Cell, Garland.

[Ptashne] A Genetic Switch.

[Davidson] Genomic Regulatory Systems.

[Milner] Communicating and Mobile Systems: the Pi-Calculus.

[Regev] Computational Systems Biology: A Calculus for Biomolecular Knowledge (Ph.D. Thesis).

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