Biological Systems as Reactive Systems

Luca Cardelli

Microsoft Research
Cambridge UK

2006-03-20 Birmingham

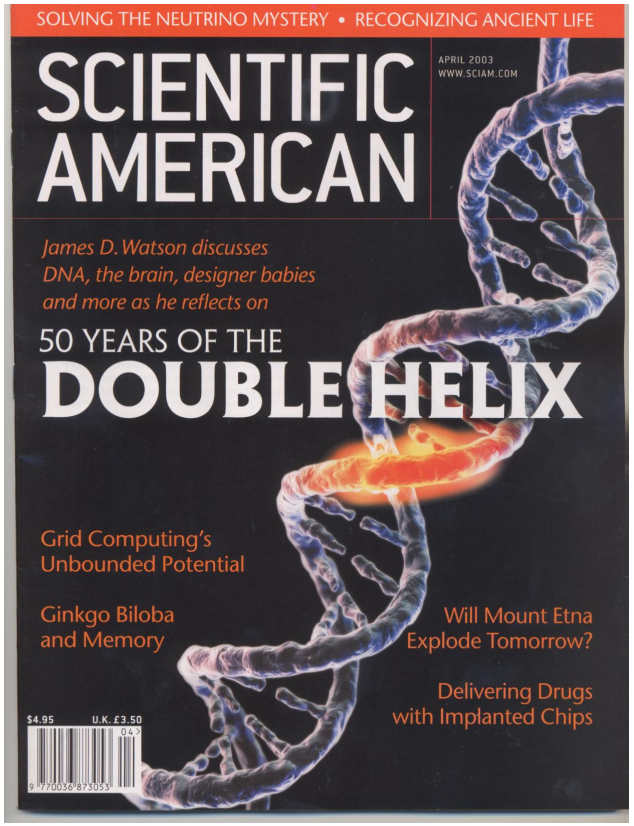
www.luca.demon.co.uk

50 Years of Molecular Cell Biology

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

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

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



Towards Systems Biology

- Biologists now understand many of the cellular components
  - A whole team of biologists will typically study a single protein for years
  - 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?

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.

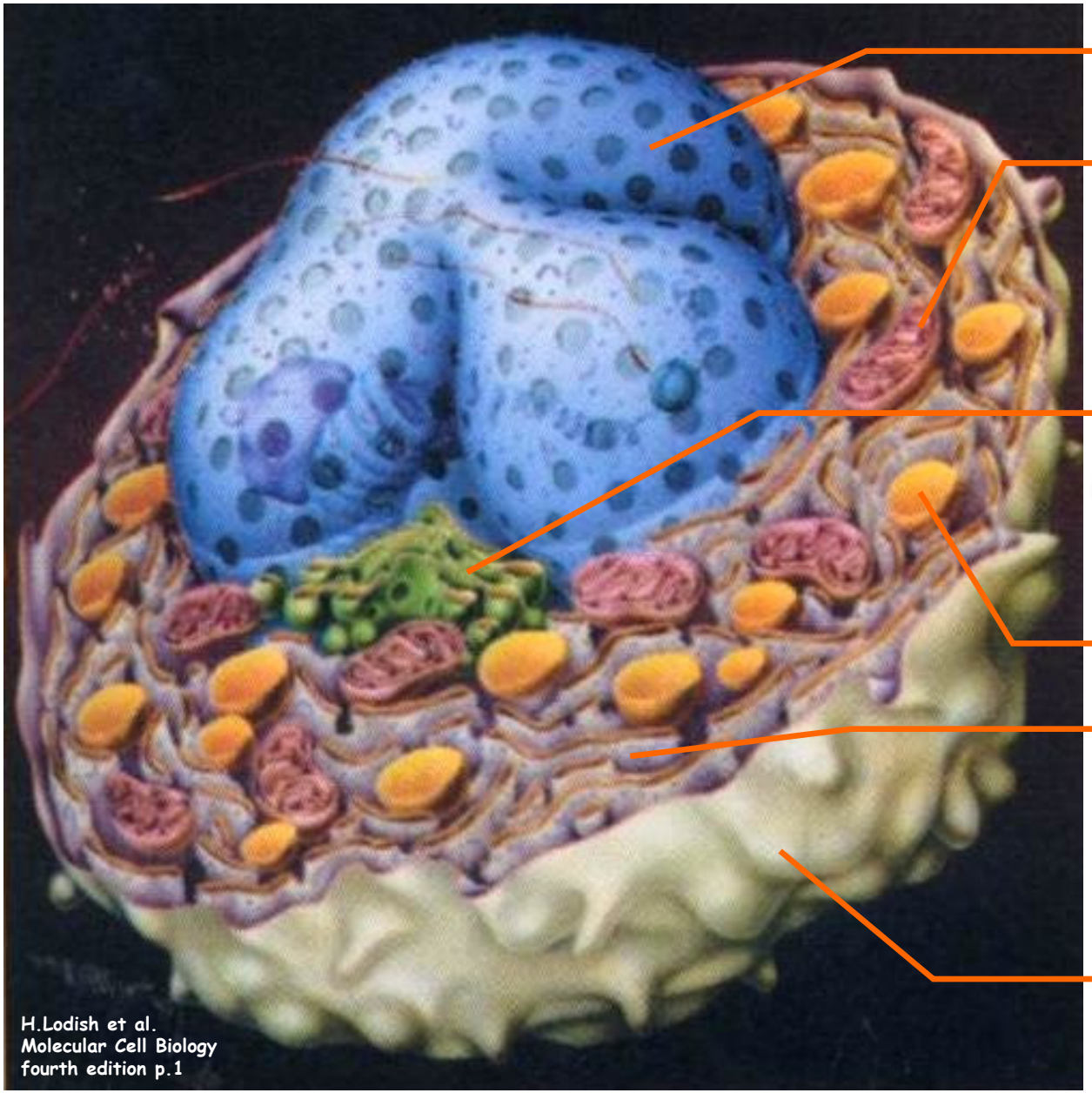
- In computing…
  - There are well-established scalable representations of dynamic reactive processes.
  - They look more or less like little, mathematically based, programming languages.

Structural Architecture

Eukaryotic Cell

(10~100 trillion in human body)

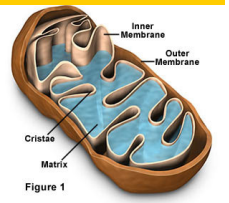
Membranes everywhere



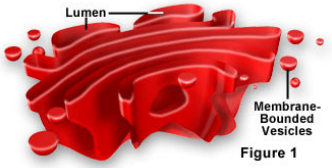
H.Lodish et al.
Molecular Cell Biology
fourth edition p.1

Nuclear membrane

Mitochondria

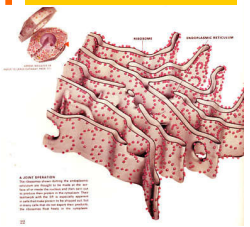


Golgi



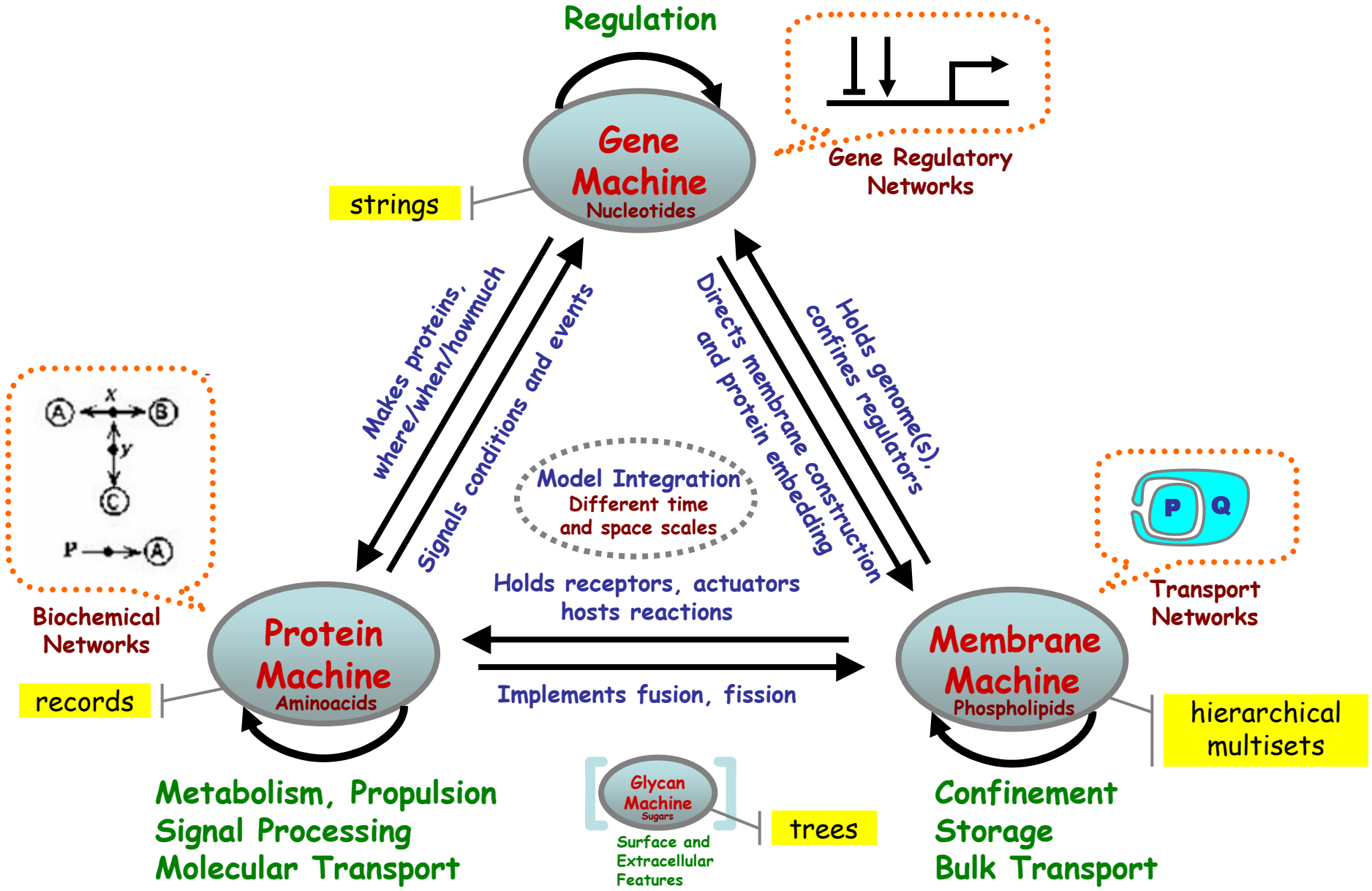
Vesicles

E.R.



Plasma membrane (<10% of all membranes)

Abstract Machines of Systems Biology



Reactive Systems

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

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

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

Methods

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

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

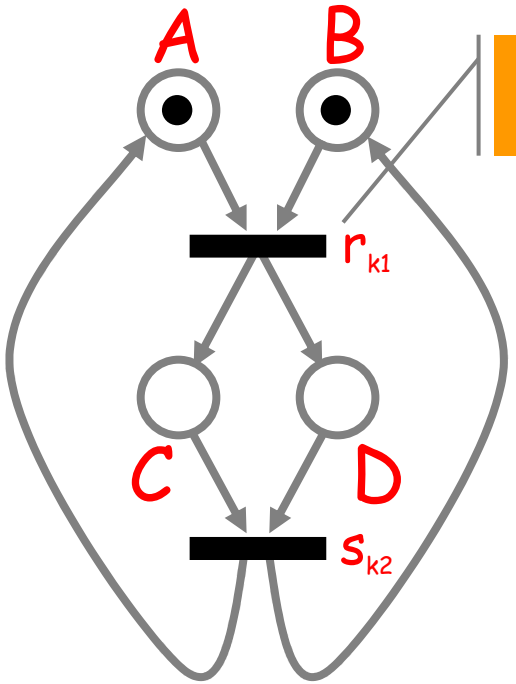
Chemistry vs. π-calculus

A process calculus (chemistry)

$r: A + B \rightarrow_{k1} C + D$

$s: C + D \rightarrow_{k2} A + B$

Does A become C or D?

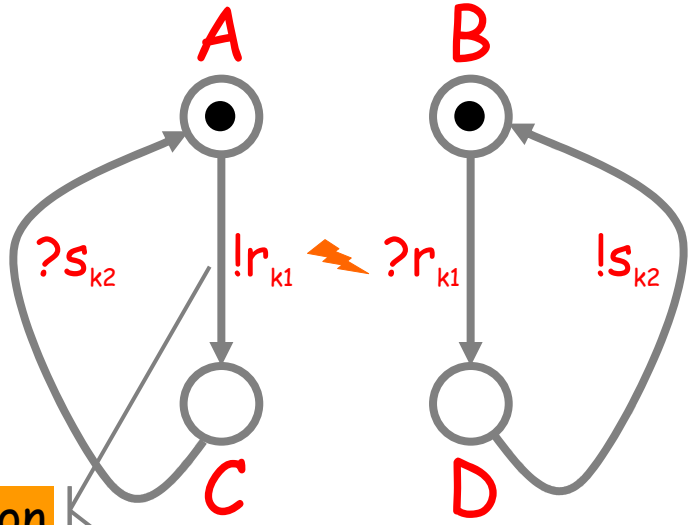


Reaction oriented

1 line per reaction

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

A different process calculus (π)



Interaction oriented

1 line per component

$A = !r_{k1}; C$

$C = ?s_{k2}; A$

$B = ?r_{k1}; D$

$D = !s_{k2}; B$

A becomes C not D!

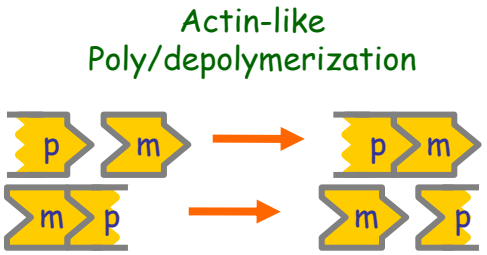
A compositional graphical representation (precise, dynamic *and* modular) and the corresponding calculus.

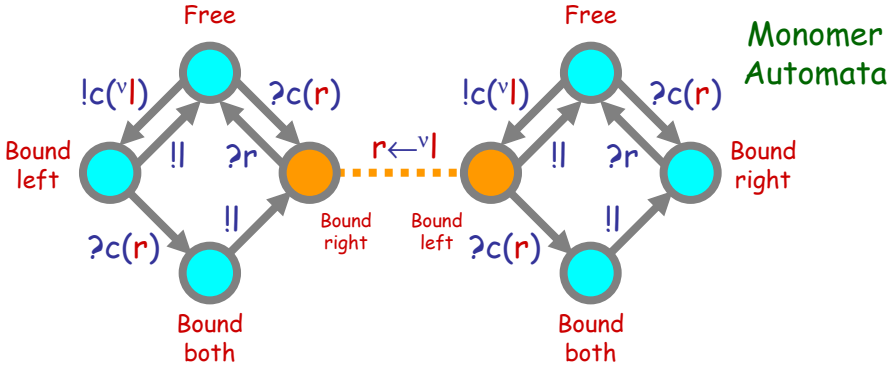
The same "model"

Maps to a CTMC | Maps to a CTMC

Why π-calculus, in particular

- Well studied, compact, precise, and general
  - A "programming language" first, a mathematical model second
  - Syntax (configurations): P ::= 0 | P+P | P|P | ?n(n).P | !n(n).P | (νn)P | *P
  - Semantics (reactions): (P' + !n(m).P) | (Q' + ?n(m').Q) → P | Q{m'←m}
- Binary interactions
  - I.e., "collisions"
- Reactive and compositional
  - Each subsystem is a separate (composition of) nondeterministic automata interacting with the environment (more automata)
- Dynamic network evolution and species evolution
  - Each subsystem can create fresh connections or spawn new subsystems
- Compact description of combinatorics (like any programming language)
  - $(Bit_1 \mid Bit_2 \mid \ldots \mid Bit_n)$ size-n description, where Bit is a 2-state subsystem
    - E.g. a protein with with 2^n phosphorilation configurations (i.e. "different chemical species")
- Complexation/polymerization
  - The most characteristic feature of π-calculus (fresh names) models "sticking"





Stochastic π-calculus

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

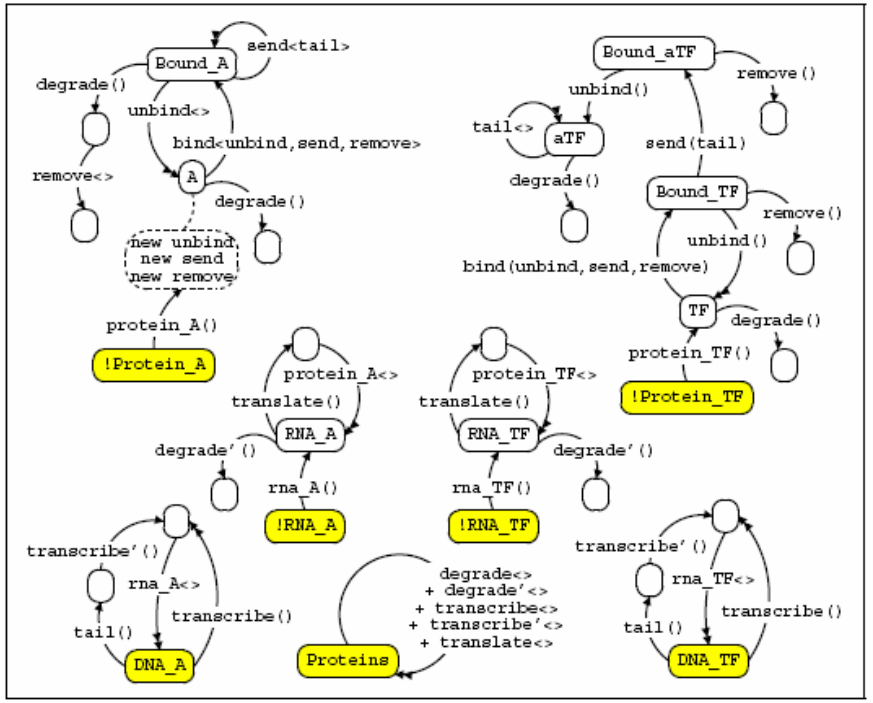


Figure 2. Regulating Gene Expression by Positive Feedback [9]

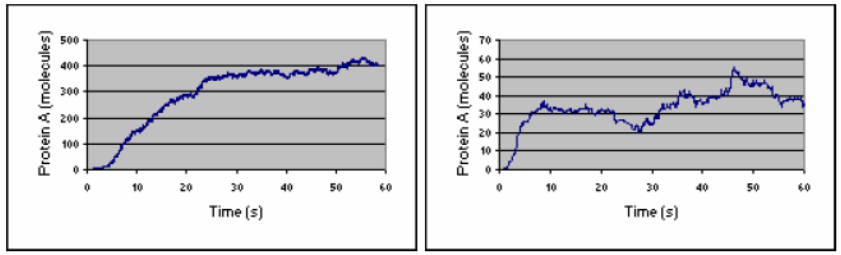


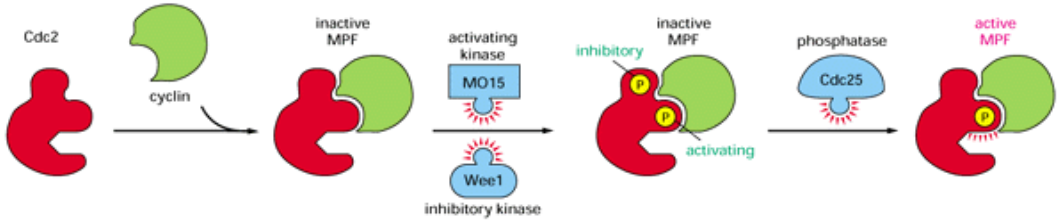
Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of TF

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

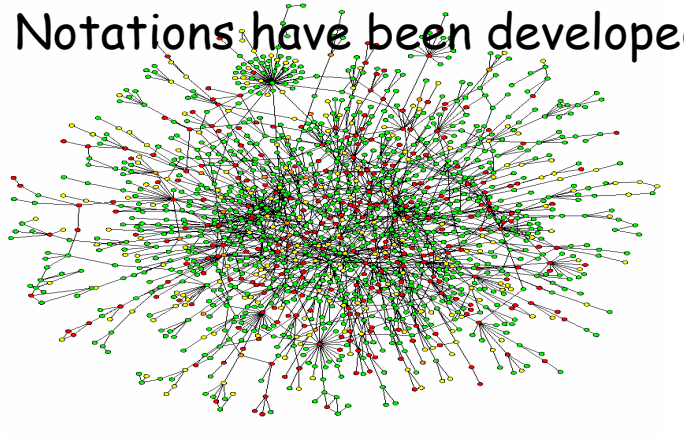
1. The Protein Machine

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.



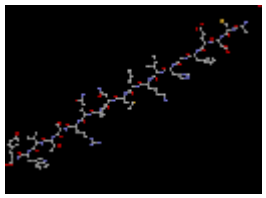
- Mapping out such networks:
  - In principle, it's "just" a very large set of chemical equations.
  - Notations have been developed to summarize and abstract.



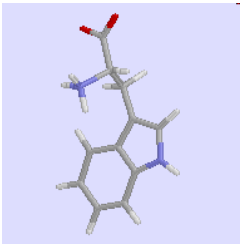
An actual molecular interaction network.
(Nodes are distinct protein kinds,
arcs mean that two kinds of proteins interact.)

Protein Structure

Primary

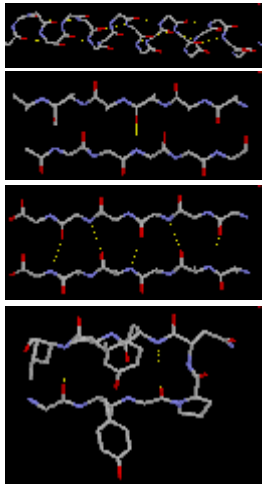


The 20 Aminoacids



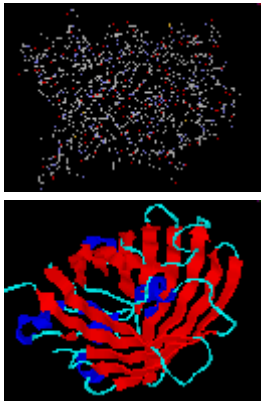
Tryptophan

Secondary



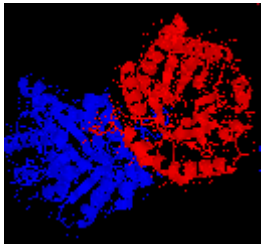
Alpha Helix, Beta Sheet

Tertiary



Green Fluorescent Protein

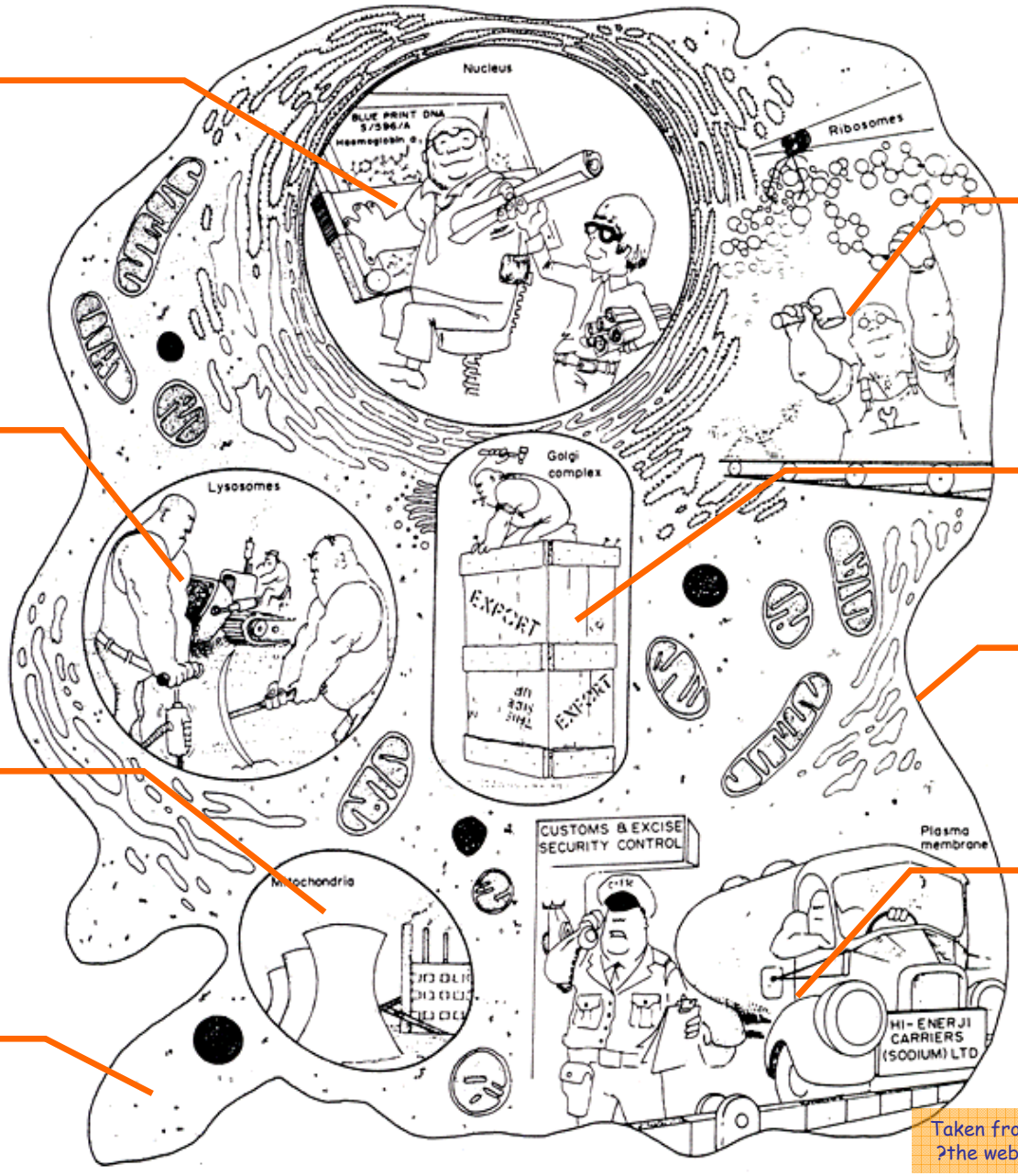
Quaternary



Triose Phosphate Isomerase

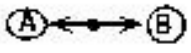
http://www.cmbi.kun.nl/gvteach/bioinformatica1/

Protein Function



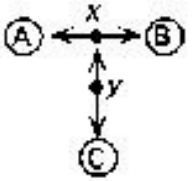
Taken from ?the web?

MIM: Molecular Interaction Maps (Kohn)

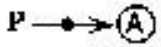


The double-arrowed line indicates that proteins **A** and **B** can bind to each other. The "node" placed on the line represents the **A:B** complex.

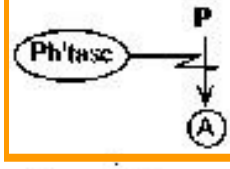
Asymmetric binding where protein **A** donates a peptide that binds to a receptor site or pocket on protein **B**.



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.



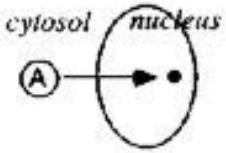
Covalent modification of protein **A**. The single-arrowed line indicates that **A** can exist in a phosphorylated state. The node represents the phosphorylated species.



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

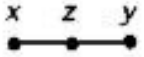
Proteolytic cleavage at a specific site within a protein.

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 (Kohn)

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

JDesigner

The p53-Mdm2 and DNA Repair Regulatory Network

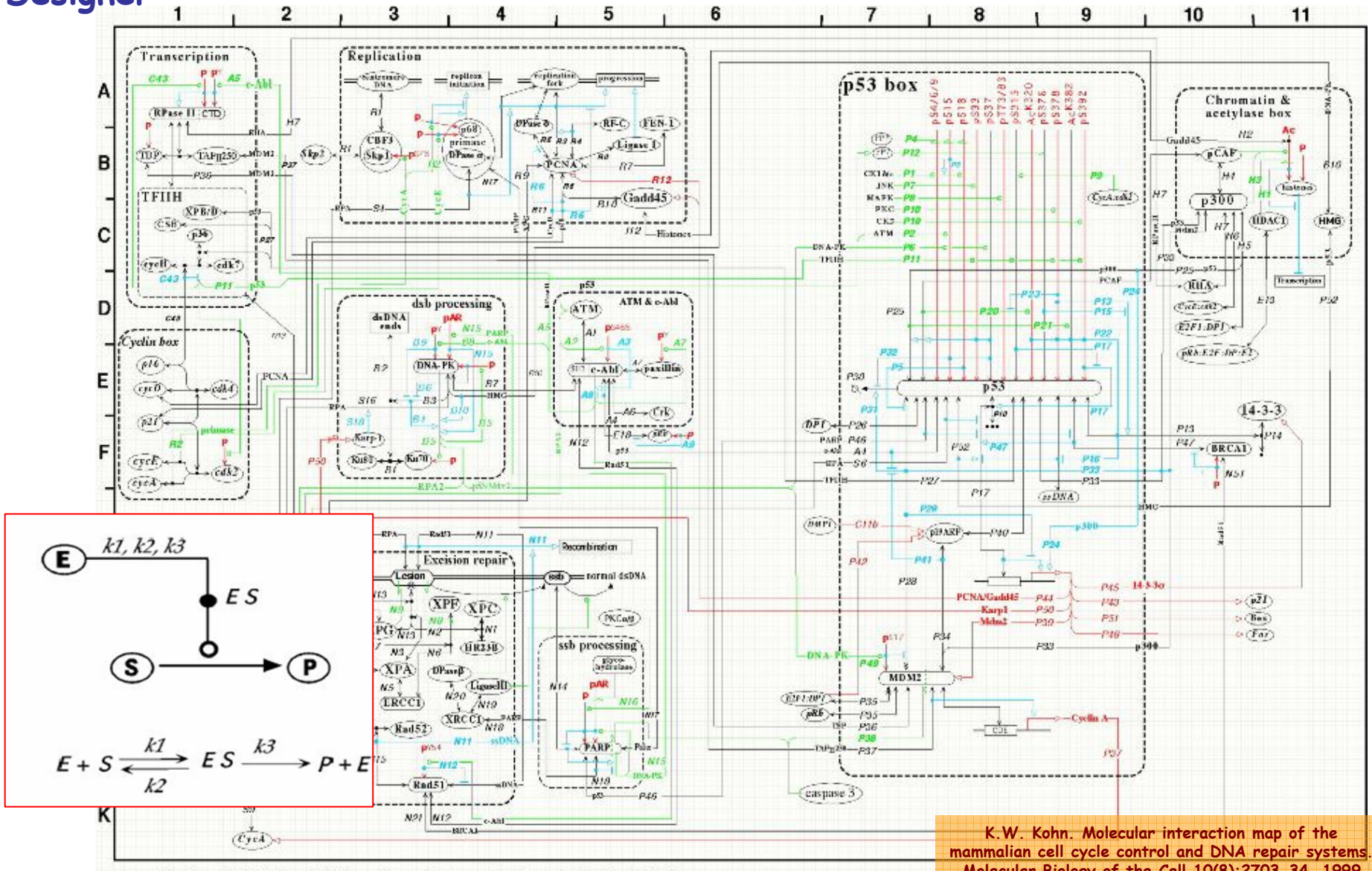
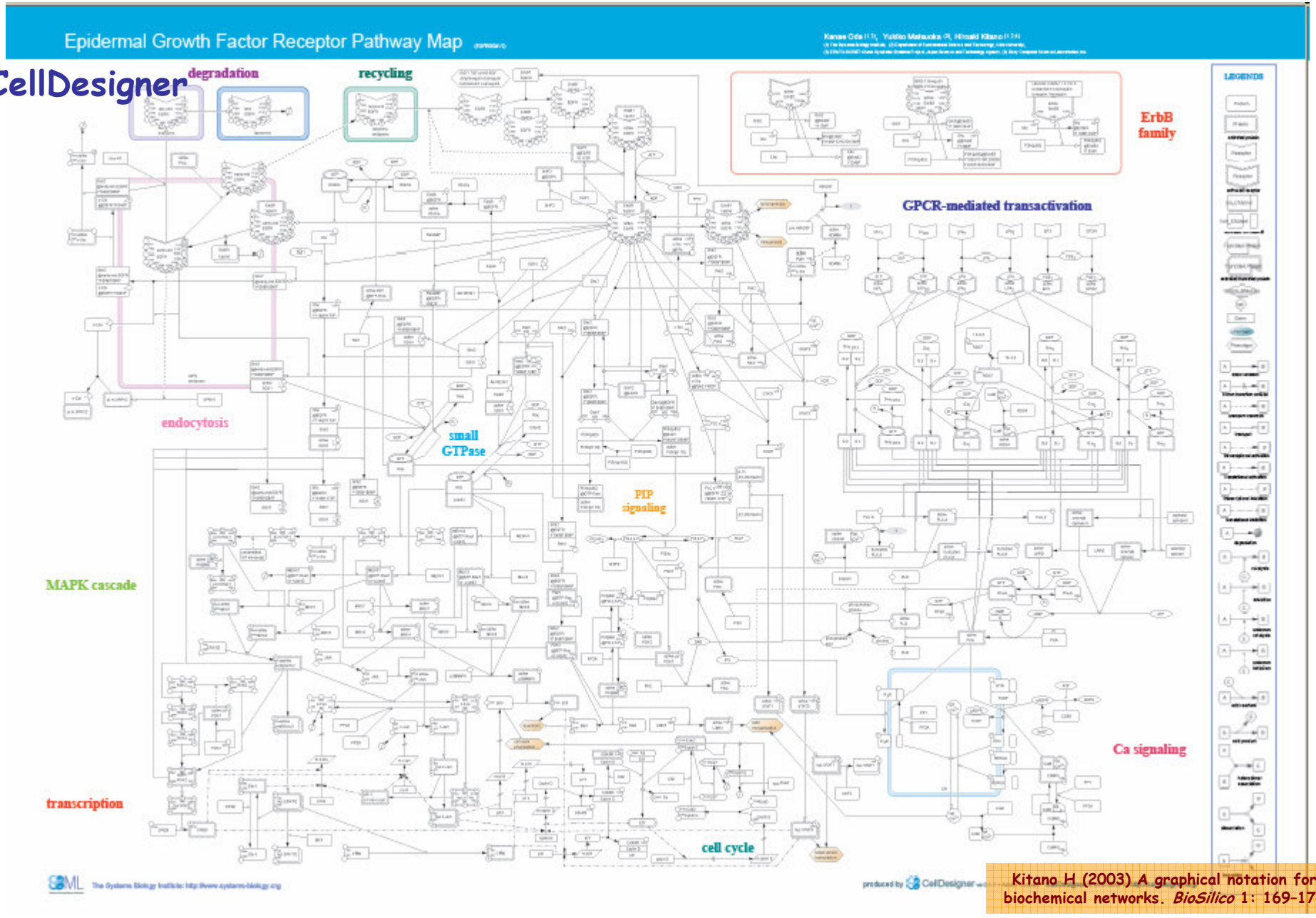


Figure 6B: The p53-Mdm2 and DNA repair regulatory network (version 2p - May 19, 1999)

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.

Molecular Interaction Maps (Kitano)

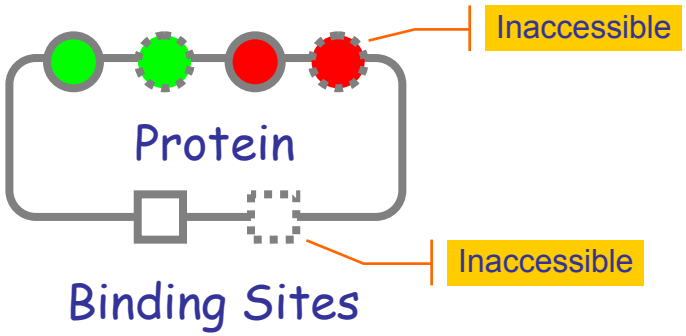


Kitano H (2003) *A graphical notation for biochemical networks. BioSilico* 1: 169-176

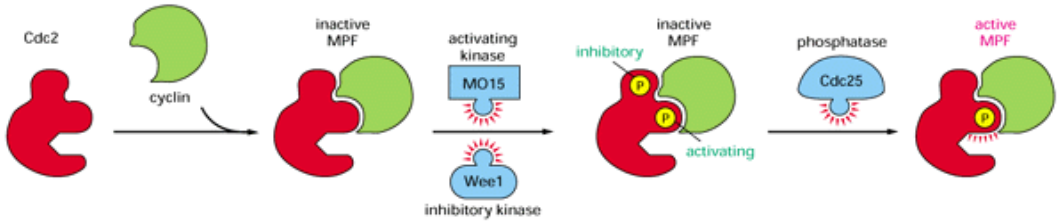
The Protein Machine "Instruction Set"

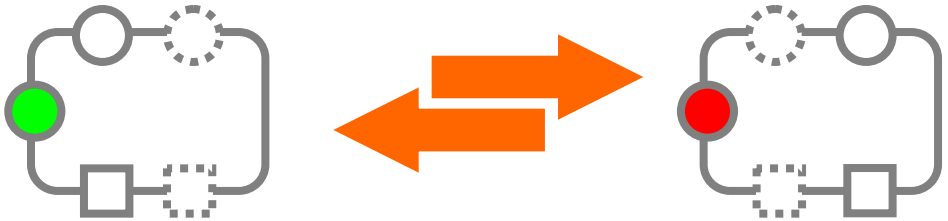
cf. BioCalculus [Kitano&Nagasaki], κ-calculus [Danos&Laneve]

On/Off switches



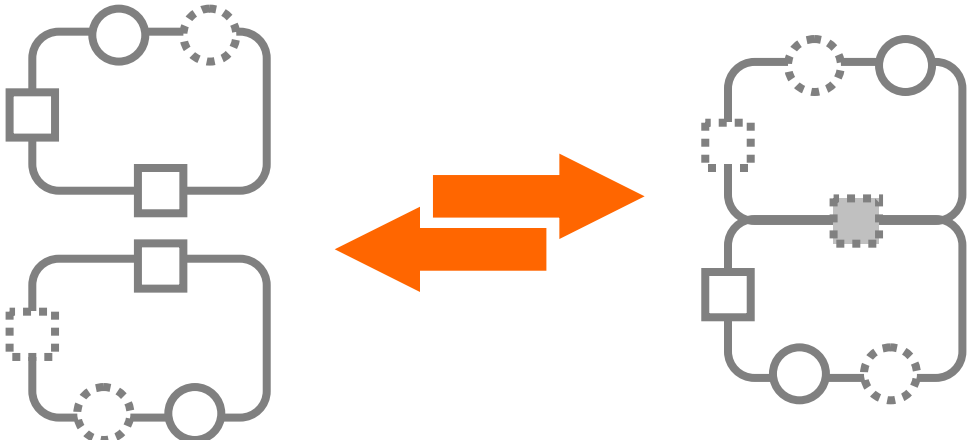
Each protein has a structure
of binary switches and binding sites.
But not all may be always *accessible*.





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

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.

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.

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

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

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

$$KK\text{-}P + KK\ P'ase \underset{d_4}{\overset{a_4}{\rightleftharpoons}} KK\text{-}P{\cdot}KK\ P'ase$$

$$\xrightarrow{k_4} KK + KK\ P'ase \quad [4]$$

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

$$KK\text{-}PP + KK\ P'ase \underset{d_6}{\overset{a_6}{\rightleftharpoons}} KK\text{-}PP{\cdot}KK\ P'ase$$

$$\xrightarrow{k_6} KK\text{-}P + KK\ P'ase \quad [6]$$

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

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

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

$$K\text{-}PP + K\ P'ase \underset{d_{10}}{\overset{a_{10}}{\rightleftharpoons}} KK\text{-}PP{\cdot}K\ P'ase$$

$$\xrightarrow{k_{10}} K\text{-}P + K\ P'ase \quad [10]$$

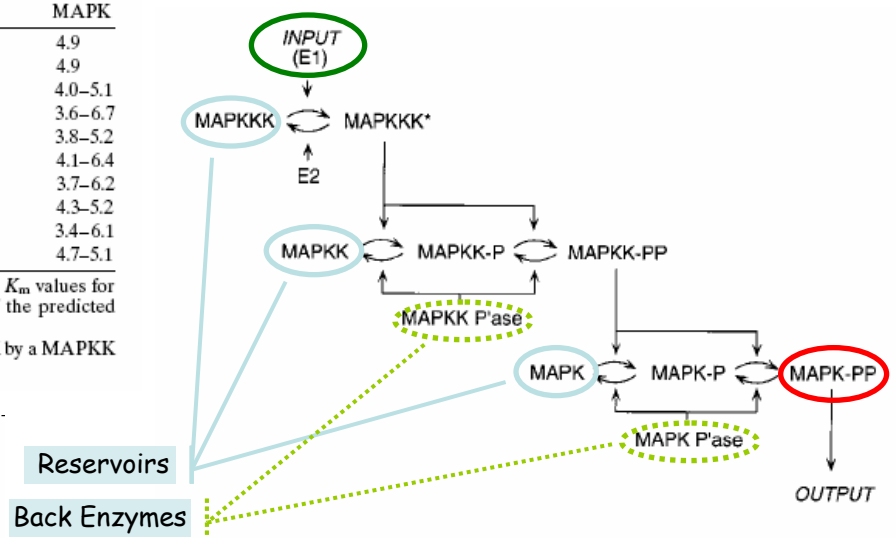
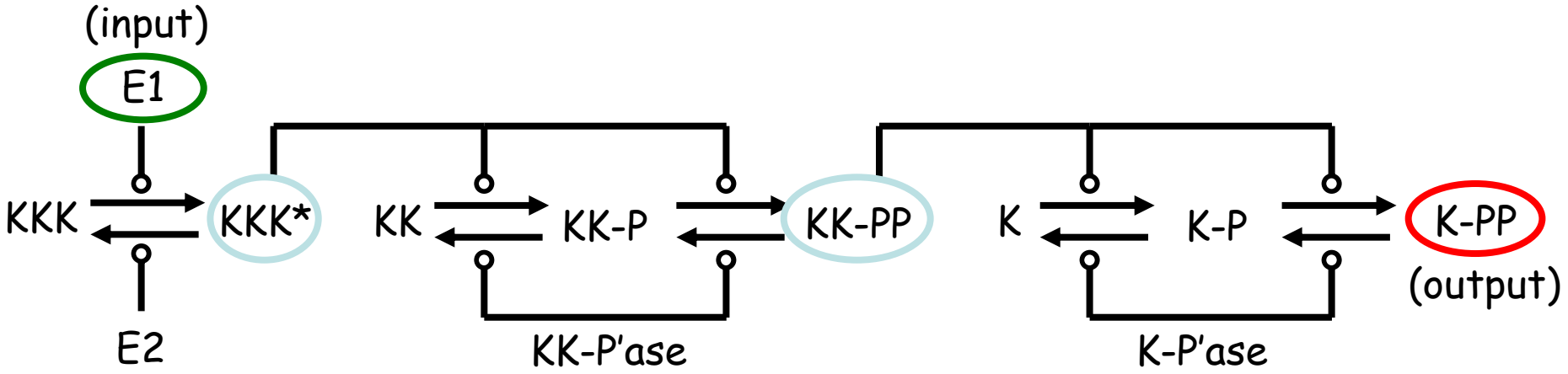


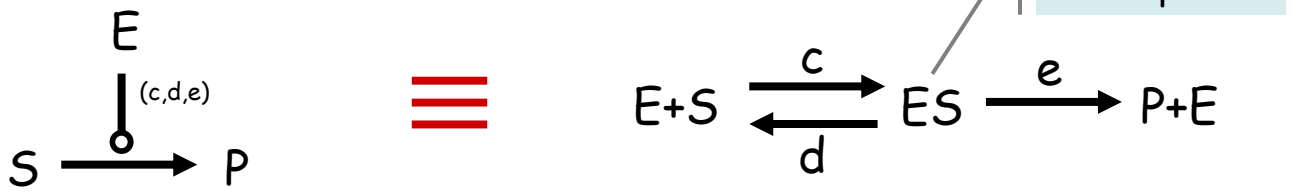
FIG. 1. Schematic view of the MAPK cascade. Activation of MAPK depends upon the phosphorylation of two conserved sites [Thr-183 and Tyr-185 in rat p42 MAPK/Erk2 (4, 5)]. Full activation of MAPKK also requires phosphorylation of two sites [Ser-218 and Ser-222 in mouse Mek-1/MKK1 (6–10)]. Detailed mechanisms 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.

The Circuit

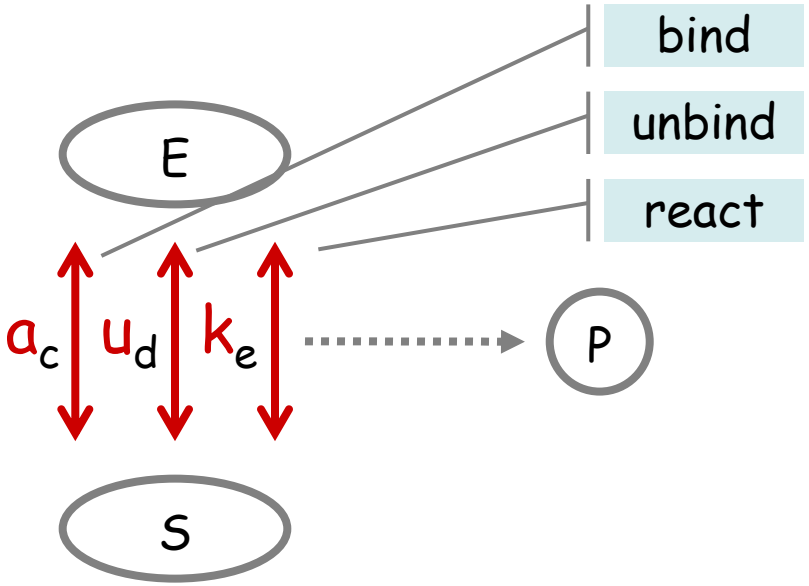


Enzymatic Reactions

Reaction View



Interaction View



private bindings between one S and one E molecule

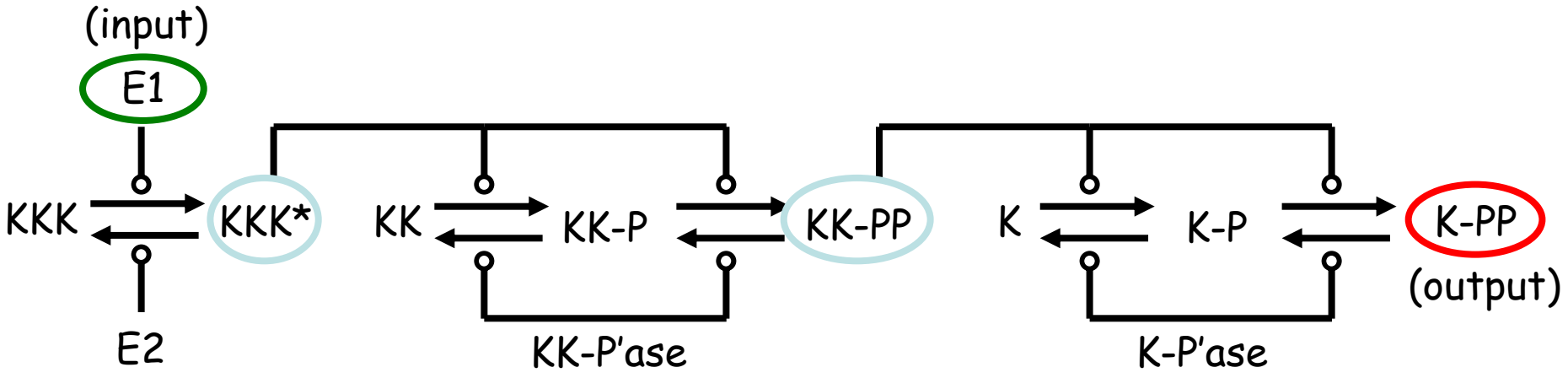
$S() \triangleq$ new u@d new k@e
$!a_c(u,k);\ (!u_d;\ S() + !k_e;\ P())$

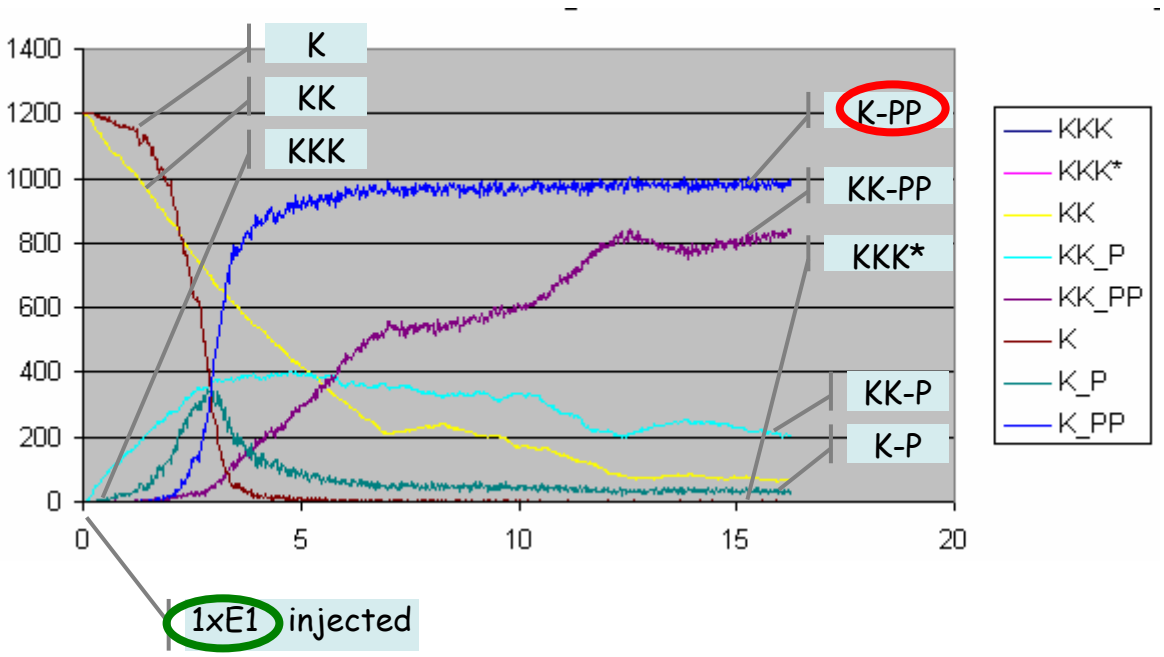
bind unbind react

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

$P() \triangleq$ ...

MAPK Cascade Simulation in SPiM





1st stage:
KKK* barely rises

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

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

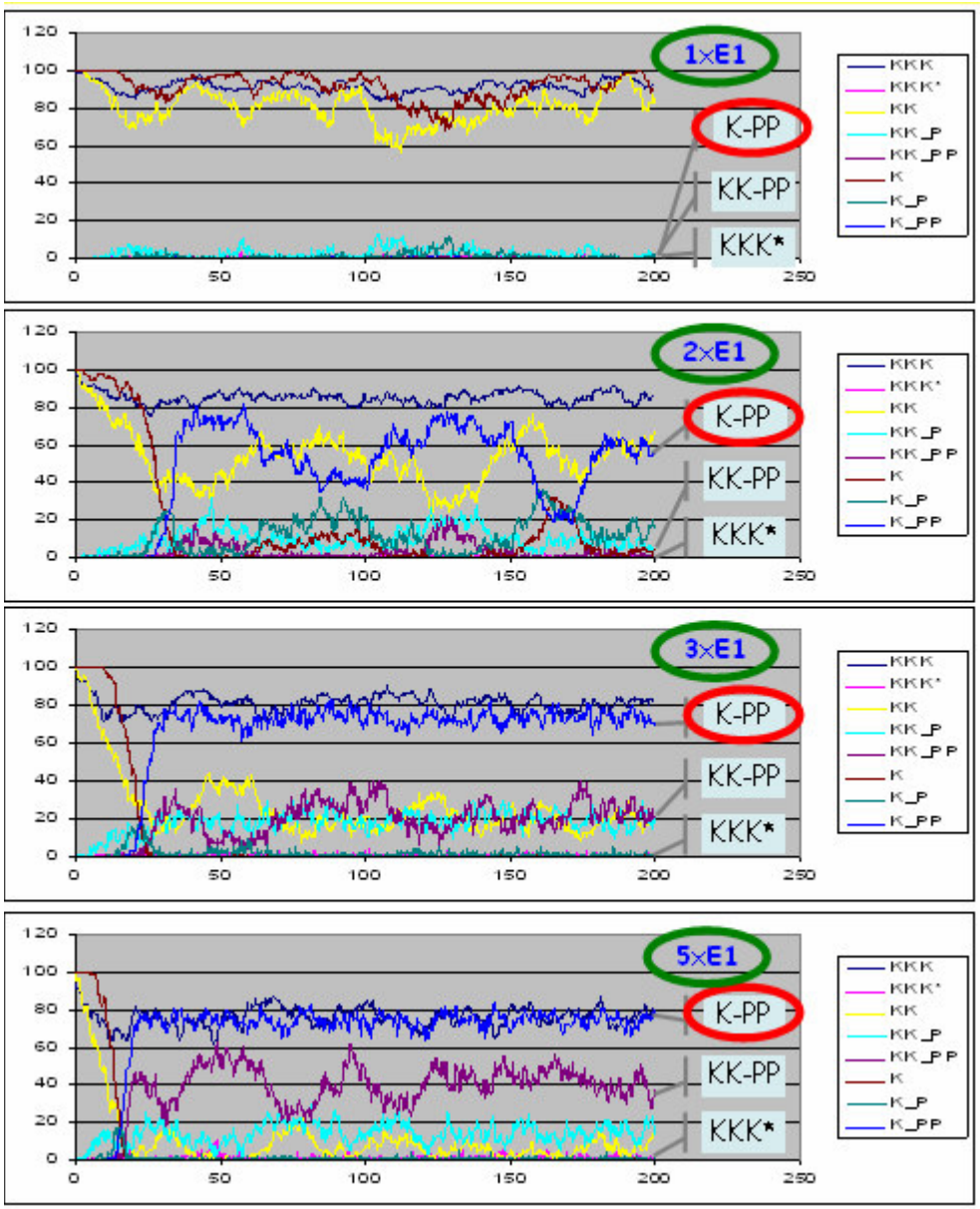
Rates and concentrations from paper:

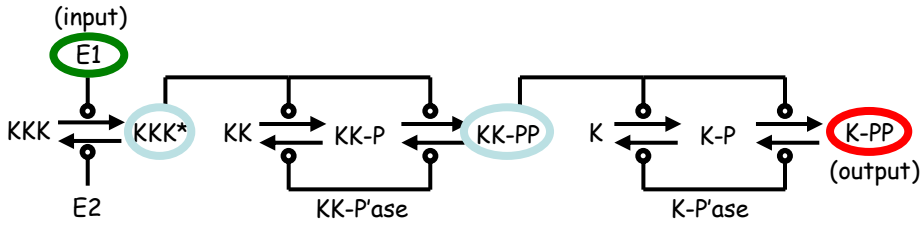
1xE2 (0.3 nM)
1xKKPase (0.3 nM)
120xKPase (120 nM)
3xKKK (3 nM)
1200xKK (1.2 uM)
1200xK (1.2 uM)

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

1xE1

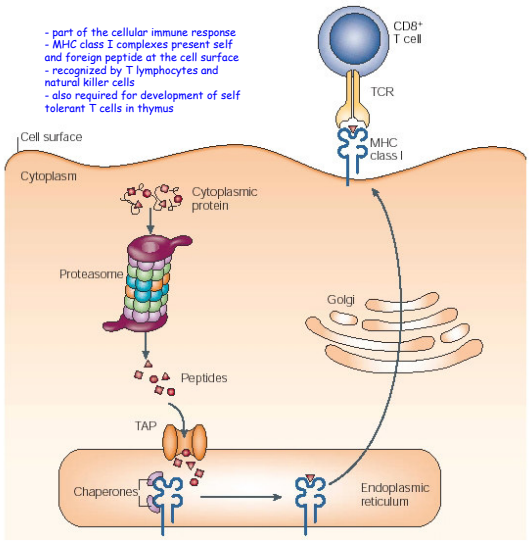
MAPK Cascade Simulation in SPiM



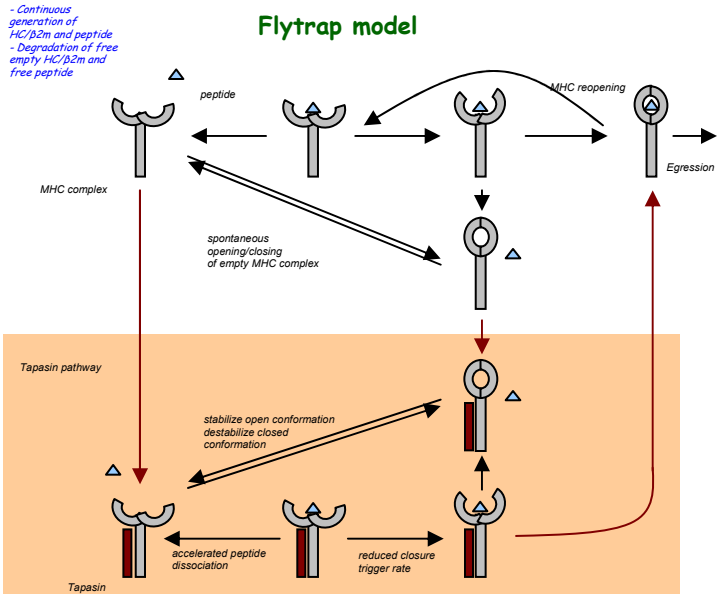


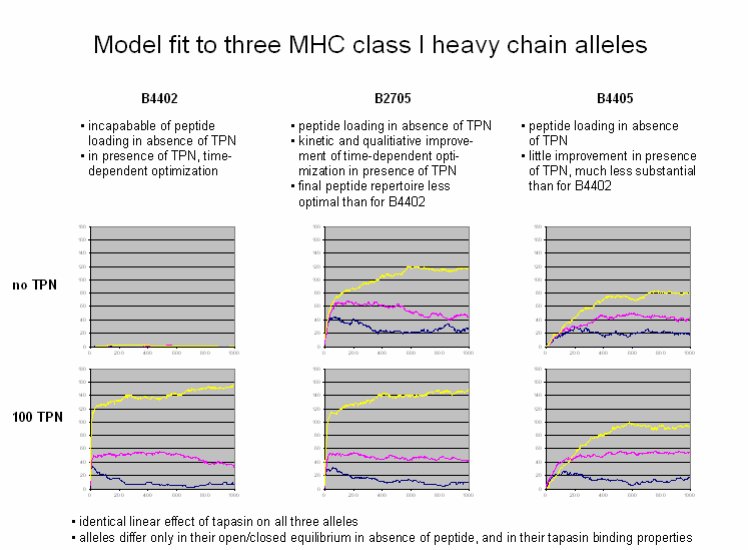
All coefficients 1.0 !!!
100xKKK, 100xKK, 100xK,
13xE2, 13xKKPse, 13xKPse.
nxE1 as indicated
(1xE1 is not sufficient to produce an output)

MHC Class I Antigen Presentation



Source: Jonathan W. Yewdell, Eric Reits, and Jacques Neefjes. Making sense of mass destruction: quantitating MHC class I antigen presentation. *Nature Reviews Immunology*, 3(12):952–961, 2003.





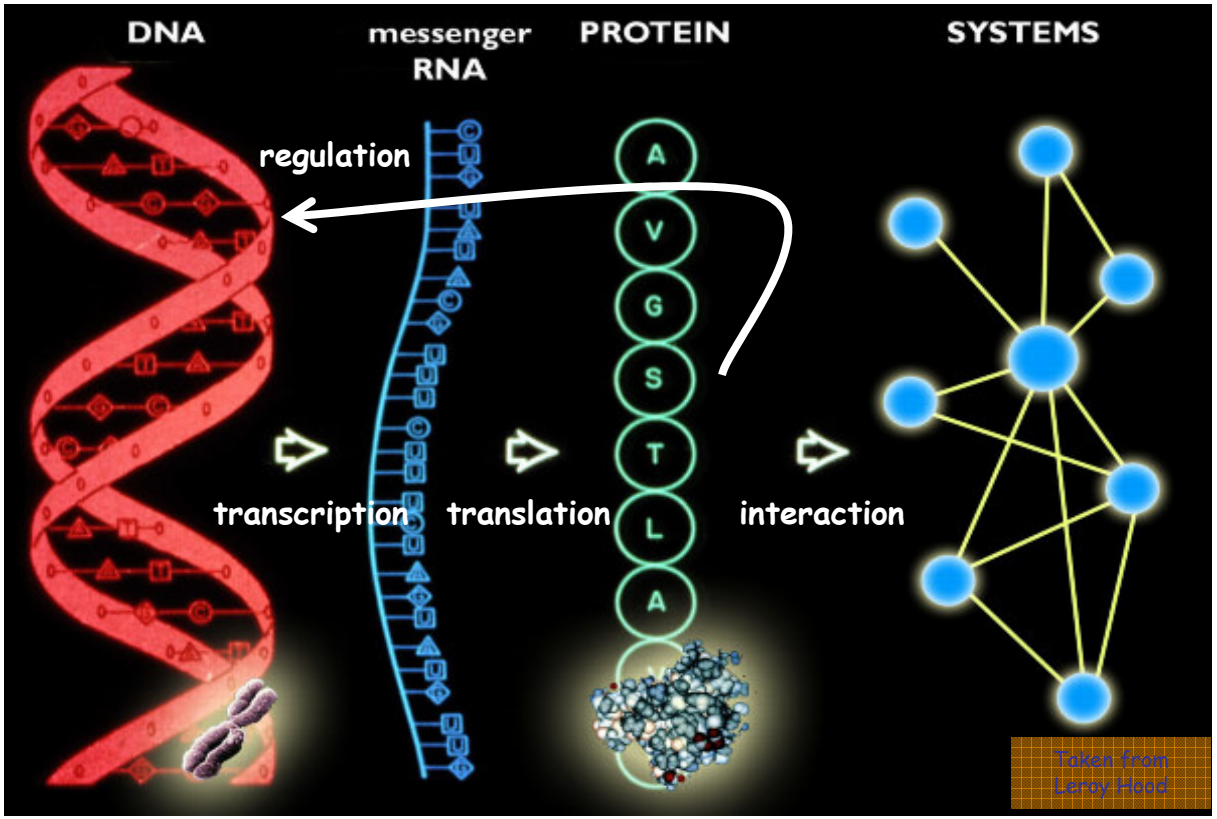
A stochastic pi-calculus model of MHC class I antigen presentation, Leonard Goldstein.

with Luca Cardelli and Andrew Phillips (Microsoft) and Tim Elliott and Joern Werner (U. Southampton)

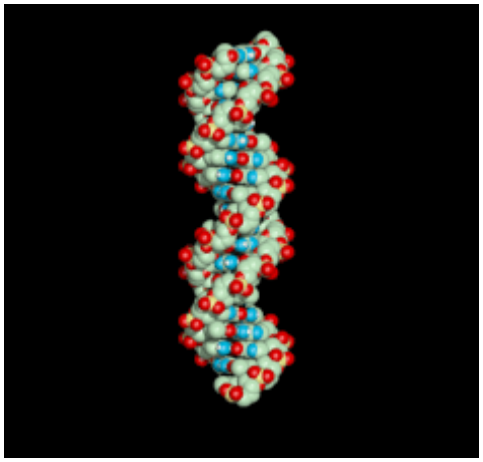
2. The Gene Machine

Pretty far from the atoms.

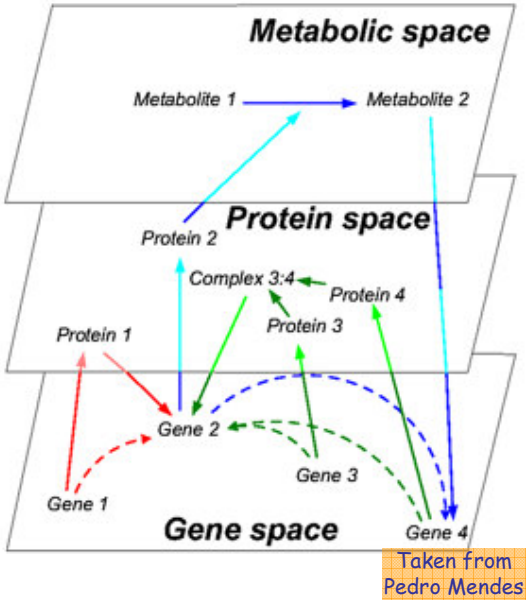
The "Central Dogma" of Molecular Biology



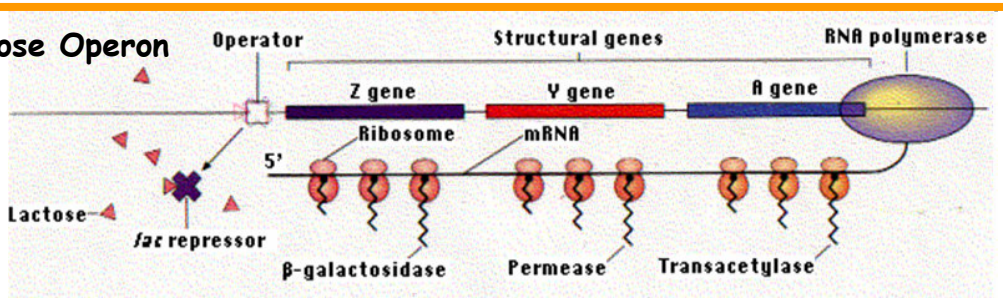
4-letter digital code | 4-letter digital code | 20-letter digital code | 50.000(?) shapes



DNA Tutorial

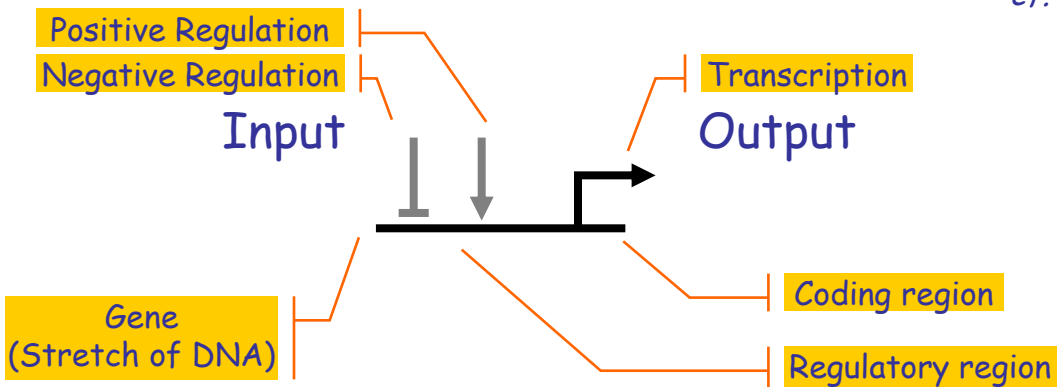


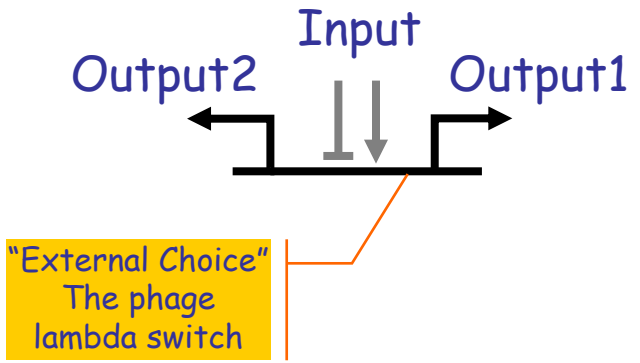
Lactose Operon



The Gene Machine "Instruction Set"

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





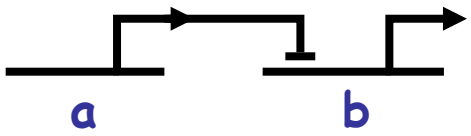
Regulation of a gene (positive and negative) influences transcription. The regulatory region has precise DNA sequences, but not meant for coding proteins: meant for binding regulators.

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

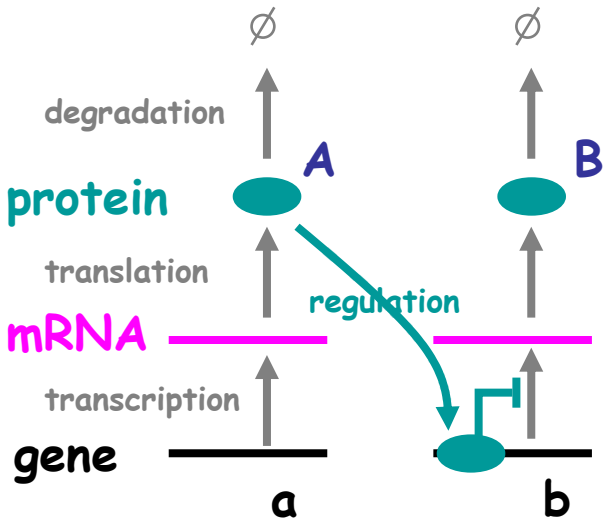
Human (and mammalian) Genome Size
3Gbp (Giga base pairs) 750MB @ 4bp/Byte (CD)
Non-repetitive: 1Gbp 250MB
In genes: 320Mbp 80MB
Coding: 160Mbp 40MB
Protein-coding genes: 30,000-40,000

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



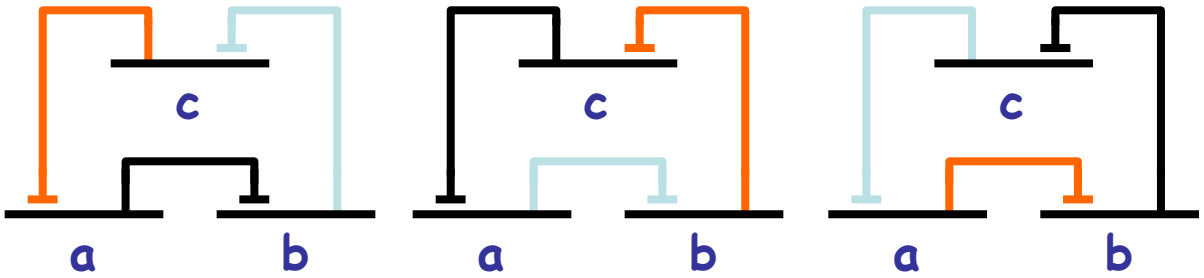
Is a shorthand for:



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

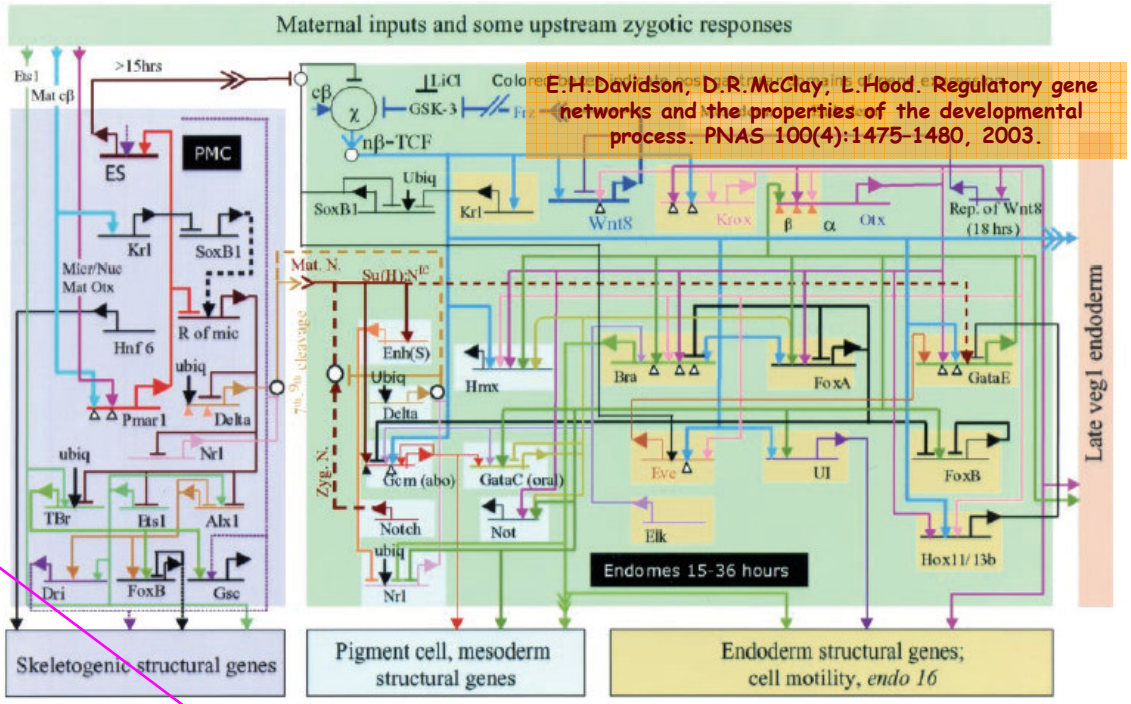


Expressed
Repressed
Expressing

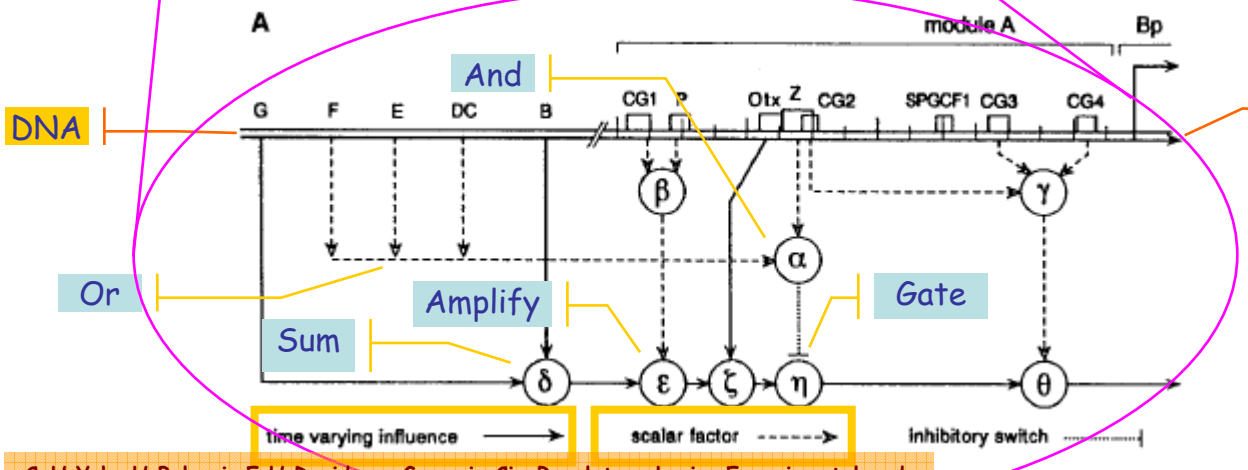
Gene Regulatory Networks

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

The Programming Model

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

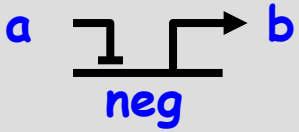
- How can it possibly work?
  - Stochastic broadcasting. The apparently crude idea of broadcasting a whole bunch of asynchronous decaying messages to activate a future gate, means there are never any "pipeline full" deadlocks, even in presence of abundant feedback loops.
  - Stochastic degradation. Degradation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

Notations for the Gene Machine

- Many of the same techniques as for the Protein Machine apply.
  - Process Calculi, Petri Nets, Term-Rewriting Systems…
- But the "programming model" is different.
  - Asynchronous stochastic control.
  - Biologically poorly understood.
  - Network "motifs" are being analyzed.
- Specific techniques:
  - Hybrid Petri Nets
    - [Matsuno, Doi, Nagasaki, Miyano] Gene Regulation
    - Genomic Object Net www.genomicobject.net
- Gene Regulation Diagrams
- Mixed Gene-Protein Diagrams

Gene Gates and Circuits

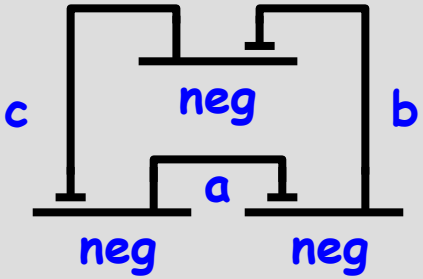
A gene gate



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

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

A genetic circuit (engineered in E.Coli)

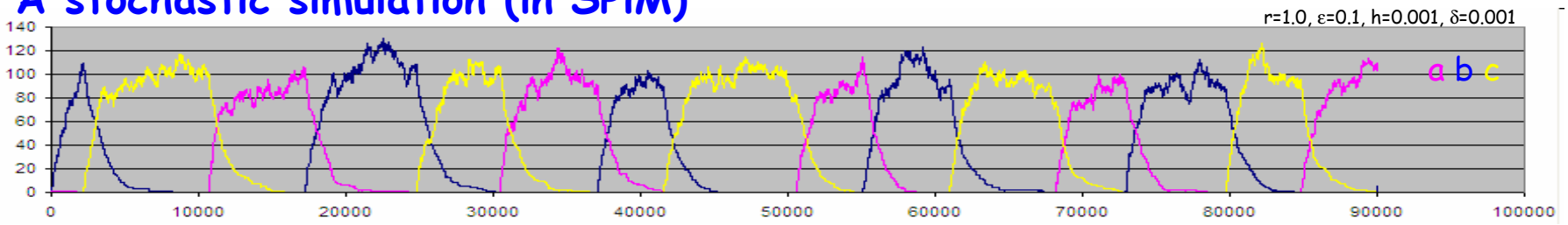


neg(a,b) |
neg(b,c) |
neg(c,a)

The stochastic-π program

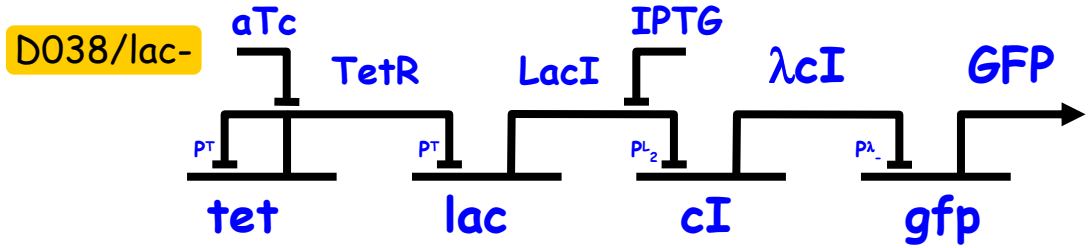
```
val dk = 0.001     (* Decay rate *)
val inh = 0.001    (* Inhibition rate *)
val cst = 0.1      (* Constitutive rate *)

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

let neg(a:chan(), b:chan()) =
  do ?a; delay@inh; 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))
```

A stochastic simulation (in SPiM)



Guet et al.: D038/lac⁻

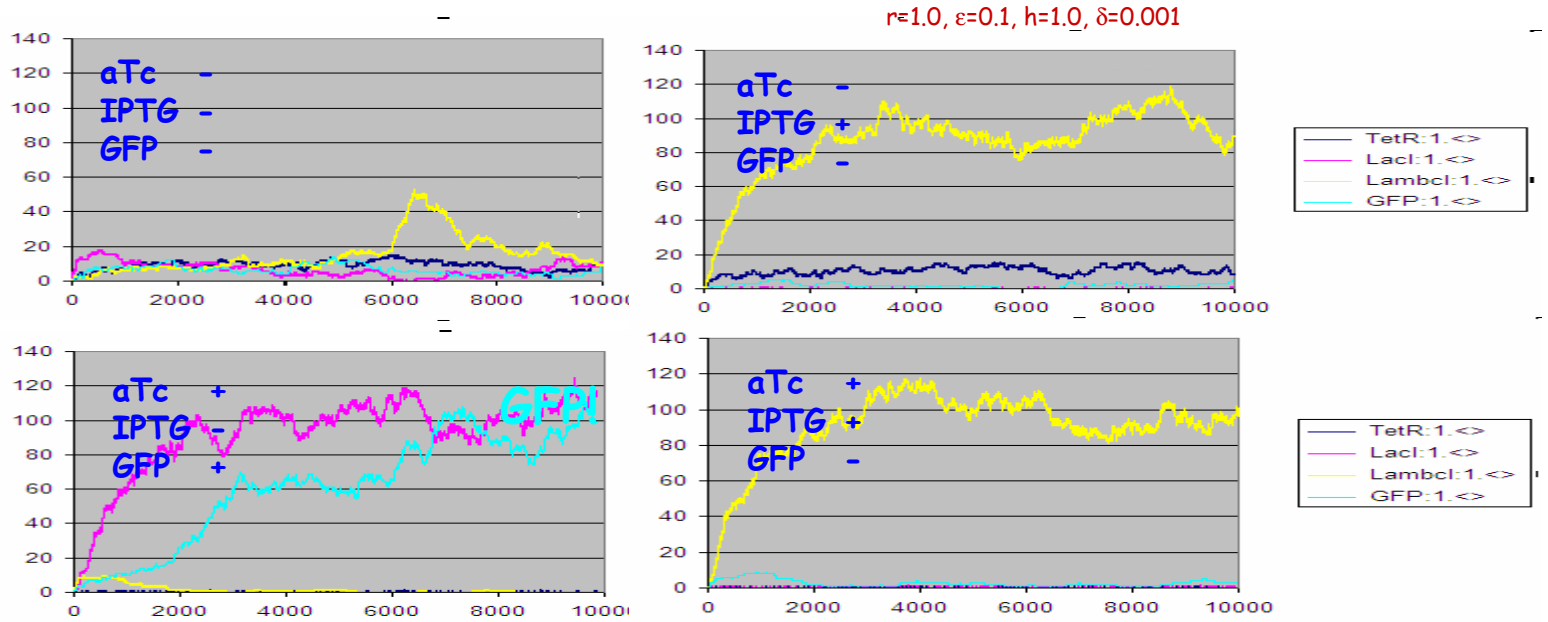
Combinatorial Synthesis of Genetic Networks, Guet, Elowitz, Hsing, Leibler, 1996, *Science*, May 2002, 1466-1470.



Experiment:	
aTc	0101
IPTG	0011
GFP	0100

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

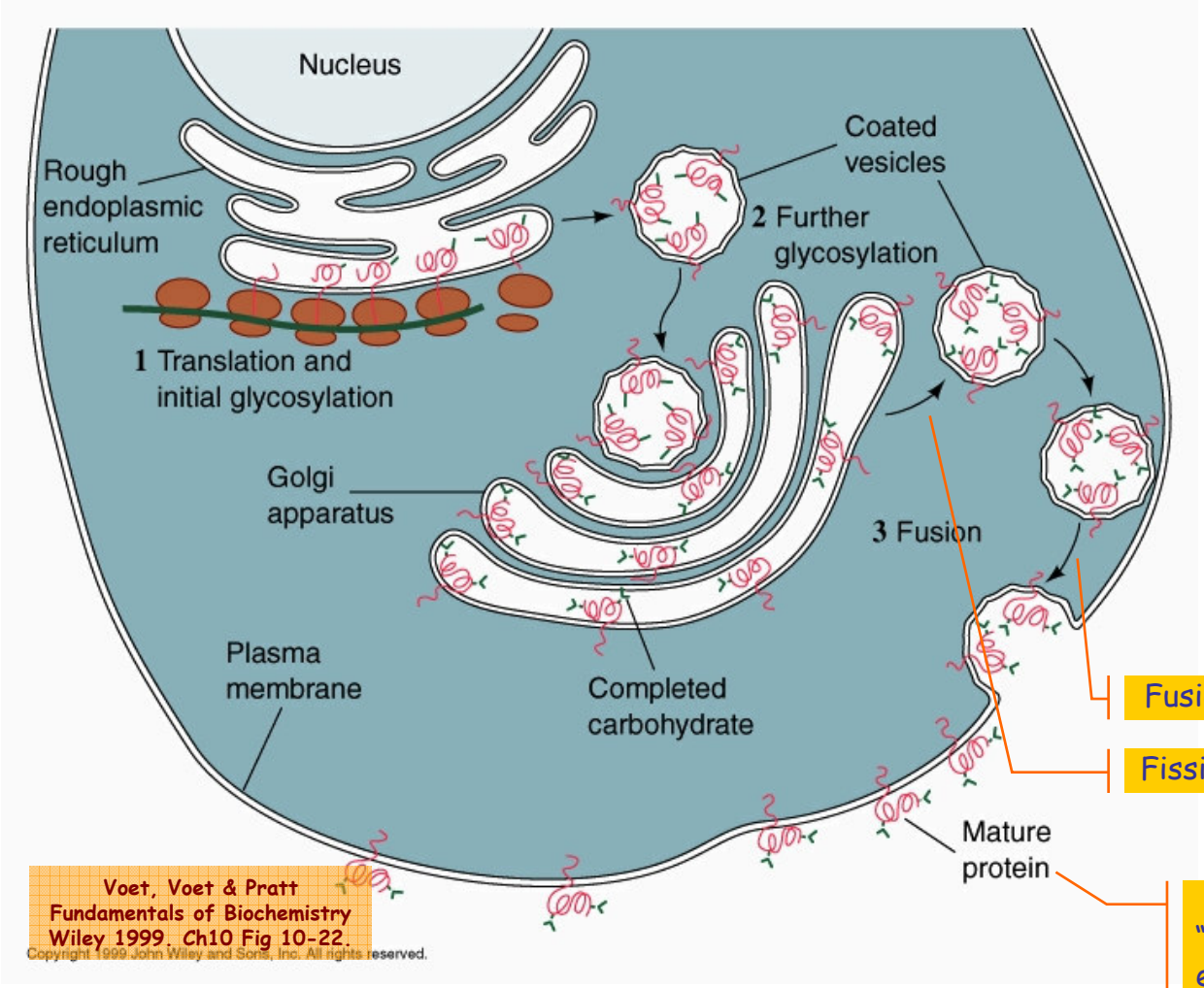
neg(TetR,TetR) | neg(TetR,LacI) | neg(LacI,λcI) | neg(λcI,GFP)



A Compositional Approach to the Stochastic Dynamics of Gene Networks, Ralf Blossey, Luca Cardelli, Andrew Phillips, *TCSB, Springer*, to appear.

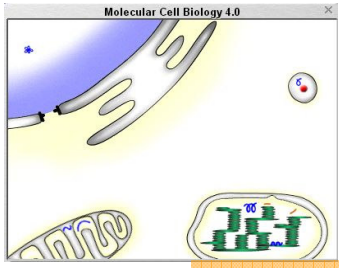
3. The Membrane Machine

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.



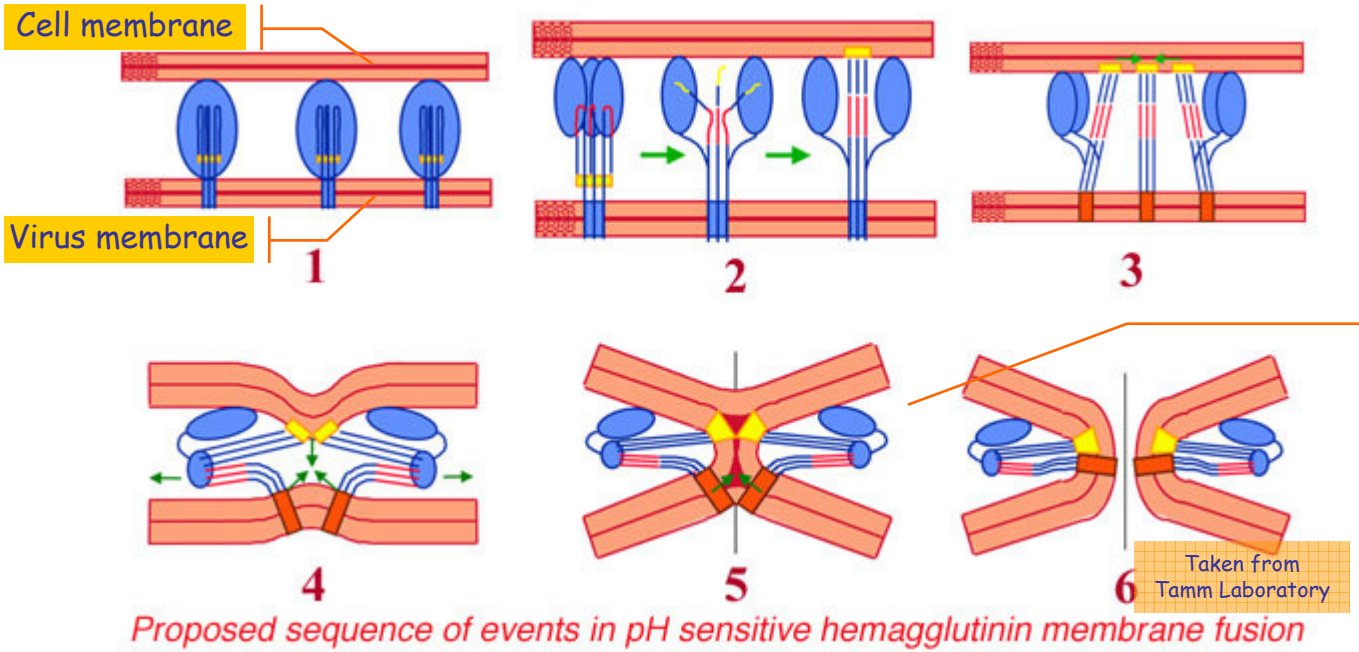
Taken from MCB CD

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]

Membrane Fusion

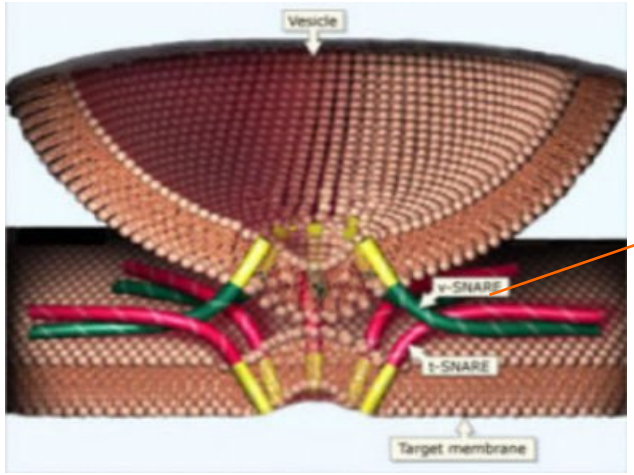
Positive curvature to Negative curvature transition in 3D



Proposed sequence of events in pH sensitive hemagglutinin membrane fusion

Aggressive fusion (virus)

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

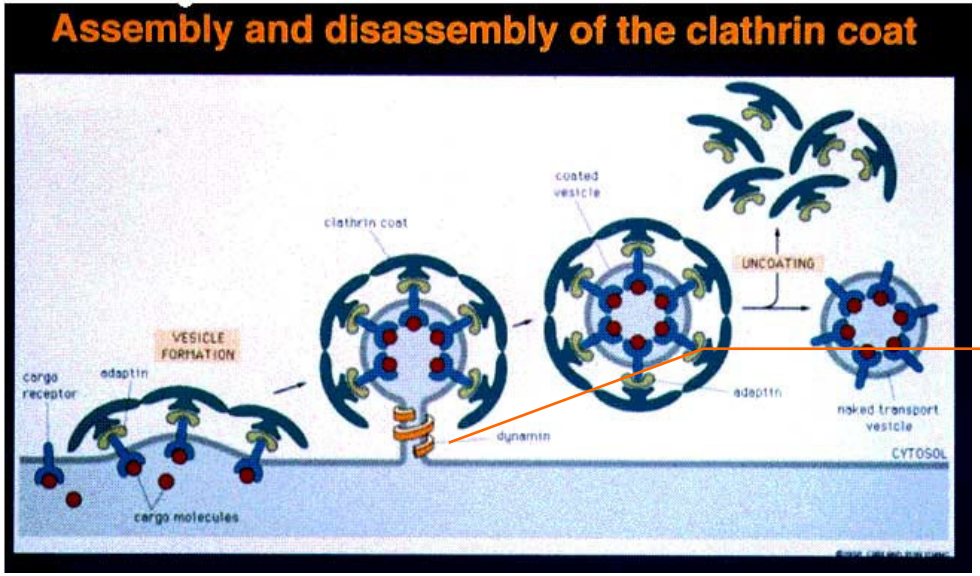


Cooperative fusion (vesicle)

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

Membrane Fission

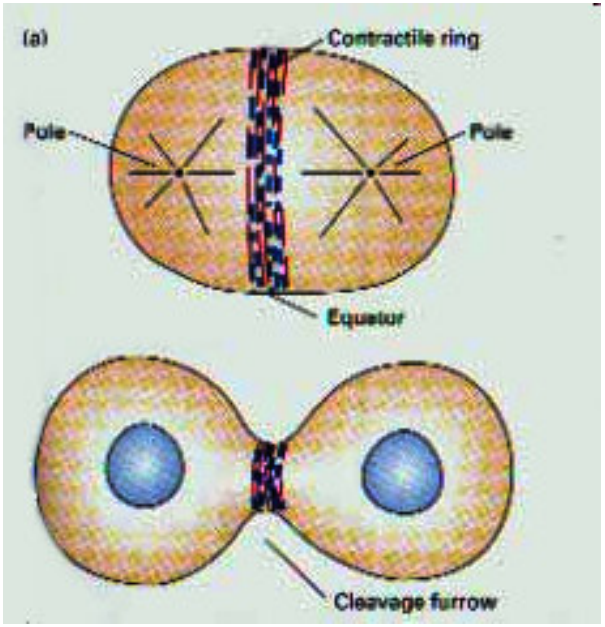
Negative curvature to Positive curvature transition in 3D



Vesicle Formation

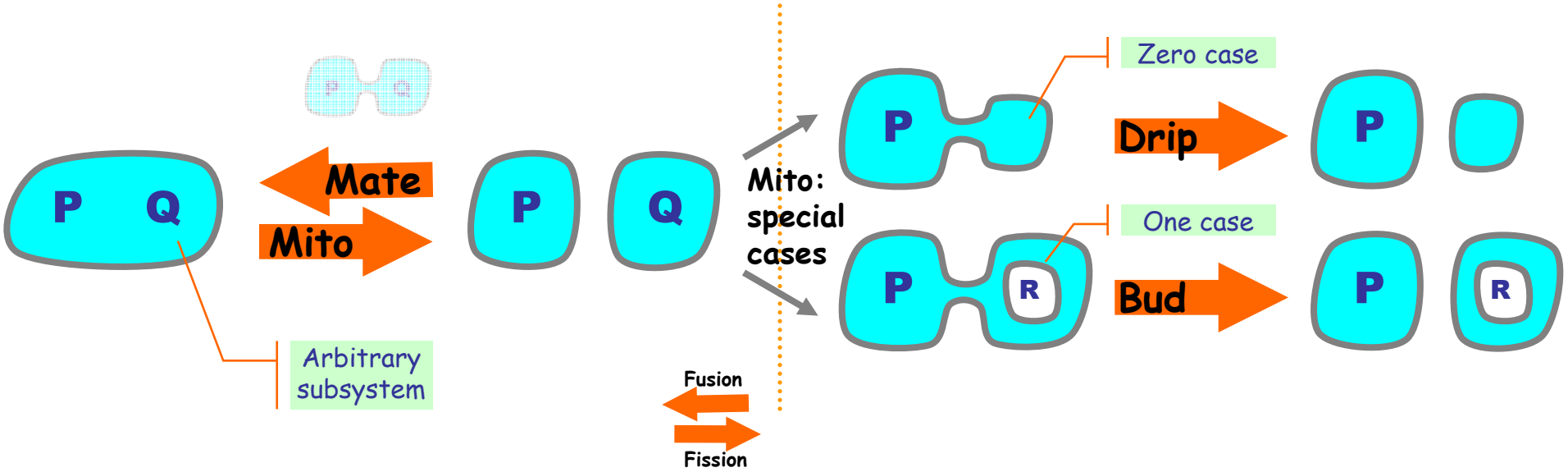
Movie by Allison Bruce

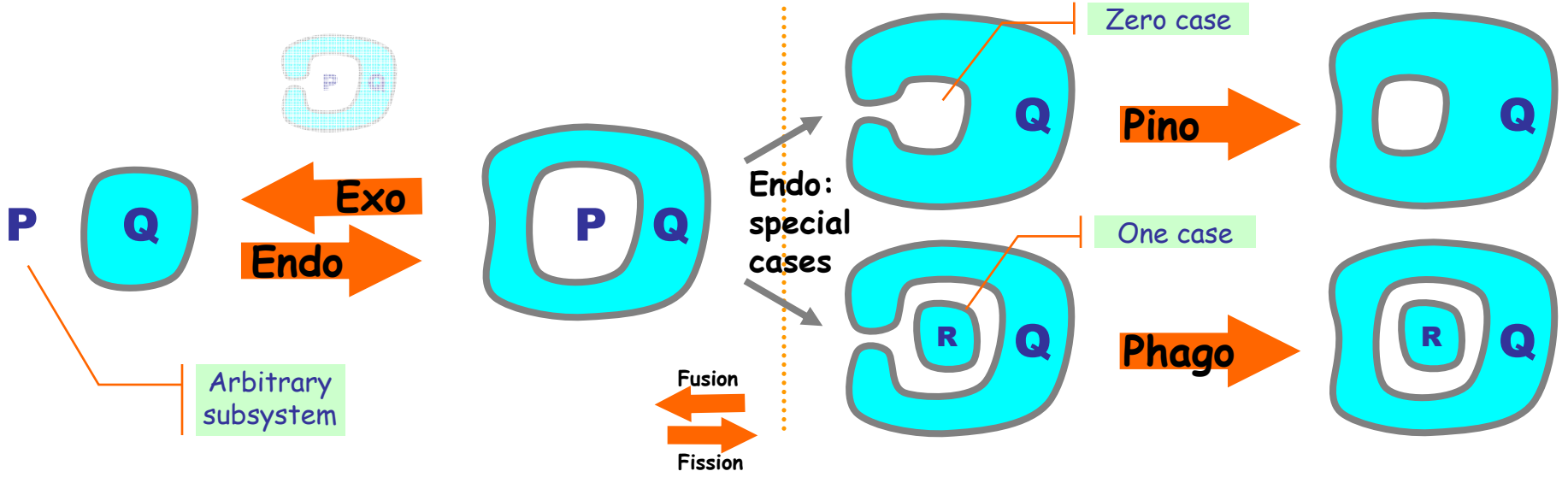
"Nonetheless, the actual process whereby a segment of phospholipid bilayer is 'pinched off' to form a pit and eventually a new vesicle is still not understood" [MCB p.746]



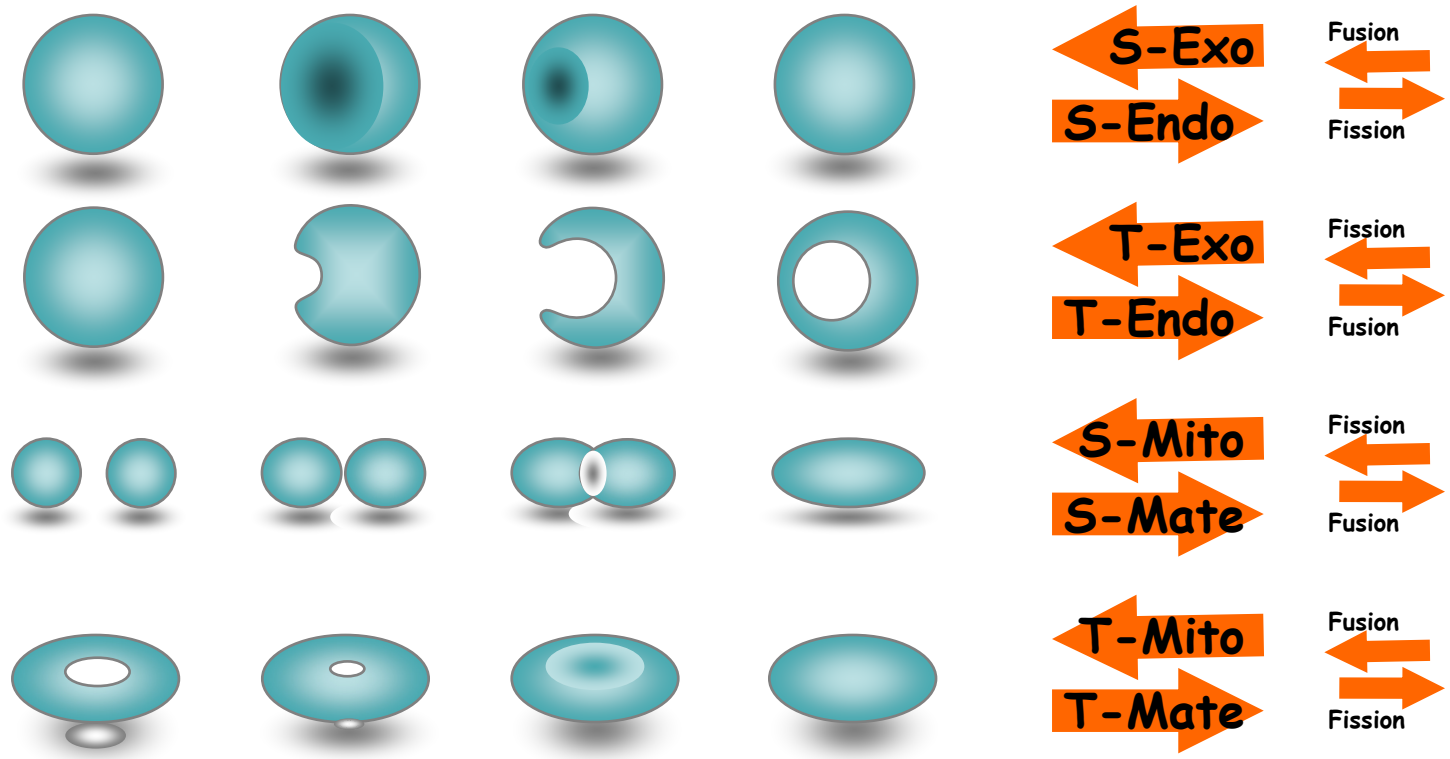
Cytokinesis (Mitosis)

The Membrane Machine "Instruction Set"

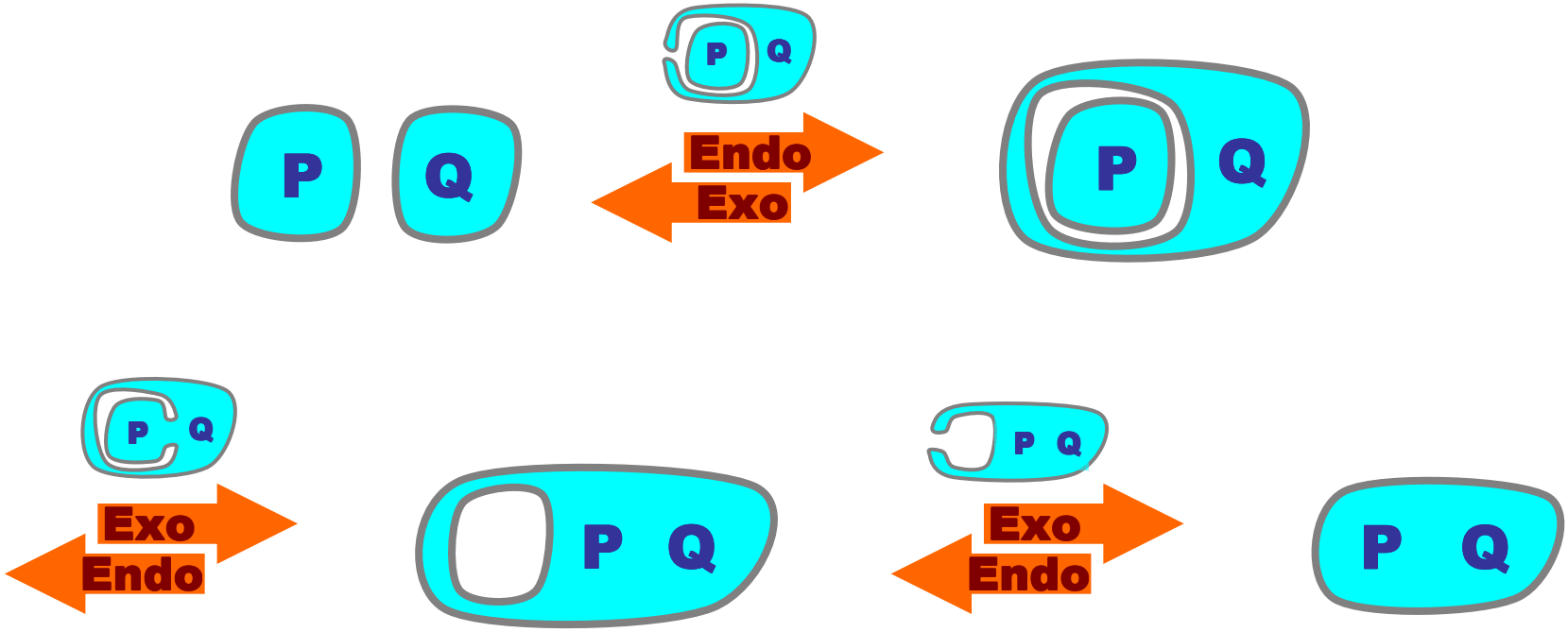




… in 3D


S-Exo
S-Endo
Fusion
Fission
T-Exo
T-Endo
Fission
Fusion
S-Mito
S-Mate
Fission
Fusion
T-Mito
T-Mate
Fusion
Fission

Mito/Mate by 3 Endo/Exo

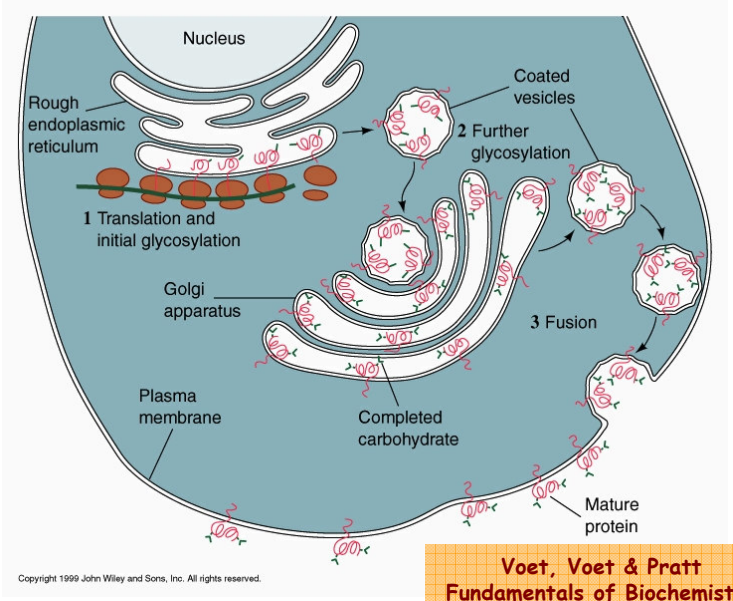


Notations for the Membrane Machine

- "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…

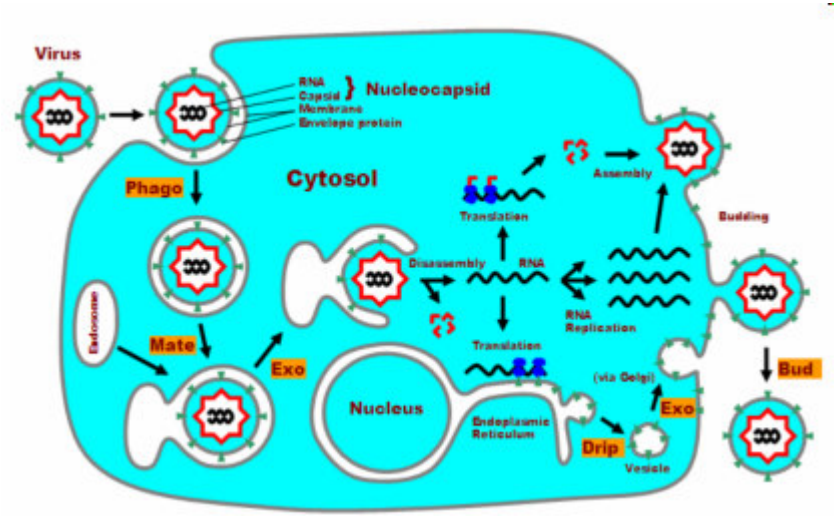
Membrane Algorithms

Protein Production and Secretion



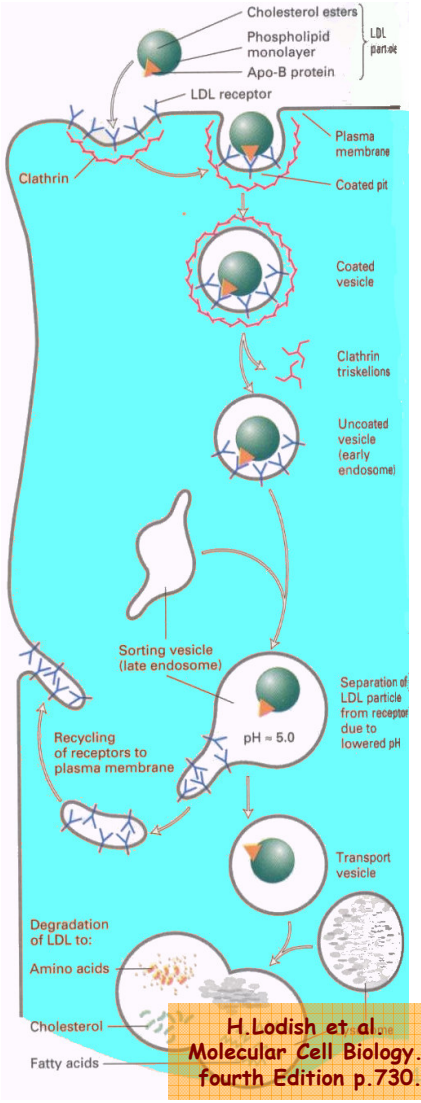
Voet, Voet & Pratt
Fundamentals of Biochemistry
Wiley 1999. Ch10 Fig 10-22.

Viral Replication



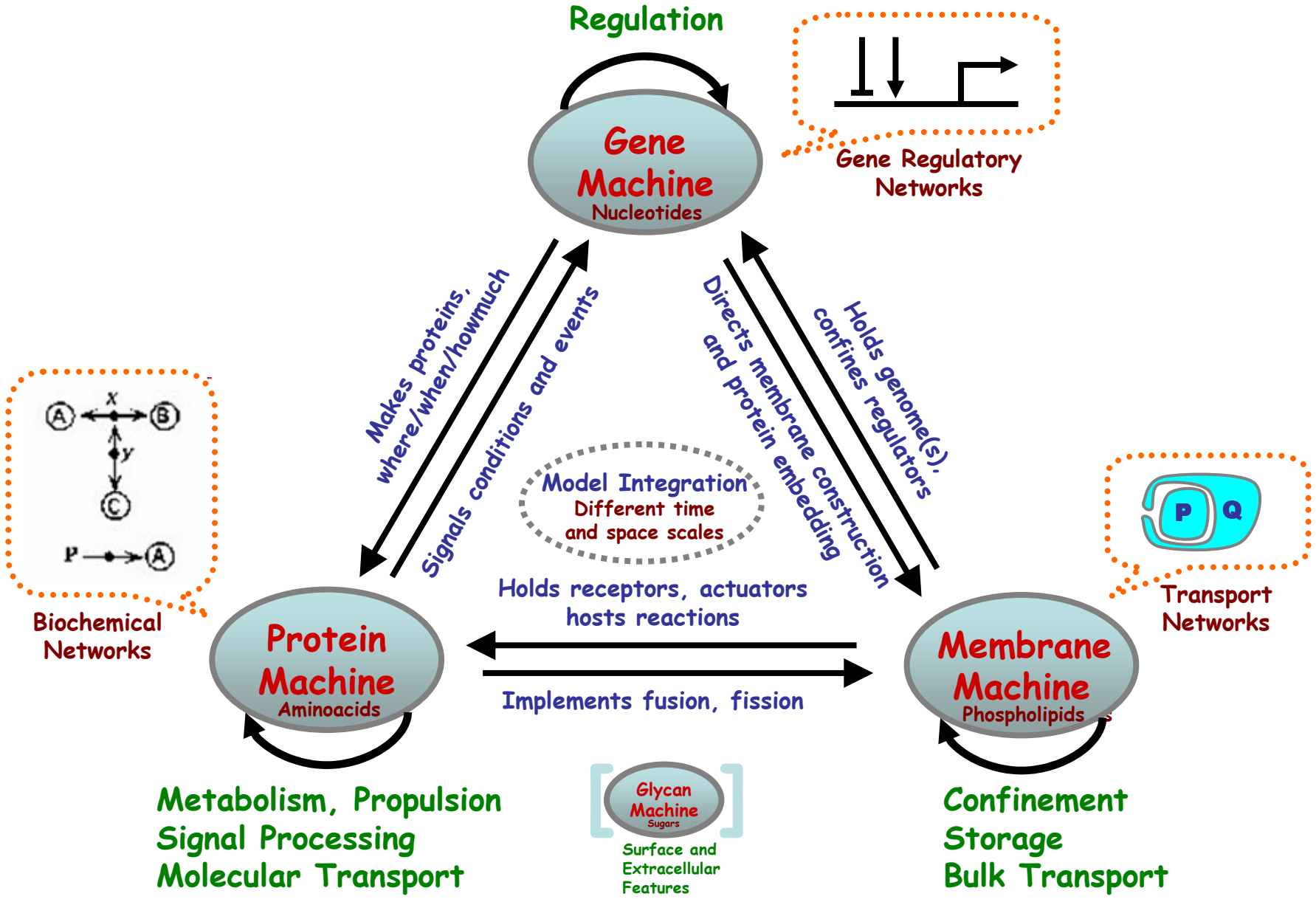
Adapted from: B.Alberts et al.
Molecular Biology of the Cell
third edition p.279.

LDL-Cholesterol Degradation



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

Abstract Machines of Systems Biology



Model Construction and Validation

Reactive Systems

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

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

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.

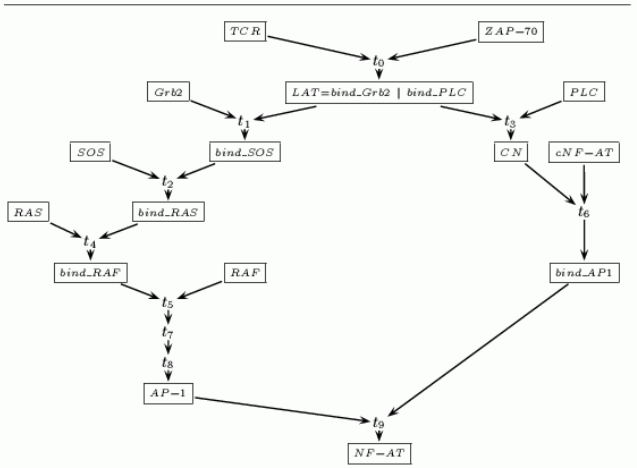


Fig. 2. A computation of *Sys*. For readability, the processes, enclosed in boxes, have no address. Causality (both on transitions and processes) is represented by the (Hasse diagram resulting from the) arrows; their absence makes it explicit concurrent activities.

- 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].

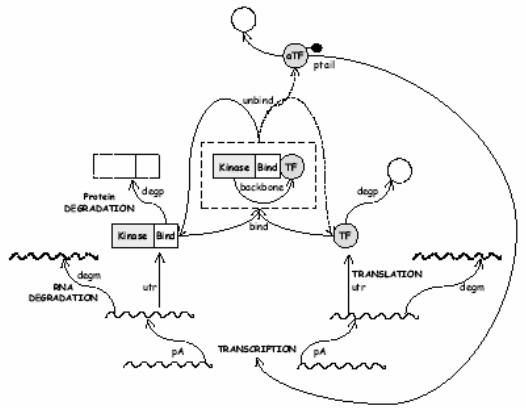


Fig. 1. Graphical presentation of Transcriptional Regulation by Positive Feedback [25].

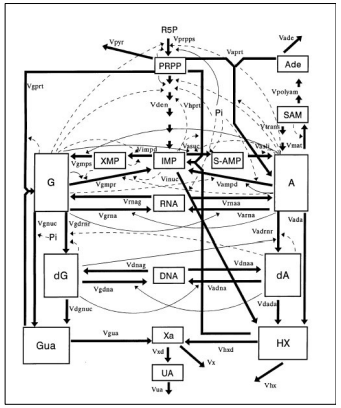
- Probabilistic Abstract Interpretation
  - [DiPierro Wicklicky].

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).

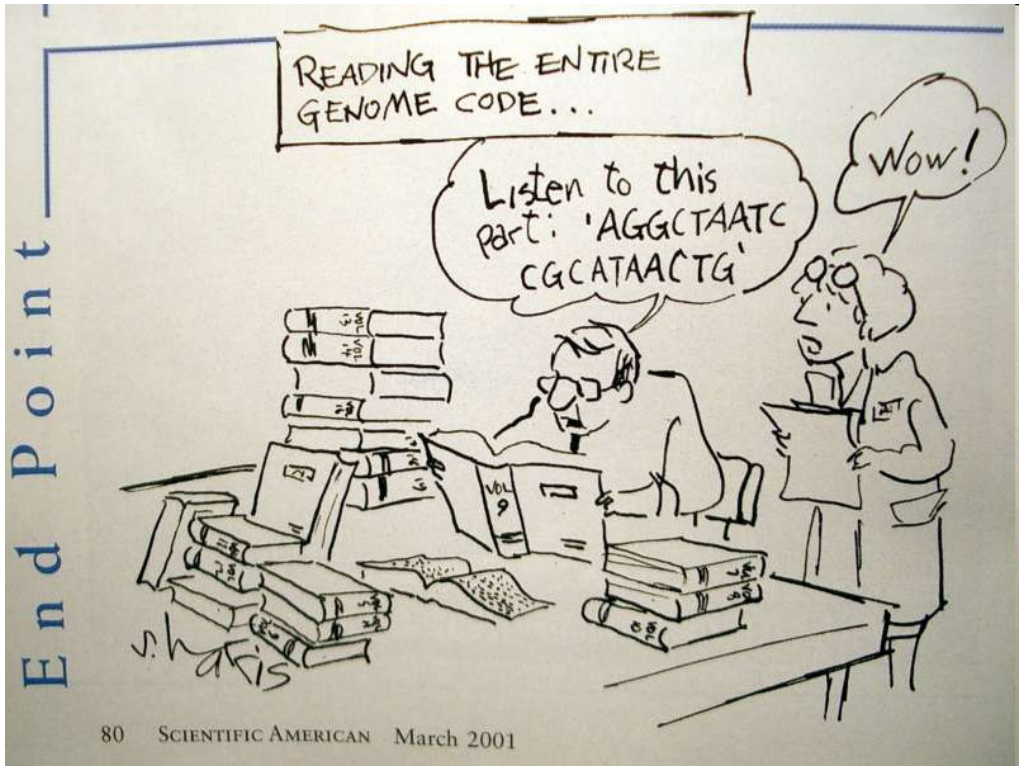
Disclaimers

- 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:

$\Rightarrow$ *Proc. Computational Methods in Systems Biology* [2003-2005]

Conclusions



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

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

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

E.Shapiro (Lecture Notes)

References

[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