Biological Networks in Stochastic π-Calculus

Luca Cardelli

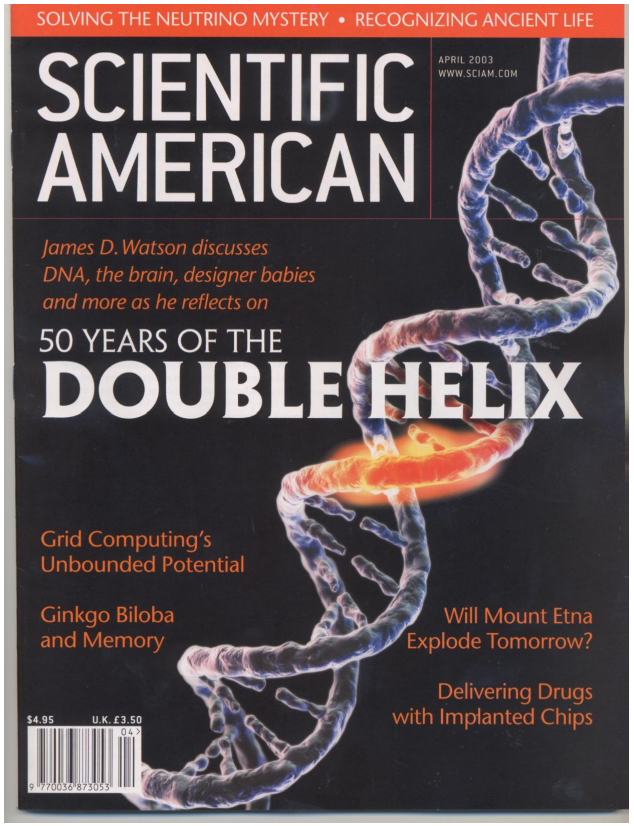
Microsoft Research
Cambridge UK

Imperial College 2005-02-17

www.luca.demon.co.uk

50 Years of Molecular Cell Biology

- How cells work:
  - DNA stores information
  - DNA instructs Ribosomes via RNA to assemble Proteins
  - Proteins (>10000) do things:
    - Process signals, activate DNA
    - Catalyze reactions to produce substances
    - Control energy production and consumption
  - Bootstrapping still a mystery
    - Happened a long time ago; not understood, not essential.



Towards Systems Biology

Biologists now understand many cellular components, but do not yet understand how "the system" works.

Molecular Biology: Understanding the components of living matter.

BioInformatics: Mining *-omics* "high-througput" whole-system data.

Systems Biology: Understanding the connectivity of the components.

Aim: Modeling biological *systems*

not as continuous systems (traditional)
but as reactive systems (information-processing)

Because they have some similar features:

Deep layering of abstractions.
Complex composition of simpler components.
Discrete (non-linear) transitions.
Digital coding of information.
Reactive information-driven behavior.
Very high degree of concurrency.
"Emergent behavior" (not obvious from part list).

Methods

Model Construction (writing things down precisely)

Studying the notations used in systems biology.

Formulating description languages, for various purposes.

Studying their kinetics (semantics).

Model Validation (using models for postdiction and prediction)

Stochastic Simulation

Stochastic = Quantitative concurrent semantics.
Based on compositional descriptions.

"Program" Analysis

Control flow analysis
Causality analysis

Modelchecking

Standard, Quantitative, Probabilistic

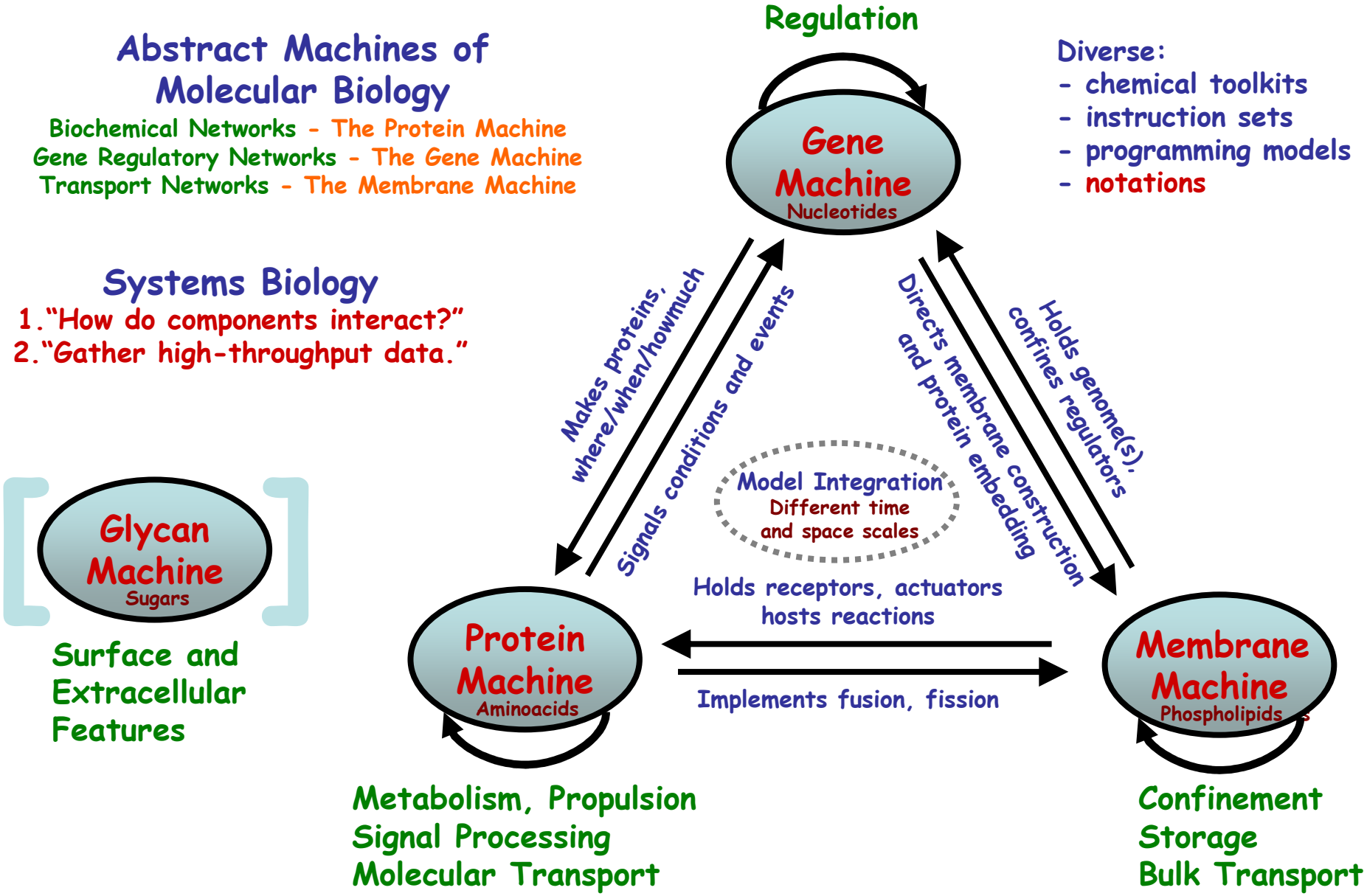
Functional Architecture of Cellular Systems

Abstract Machines of Molecular Biology

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

Systems Biology

1."How do components interact?"
2."Gather high-throughput data."



Stochastic π-calculus Executive Summary

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

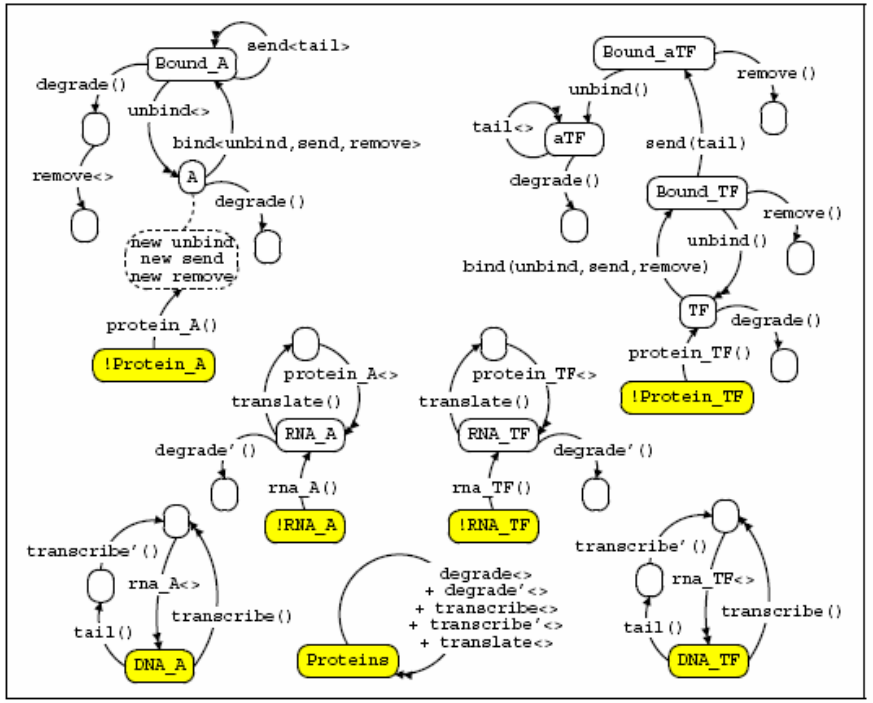


Figure 2. Regulating Gene Expression by Positive Feedback [9]

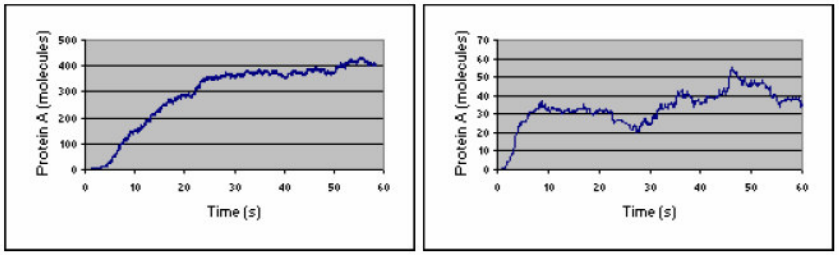


Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of TF

Regev-Shapiro: "Molecules as Computation"

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

π-calculus

Syntax

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

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

Structural congruence

Renaming of bound variables

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

Structural congruence laws

$$\begin{array}{rcll} P|Q & \equiv & Q|P & \text{commutativity of parallel composition} \\ (P|Q)|R & \equiv & P|(Q|R) & \text{associativity of parallel composition} \\ P+Q & \equiv & Q+P & \text{commutativity of summation} \\ (P+Q)+R & \equiv & P+(Q+R) & \text{associativity of summation} \\ (\text{new } x)0 & \equiv & 0 & \text{restriction of inert processes} \\ (\text{new } x)(\text{new } y)P & \equiv & (\text{new } y)(\text{new } x)P & \text{polyadic restriction} \\ ((\text{new } x)P)|Q) & \equiv & (\text{new } x)(P|Q) \quad \text{if } x \notin FN(Q) & \text{scope extrusion} \\ !P & \equiv & P|!P & \text{replication} \end{array}$$

Reaction rules

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

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

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

$$\frac{Q \equiv P \ P \rightarrow P' \ P' \equiv Q'}{Q \rightarrow Q'} \qquad \text{structural congruence (STRUCT)}$$

Syntax

Chemical Mixing

Reactions

Stochastic π-calculus

- Stochastic extension of p-calculus. [C.Priami]

  Associate a **rate** $r \in (0, \infty]$ of an **exponential distribution** to each activity a; it 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.

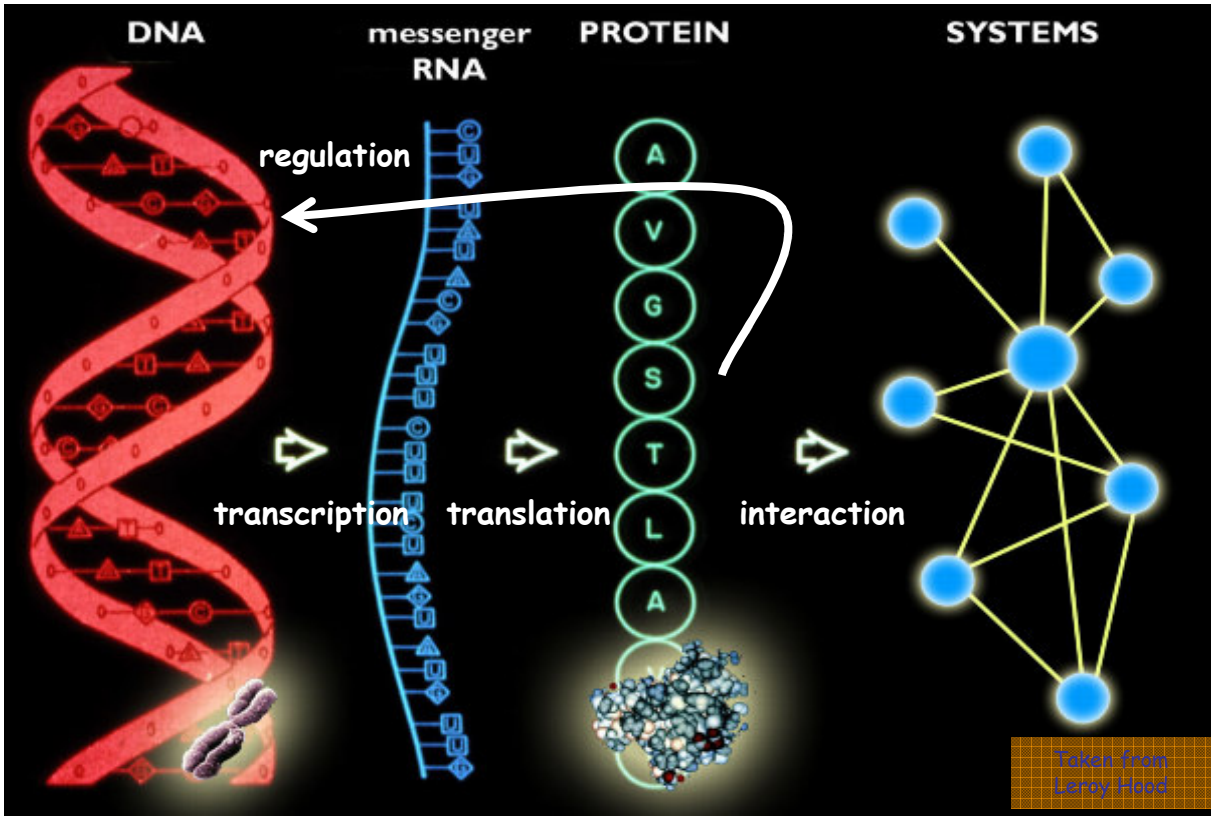
  Race condition is defined in a **probabilistic competitive** context: all the activities that are enabled in a state compete and the fastest one succeeds.

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

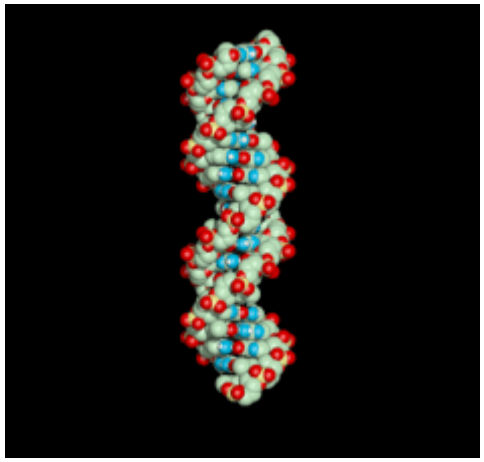
Gene Networks

The Gene Machine

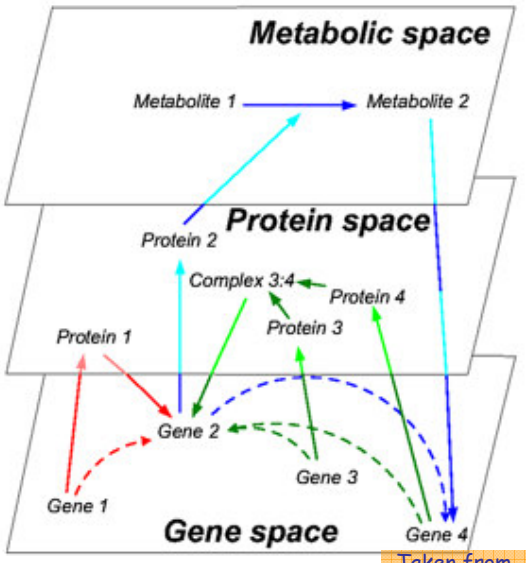
The "Central Dogma" of Molecular Biology



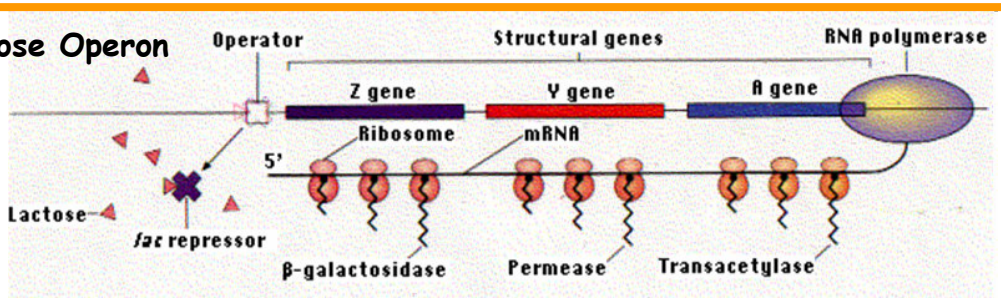
4-letter digital code | 4-letter digital code | 20-letter digital code | 50.000(?) shapes



DNA Tutorial

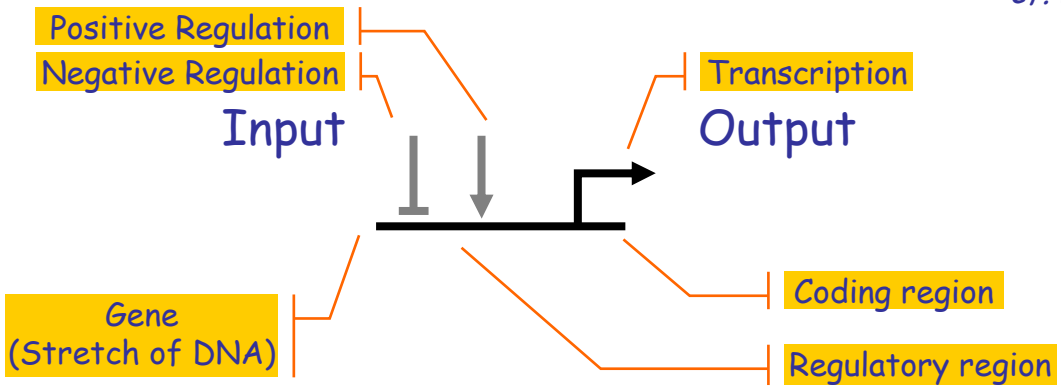


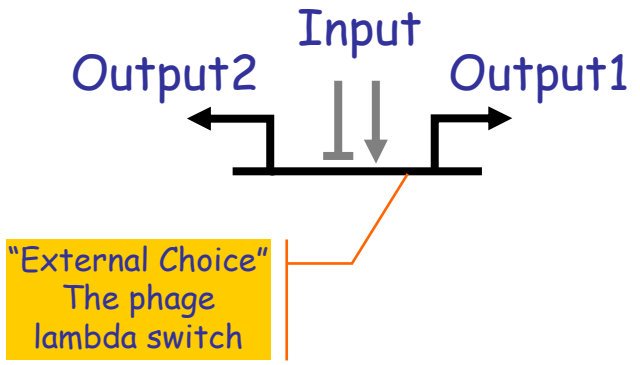
Lactose Operon



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).

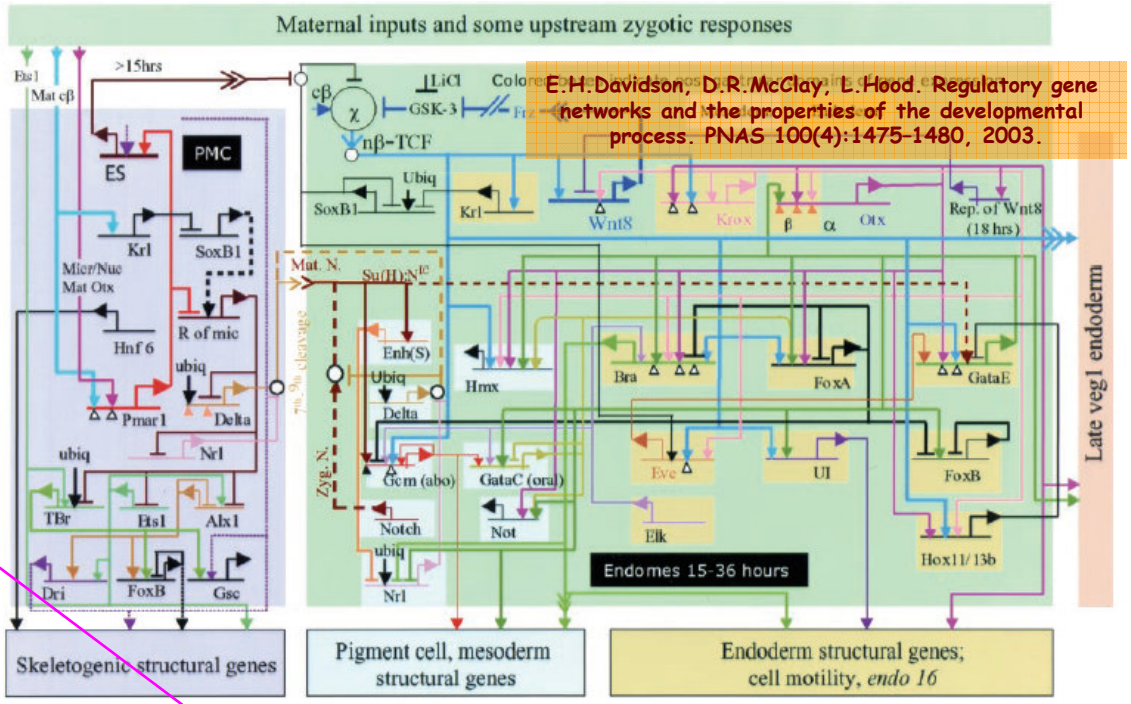
Human (and mammalian) Genome Size
3Gbp (Giga base pairs) 750MB @ 4bp/Byte (CD)
Non-repetitive: 1Gbp 250MB
In genes: 320Mbp 80MB
Coding: 160Mbp 40MB
Protein-coding genes: 30,000-40,000

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)

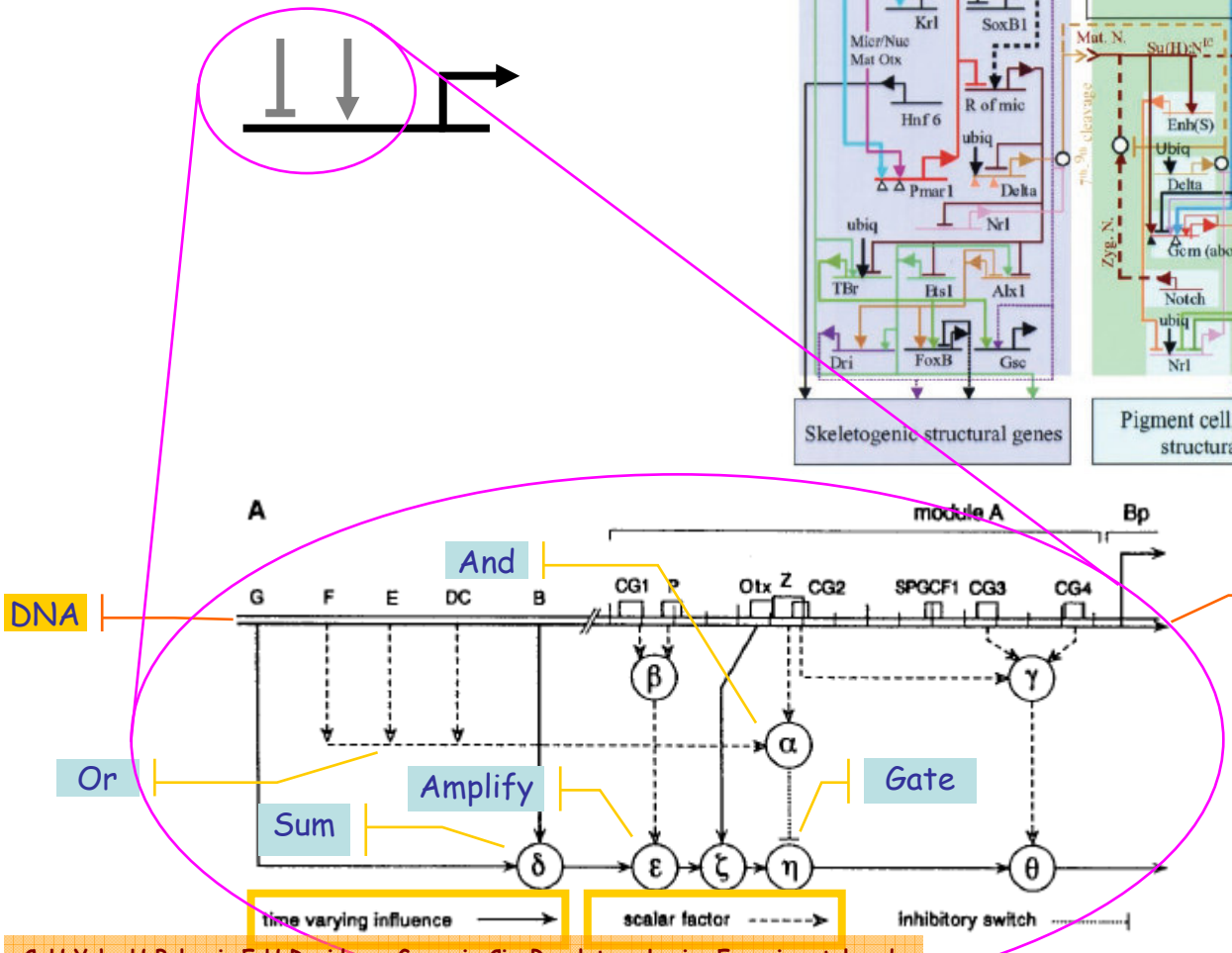
Gene Regulatory Networks

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

NetBuilder



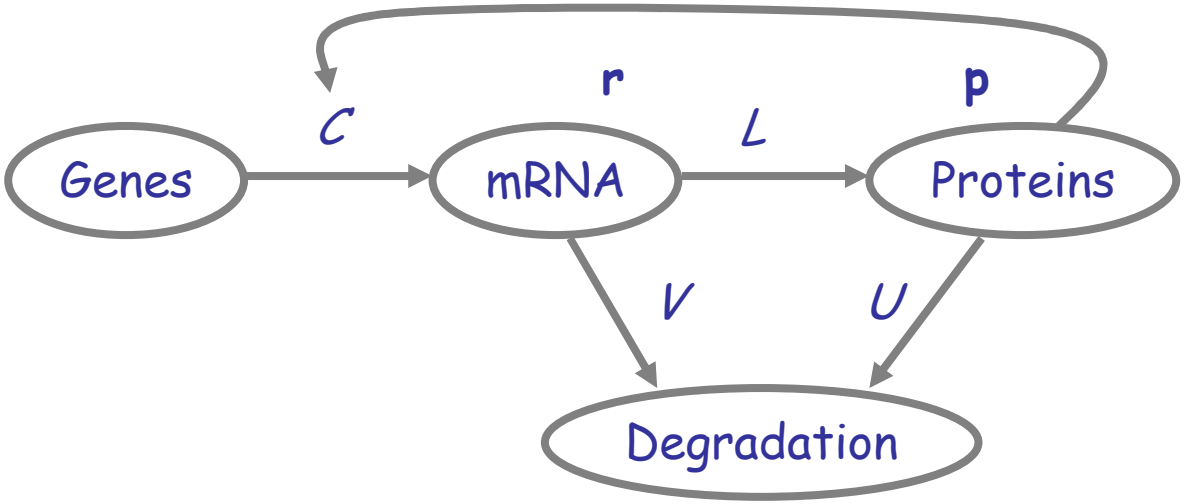
E.H.Davidson, D.R.McClay, L.Hood. Regulatory gene networks and the properties of the developmental process. PNAS 100(4):1475–1480, 2003.



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 Classical ODE Approach)

[Chen, He, Church]



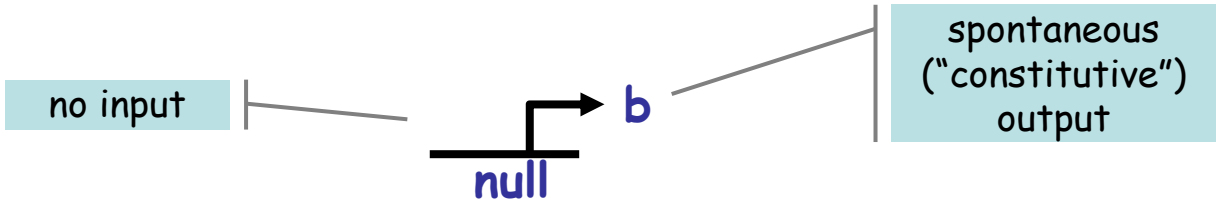
$$\frac{d\mathbf{r}}{dt} = f(\mathbf{p}) - V\mathbf{r}$$

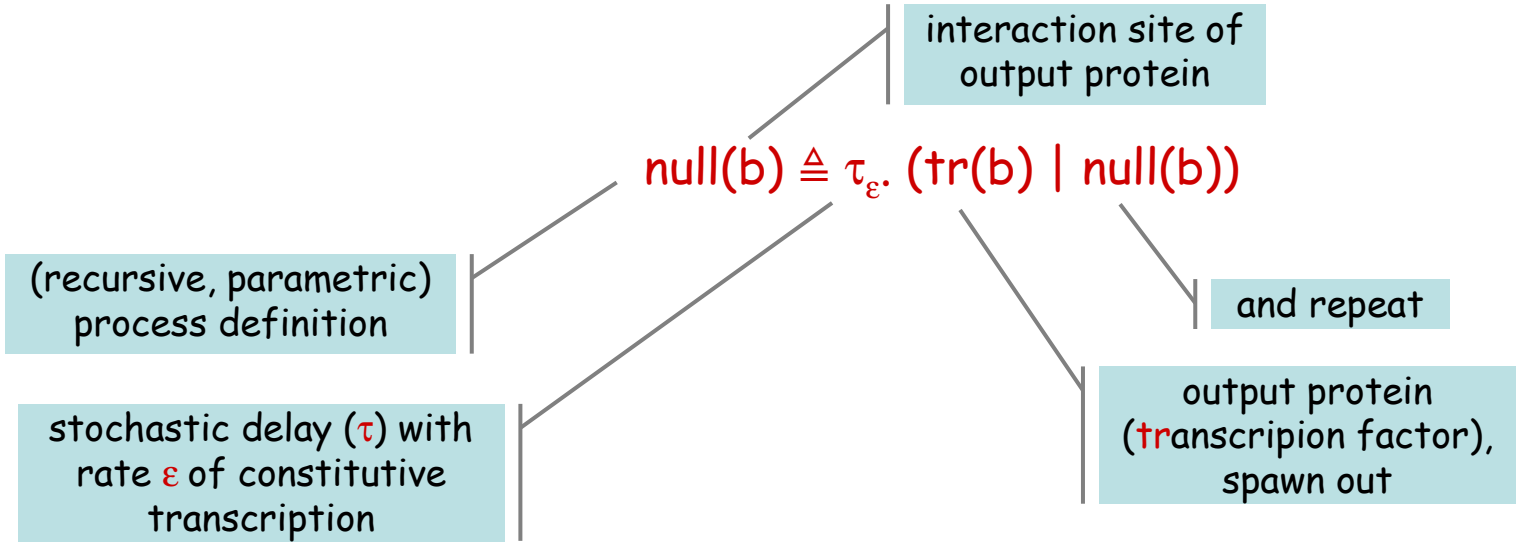
$$\frac{d\mathbf{p}}{dt} = L\mathbf{r} - U\mathbf{r}$$

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

$f(\mathbf{p})$ transcription functions:
(n-dim vector polynomials on **p**)

Nullary Gate

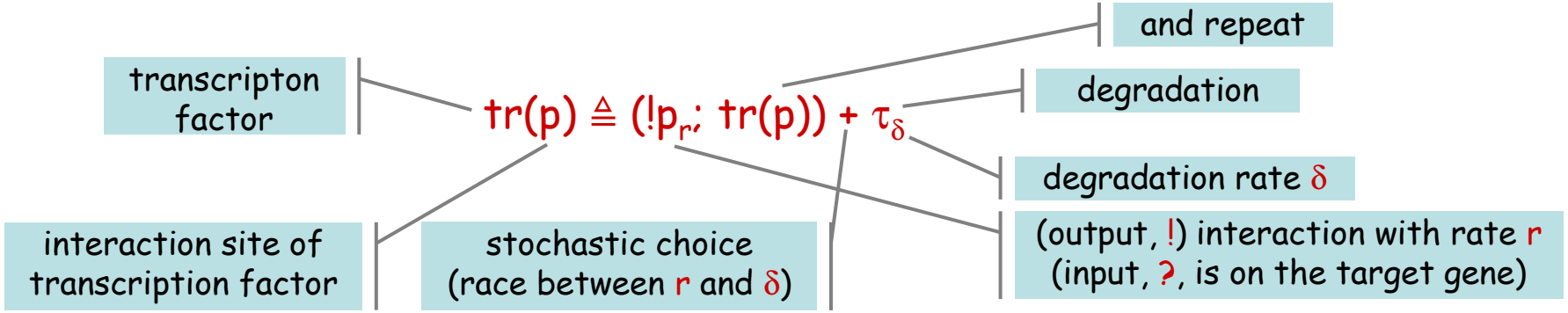




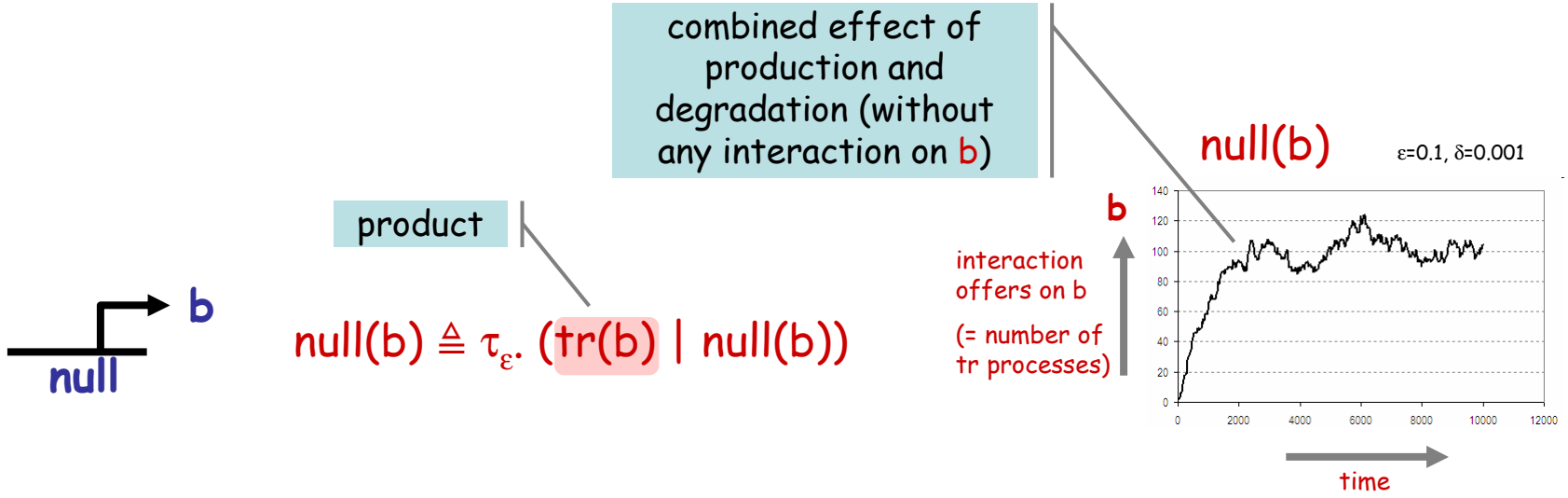
A stochastic rate r is always associated with each channel a_r (at creation time) and delay τ_r, but is often omitted when unambiguous.

Production and Degradation

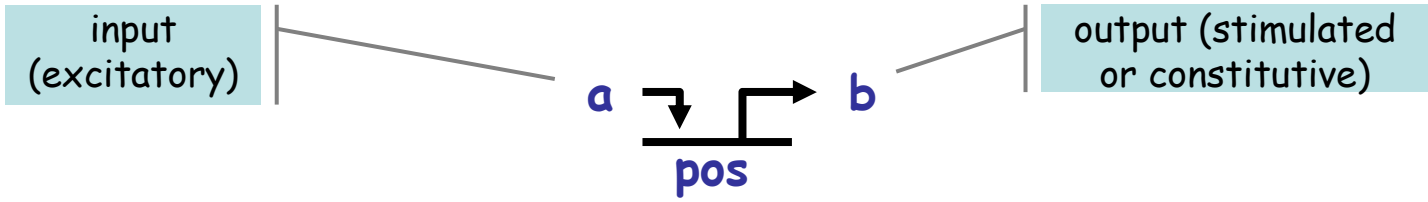
Degradation is extremely important and often deliberate; it changes unbounded growth into (roughly) stable signals.

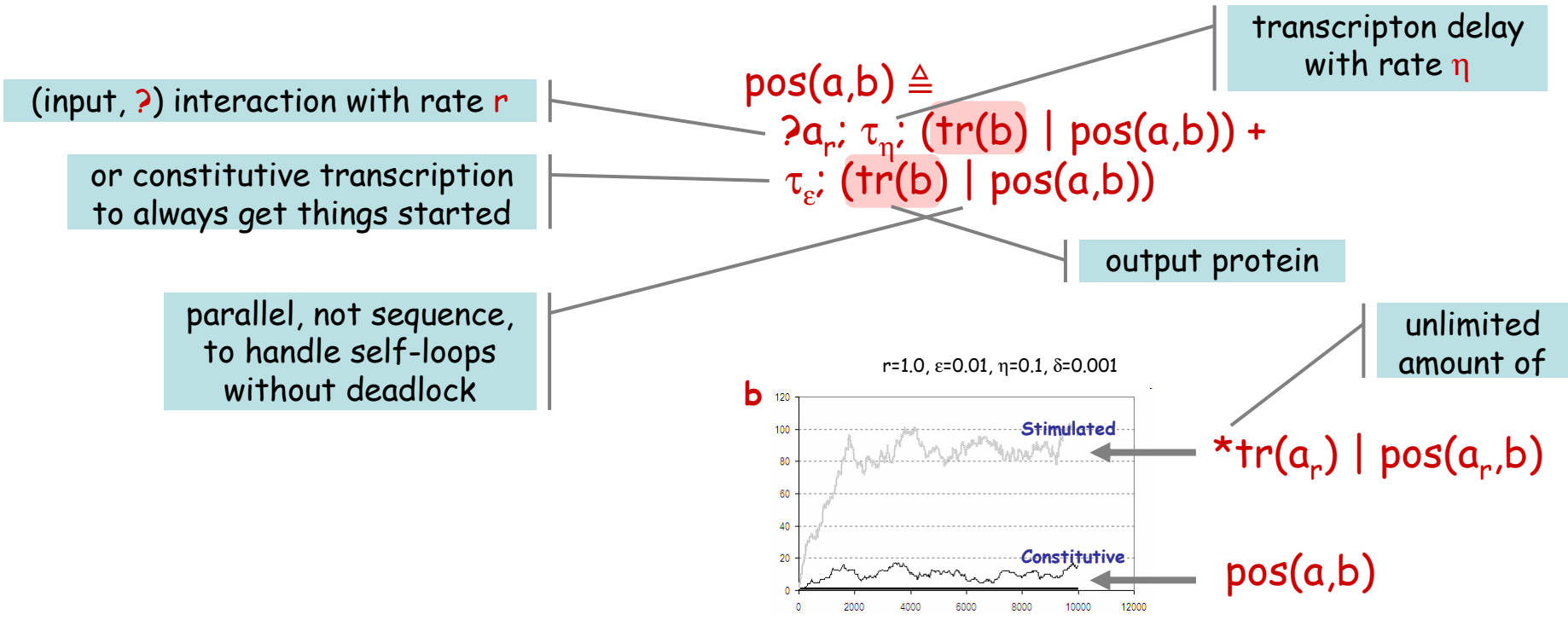


A transcription factor is a process (not a message or a channel): it has behavior such as interaction on p and degradation.

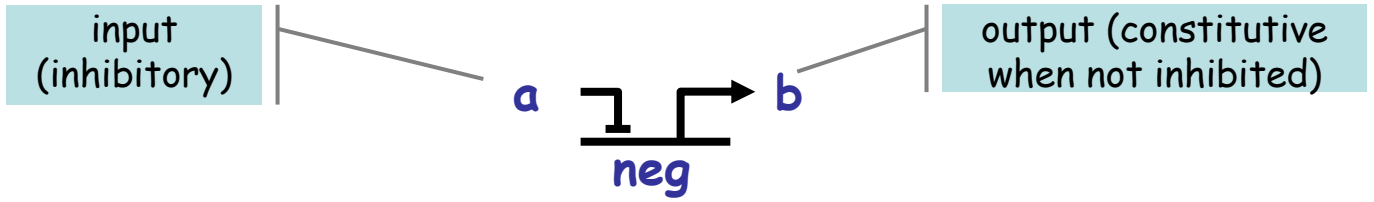


Unary Pos Gate





Unary Neg Gate

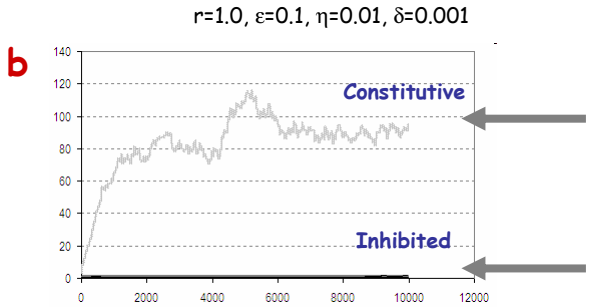


(input, ?) interaction with rate r

or constitutive transcription
to always get things started

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

inhibition delay
with rate η

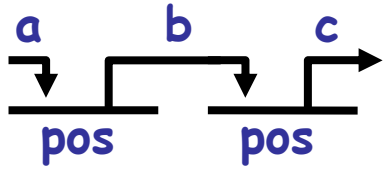


$neg(a_r,b)$

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

Signal Amplification

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



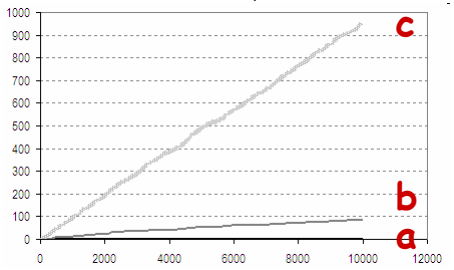
$pos(a,b) \triangleq$
$?a_r.\ \tau_\eta.\ (tr(b) \mid pos(a,b)) +$
$\tau_\varepsilon.\ (tr(b) \mid pos(a,b))$

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

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

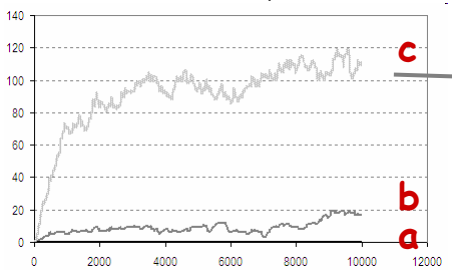
With little degradation

r=1.0, ε=0.01, η=0.1, δ=0.00001



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

r=1.0, ε=0.01, η=0.1, δ=0.001

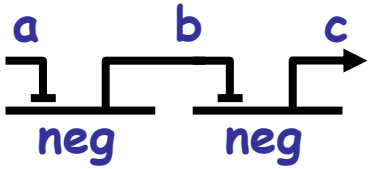


pos(a,b) | pos(b,c)

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

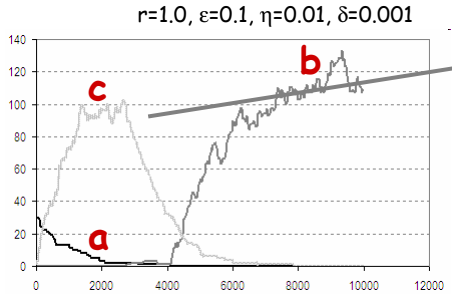
Signal Normalization

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



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

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

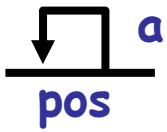


a non-zero input level, a, whether weak or strong, is renormalized to a standard level, c.

$30^*tr(a) \mid neg(a,b) \mid neg(b,c)$

Self Feedback Circuits

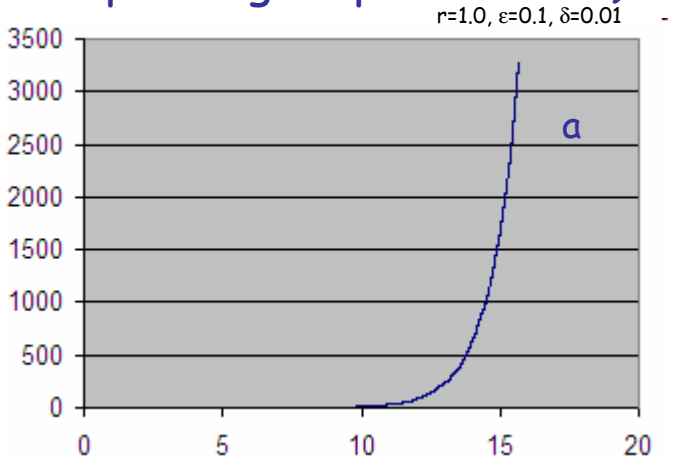
pos(a,a)



pos(a,b) ≜
  ?a_r; (tr(b) | pos(a,b)) +
  τ_ε; (tr(b) | pos(a,b))

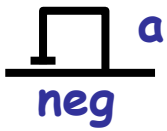
tr(p) ≜ (!p_r; tr(p)) + τ_δ

(Can overwhelm degradation, depending on parameters)



pos(a,a)

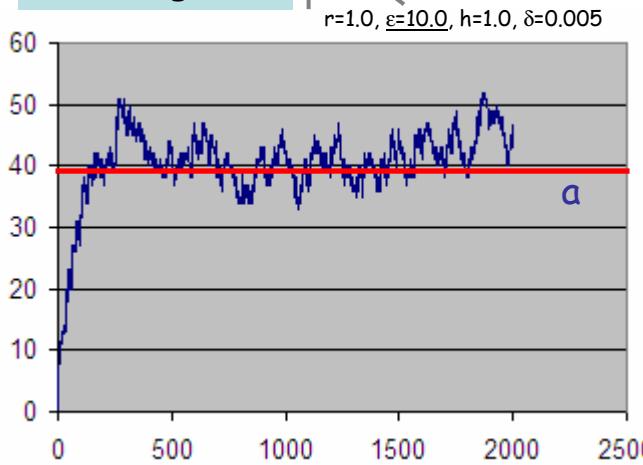
neg(a,a)



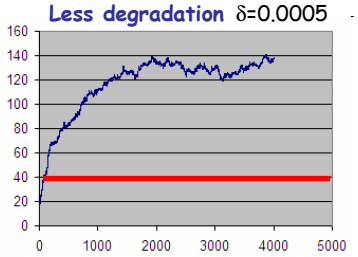
neg(a,b) ≜
  ?a_r; τ_h; neg(a,b) +
  τ_ε; (tr(b) | neg(a,b))

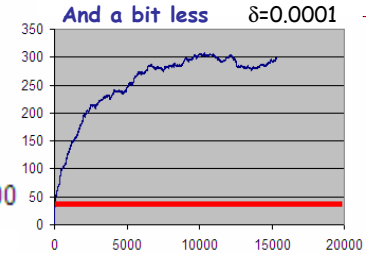
tr(p) ≜ (!p_r; tr(p)) + τ_δ

high, to raise the signal



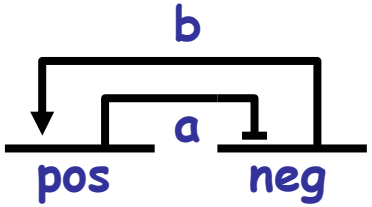
neg(a,a)



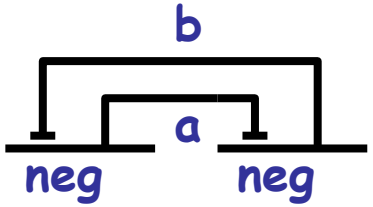


Two-gate Feedback Circuits

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

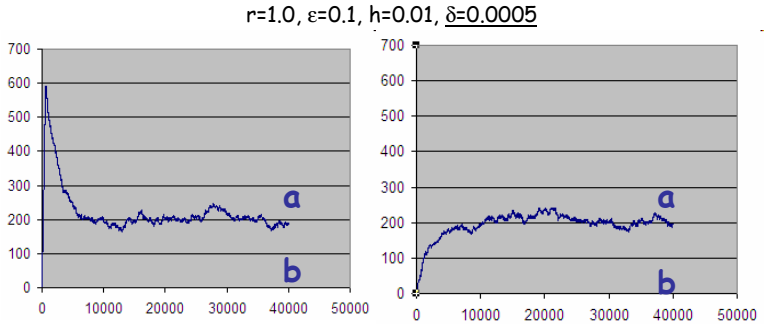


neg(b,a) |
neg(a,b)



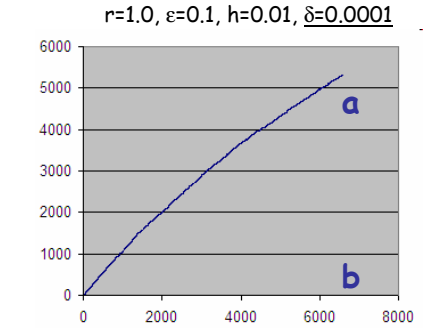
Monostable:

For some degradation rates is quite stable:



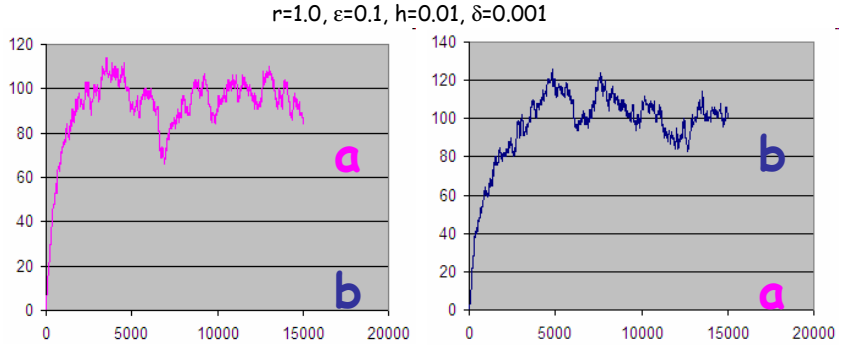
pos(b,a) | neg(a,b)

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

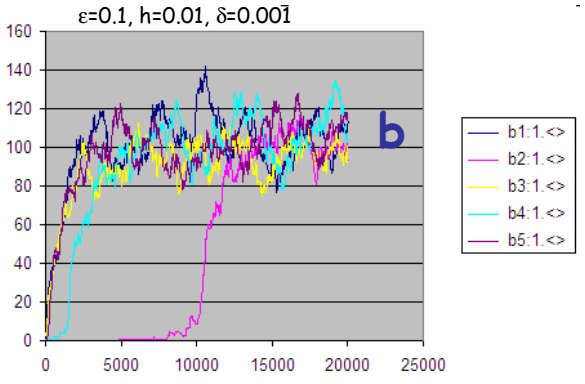


pos(b,a) | neg(a,b)

Bistable:



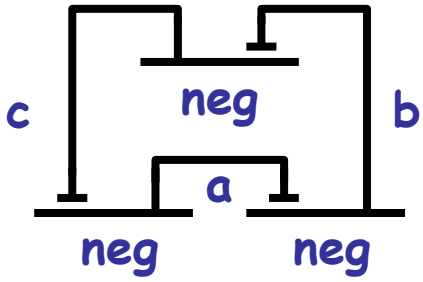
neg(b,a) | neg(a,b)



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)

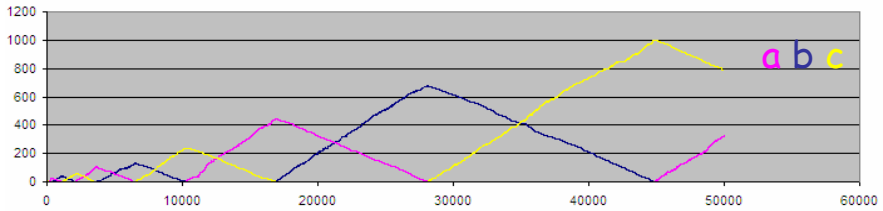


$neg(a,b) \triangleq$
$?a_r; \tau_h; neg(a,b) +$
$\tau_\varepsilon; (tr(b) \mid neg(a,b))$

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

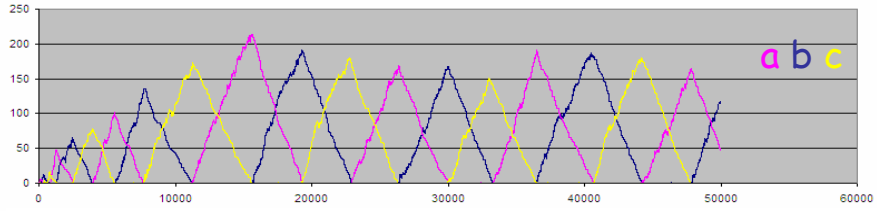
$tr(p) \triangleq !p_r$ — interact once and die otherwise stick around

r=1.0, ε=0.1, h=0.04



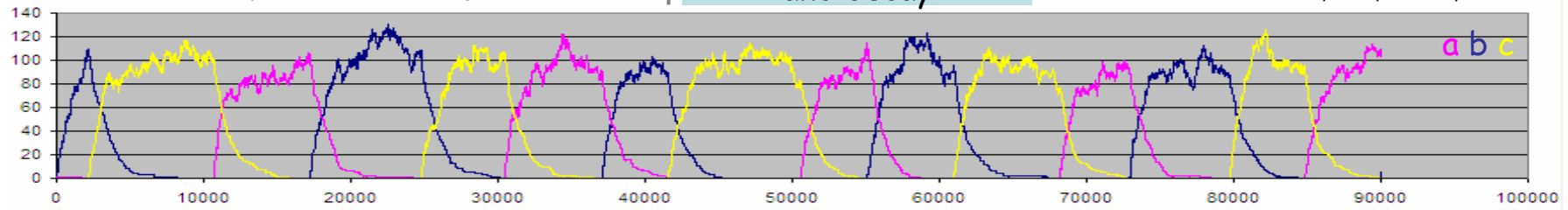
$tr(p) \triangleq !p_r + \tau_\delta$ — interact once and die otherwise decay

r=1.0, ε=0.1, h=0.04, δ=0.0001



$tr(p) \triangleq (!p_r; tr(p)) + \tau_\delta$ — interact many times and decay

r=1.0, ε=0.1, h=0.001, δ=0.001



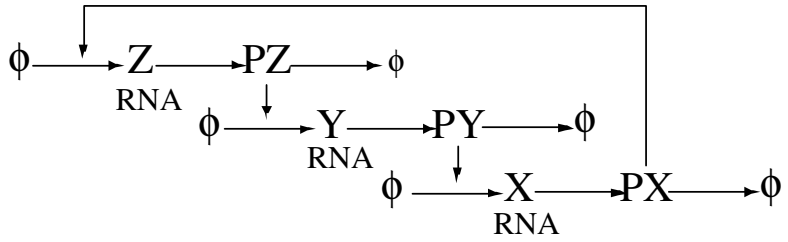
Subtle… at any point one gate is inhibited and the other two can fire constitutively. If one of them fires first, nothing really changes, but if the other one fires first, then the cycle progresses.

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))
```

Repressilator ODE Model and Simulation

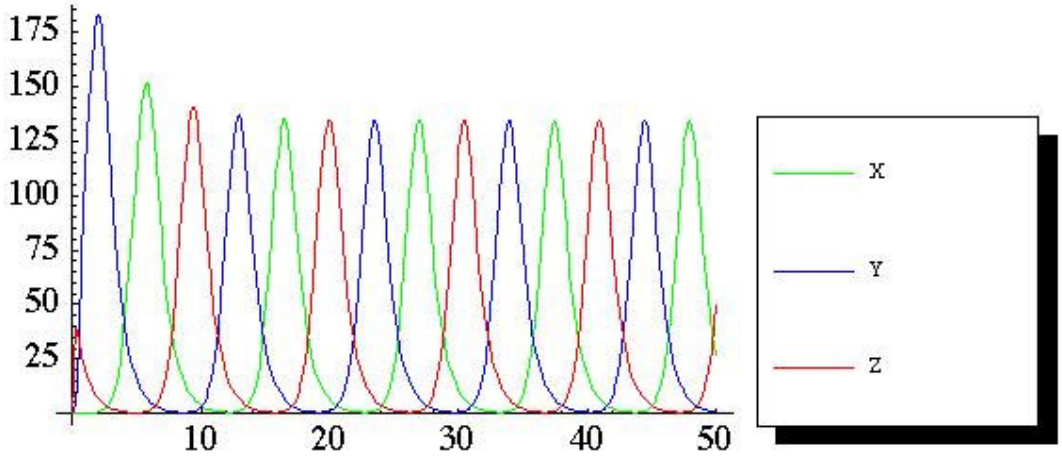
Bruce E Shapiro
Cellerator



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

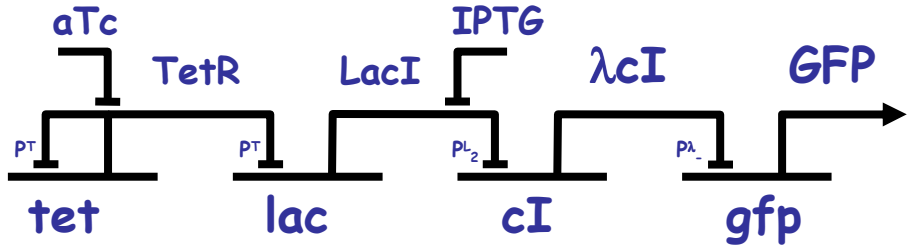
$$\frac{d[\mathrm{Y}]}{dt} = \alpha_0 + \frac{\alpha + \alpha_1 [\mathrm{PZ}]^n}{K^n + [\mathrm{PZ}]^n} - k[\mathrm{Y}], \quad \frac{d[\mathrm{PY}]}{dt} = \beta\{[\mathrm{Y}] - [\mathrm{PY}]\}$$

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



Guet et al.: D038/lac⁻

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

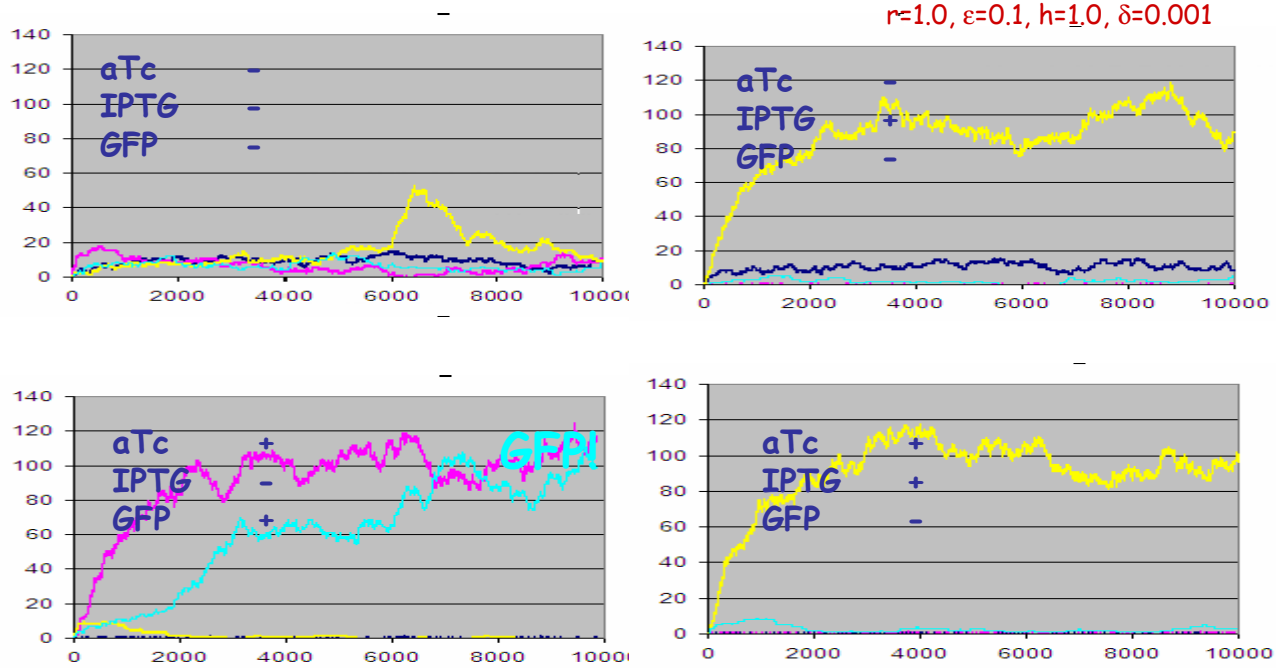


experiment:

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

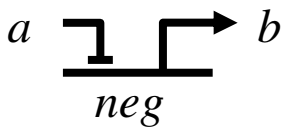


We can model an inducer like aTc as something that competes for the transcription factor.

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

Neg Gate Signal Response

Neg Gate



inhibitory input

constitutive transcription

$neg(a,b) =$

$?a;\ \tau_\eta;\ neg(a,b) +$

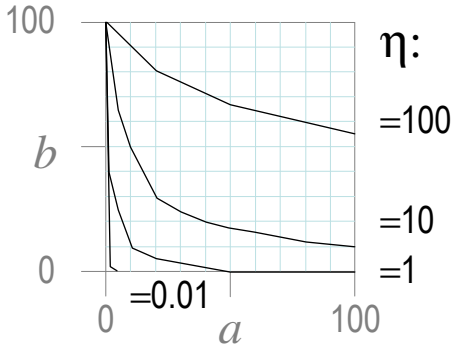
$\tau_\varepsilon;\ (tr(b) \mid neg(a,b))$

inhibition delay

back to initial state

Gate Response

($\varepsilon = 0.1$, $\delta = 0.001$)



transcription factor

$tr(b) =$

binding → $!b;\ tr(b) +$

delay → τ_δ

back to initial state

choice

degradation

η=1 : b~100/a (at the fixpoint) matches Alon's numbers

η=0.01 : b~1/a is the self-feedback instability point

iη=10 : b~900/a

hence b ~ 100*η/a

Protein Networks

MAPK Cascade - Huang&Ferrell

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.

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.

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

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

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

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

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

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

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

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

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

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

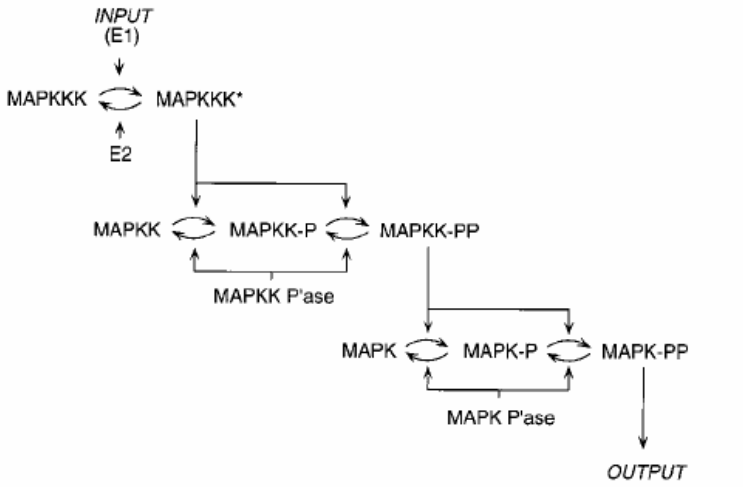


FIG. 1. Schematic view of the MAPK cascade. Activation of MAPK depends upon the phosphorylation of two conserved sites [Thr-183 and Tyr-185 in rat p42 MAPK/Erk2 (4, 5)]. Full activation of MAPKK also requires phosphorylation of two sites [Ser-218 and Ser-222 in mouse Mek-1/MKK1 (6–10)]. Detailed mechanisms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

As 18 Ordinary Differential Equations

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

One for each species (8) and complex (10) but not for constant concentration enzymes (4)

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

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

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

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

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

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

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

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

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

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

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

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

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

… Plus 7 conservation equations

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

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

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

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

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

Each molecule in exactly one state

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

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

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

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

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

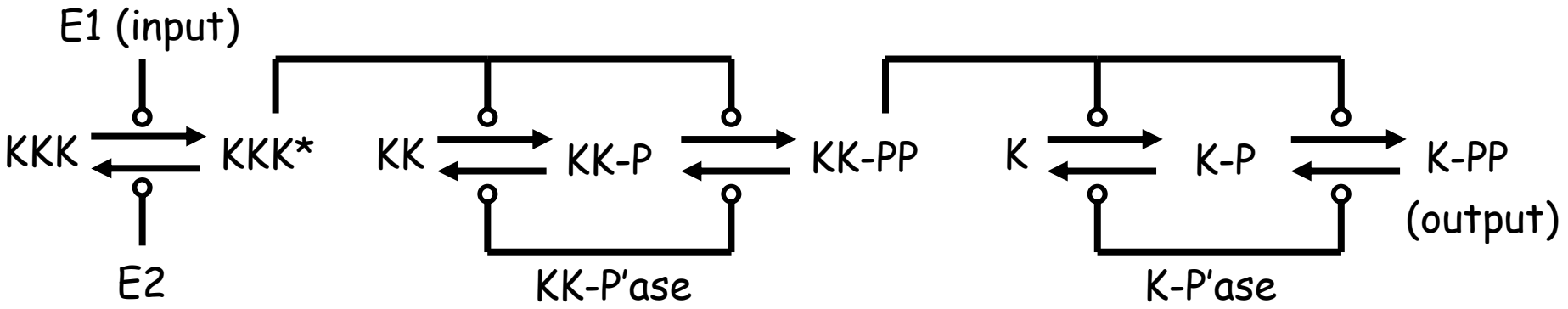
$$[\text{KK P'ase}_{tot}] = [\text{KK P'ase}] + [\text{KK P'ase}\cdot\text{KK-P}] + [KK\,P'ase\cdot KK\text{-}PP] \quad [33]$$

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

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

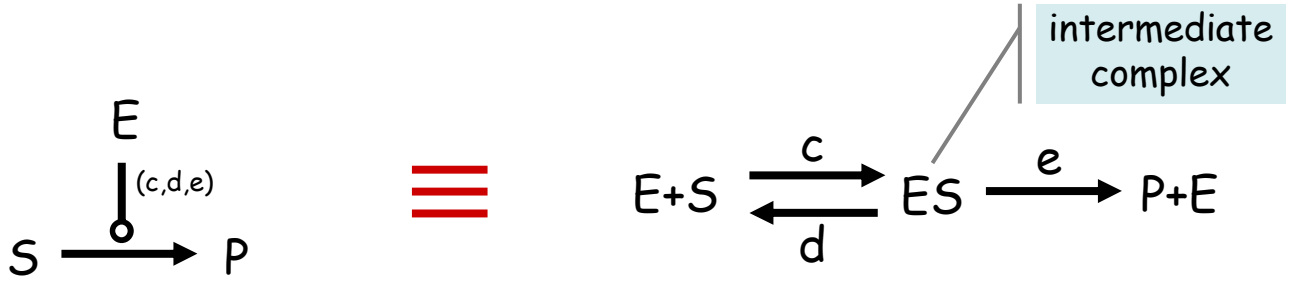
These equations were solved numerically using the Runge–Kutta-based NDSolve algorithm in Mathematica (Wolfram Research, Champaign, IL). An annotated copy of the Mathematica code for the MAPK cascade rate equations can be obtained from J.E.F.

The Circuit

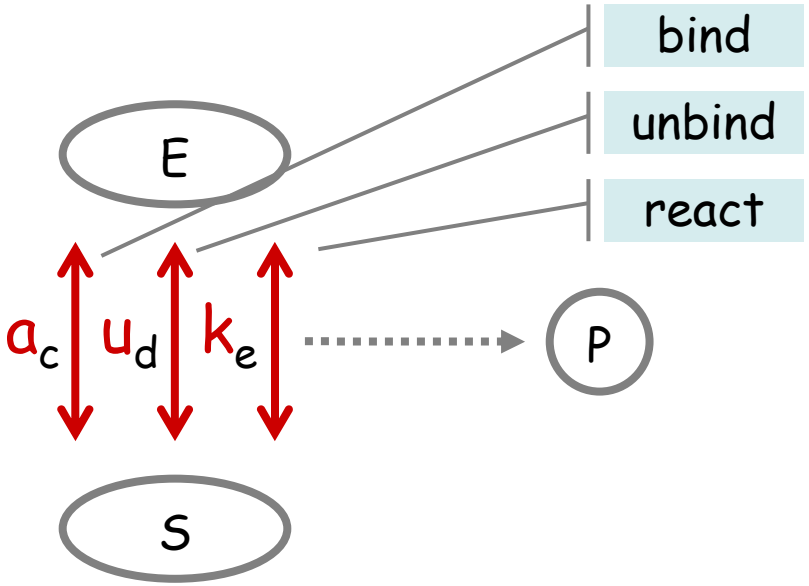


Enzymatic Reactions

Reaction View



Interaction View



private binding between one S and one E molecule

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

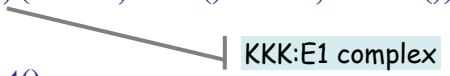
bind unbind react

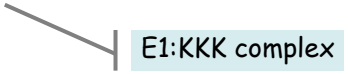
$E() \triangleq ?a_c(u); (?u_d; E() + ?k_e; E())$

$P() \triangleq$ ...

MAPK Cascade in SPiM

```
let KKK() =
   (new u1@d1:Release
    !a1(u1); (do !u1;KKK() or !k1;KKKst()))          [1]substrate
and KKKst() =
   (new u2@d2:Release
    do !a2(u2); (do !u2;KKKst() or !k2;KKK())        [2]substrate
    or ?a3(u3); (do ?u3;KKKst() or ?k3;KKKst())      [3]kinase
    or ?a5(u5); (do ?u5;KKKst() or ?k5;KKKst()))     [5]kinase

let E1() =
   ?a1(u1); (do ?u1;E1() or ?k1;E1())                [1]enzyme
let E2() =
   ?a2(u2); (do ?u2;E2() or ?k2;E2())                [2]enzyme

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

and KK_P() =
   (new u4@d4:Release  new u5@d5:Release
    do !a4(u4); (do !u4;KK_P() or !k4;KK())          [4]substrate
    or !a5(u5); (do !u5;KK_P() or !k5;KK_PP()))      [5]substrate

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

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

let K() =
   (new u7@d7:Release
    !a7(u7); (do !u7;K() or !k7;K_P()))              [7]substrate

and K_P() =
   (new u8@d8:Release  new u9@d9:Release
    do !a8(u8); (do !u8;K_P() or !k8;K())            [8]substrate
    or !a9(u9); (do !u9;K_P() or !k9;K_PP()))        [9]substrate

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

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

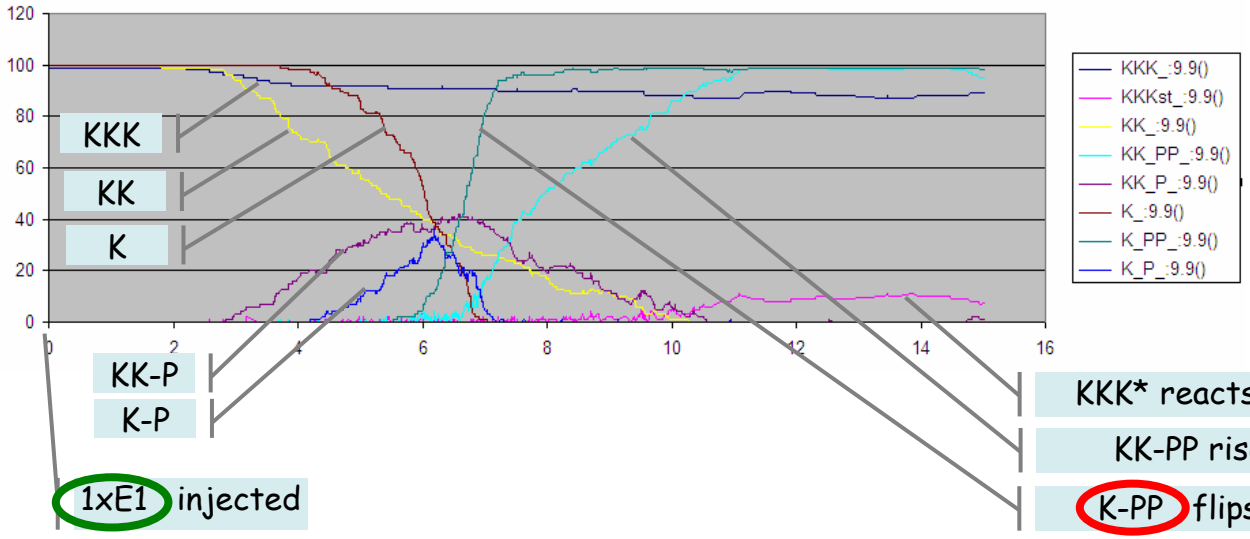


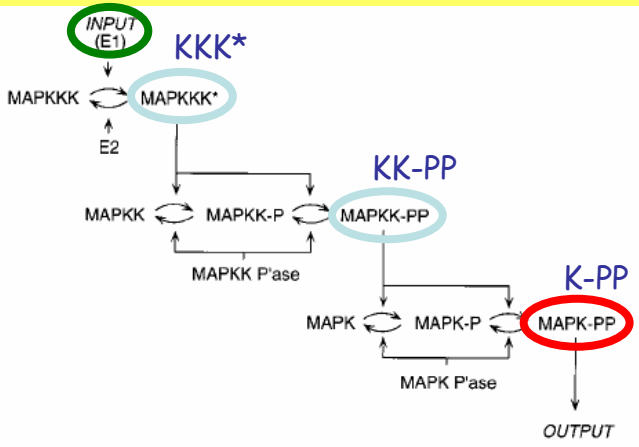


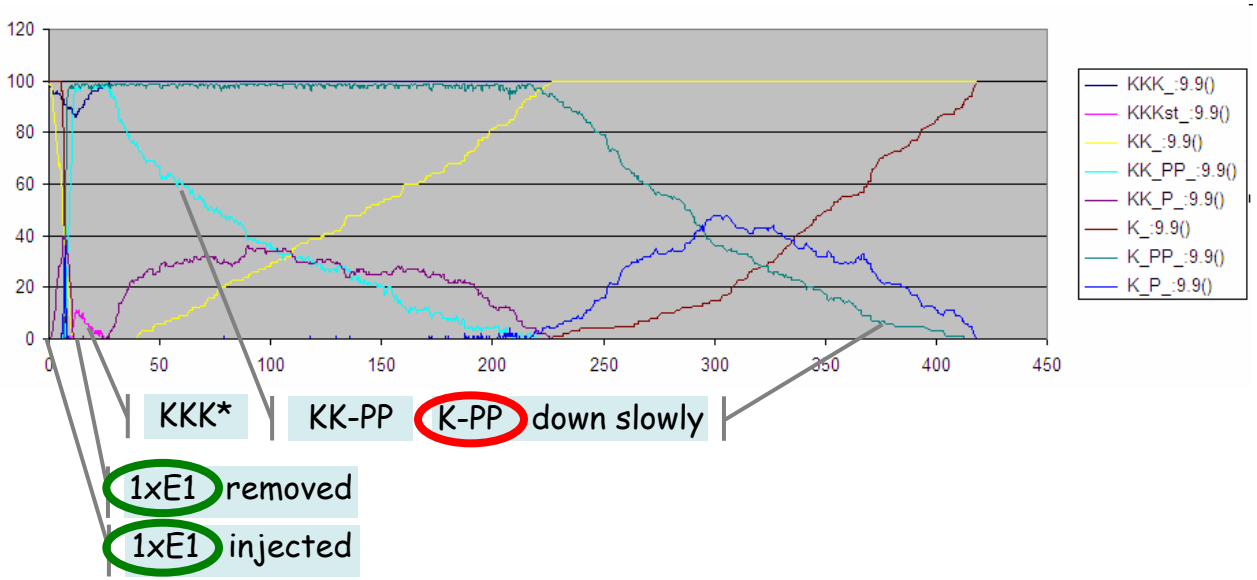
… globals

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

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

…

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

MAPK Cascade Simulation in SPiM



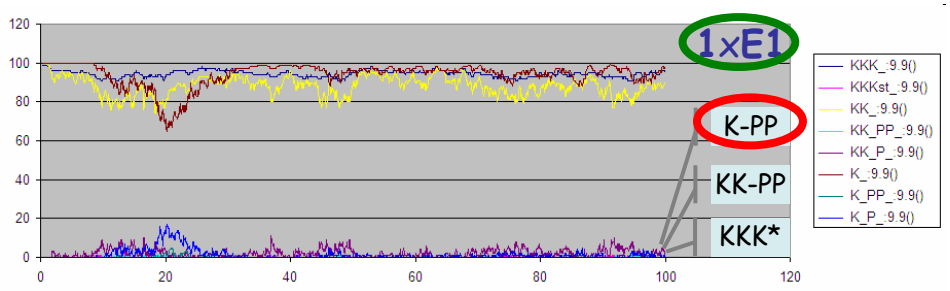


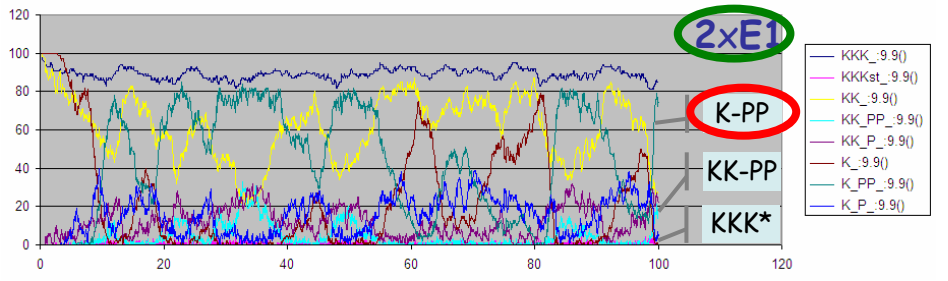


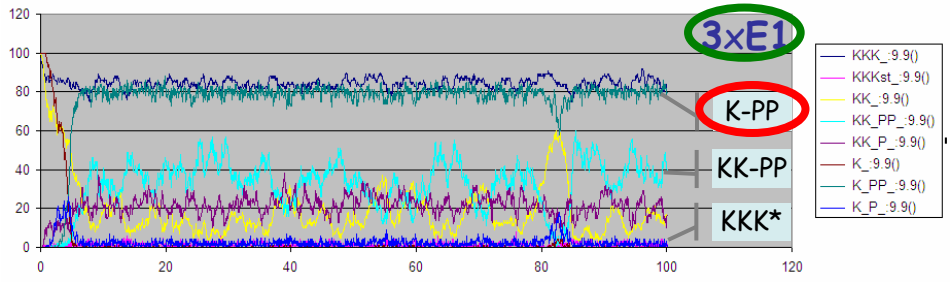
All coefficients 1.0 !!!
100xKKK, 100xKK, 100xK,
1xE2, 1xKKPse, 1xKPse.

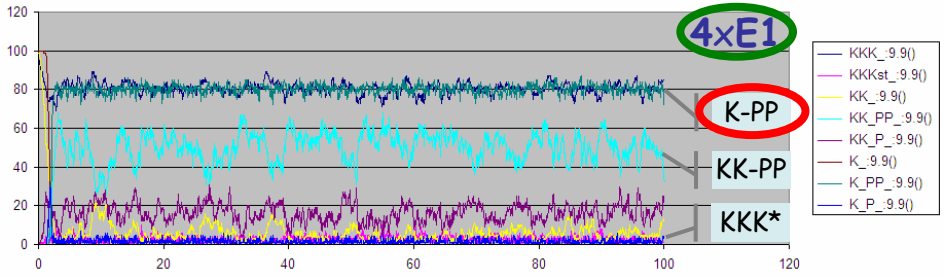
Input is **1xE1**.
Output is **100xK-PP**
(ultrasensitivity).

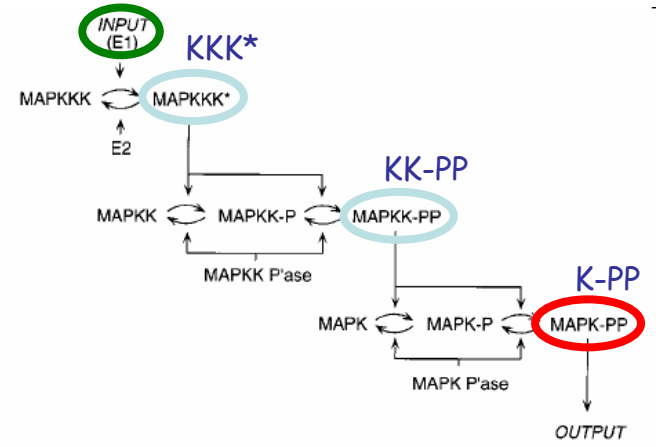
MAPK Cascade Simulation in SPiM



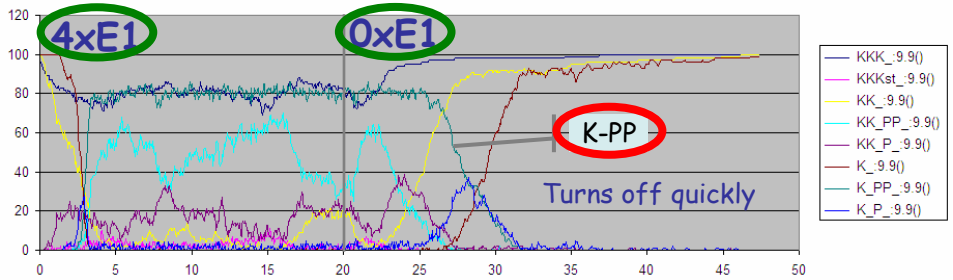








All coefficients 1.0 !!!
100xKKK, 100xKK, 100xK,
10xE2, 10xKKPse, 10xKPse.
(so 1xE1 is no longer sufficient to produce an output)

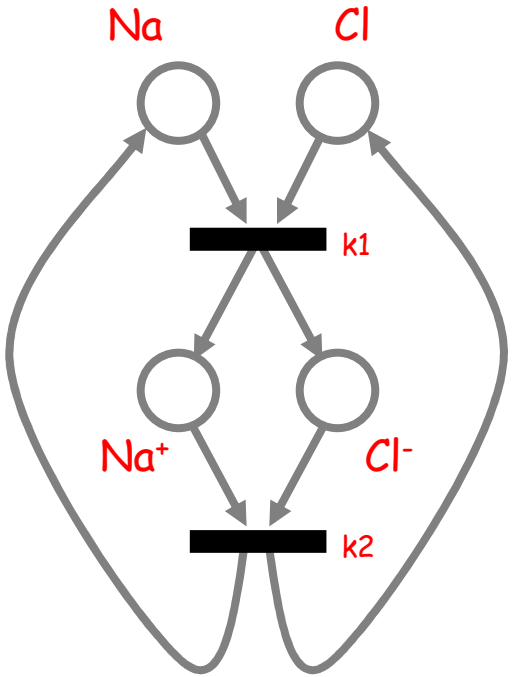


Scaling up: ODE vs Process Descriptions

Chemistry vs. π-calculus

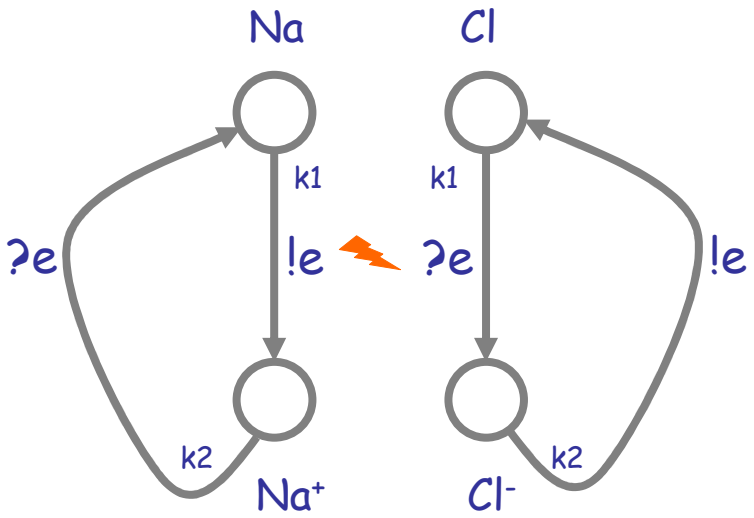
A process calculus (chemistry, or SBML)

$$Na + Cl \rightarrow_{k1} Na^+ + Cl^-$$
$$Na^+ + Cl^- \rightarrow_{k2} Na + Cl$$



(Can be converted to a CTMC)

A compositional graphical representation, and the corresponding calculus.



(Can be converted to a CTMC)

The same "model"

$$Na = !e_{k1}; \overbrace{?e_{k2}; Na}^{Na^+}$$
$$Cl = \overbrace{?e_{k1}; !e_{k2}; Cl}^{Cl^-}$$

A different process calculus (π)

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

From Reactions to ODE's

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

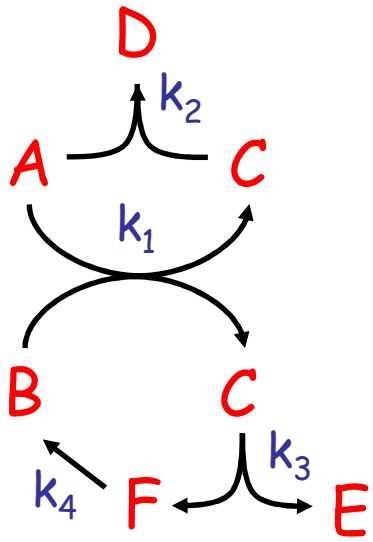
reactions

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

species

x

Stoichiometric Matrix



Concentration changes

Stoichiometric matrix

Rate laws

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

Read the concentration changes from the rows

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

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

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

$d[D]/dt = v_2$

$d[E]/dt = v_3$

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

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

Read the rate laws from the columns

$v_i(x, e_i, k_i)$

	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 Reactions to Processes

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

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

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

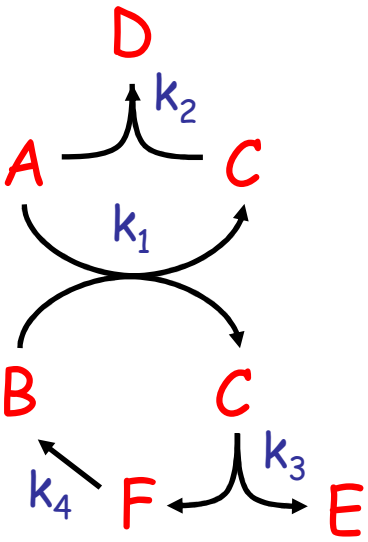
r_4: $F \rightarrow_{k_4} B$

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



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

Read the process interactions from the rows

(Rate laws are implicit in stochastic semantics)

From Reactions to (join)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 in the columns

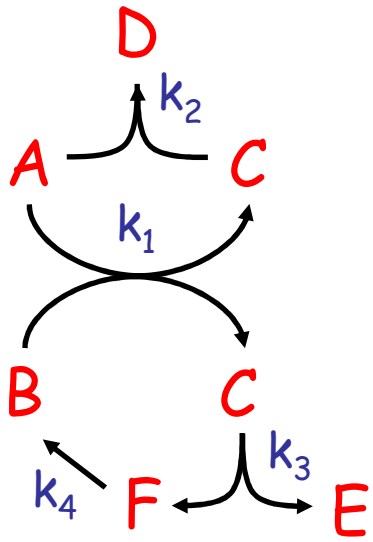
interactions

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

v

x



Would support arbitrary n-ary reactions.

Read the species from the rows

$A = !v_1a + !v_2a$

$B = !v_1b$

$C = !v_2c + !v_3c$

$D = 0$

$E = 0$

$F = !v_4f$

Read the reactions from the columns

Stochastic join calculus ?!?

$?v_1a \,\&\, ?v_1b \rightarrow_{k_1} C|C$

$?v_1a \,\&\, ?v_2c \rightarrow_{k_2} D$

$?v_3c \rightarrow_{k_3} E|F$

$?v_4f \rightarrow_{k_4} B$

Not What We Want

- Stoichiometric matrices
  - It is standard to go from chemical equations to ODE's via stoichiometric matrices.
  - It is possible to go from chemical equations to processes via stoichiometric matrices.
- But there is a better way:
  - Stoichiometric matrices blow-up exponentially for biochemical systems (unlike for ordinary chemical systems) because *proteins have combinatorial state.*
  - We should describe biochemical systems compositionally without going through stochiometric matrices (and hence without ODE's).

Complexes: From Reactions to ODE's

n domains: A, B, C

2n domain reactions:
$A \leftrightharpoons A_p$
$B \leftrightharpoons B_p$
$C \leftrightharpoons C_p$

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):
$ABC \leftrightharpoons A_pBC$
$ABC \leftrightharpoons AB_pC$
$ABC \leftrightharpoons ABC_p$
$A_pBC \leftrightharpoons A_pB_pC$
$A_pBC \leftrightharpoons A_pBC_p$
$AB_pC \leftrightharpoons A_pB_pC$
$AB_pC \leftrightharpoons AB_pC_p$
$ABC_p \leftrightharpoons A_pBC_p$
$ABC_p \leftrightharpoons AB_pC_p$
$A_pB_pC \leftrightharpoons A_pB_pC_p$
$A_pBC_p \leftrightharpoons A_pB_pC_p$
$AB_pC_p \leftrightharpoons A_pB_pC_p$

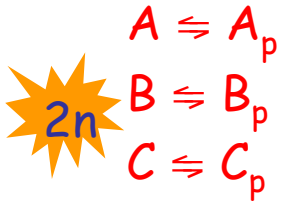
The matrix is sparse, so the corresponding ODE system is not dense, but it still has 2^n equations, one per species, plus conservation equations ([ABC]+[A_pBC]=constant, etc.).

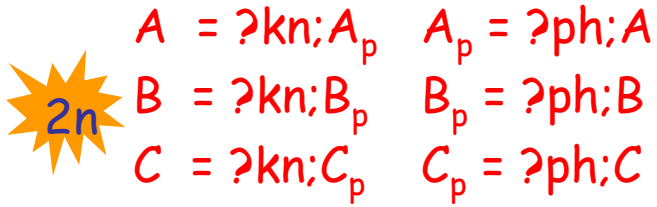
Stoichiometric Matrix

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: From Reactions to Processes

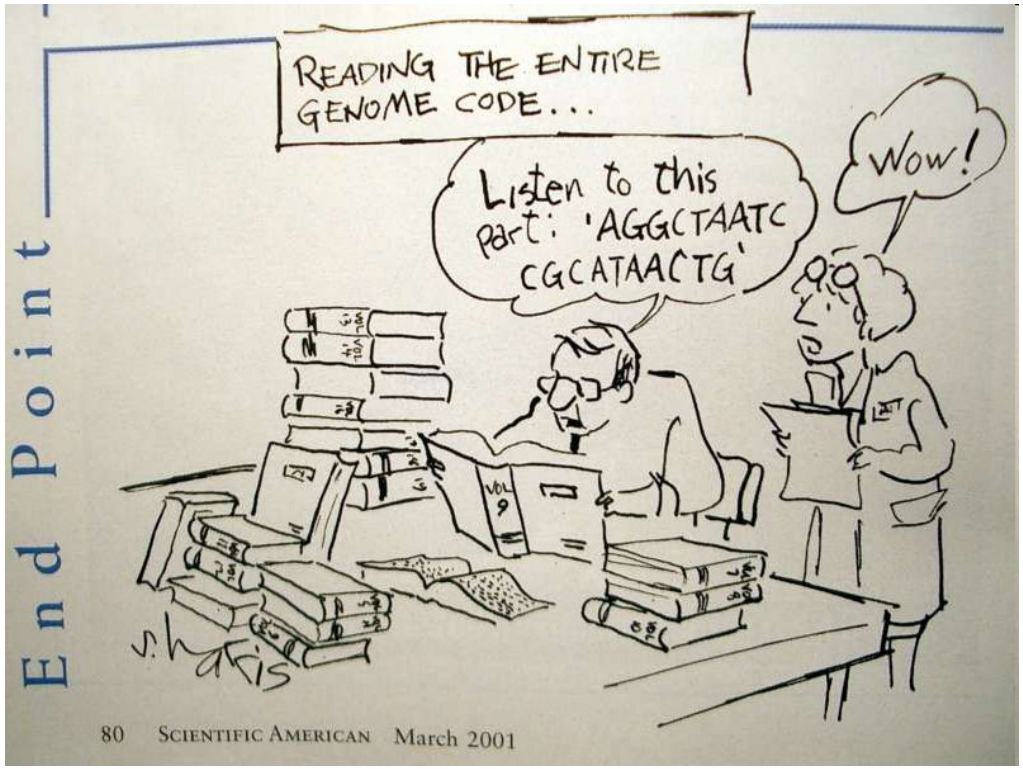




A | B | C | kinase | phtase

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

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)