Biological Systems as Reactive Systems

Luca Cardelli

Microsoft Research
Cambridge UK

Cambridge 2005-07-06
Anatomy Department

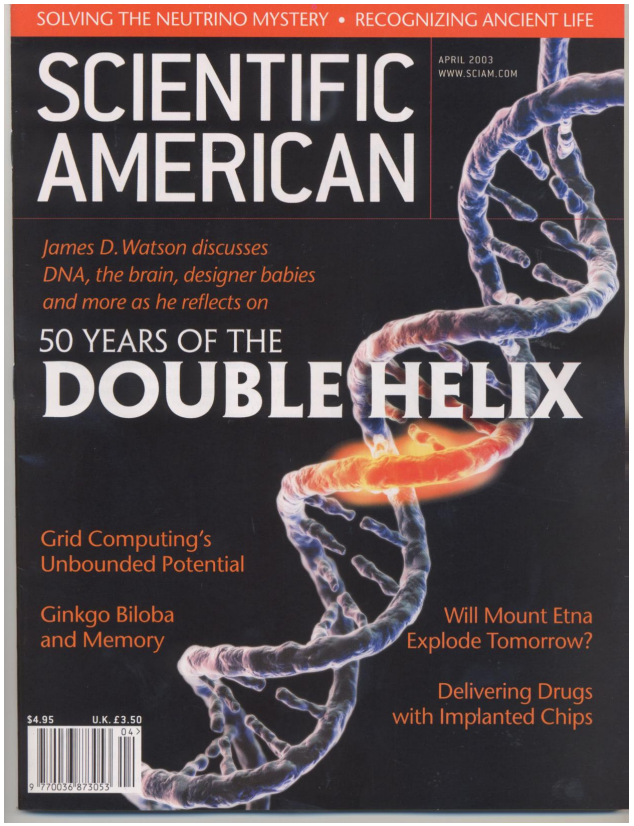
www.luca.demon.co.uk/BioComputing.htm

50 Years of Molecular Cell Biology

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

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

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



Towards Systems Biology

- Biologists now understand many of the cellular components
  - A whole team of biologists will typically study a single protein for years
  - When each component and each reaction is understood, the system is understood (?)

- But this has not led to understand how "the system" works
  - Behavior comes from complex chains of interactions between components
  - Predictive biology and pharmacology still rare
  - Synthetic biology still unreliable

- New approach: try to understand "the system"
  - Experimentally: massive data gathering and data mining (e.g. Genome projects)
  - Conceptually: modeling and analyzing networks (i.e. interactions) of components

- What kind of a system?
  - Just beyond the basic chemistry of energy and materials processing…
  - Built right out of digital information (DNA)
  - Based on information processing for both survival and evolution

- Can we fix it when it breaks?
  - 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
  - Signalling pathway data
  - …

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

- How can we represent, store, and analyze *biological processes*?
  - Scalable, precise, dynamic, highly structured, maintainable representations for *systems biology*.
  - Not just huge lists of chemical reactions or differential equations.

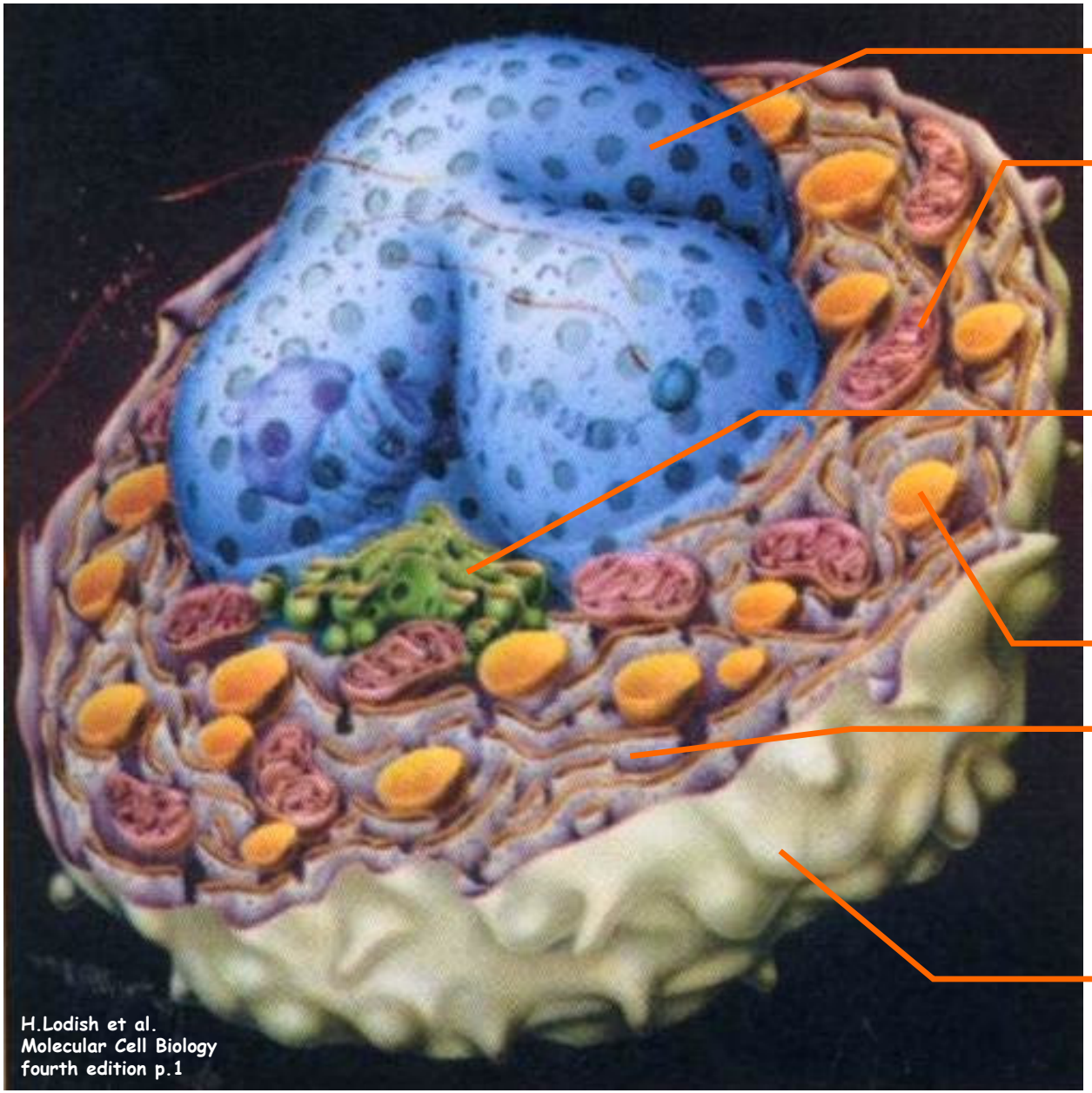
- In computing…
  - There are well-established scalable representations of dynamic reactive processes.
  - They look more or less like little, mathematically based, programming languages.

Structural Architecture

Eukaryotic Cell

(10~100 trillion in human body)

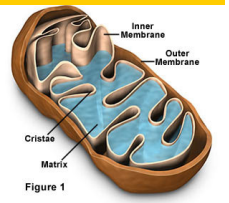
Membranes everywhere



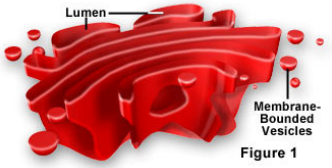
H.Lodish et al.
Molecular Cell Biology
fourth edition p.1

Nuclear membrane

Mitochondria

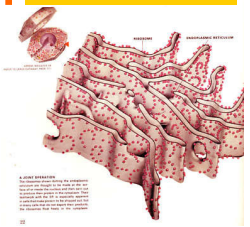


Golgi



Vesicles

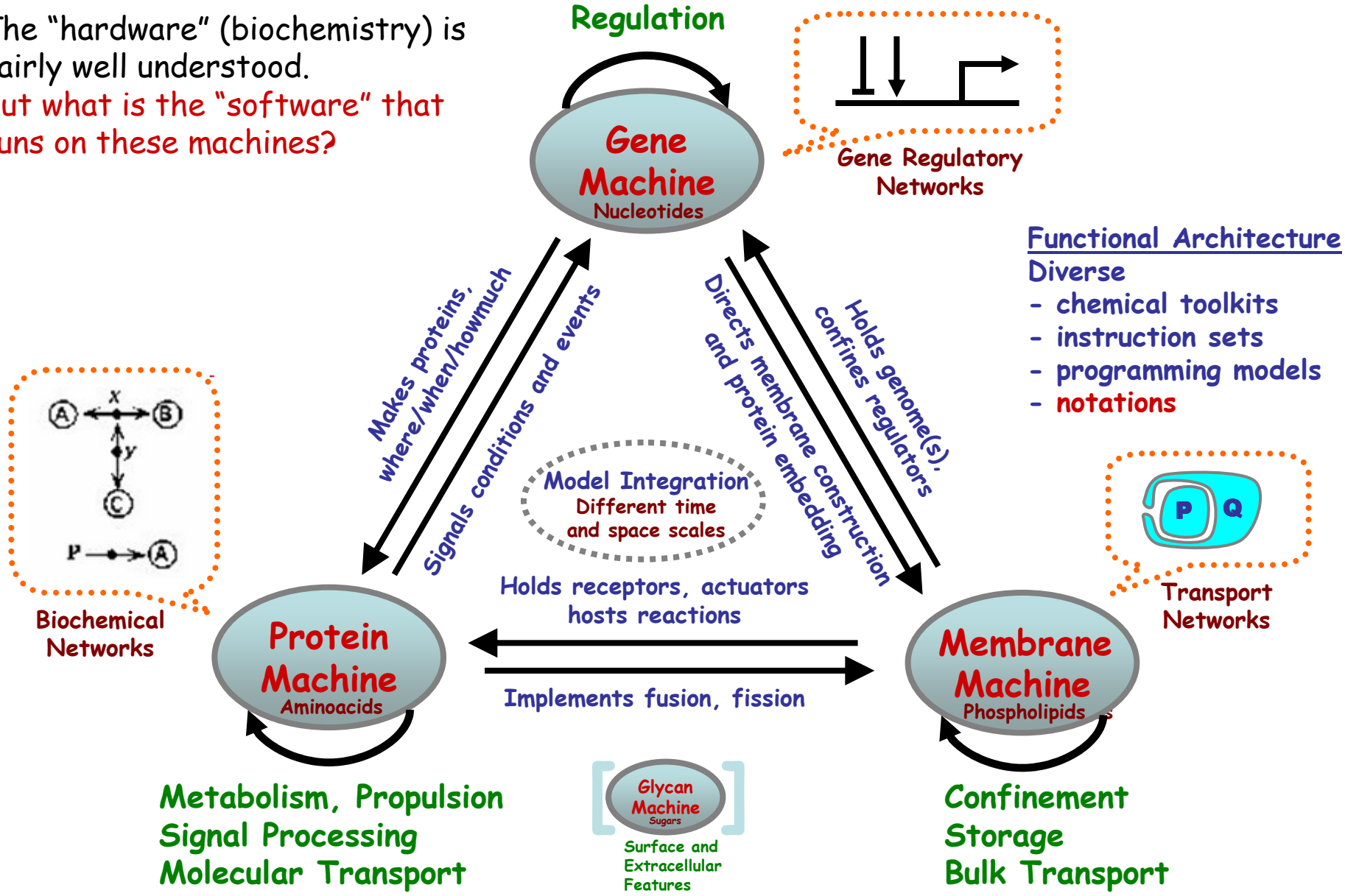
E.R.



Plasma membrane (<10% of all membranes)

Abstract Machines of Systems Biology

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



Reactive Systems

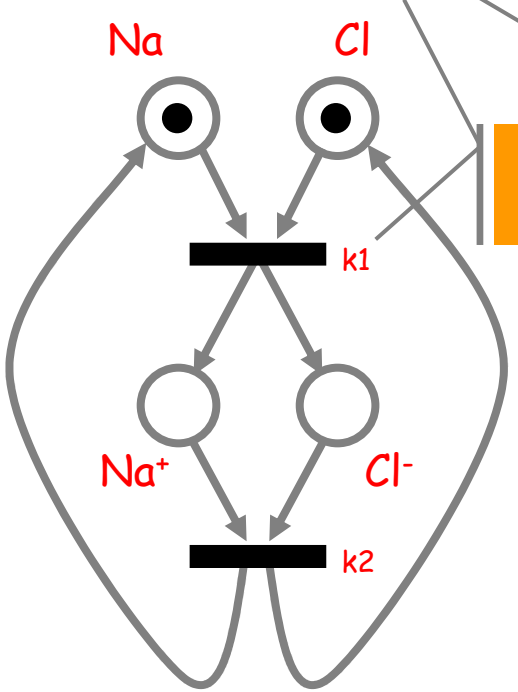
- Modeling biological systems
  - Not as continuous systems (often highly nonlinear)
  - But as discrete reactive systems; abstract machines with:
    - States represent situations
    - Event-driven transitions between states represent dynamics
  - The adequacy of describing (discrete) complex systems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

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

Chemistry vs. π-calculus

A process calculus (chemistry, or SBML)

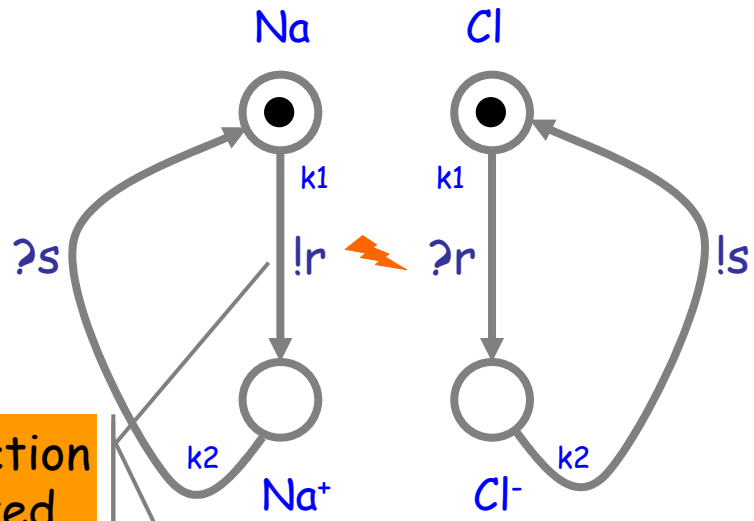
$$Na + Cl \rightarrow_{k1} Na^+ + Cl^-$$
$$Na^+ + Cl^- \rightarrow_{k2} Na + Cl$$



1 line per reaction

Reaction oriented

A compositional graphical representation, and the corresponding calculus.



Interaction oriented

1 line per component

$$Na = !r_{k1}; ?s_{k2}; Na$$
$$Cl = ?r_{k1}; !s_{k2}; Cl$$

Na^+ Cl^-

The same "model"

Maps to a CTMC	Maps to a CTMC

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

A different process calculus (π)

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

Basic Modeling Guidelines

- Regev-Shapiro: "Molecules as Processes":

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

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

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

π-calculus Executive Summary

- It's for:
  - The modular description of concurrent, nondeterministic systems
  - Study of such systems based on their descriptions
- It's got:
  - Processes
  - Channels
  - A minimalistic syntax (it's a *language* and also a model)
- You can:
  - Fork new processes
  - Create new channels
  - Do I/O over channels (synchronous and asynchronous) including passing channels over channels
  - Make nondeterministic choices
  - Define processes recursively
- That's it.
  - Except for extensive model theory and metatheory.
  - Cannot pass processes over channels (simulated by passing channels to them)
  - Cannot define procedures (simulated by supplying reply channels)

π-calculus

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 Executive Summary

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

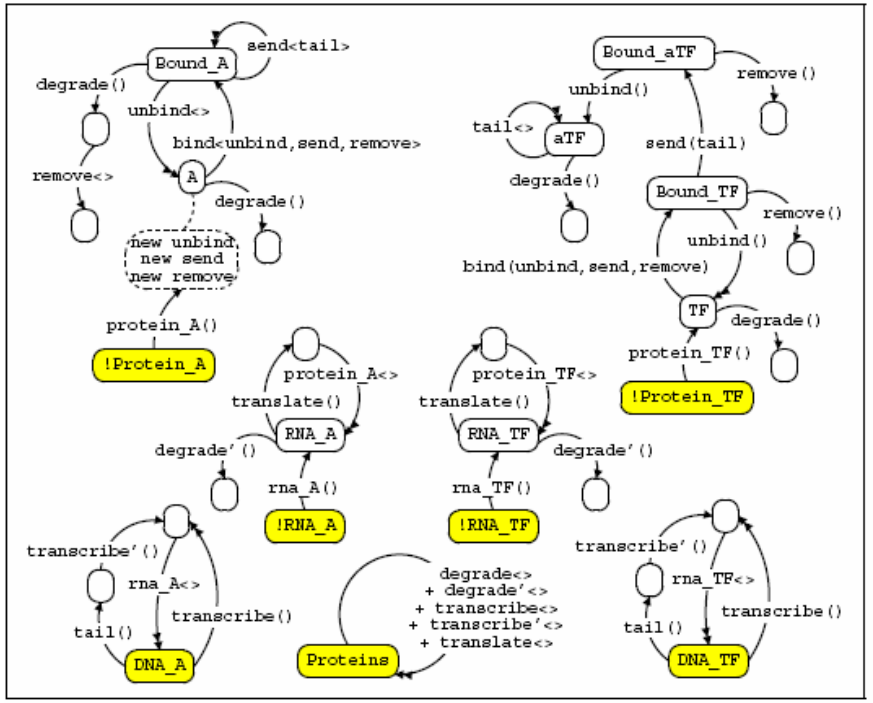


Figure 2. Regulating Gene Expression by Positive Feedback [9]

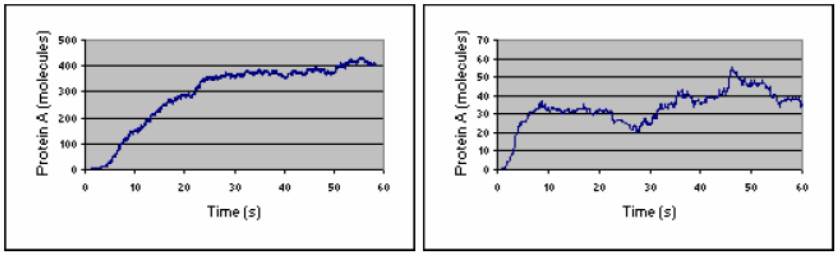


Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of TF

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

Importance of Stochastic Effects

- A deterministic system:
  - May get "stuck in a fixpoint".
  - And hence never oscillate.

- A similar stochastic system:
  - May be "thrown off the fixpoint" by stochastic noise, entering a long orbit that will later bring it back to the fixpoint.
  - And hence oscillate.

Surprisingly enough, we have found that parameter values that give rise to a stable steady state in the deterministic limit continue to produce reliable oscillations in the stochastic case, as shown in Fig. 5. Therefore, the presence of noise not only changes the behavior of the system by adding more disorder but can also lead to marked qualitative differences.

Mechanisms of noise-resistance in genetic oscillators

Jose´ M. G. Vilar, Hao Yuan Kueh, Naama Barkai, Stanislas Leibler

PNAS April 30, 2002 vol. 99 no. 9 p.5991

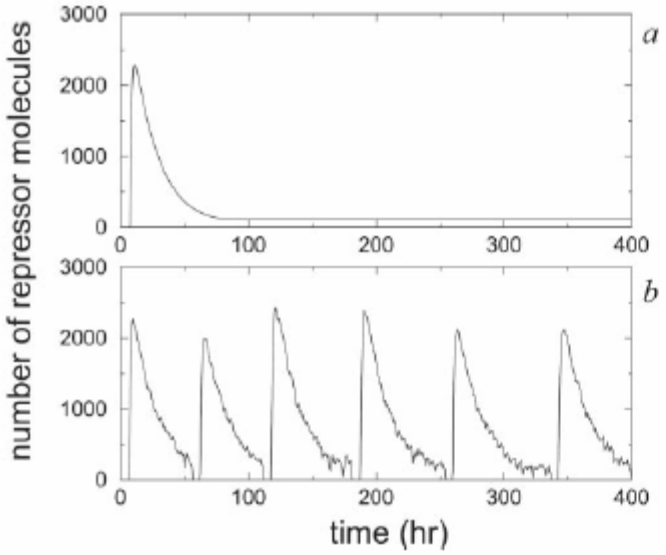


Fig. 5. Time evolution of R for the deterministic Eq. [1] (*a*) and stochastic (*b*) versions of the model. The values of the parameters are as in the caption of Fig. 1, except that now we set $\delta_R = 0.05\ h^{-1}$. For these parameter values, $\tau < 0$, so that the fixed point is stable.

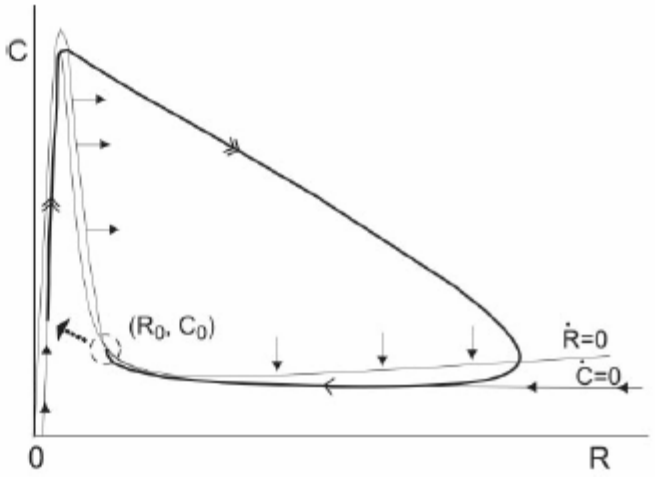
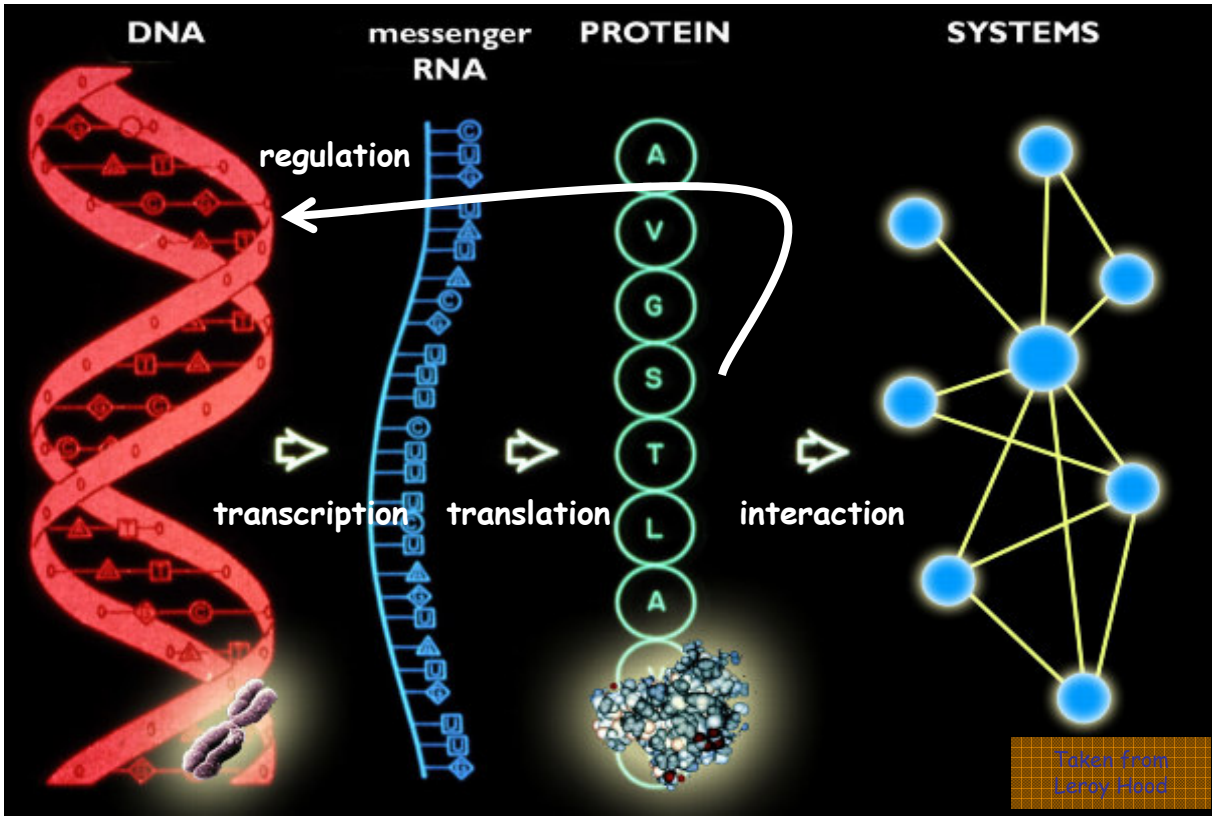


Fig. 6. Phase portrait as in Fig. 4 but for a situation in which the system falls into the stable fixed point (R_0,C_0). The dotted arrow to the left of the fixed point illustrates a perturbation that would initiate a single sweep of the (former) oscillatory trajectory.

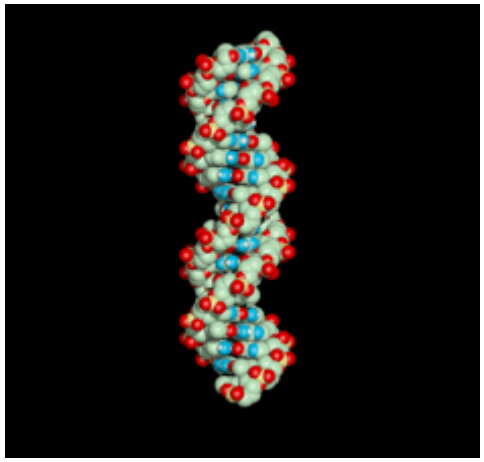
Gene Networks

The Gene Machine

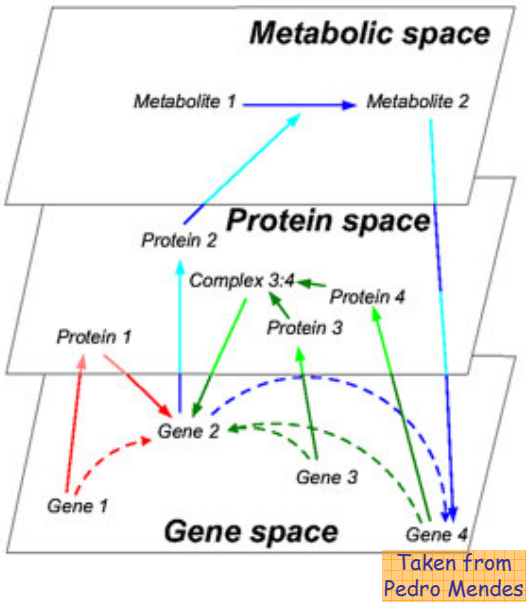
The "Central Dogma" of Molecular Biology



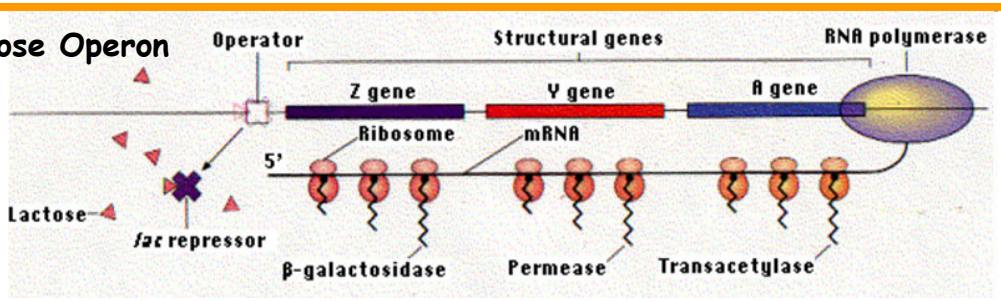
4-letter digital code | 4-letter digital code | 20-letter digital code | 50.000(?) shapes



DNA Tutorial

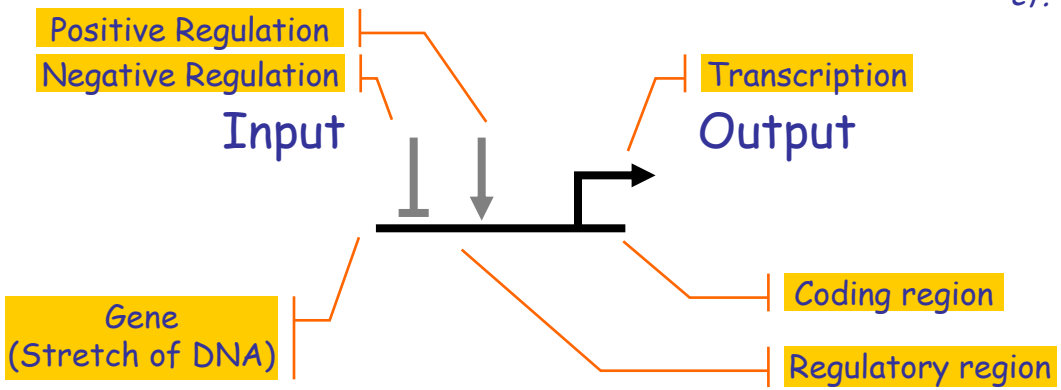


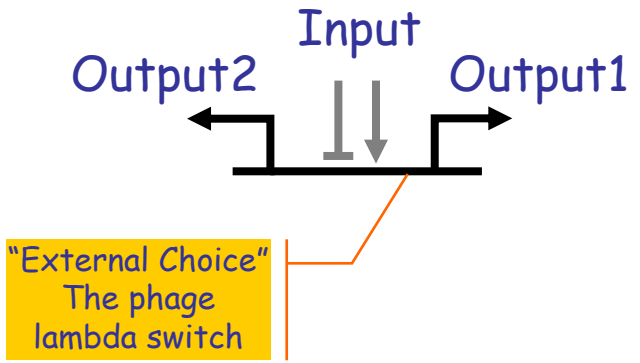
Lactose Operon



The Gene Machine "Instruction Set"

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





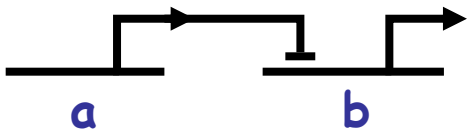
Regulation of a gene (positive and negative) influences transcription. The regulatory region has precise DNA sequences, but not meant for coding proteins: meant for binding regulators.

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

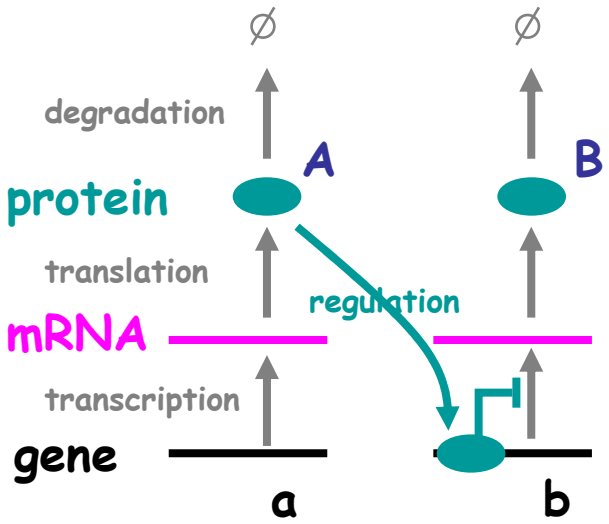
Human (and mammalian) Genome Size
3Gbp (Giga base pairs) 750MB @ 4bp/Byte (CD)
Non-repetitive: 1Gbp 250MB
In genes: 320Mbp 80MB
Coding: 160Mbp 40MB
Protein-coding genes: 30,000-40,000

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 Composition



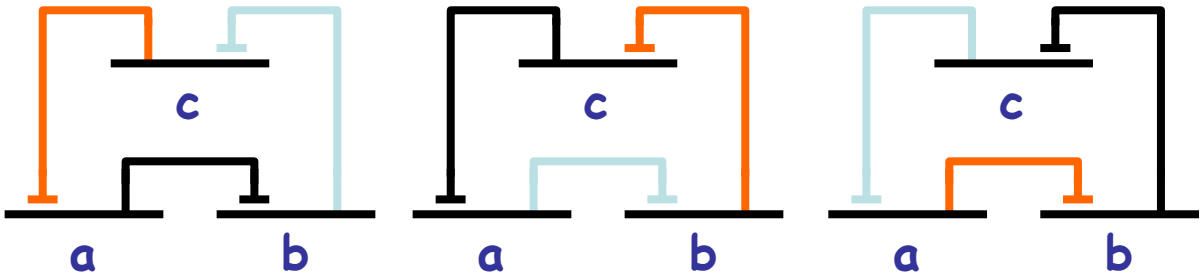
Is a shorthand for:



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

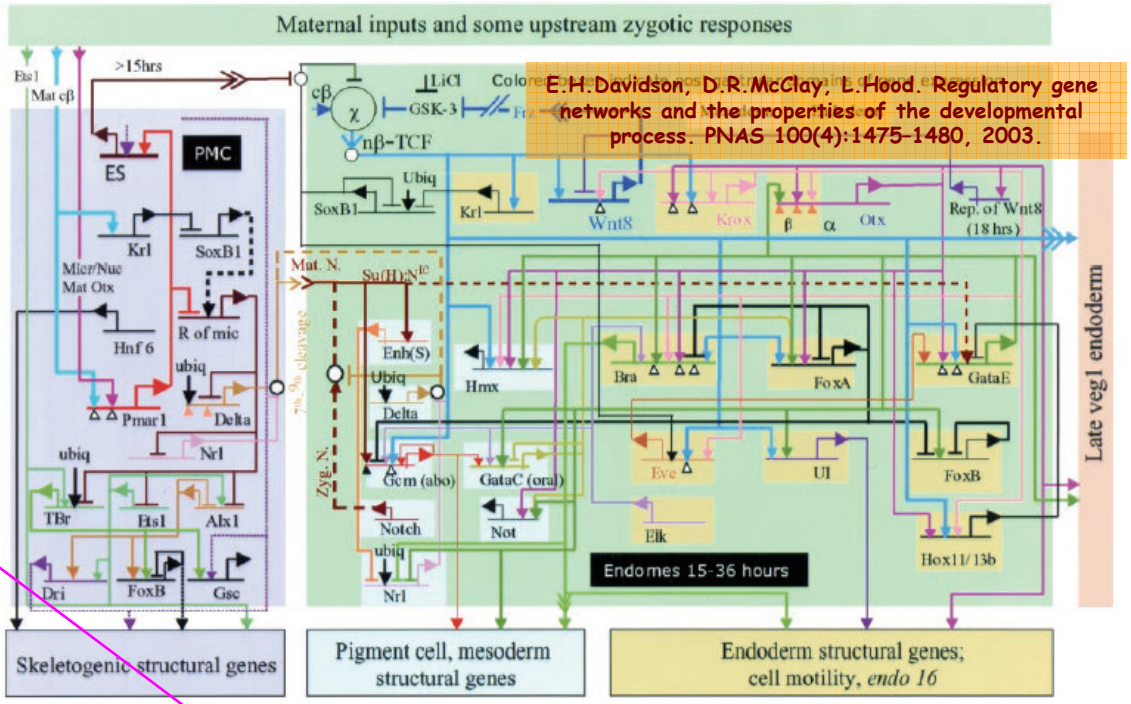


Expressed
Repressed
Expressing

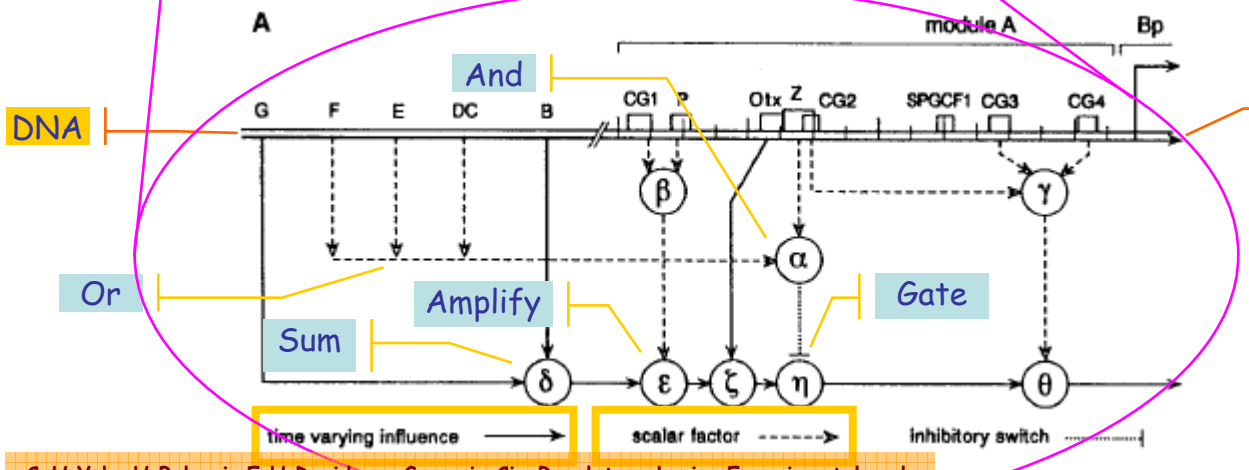
Gene Regulatory Networks

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

NetBuilder



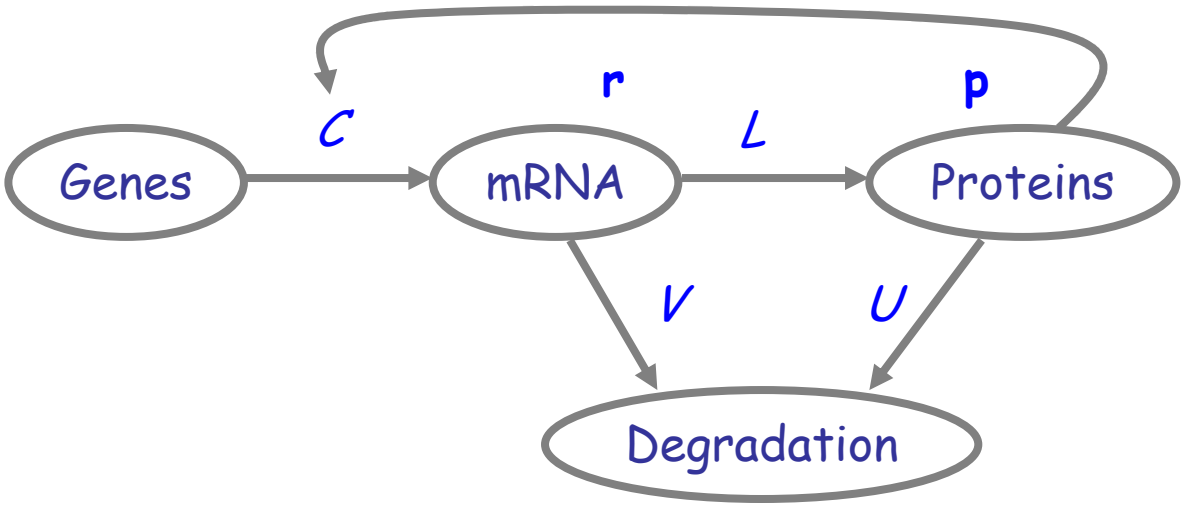
E.H.Davidson, D.R.McClay, L.Hood. Regulatory gene networks and the properties of the developmental process. PNAS 100(4):1475-1480, 2003.



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 Classical ODE Approach)

[Chen, He, Church]



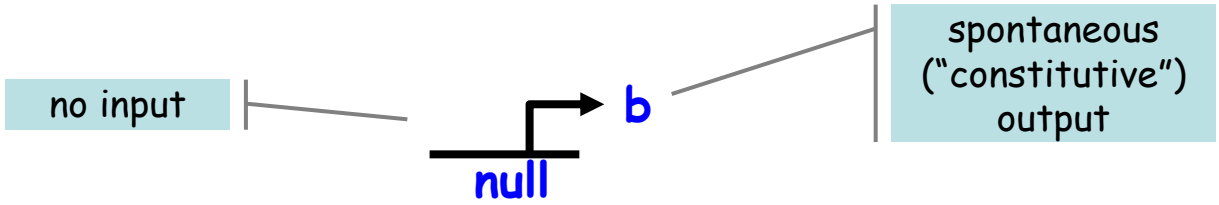
$$\frac{d\mathbf{r}}{dt} = f(\mathbf{p}) - V\mathbf{r}$$

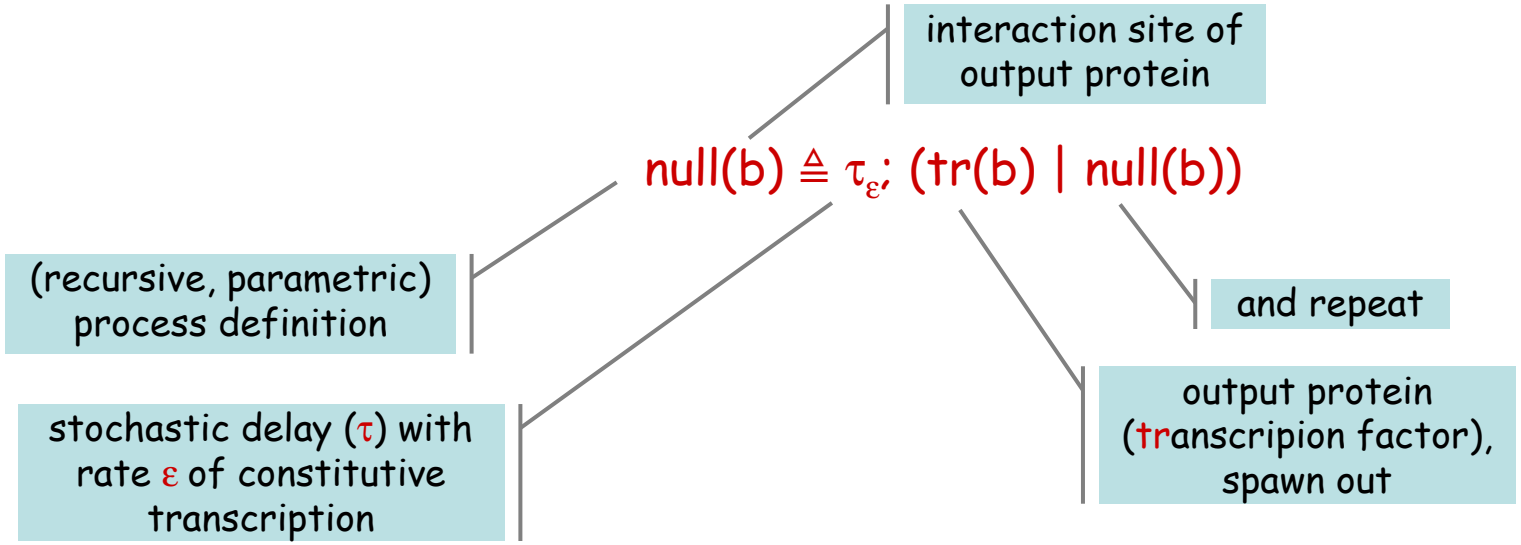
$$\frac{d\mathbf{p}}{dt} = L\mathbf{r} - U\mathbf{r}$$

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

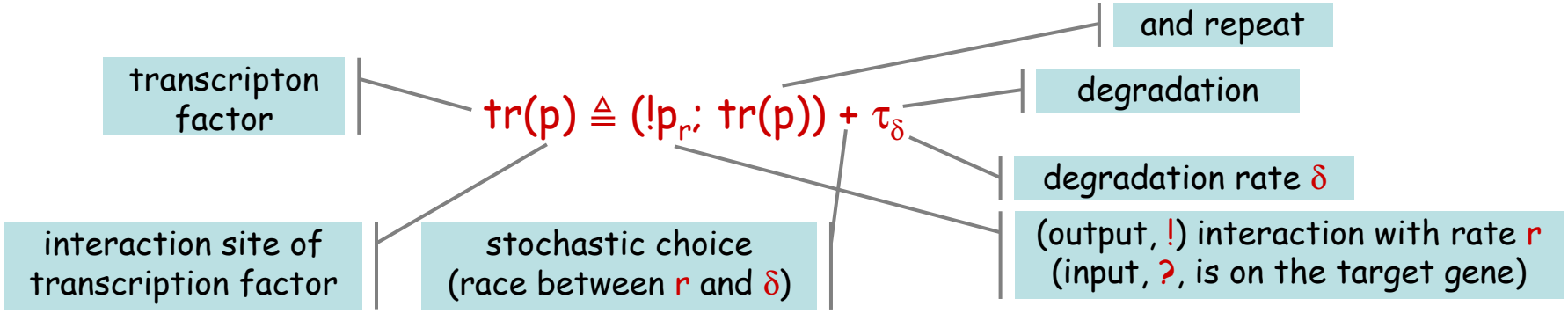




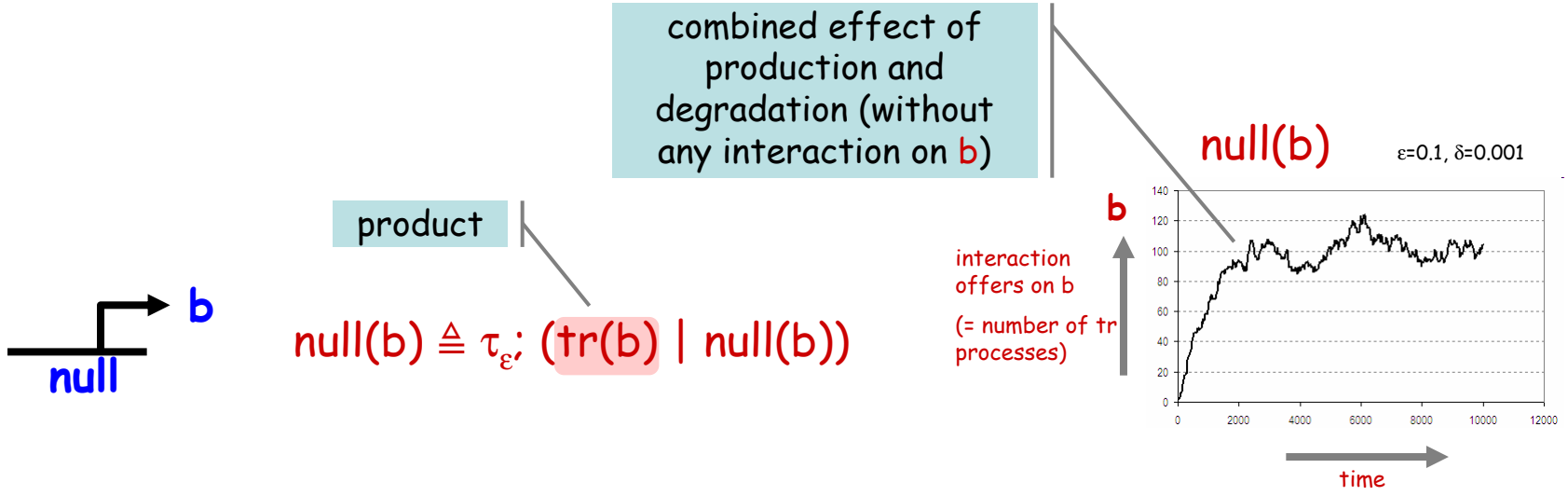
A stochastic rate r is always associated with each channel a_r (at channel creation time) and delay τ_r, but is often omitted when unambiguous.

Production and Degradation

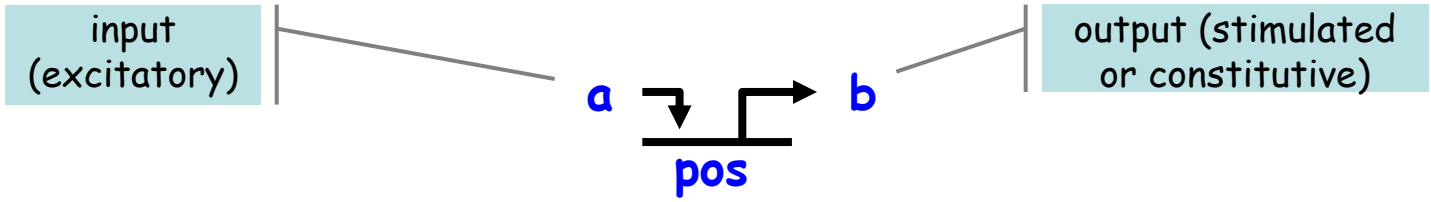
Degradation is extremely important and often deliberate; it changes unbounded growth into (roughly) stable signals.

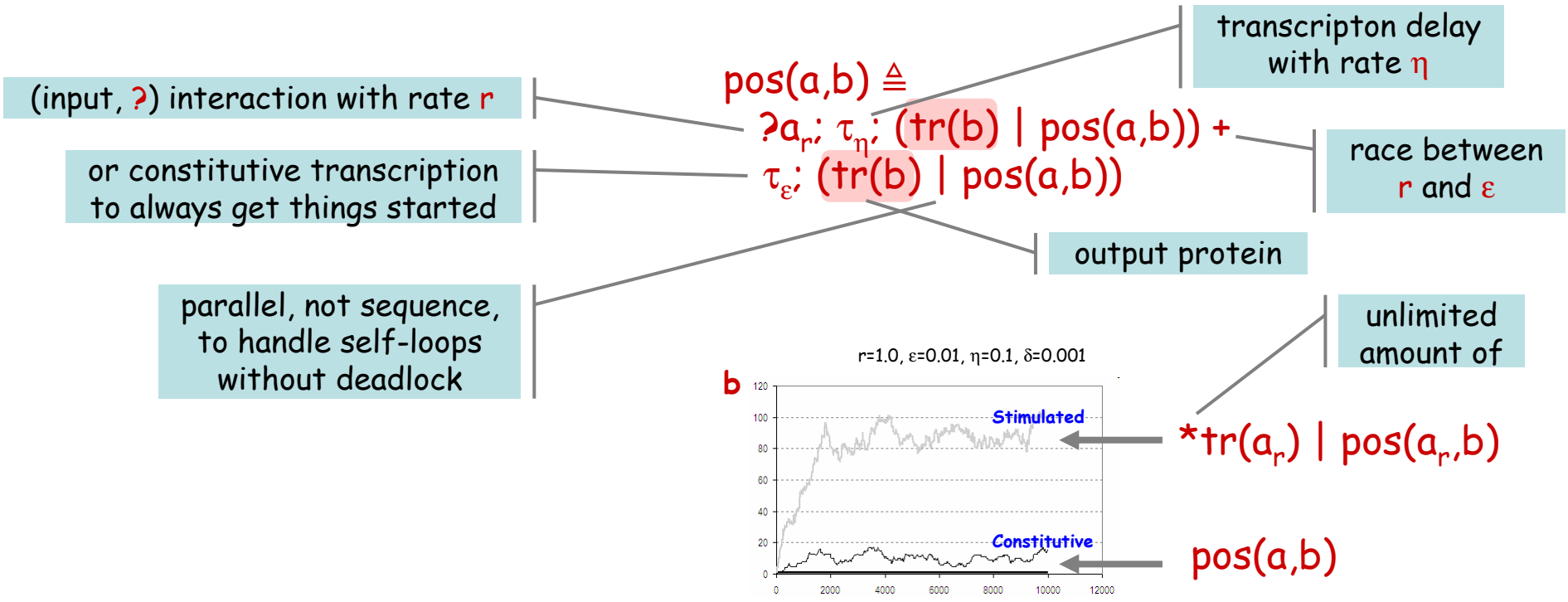


A transcription factor is a process (not a message or a channel): it has behavior such as interaction on p and degradation.

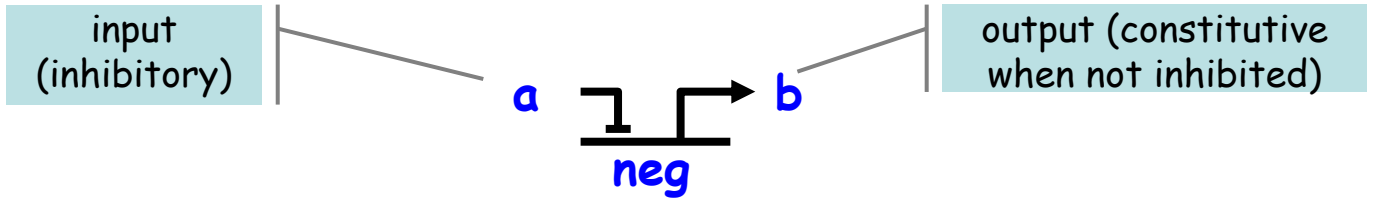


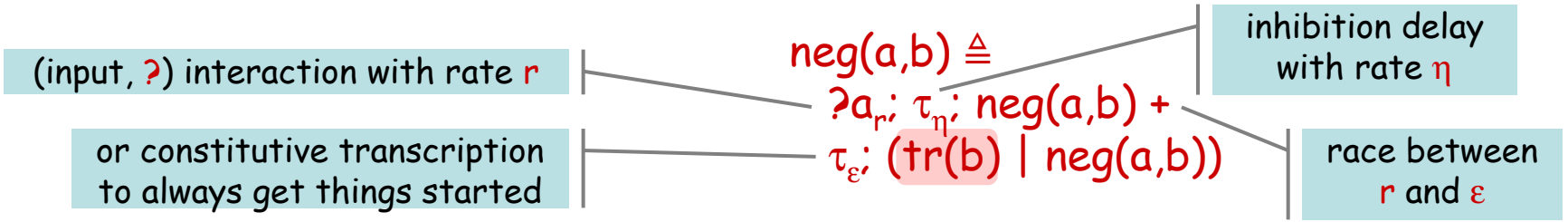
Unary Pos Gate

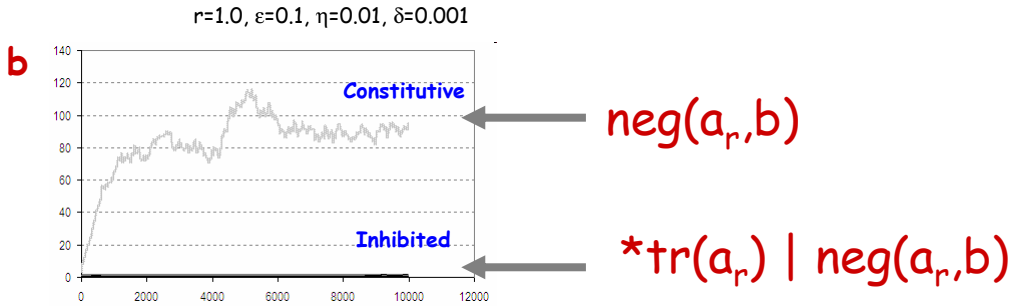




Unary Neg Gate

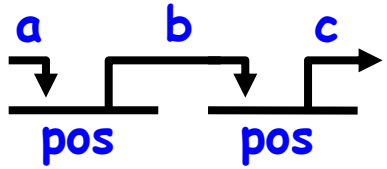






Signal Amplification

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

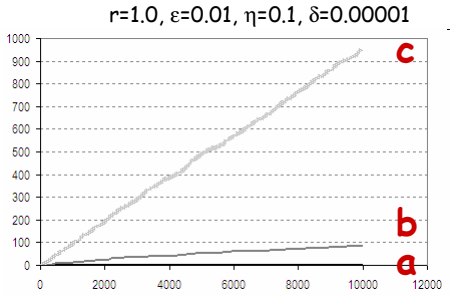


$$pos(a,b) \triangleq ?a_r;\ \tau_\eta;\ (tr(b) \mid pos(a,b)) + \tau_\varepsilon;\ (tr(b) \mid pos(a,b))$$

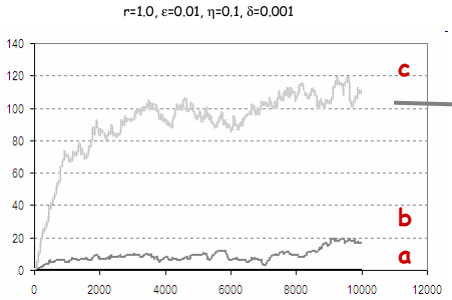
$$tr(p) \triangleq (!p_r;\ tr(p)) + \tau_\delta$$

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



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

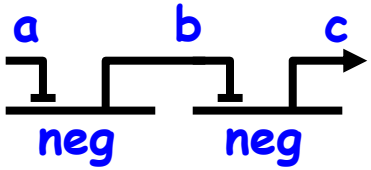


pos(a,b) | pos(b,c)

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

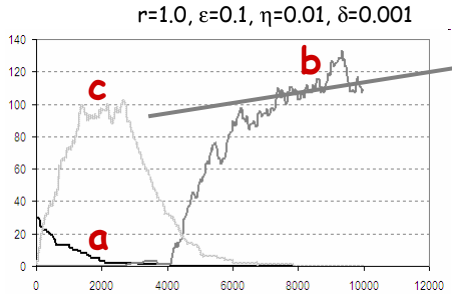
Signal Normalization

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



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

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

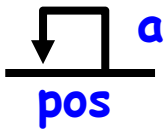


a non-zero input level, a, whether weak or strong, is renormalized to a standard level, c.

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

Self Feedback Circuits

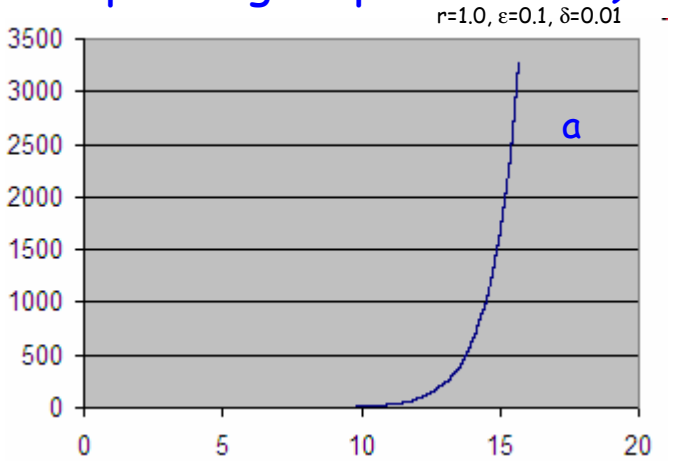
pos(a,a)



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

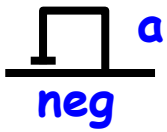
$tr(p) \triangleq (!p_r;\ tr(p)) + \tau_\delta$

(Can overwhelm degradation, depending on parameters)



pos(a,a)

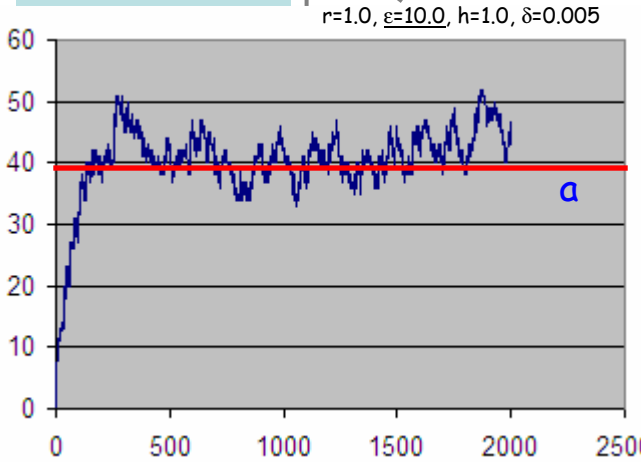
neg(a,a)



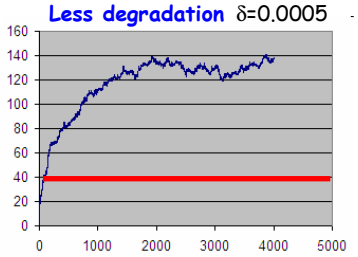
$neg(a,b) \triangleq$
$?a_r;\ \tau_h;\ neg(a,b) +$
$\tau_\varepsilon;\ (tr(b) \mid neg(a,b))$

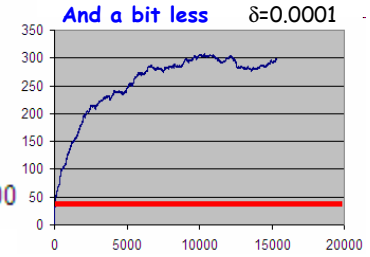
$tr(p) \triangleq (!p_r;\ tr(p)) + \tau_\delta$

high, to raise the signal



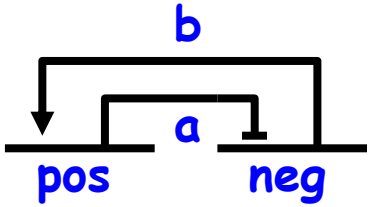
neg(a,a)



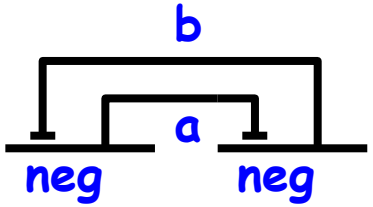


Two-gate Feedback Circuits

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

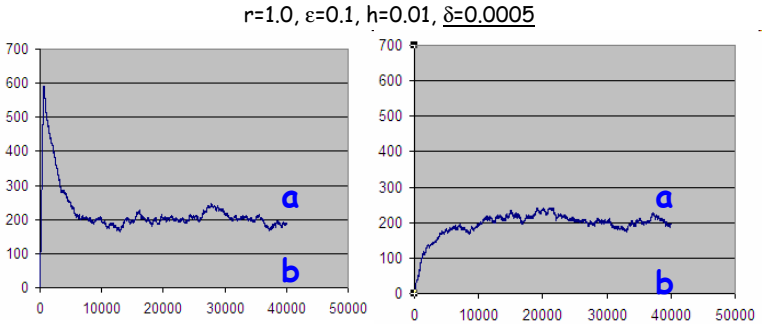


neg(b,a) |
neg(a,b)



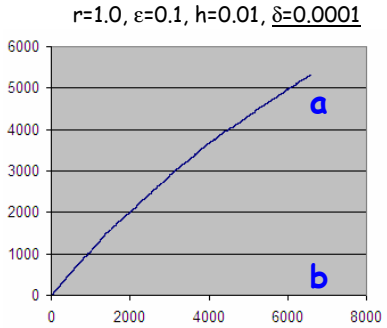
Monostable:

For some degradation rates is quite stable:



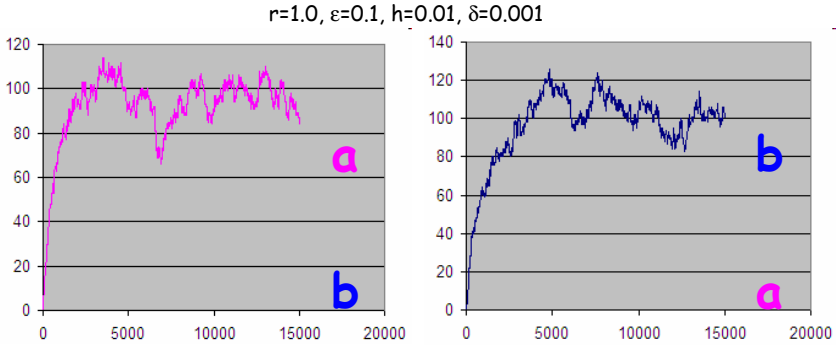
pos(b,a) | neg(a,b)

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

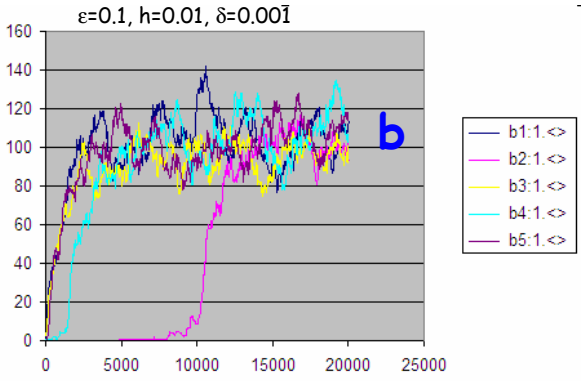


pos(b,a) | neg(a,b)

Bistable:



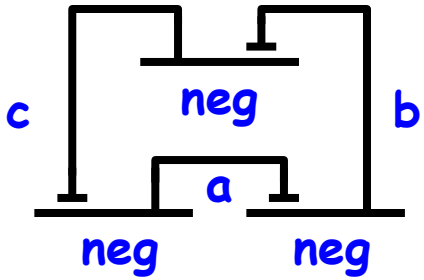
neg(b,a) | neg(a,b)



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)

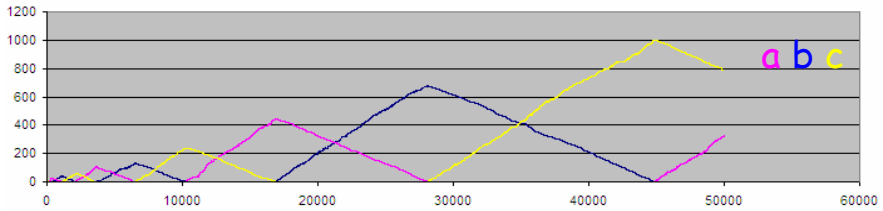


neg(a,b) ≜
?a_r; τ_h; neg(a,b) +
τ_ε; (tr(b) | neg(a,b))

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

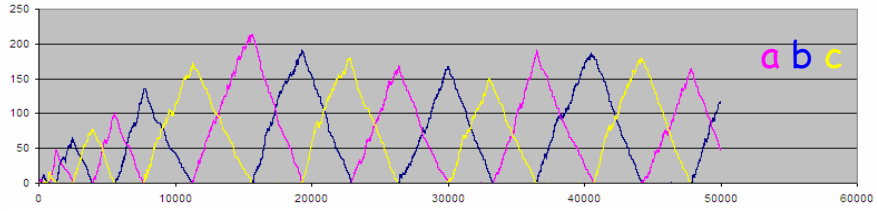
tr(p) ≜ !p_r — interact once and die otherwise stick around

r=1.0, ε=0.1, h=0.04



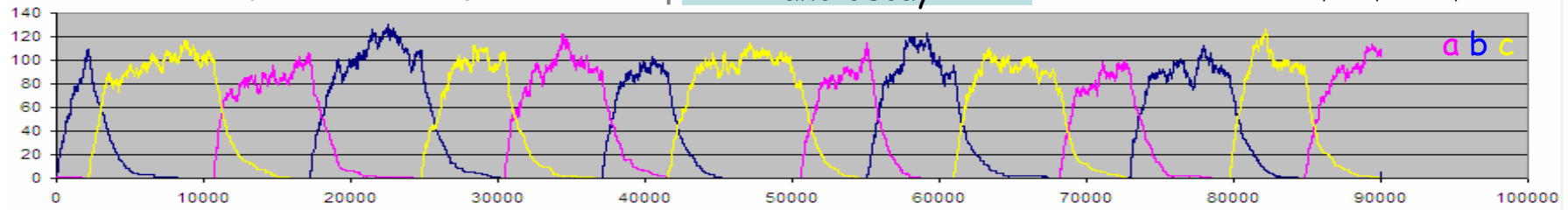
tr(p) ≜ !p_r + τ_δ — interact once and die otherwise decay

r=1.0, ε=0.1, h=0.04, δ=0.0001



tr(p) ≜ (!p_r; tr(p)) + τ_δ — interact many times and decay

r=1.0, ε=0.1, h=0.001, δ=0.001



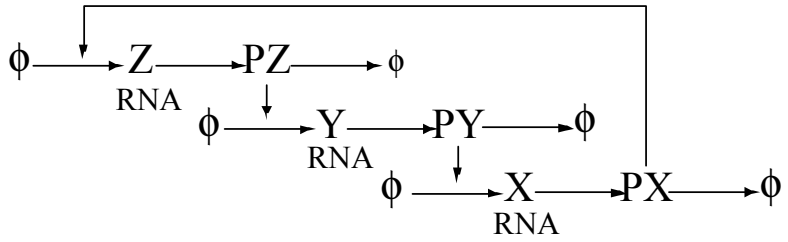
Subtle… at any point one gate is inhibited and the other two can fire constitutively. If one of them fires first, nothing really changes, but if the other one fires first, then the cycle progresses.

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

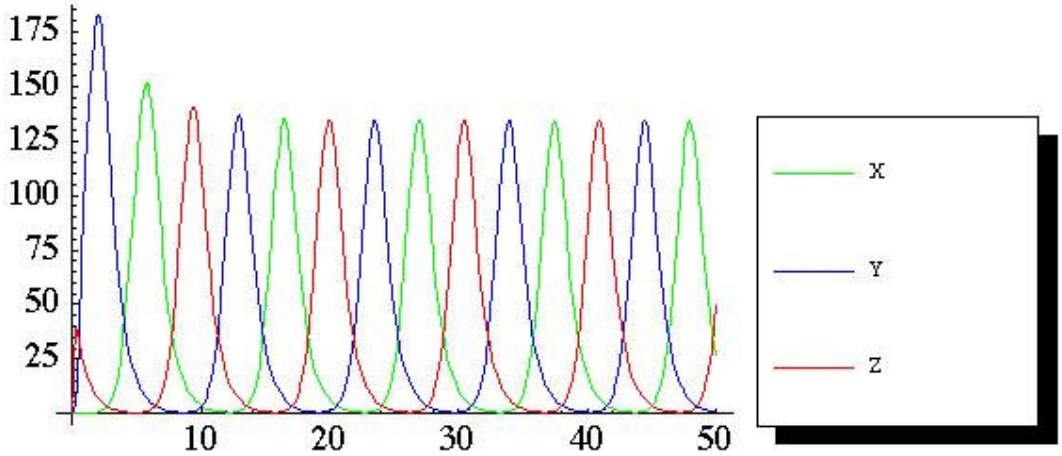
Bruce E Shapiro
Cellerator



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

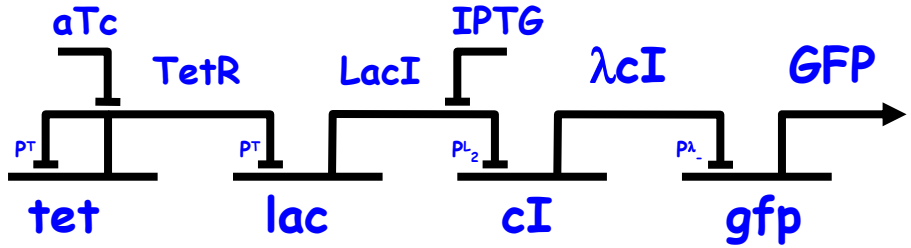
$$\frac{d[\mathrm{Y}]}{dt} = \alpha_0 + \frac{\alpha + \alpha_1 [\mathrm{PZ}]^n}{K^n + [\mathrm{PZ}]^n} - k[\mathrm{Y}], \quad \frac{d[\mathrm{PY}]}{dt} = \beta\{[\mathrm{Y}] - [\mathrm{PY}]\}$$

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



Guet et al.: D038/lac⁻

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

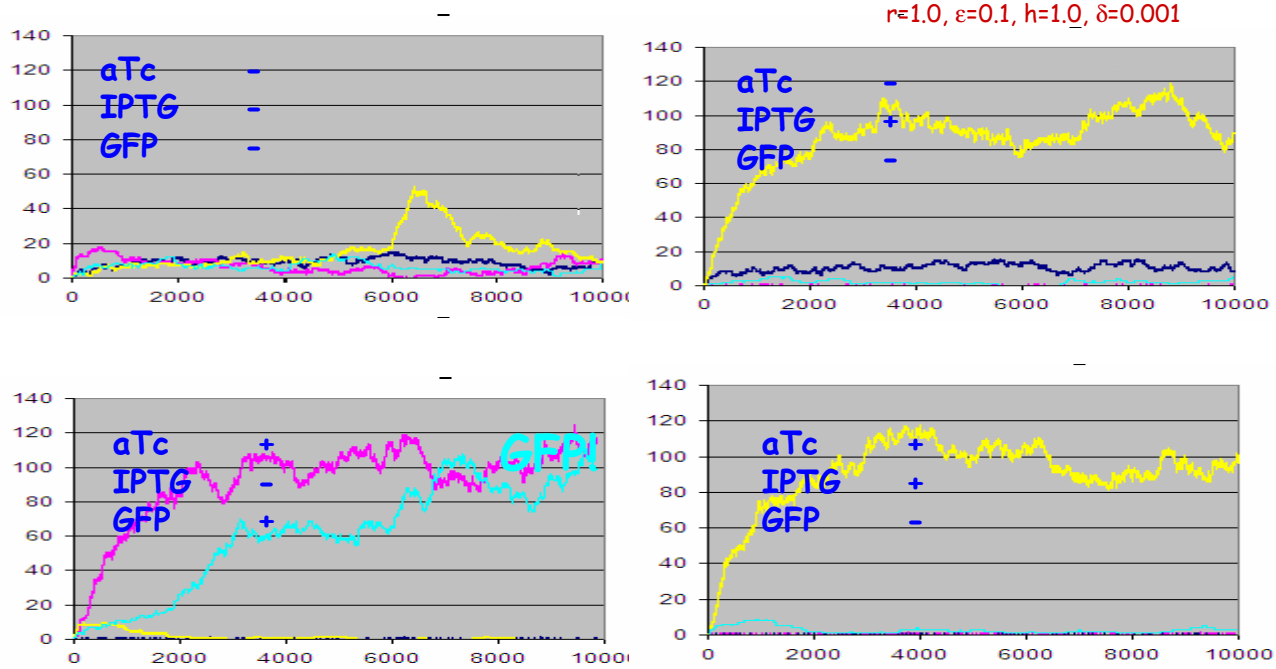


experiment:

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.

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.

10 chemical reactions

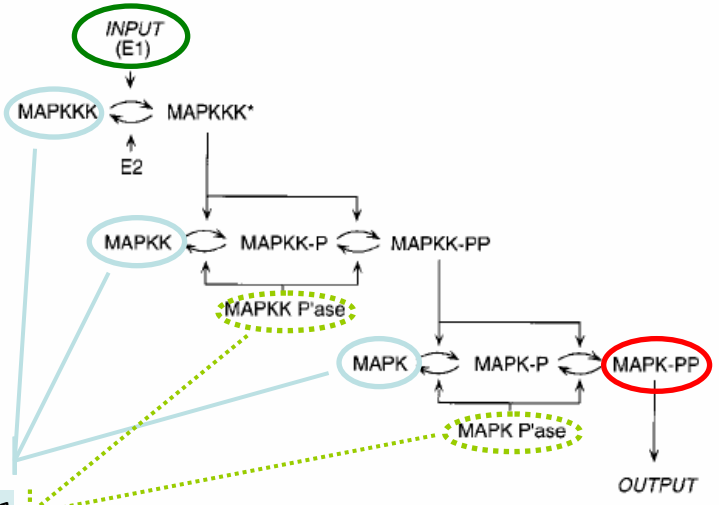


FIG. 1. Schematic view of the MAPK cascade. Activation of MAPK depends upon the phosphorylation of two conserved sites [Thr-183 and Tyr-185 in rat p42 MAPK/Erk2 (4, 5)]. Full activation of MAPKK also requires phosphorylation of two sites [Ser-218 and Ser-222 in mouse Mek-1/MKK1 (6–10)]. Detailed mechanisms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

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.

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

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

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

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

$$\mathrm{KK\text{-}P{\cdot}KK\ P'ase} \xrightarrow{k_4} \mathrm{KK} + \mathrm{KK\ P'ase} \quad [4]$$

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

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

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

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

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

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

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

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

As 18 Ordinary Differential Equations Plus 7 conservation equations

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

$$[E1_{tot}] = [E1] + [KKK{\cdot}E1] \quad [30]$$

$$[E2_{tot}] = [E2] + [KKK^*{\cdot}E2] \quad [31]$$

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

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

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

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

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.

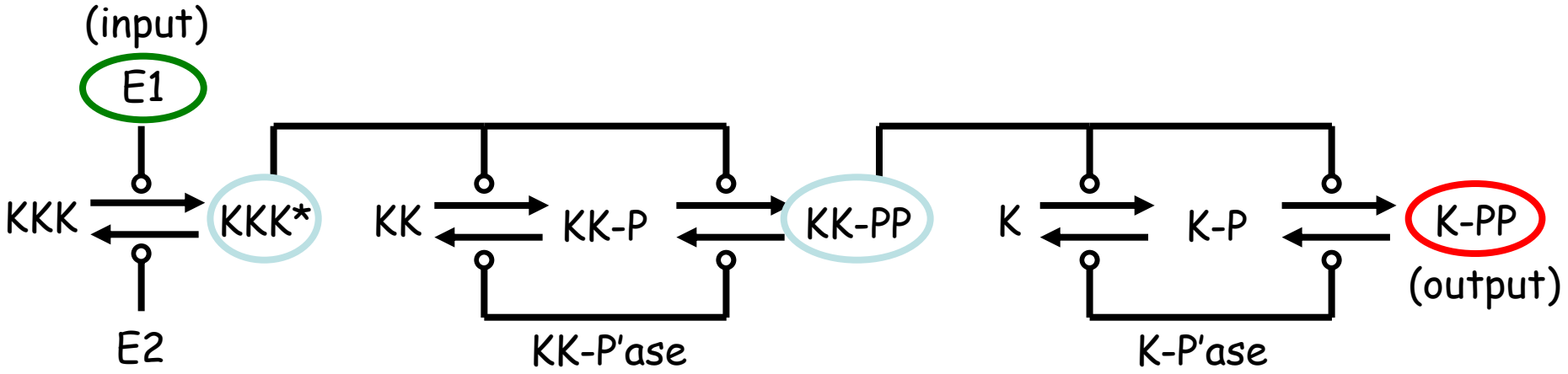
In addition, there are seven conservation equations (Eqs. 29-35).

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

in exactly one state

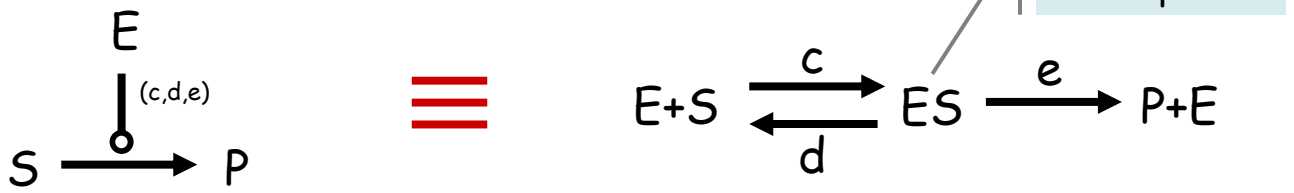
Each molecule

The Circuit

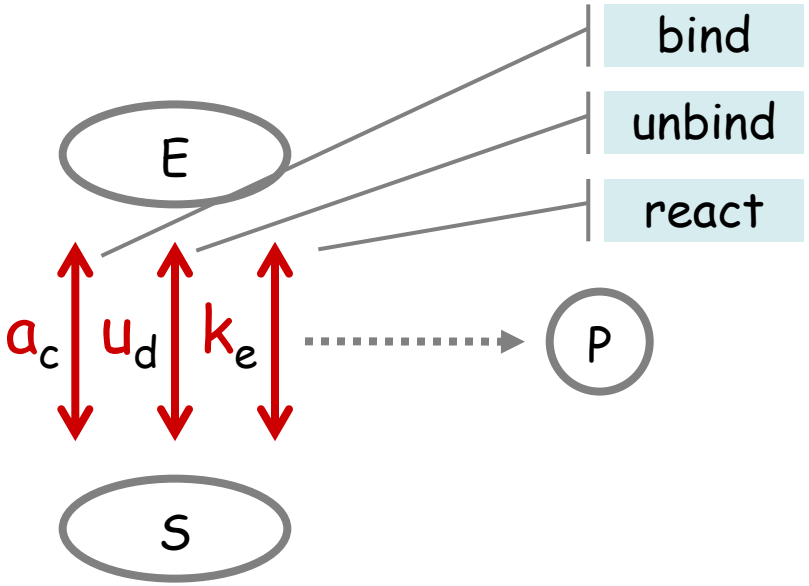


Enzymatic Reactions

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 in SPiM

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

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

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

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

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

and KK_P() =
   (new u4@d4:Release new k4@r4:React
    new u5@d5:Release new k5@r5:React
    do !a4(u4,k4); (do !u4;KK_P() or !k4;KK())          [4]substrate
    or !a5(u5,k5); (do !u5;KK_P() or !k5;KK_PP()))      [5]substrate
```

KKK:E1 complex

E1:KKK complex

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

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

let K() =
   (new u7@d7:Release new k7@r7:React
    !a7(u7,k7); (do !u7;K() or !k7;K_P()))              [7]substrate

and K_P() =
   (new u8@d8:Release new k8@r8:React
    new u9@d9:Release new k9@r9:React
    do !a8(u8,k8); (do !u8;K_P() or !k8;K())            [8]substrate
    or !a9(u9,k9); (do !u9;K_P() or !k9;K_PP()))        [9]substrate

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

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

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

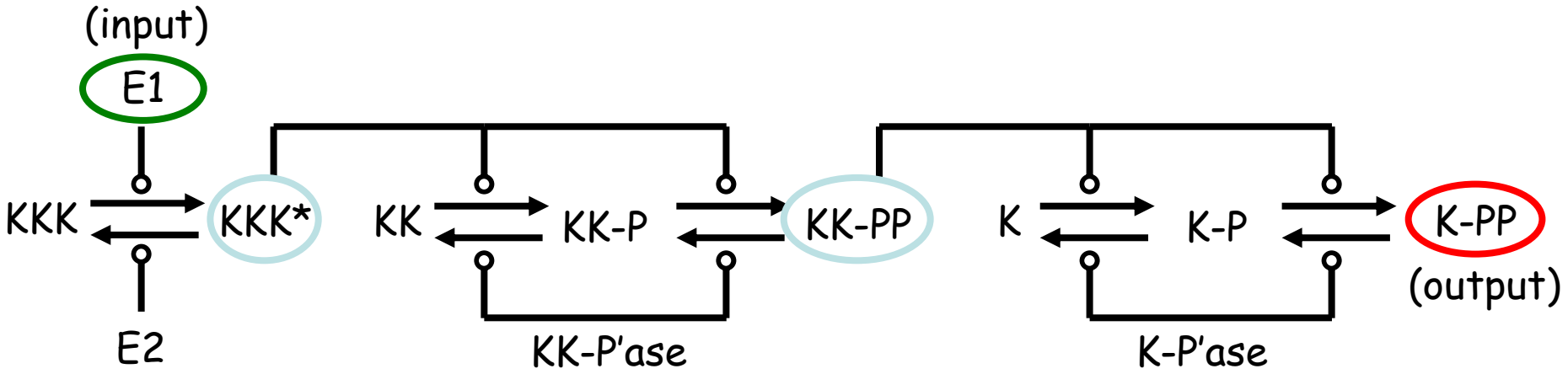
No need for conservation equations: implicit in "choice" operator in the calculus.

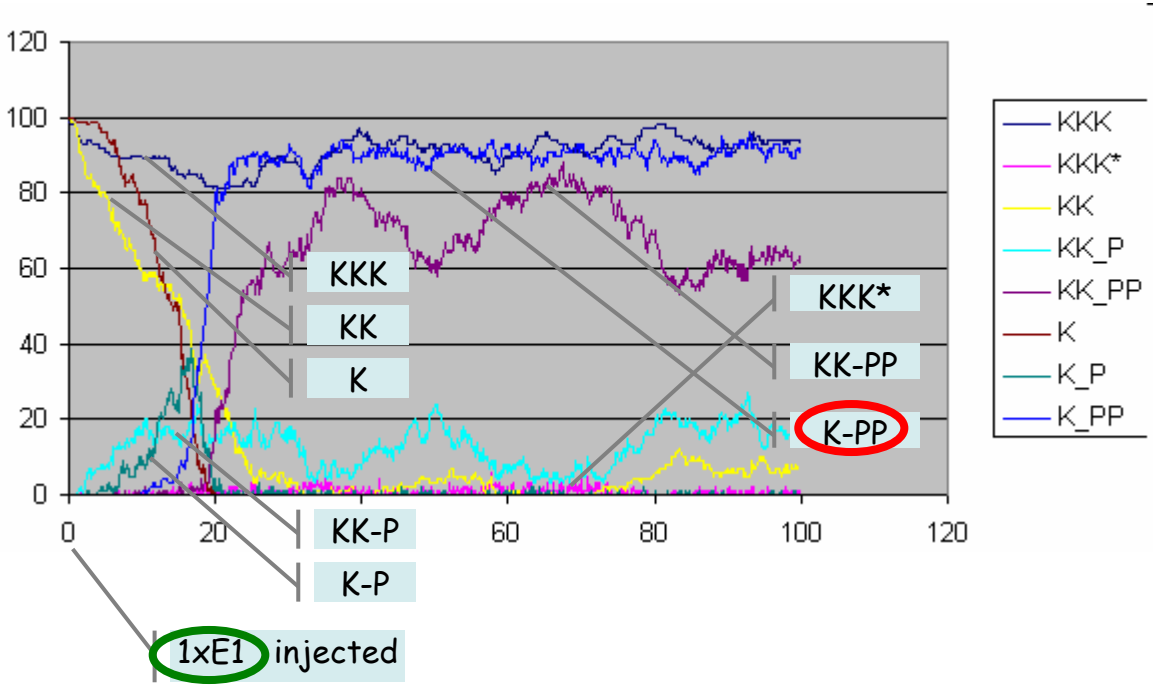
… globals

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

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

…

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

$a_i(u_i,k_i)$: release (u_i@d_i) and react (k_i@r_i) channels passed over bond (a_i) channel. (No behavior attached to channels except interaction rate.)

MAPK Cascade Simulation in SPiM





1st stage:
KKK* barely rises

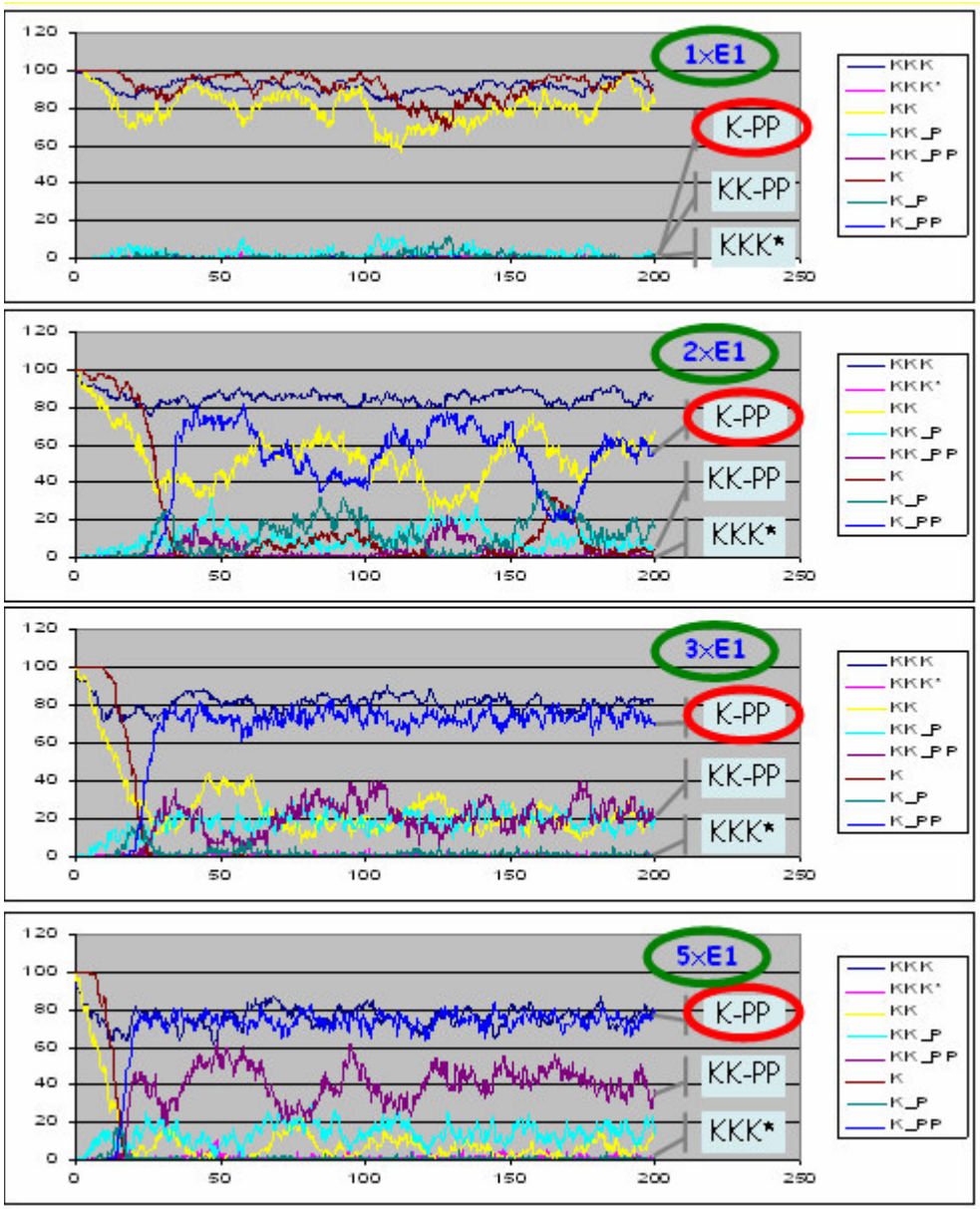
2nd stage:
KK-PP rises, but is not stable

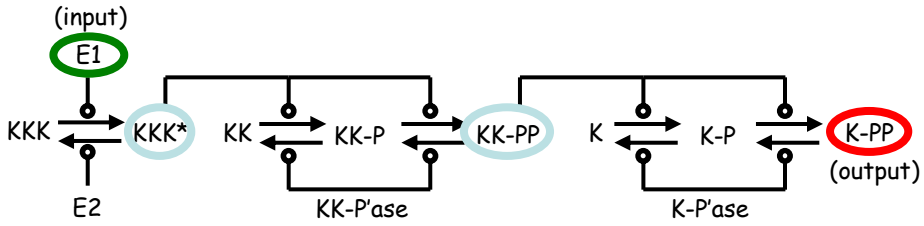
3rd stage:
K-PP flips up to max
even anticipating 2nd stage

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

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

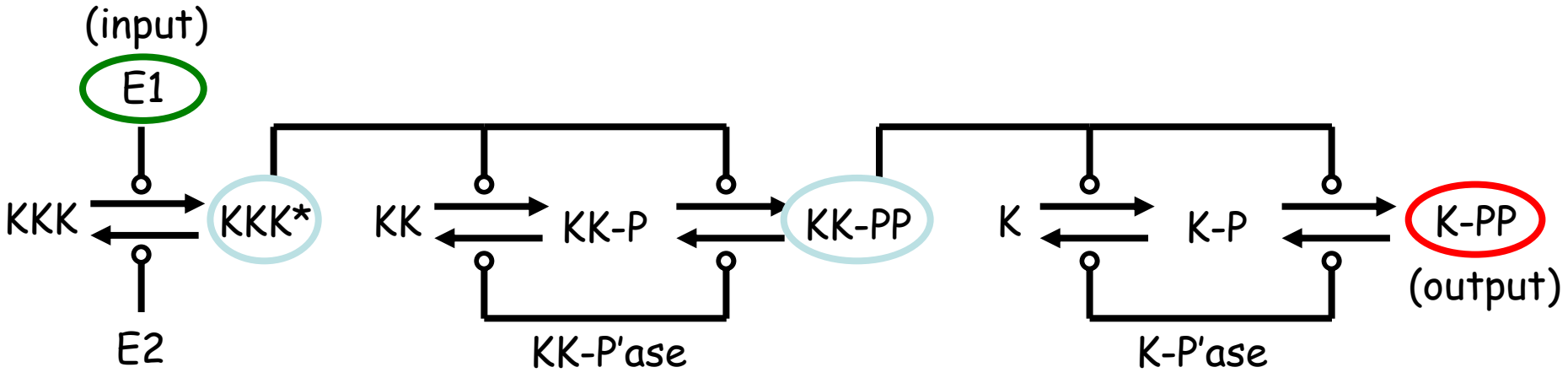
MAPK Cascade Simulation in SPiM

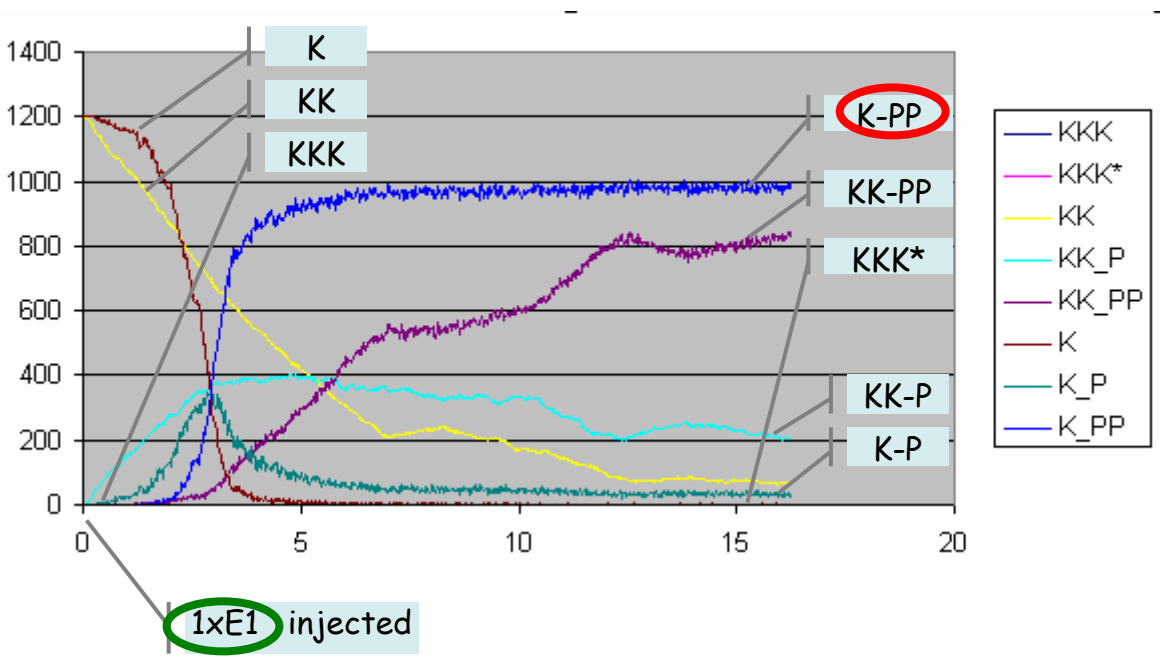




All coefficients 1.0 !!!
100xKKK, 100xKK, 100xK,
13xE2, 13xKKPse, 13xKPse.
nxE1 as indicated
(1xE1 is not sufficient to produce an output)

MAPK Cascade Simulation in SPiM





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)

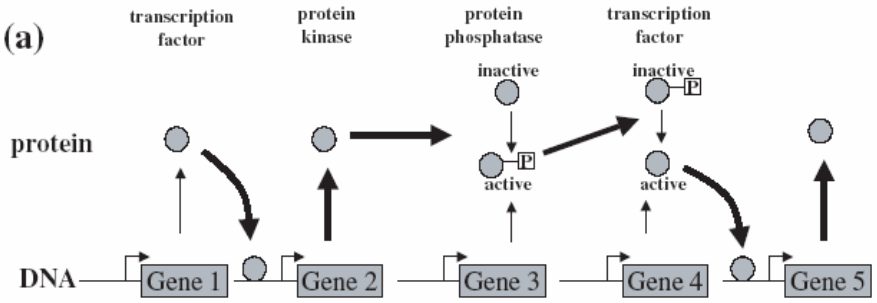
dx = rx = 150, ax = 1
(Kmx = (dx + rx) / ax, Km = 300 nM)

1xE1

Gene-Protein Networks

Indirect Gene Effects

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



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

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

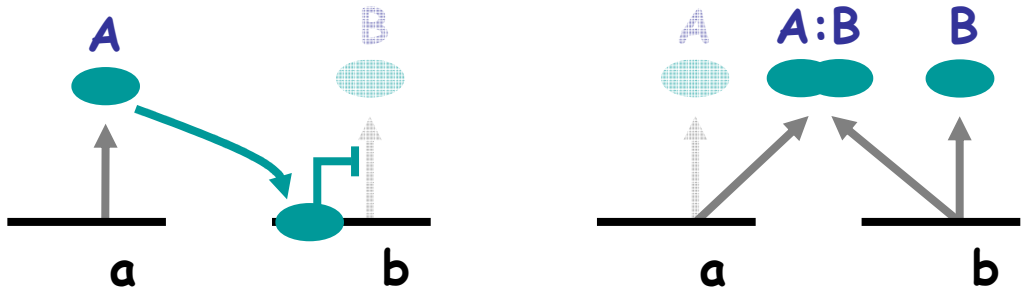
(c)
Aspect of gene activity: phosphorylation state
Genetic perturbation: gene deletion

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

Taken from Andreas Wagner

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

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



François & Hakim Fig3A

PNAS (101)2, 580-585, 2004

Design of genetic networks with specified functions by evolution in silico

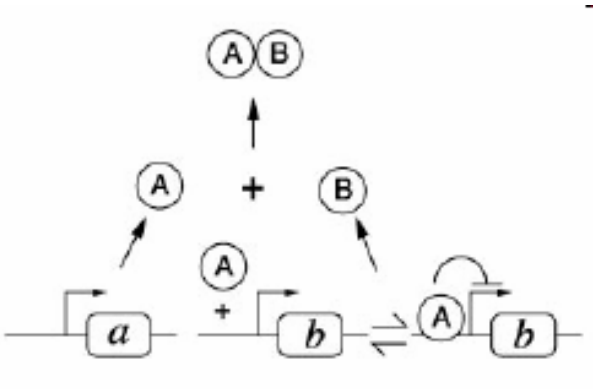
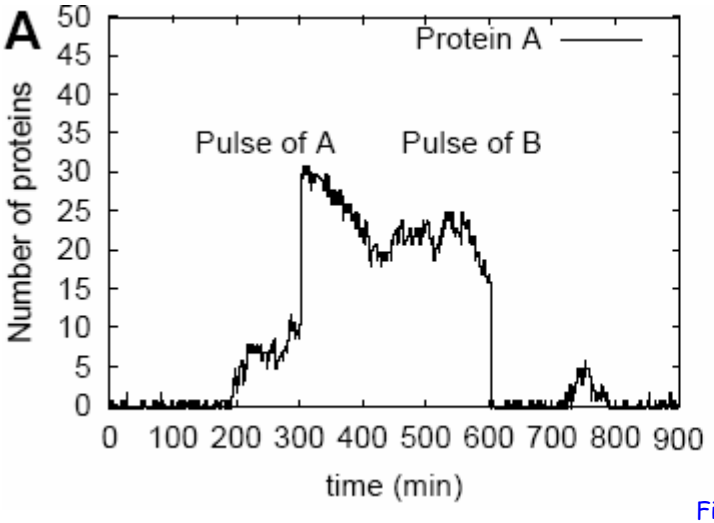


Fig 3A

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

Reaction oriented



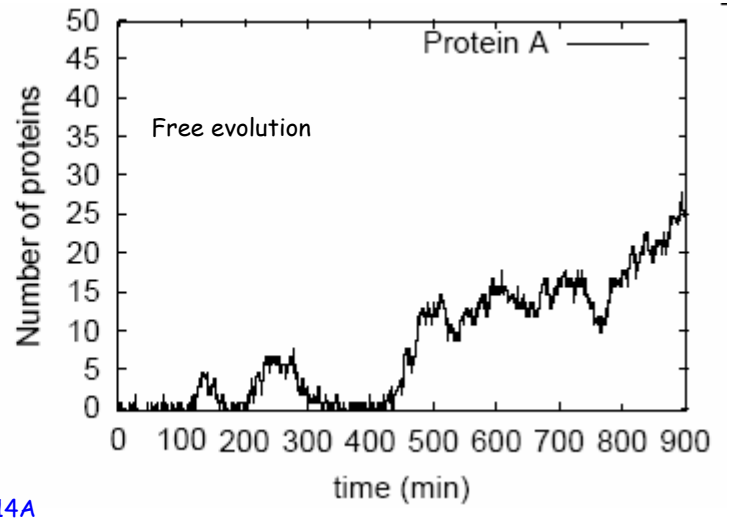
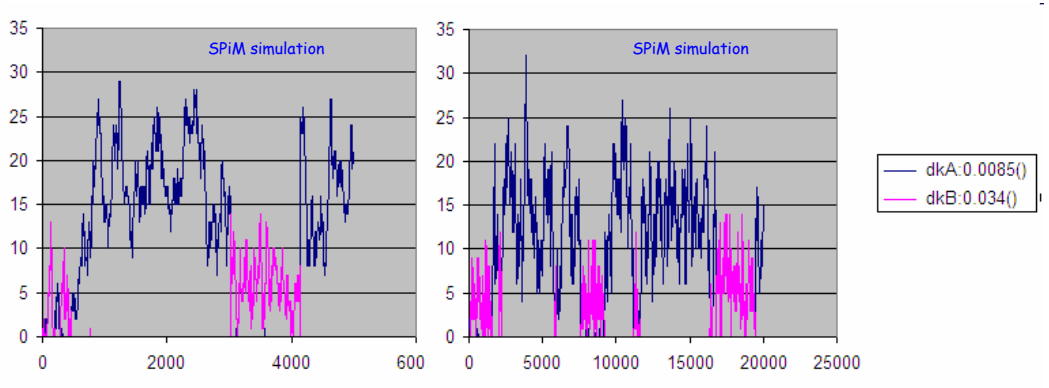


Fig 14A

François & Hakim Fig3A, SPiM simulation

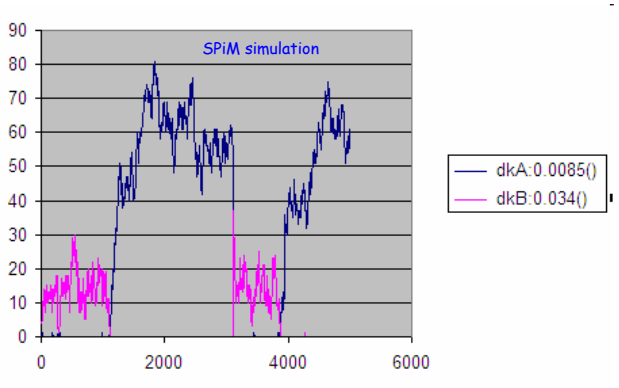
Parameters as in paper



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

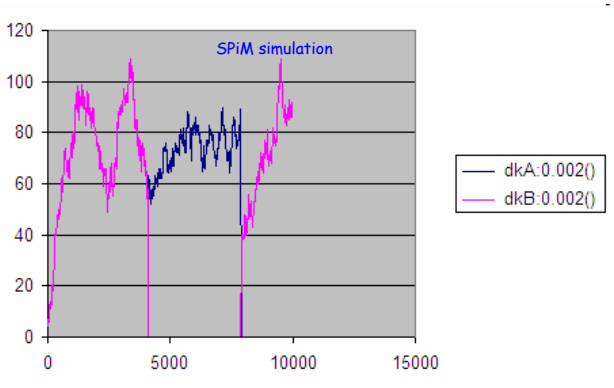
Free evolution

3 copies of each gene.

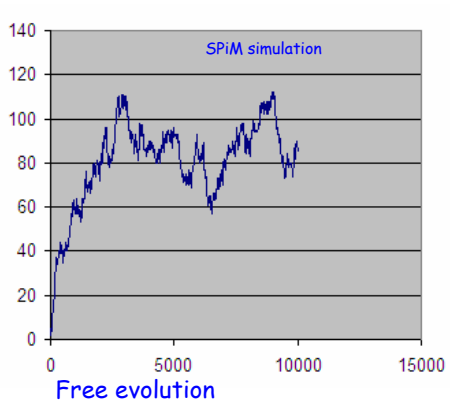


Spontaneous switch at ~1100
100xB injected at ~3000
30xA injected at ~4000

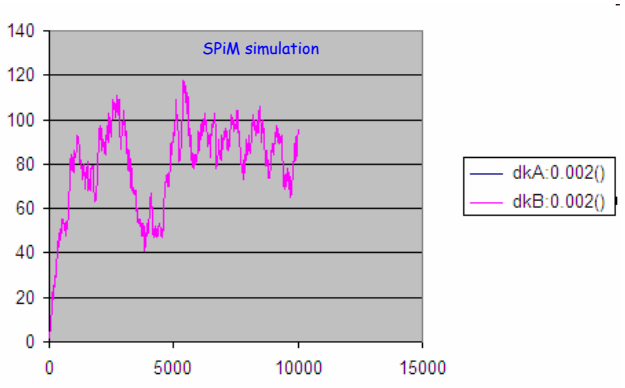
Modified for stability: dkA = 0.02, dkB = 0.02



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



Free evolution



François & Hakim Fig3Ast8

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

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

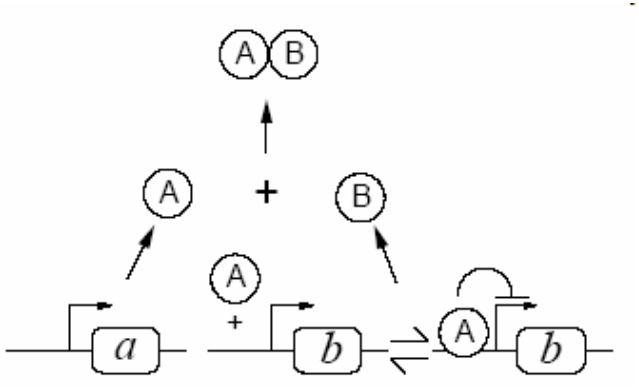


Fig 8

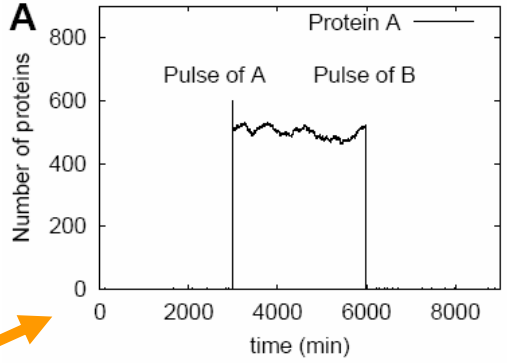
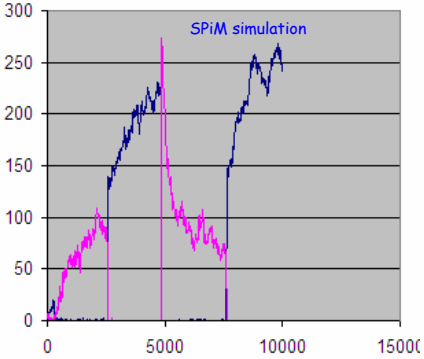
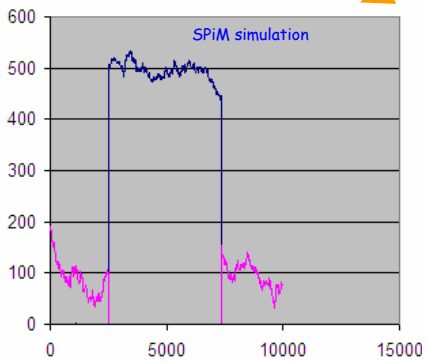


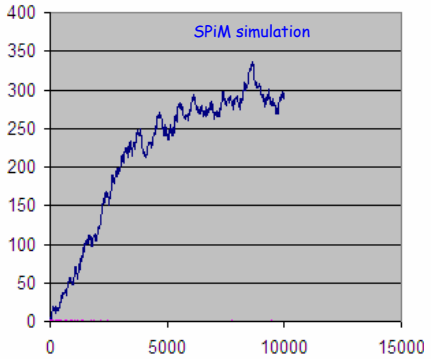
Fig 13A

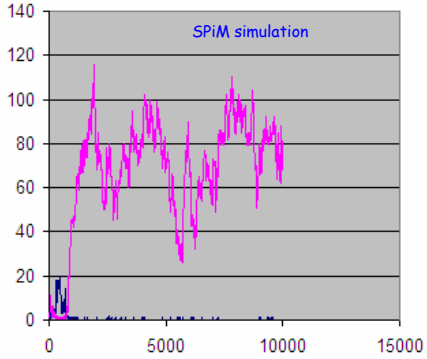


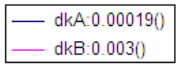
200xA injected at ~2500
500xB injected at ~5000
200xA injected at ~7500



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







Free evolution

François & Hakim 3A in SPiM

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

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

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

let ptnB() =
    do delay@dkB or ?AB;cpxAB()

let cpxAB() = delay@dkAB

let geneA() =
    delay@geneACst; (ptnA() | geneA())

let geneBfree() =
    do delay@geneBCst; (ptnB() | geneBfree())
    or ?bA(unb); geneBbound(unb)

and geneBbound(unb:ch()) =
    do delay@geneBInh; (ptnB() | geneBbound(unb))
    or !unb; geneBfree()

run (geneA() | geneBfree())
```

Interaction oriented

Scaling up: ODE vs Process Descriptions

From Chemical 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

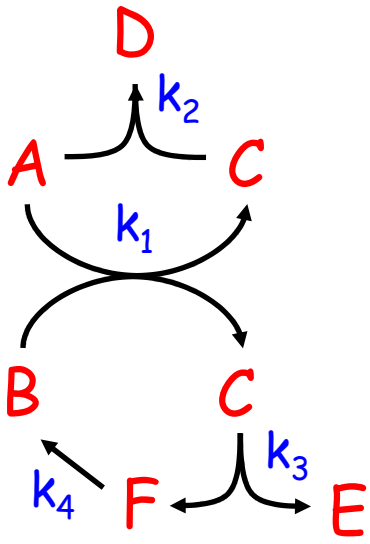
reactions

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

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

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 Chemical Reactions to Processes

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

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

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

r_4: $F \rightarrow_{k_4} B$

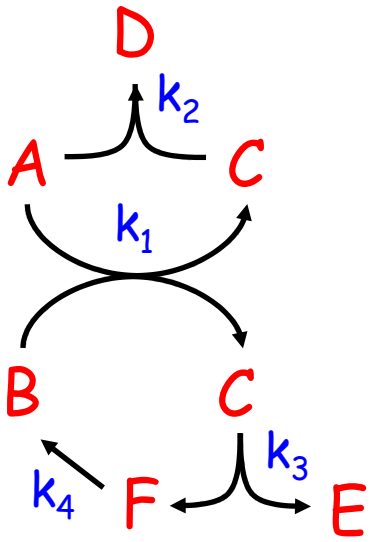
Write the coefficients by columns

interactions

processes

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

Stoichiometric Matrix



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

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

$B = !v_1k_1 + ?b$

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

$D = 0 + ?d$

$E = 0 + ?e$

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

Add a barb for counting and plotting

Read the process interactions from the rows

(Rate laws are implicit in stochastic semantics)

Stoichiometric Matrices Blow Up

- We can translate Chemistry to ODE's or Processes
  - It is standard to go from chemical equations to ODE's via a stoichiometric matrix.
  - It is similarly possible to go from chemical equations to processes via a stoichiometric matrix.
- But there is a better way:
  - Stoichiometric matrices blow-up exponentially for biochemical systems (unlike for ordinary chemical systems) because *proteins have combinatorial state* and *complexed states are common*.
  - To avoid this explosion, we should describe biochemical systems compositionally without going through a stochiometric matrix (and hence without ODE's).

Complexes: The ODE Way

n domains: A, B, C

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

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

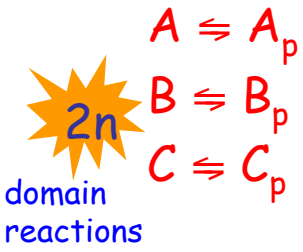
System description is exponential in the number of basic components.

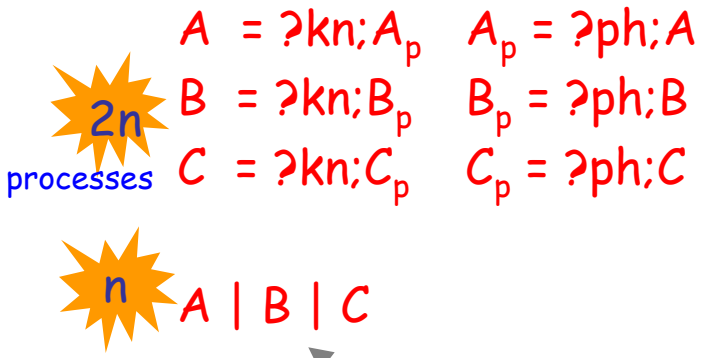
Stoichiometric Matrix

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

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

Complexes: The Reactive System Way





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

System description is linear in the number of basic components.

(Its "run-time" behavior or analysis potentially blows-up just as in the previous case, but its description does not.)

Model Validation

Model Validation: Simulation

- Basic stochastic algorithm: Gillespie
  - Exact (i.e. based on physics) stochastic simulation of chemical kinetics.
  - Can compute concentrations and reaction times for biochemical networks.

- Stochastic Process Calculi
  - BioSPi [Shapiro, Regev, Priami, et. al.]
    - Stochastic process calculus based on Gillespie.
  - BioAmbients [Regev, Panina, Silverma, Cardelli, Shapiro]
    - Extension of BioSpi for membranes.
  - Case study: Lymphocytes in Inflamed Blood Vessels [Lecaa, Priami, Quaglia]
    - Original analysis of lymphocyte rolling in blood vessels of different diameters.
  - Case study: Lambda Switch [Celine Kuttler, IRI Lille]
    - Model of phage lambda genome (well-studied system).
  - Case study: VICE [U. Pisa]
    - Minimal prokaryote genome (180 genes) and metabolism of *whole* VIrtual CEll, in stochastic π-calculus, simulated under stable conditions for 40K transitions.

- Hybrid approaches
  - Charon language [UPenn]
    - Hybrid systems: continuous differential equations + discrete/stochastic mode switching.
  - Etc.

Model Validation: "Program" Analysis

- Causality Analysis
  - *Biochemical pathways*, ("concurrent traces" such as the one here), are found in biology publications, summarizing known facts.
  - This one, however, was automatically generated from a program written in BioSpi by comparing traces of all possible interactions. [Curti, Priami, Degano, Baldari]
  - One can play with the program to investigate various hypotheses about the pathways.

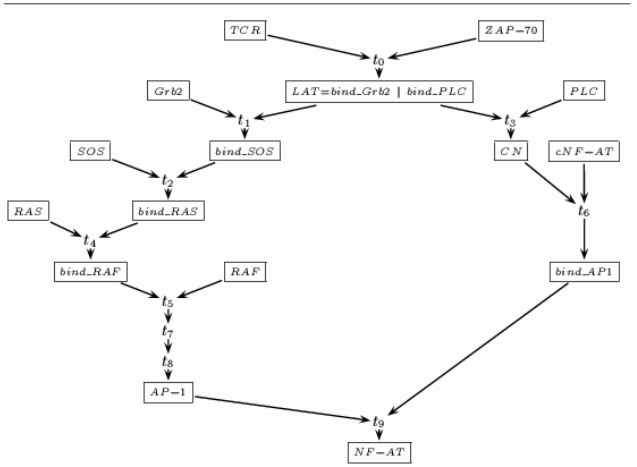


Fig. 2. A computation of *Sys*. For readability, the processes, enclosed in boxes, have no address. Causality (both on transitions and processes) is represented by the (Hasse diagram resulting from the) arrows; their absence makes it explicit concurrent activities.

- 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].

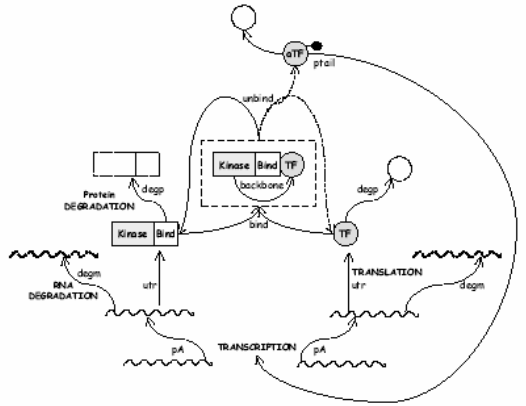


Fig. 1. Graphical presentation of Transcriptional Regulation by Positive Feedback [25].

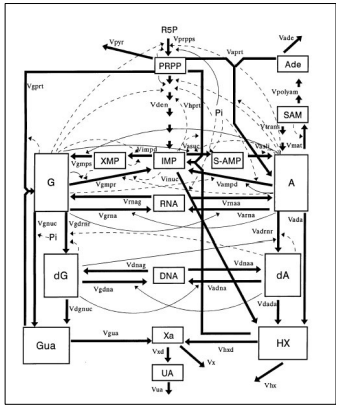
- Probabilistic Abstract Interpretation
  - [DiPierro Wicklicky].

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).

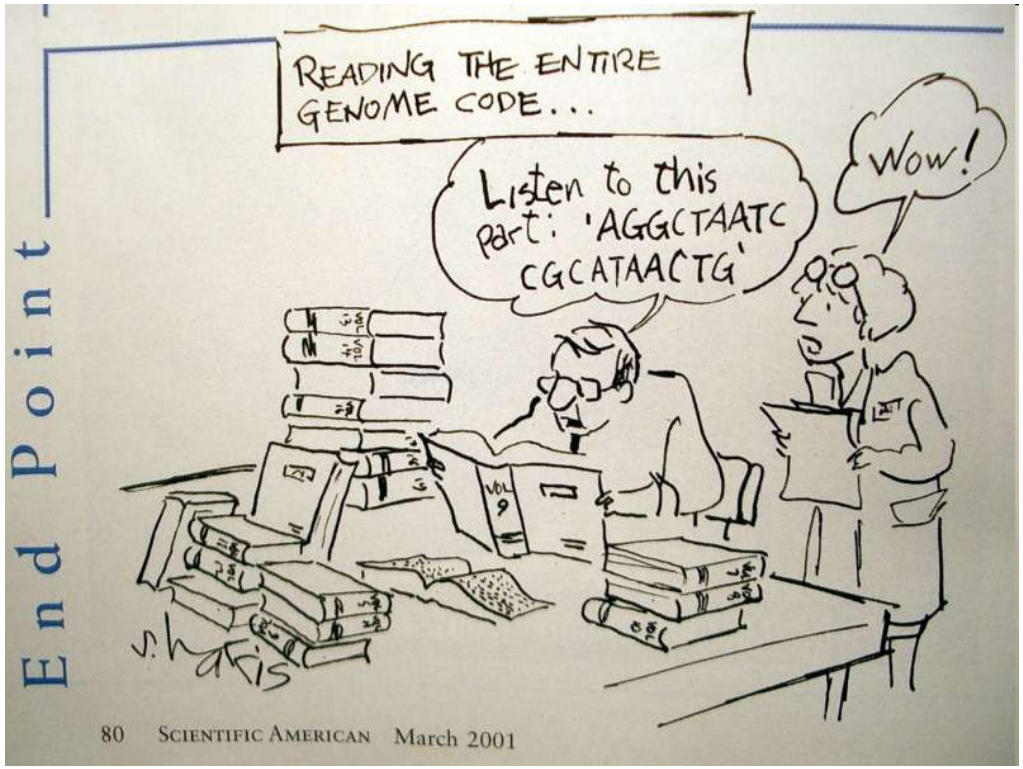
Disclaimers

- 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