

Biological Systems as Reactive Systems

Luca Cardelli

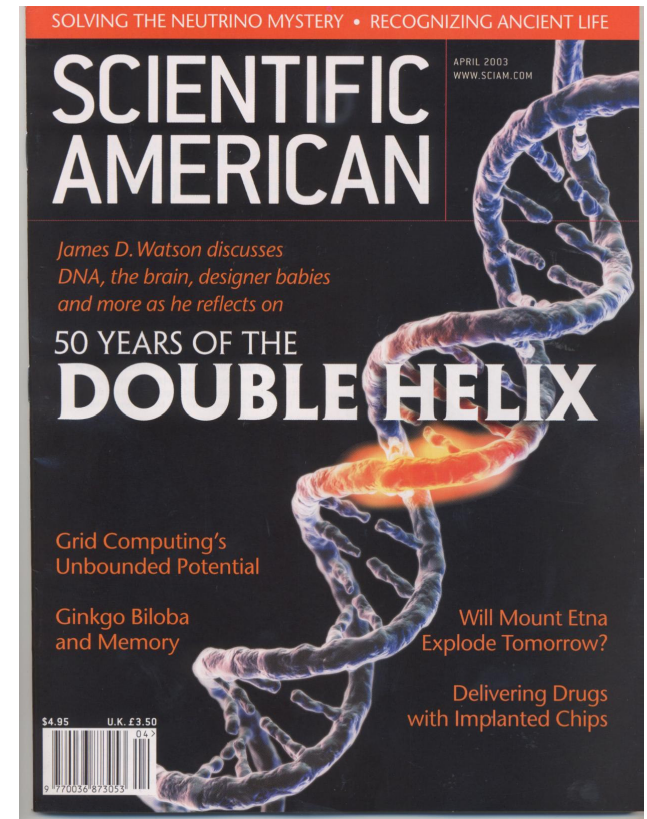
Microsoft Research
Cambridge UK

Cambridge 2005-05-04

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

50 Years of Molecular Cell Biology

- **Genes are made of DNA**
 - Store digital information as sequences of 4 different nucleotides
 - Direct protein assembly through RNA and the Genetic Code
- **Proteins (>10000) are made of amino acids**
 - Process signals
 - Activate genes
 - Move materials
 - Catalyze reactions to produce substances
 - Control energy production and consumption
- **Bootstrapping still a mystery**
 - DNA, RNA, proteins, membranes are today interdependent. Not clear who came first
 - Separation of tasks happened a long time ago
 - Not understood, not essential



Towards Systems Biology

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

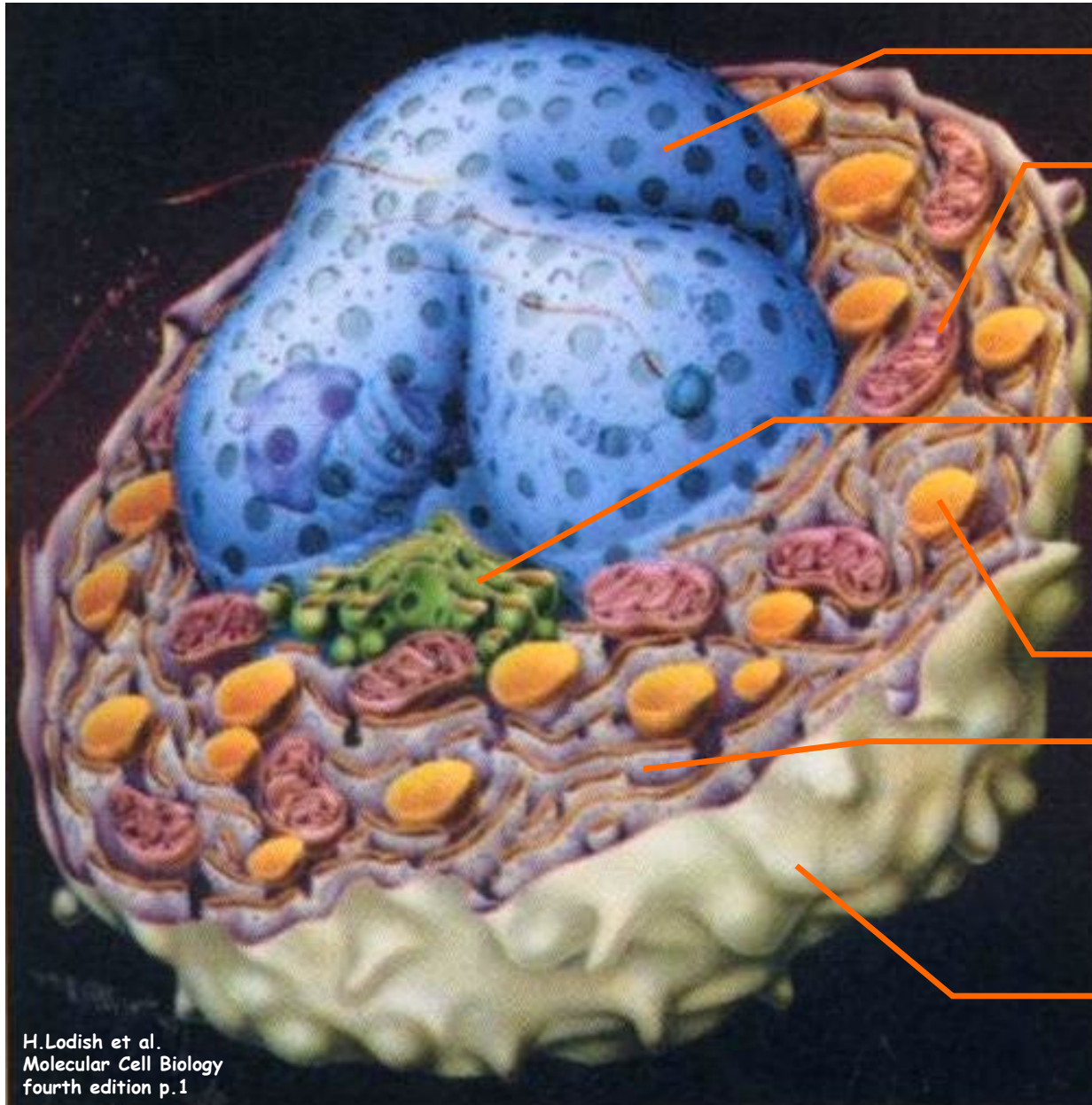
- Today we represent, store, search, and analyze:
 - Gene sequence data
 - Protein structure data
 - Metabolic network data
 - Signalling pathway data
 - ...
- How can we represent, store, and analyze *biological processes*?
 - Scalable, precise, dynamic, highly structured, maintainable representations 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

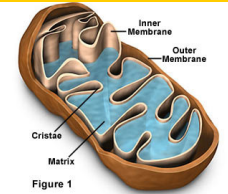
(10~100 trillion in human body)

Membranes everywhere

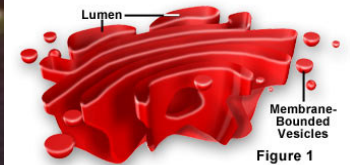


Nuclear membrane

Mitochondria

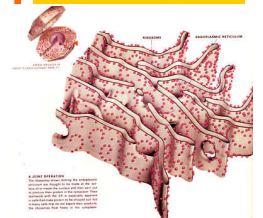


Golgi



Vesicles

E.R.



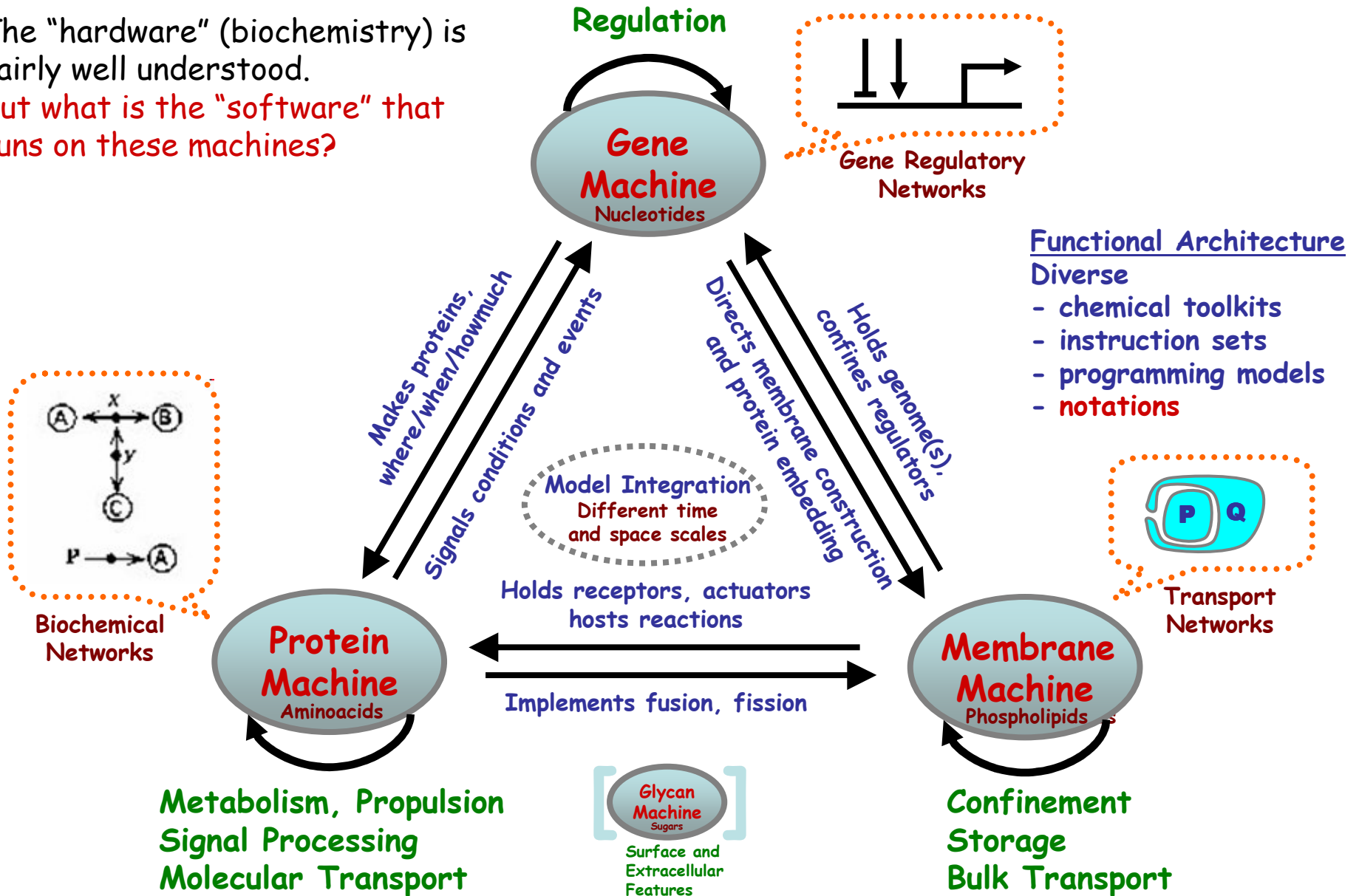
Plasma membrane (<10% of all membranes)



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

Abstract Machines of Systems Biology

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



Metabolism, Propulsion
 Signal Processing
 Molecular Transport

Confinement
 Storage
 Bulk Transport

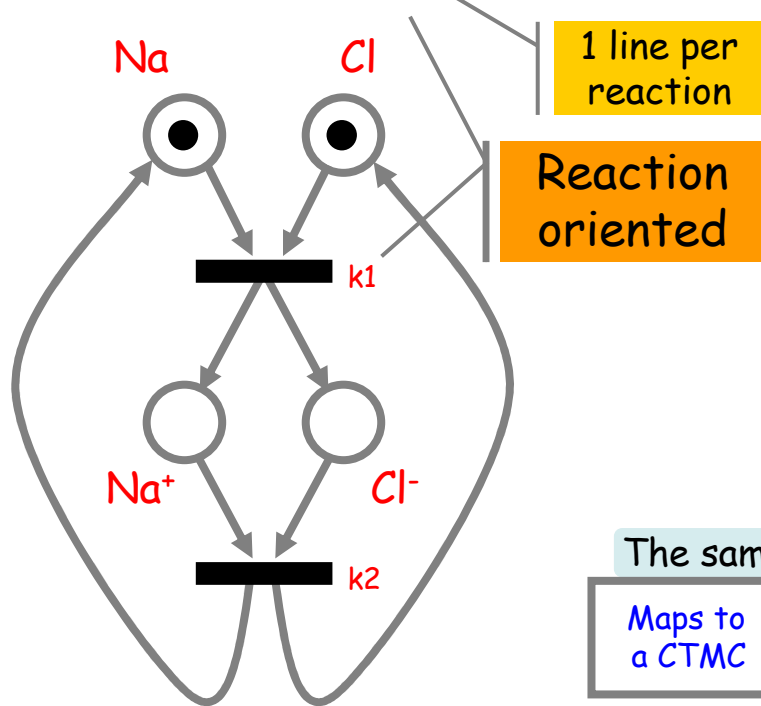
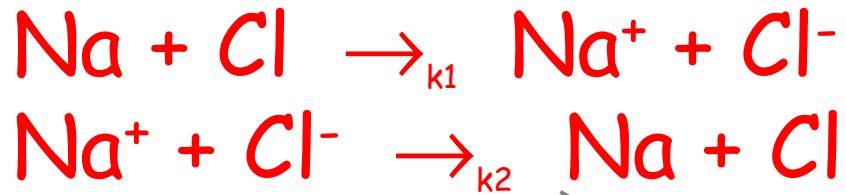
Surface and Extracellular Features

Reactive Systems

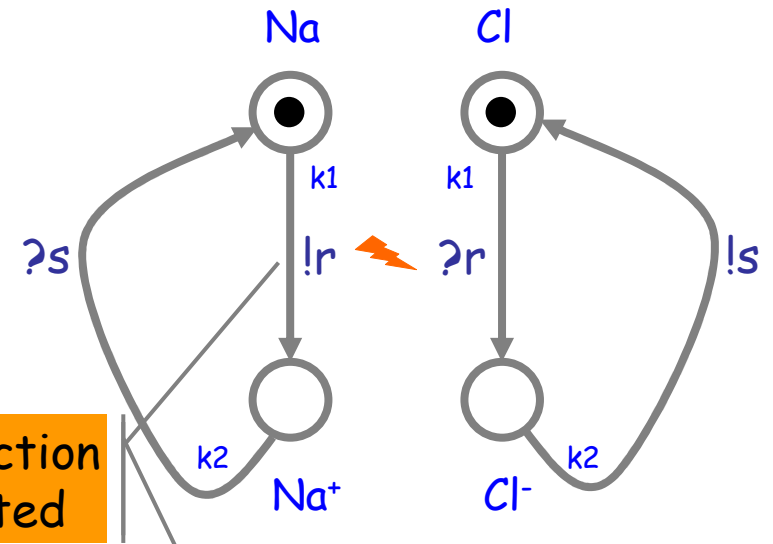
- **Modeling biological systems**
 - Not as continuous systems (often highly nonlinear)
 - But as discrete **reactive systems**; abstract machines with:
 - **States** represent situations
 - Event-driven **transitions** between states represent dynamics
 - The adequacy of describing (discrete) complex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



A compositional graphical representation, and the corresponding calculus.



Interaction oriented

1 line per component

The same "model"

Maps to a CTMC	Maps to a CTMC
----------------	----------------

$$\text{Na} = !r_{k_1}; ?s_{k_2}; \text{Na}$$

$$\text{Cl} = ?r_{k_1}; !s_{k_2}; \text{Cl}$$

A different process calculus (π)

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

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 Computation":

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

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

$$\pi ::= x(y) \text{ receive } y \text{ along } x \\ \bar{x}(y) \text{ send } y \text{ along } x$$

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

Structural congruence

Renaming of bound variables

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

Structural congruence laws

$P \mid Q \equiv Q \mid P$	commutativity of parallel composition
$(P \mid Q) \mid R \equiv P \mid (Q \mid R)$	associativity of parallel composition
$P + Q \equiv Q + P$	commutativity of summation
$(P + Q) + R \equiv P + (Q + R)$	associativity of summation
$(\text{new } x)0 \equiv 0$	restriction of inert processes
$(\text{new } x)(\text{new } y)P \equiv (\text{new } y)(\text{new } x)P$	polyadic restriction
$((\text{new } x)P) \mid Q \equiv (\text{new } x)(P \mid Q)$	scope extrusion
$!P \equiv P \mid P$	replication

Reaction rules

$$(\dots + \bar{x}(z).Q) \mid (\dots + x(y).P) \rightarrow Q \mid P \{z/y\} \quad \text{communication (COMM)}$$

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

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

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

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).

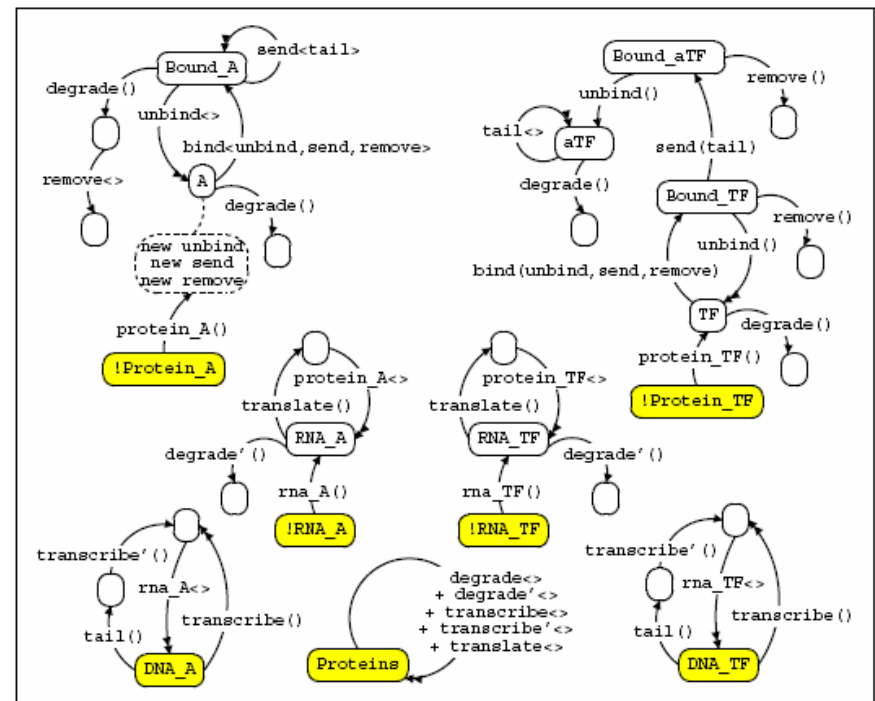


Figure 2. Regulating Gene Expression by Positive Feedback [9]

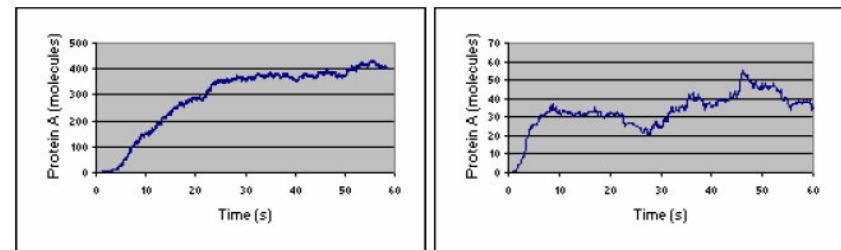


Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of TF

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**.

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

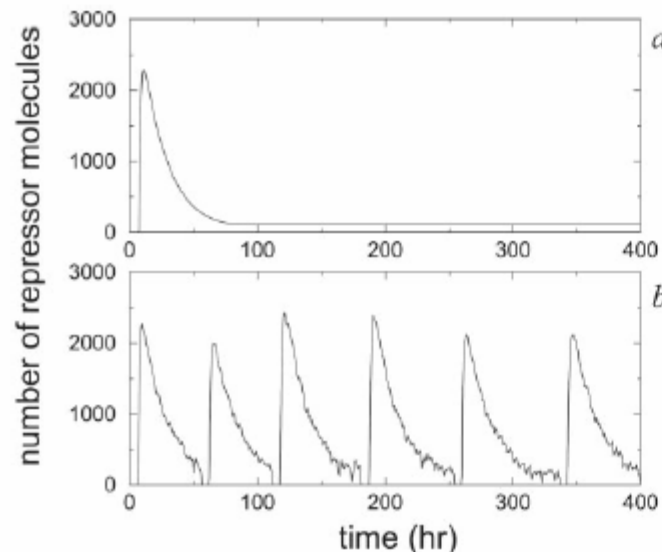


Fig. 5. Time evolution of R for the deterministic Eq. [1] (a) and stochastic (b) versions of the model. The values of the parameters are as in the caption of Fig. 1, except that now we set $\delta_R = 0.05 \text{ h}^{-1}$. For these parameter values, $\tau < 0$, so that the fixed point is stable.

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

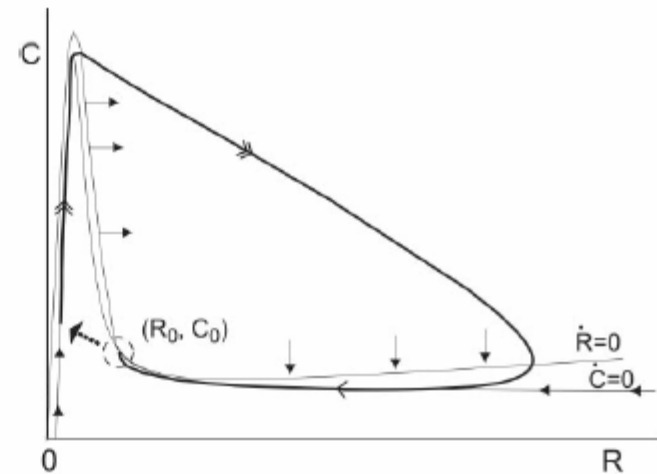
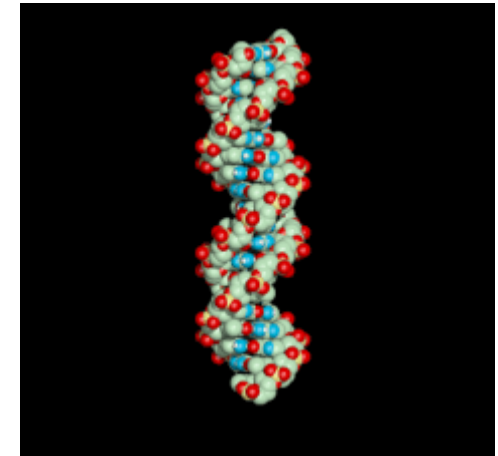
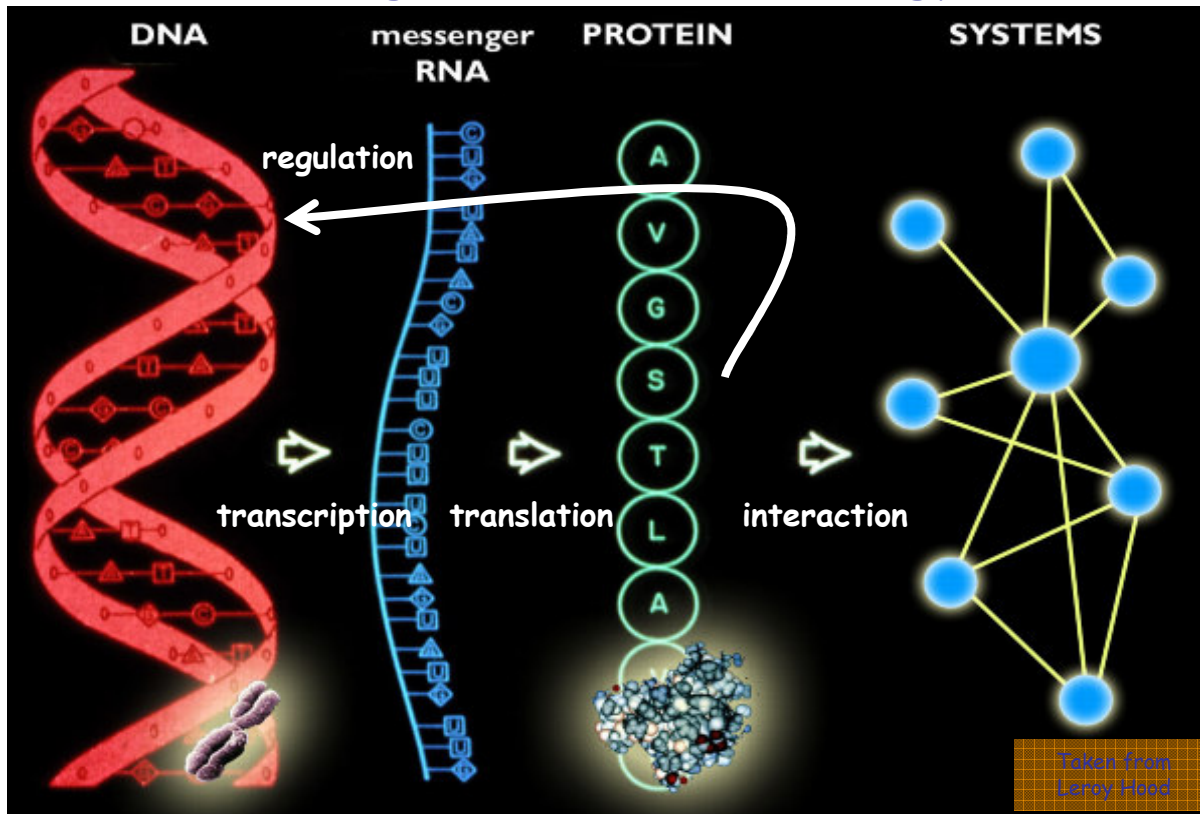


Fig. 6. Phase portrait as in Fig. 4 but for a situation in which the system falls into the stable fixed point (R_0, C_0) . The dotted arrow to the left of the fixed point illustrates a perturbation that would initiate a single sweep of the (former) oscillatory trajectory.

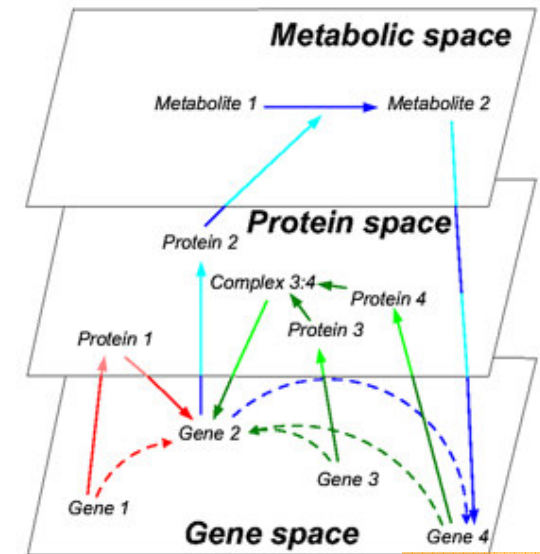
Gene Networks

The Gene Machine

The "Central Dogma" of Molecular Biology

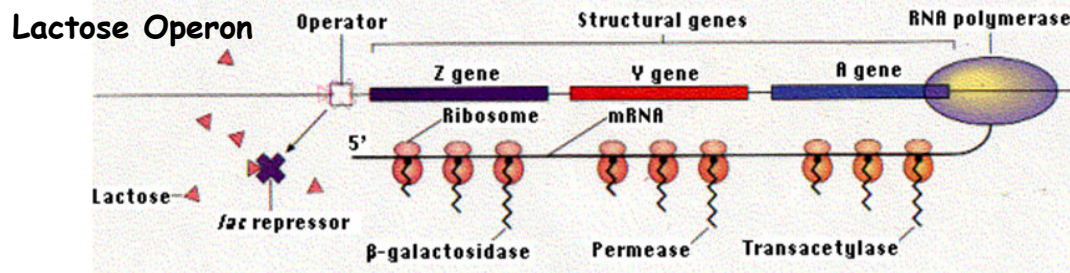


[DNA Tutorial](#)



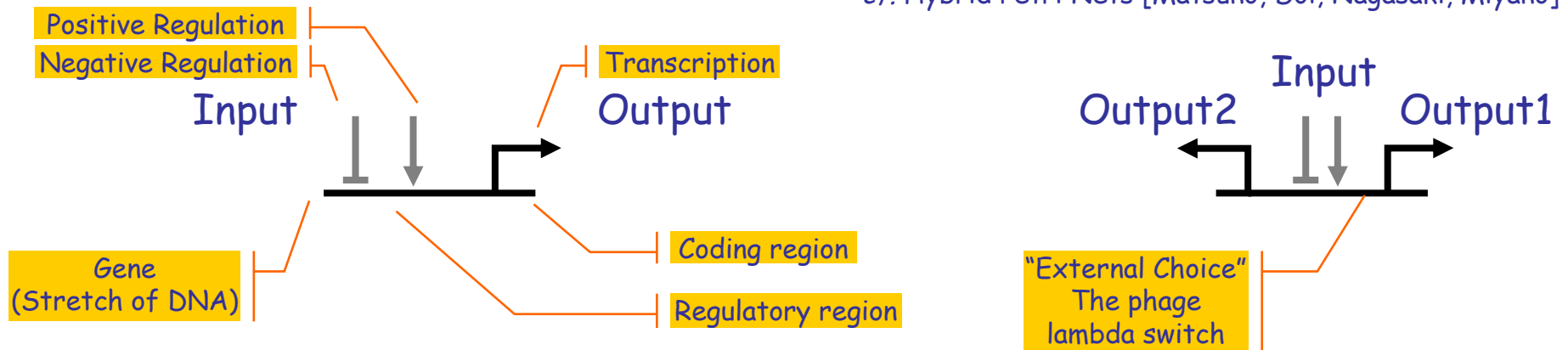
Taken from Pedro Mendes

2005-05-04



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)

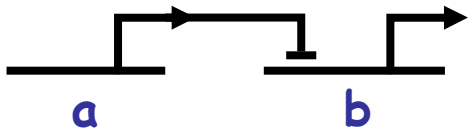
580,073bp 145KB (eBook)

E.Coli (bacteria): 4Mbp 1MB (floppy)

Yeast (eukarya): 12Mbp 3MB (MP3 song)

Wheat 17Gbp 4.25GB (DVD)

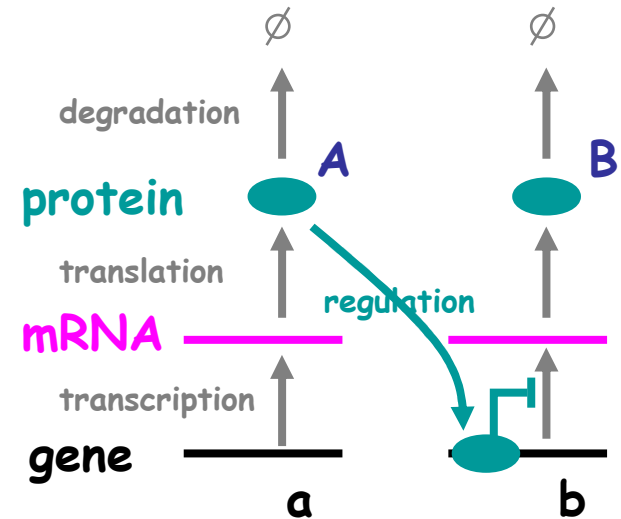
Gene Composition



Is a shorthand for:

Under the assumptions [Kim & Tidor]

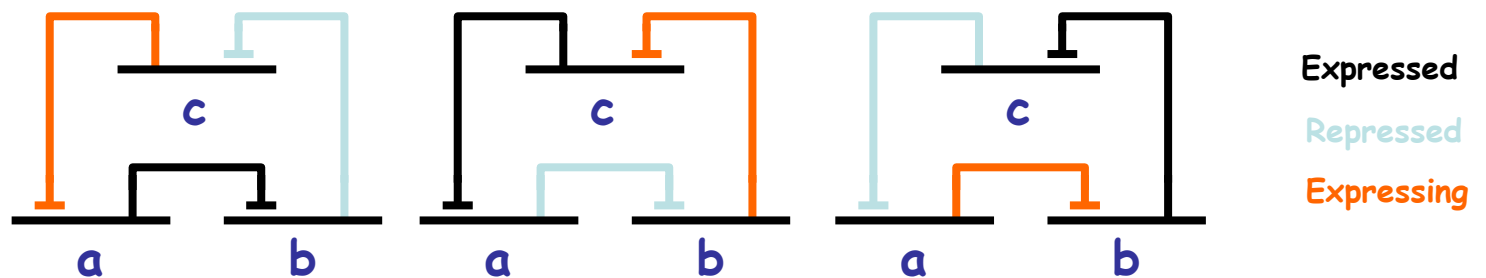
- 1) The solution is well-stirred
(no spatial dependence on concentrations or rates).
- 2) There is no regulation cross-talk.
- 3) Control of expression is at transcription level only
(no RNA-RNA or RNA-protein effects)
- 4) Transcriptions and translation rates monotonically affect mRNA and protein concentrations resp.



Ex: Bistable Switch



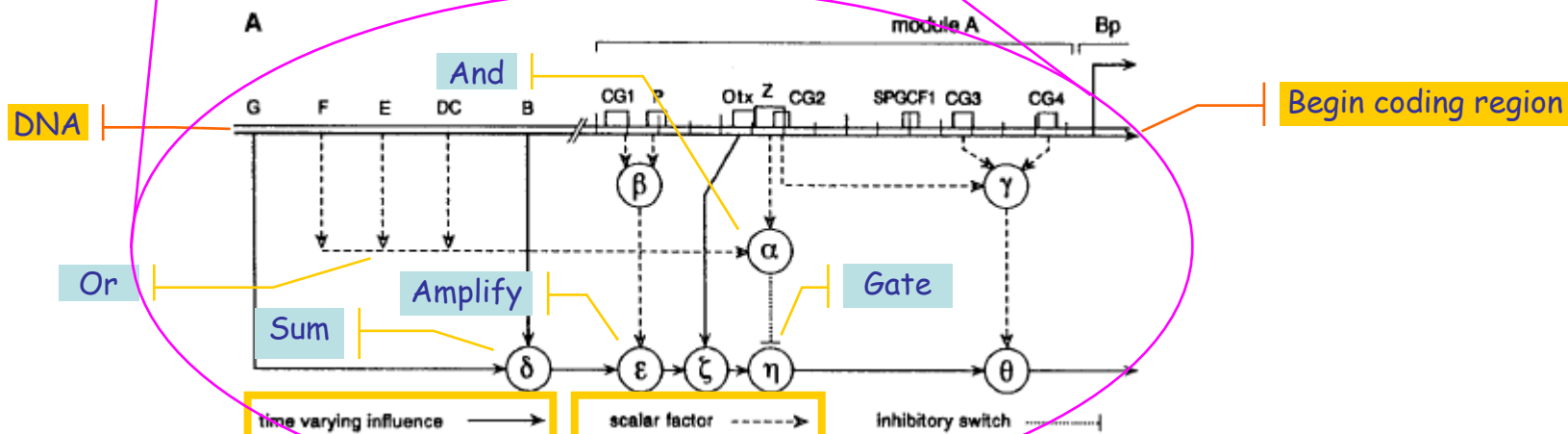
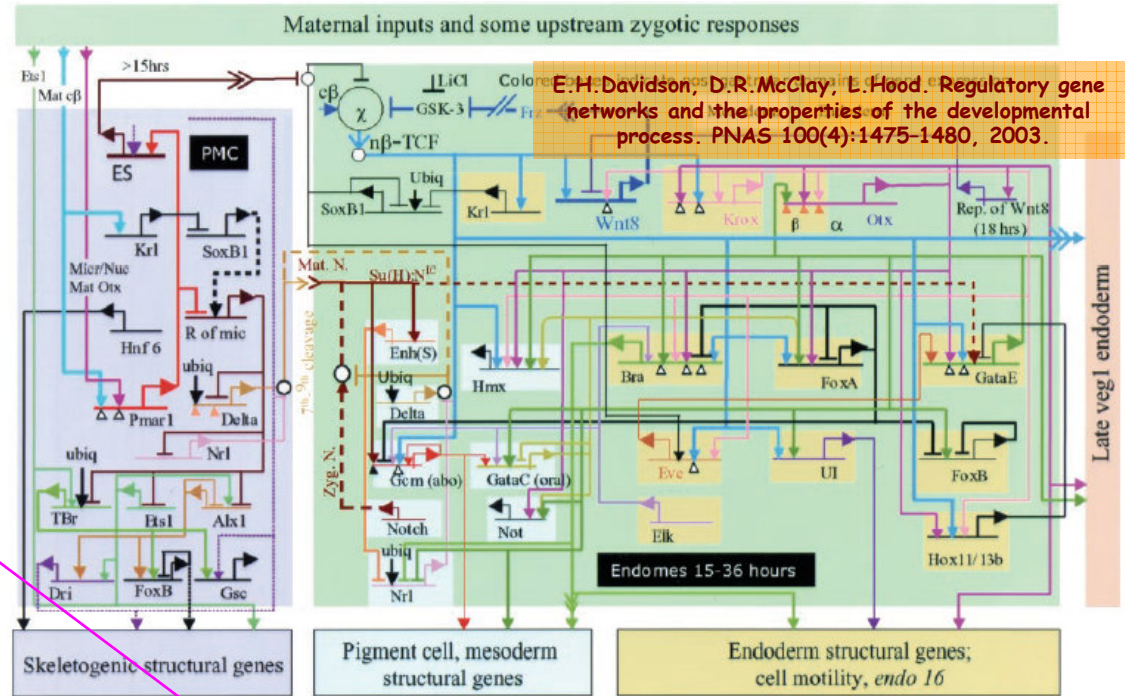
Ex: Oscillator



Gene Regulatory Networks

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

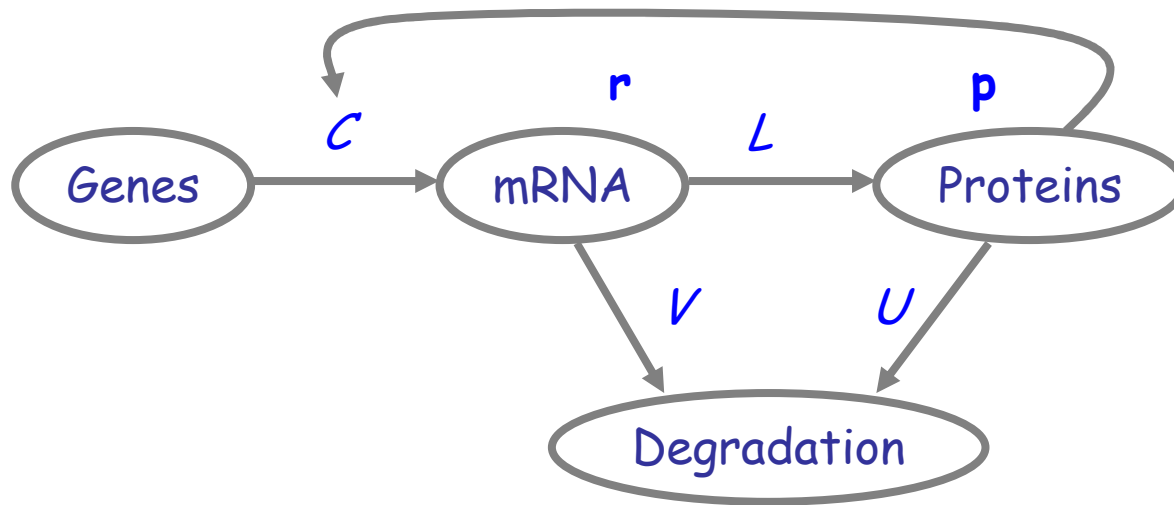
NetBuilder



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]



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

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

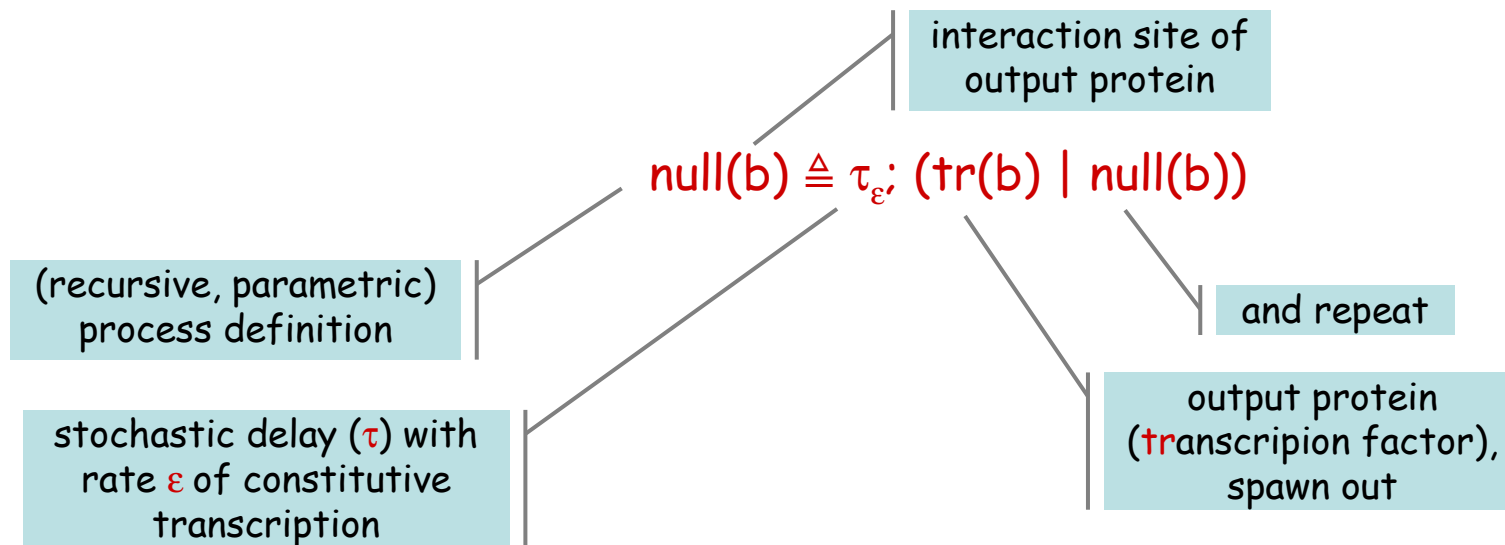
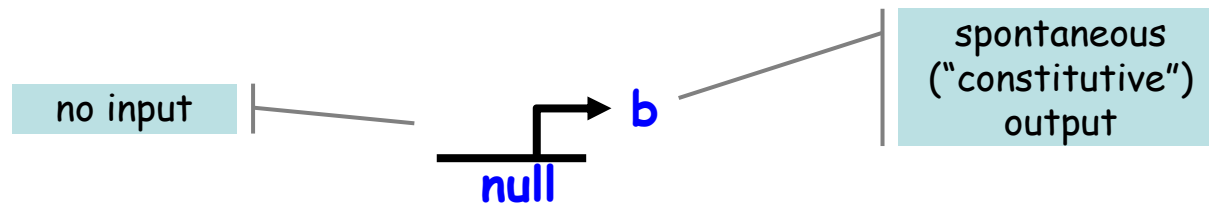
n : number of genes

\mathbf{r} mRNA concentrations (n-dim vector)

\mathbf{p} protein concentrations (n-dim vector)

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

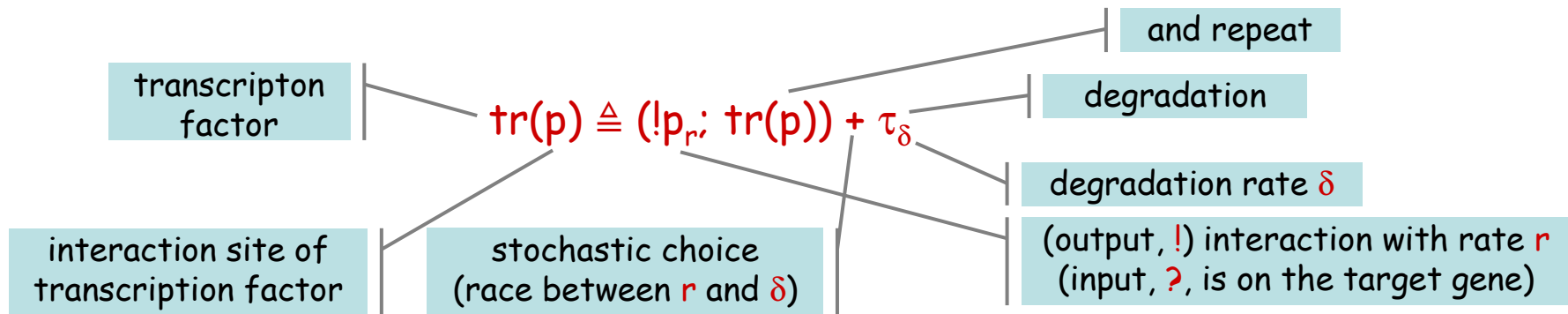
Nullary Gate



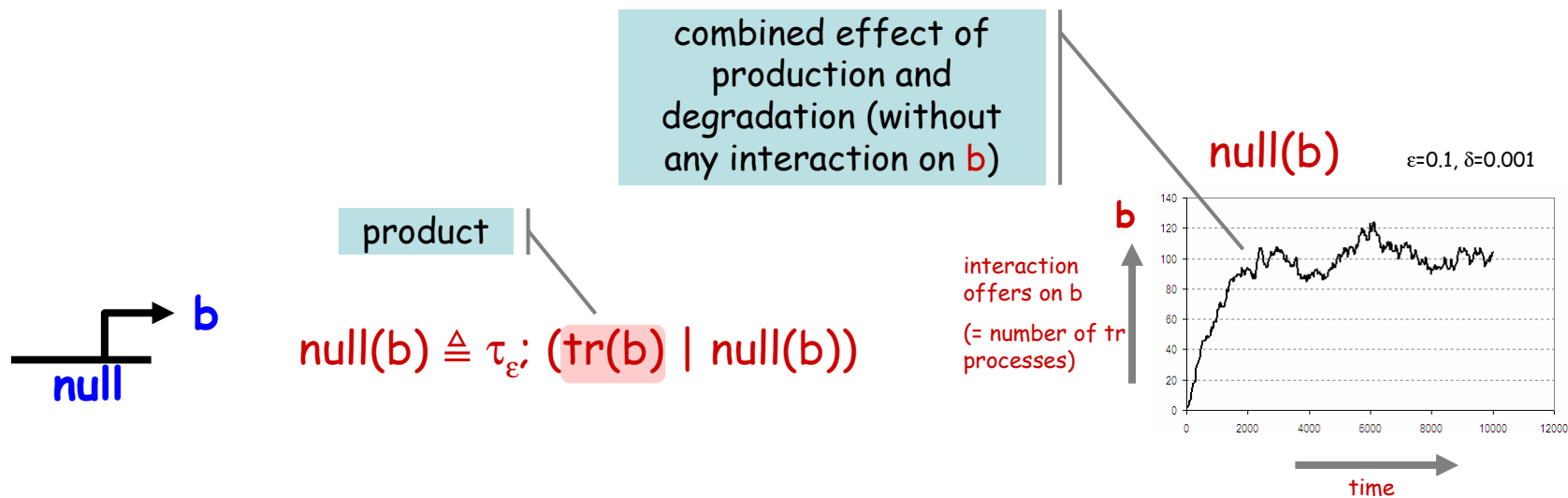
A stochastic rate r is always associated with each channel a_r (at channel creation time) and delay τ_r , but is often omitted when unambiguous.

Production and Degradation

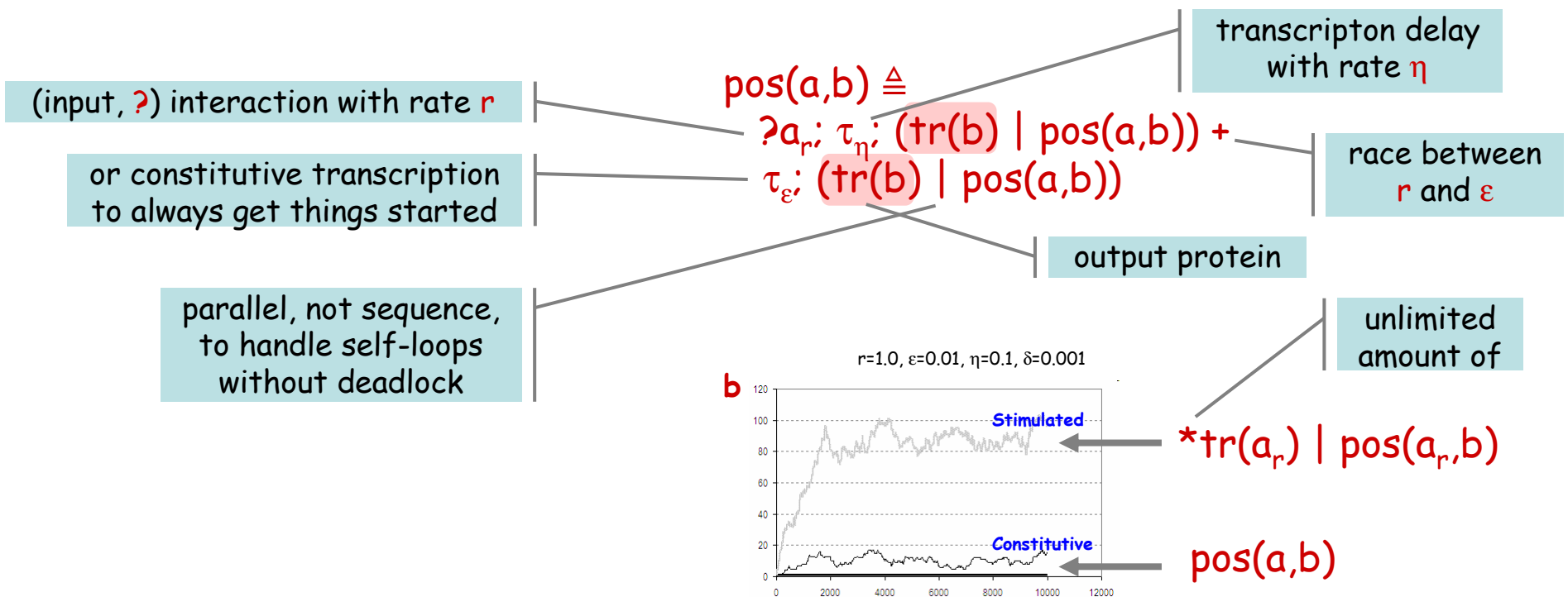
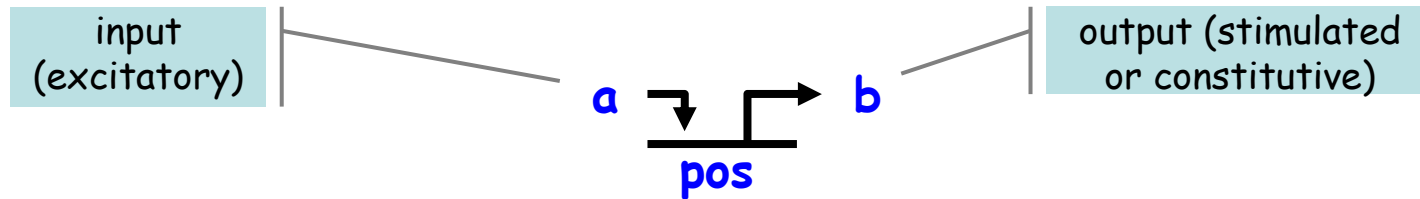
Degradation is extremely important and often deliberate; it changes unbounded growth into (roughly) stable signals.



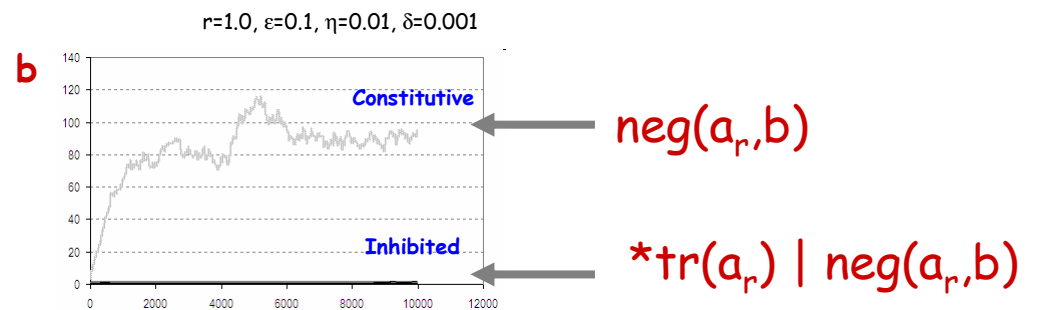
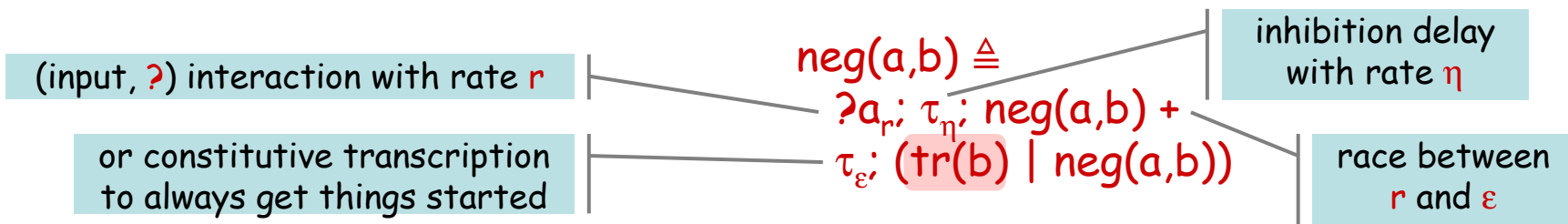
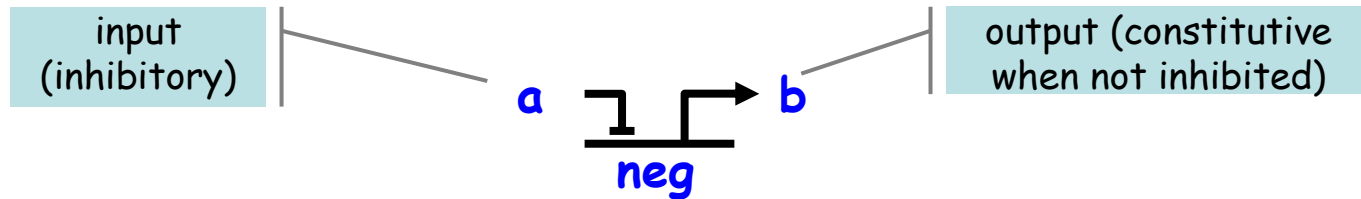
A transcription factor is a *process* (not a message or a channel): it has behavior such as interaction on p and degradation.



Unary Pos Gate

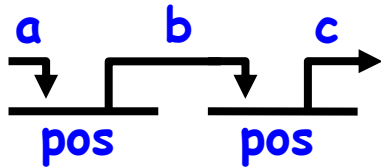


Unary Neg Gate



Signal Amplification

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



$$\text{pos}(a,b) \triangleq$$

$$\tau_{a_r}; \tau_{\eta}; (\text{tr}(b) | \text{pos}(a,b)) +$$

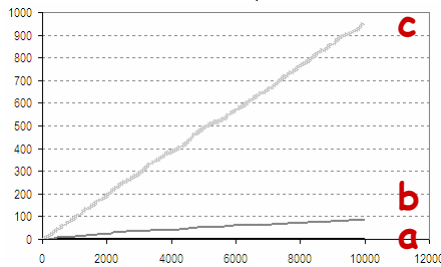
$$\tau_{\varepsilon}; (\text{tr}(b) | \text{pos}(a,b))$$

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

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

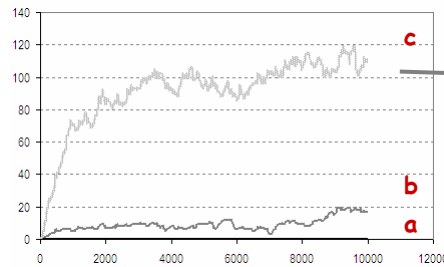
With little degradation

$r=1.0, \varepsilon=0.01, \eta=0.1, \delta=0.00001$



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

$r=1.0, \varepsilon=0.01, \eta=0.1, \delta=0.001$

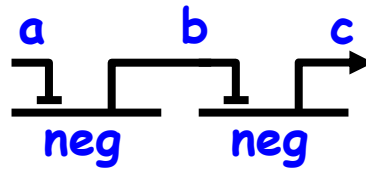


pos(a,b) | pos(b,c)

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

Signal Normalization

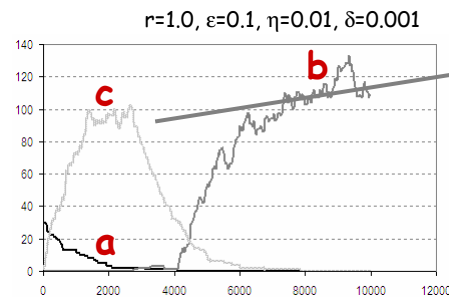
neg(a,b) |
neg(b,c)



$$\text{neg}(a,b) \triangleq$$

$$\begin{aligned} & \tau_{a_r}; \tau_{h_r}; \text{neg}(a,b) + \\ & \tau_{\epsilon_r}; (\text{tr}(b) | \text{neg}(a,b)) \end{aligned}$$

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

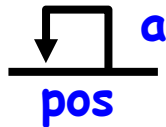


a non-zero input level, **a**,
whether weak or strong,
is renormalized to a
standard level, **c**.

30*tr(a) | neg(a,b) | neg(b,c)

Self Feedback Circuits

pos(a,a)



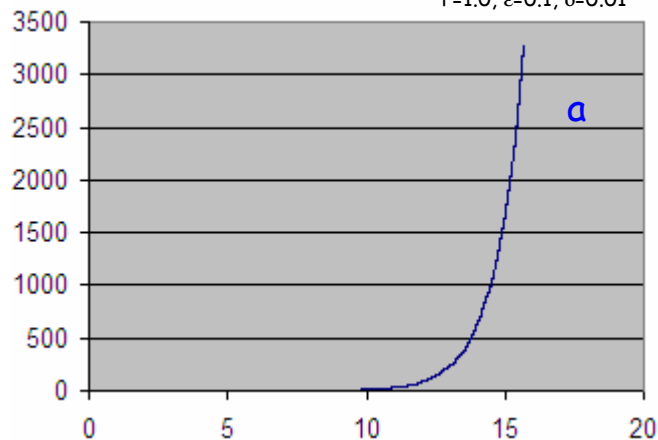
$$\text{pos}(a,b) \triangleq$$

$$\begin{aligned} & \tau_{a_r}; (\text{tr}(b) \mid \text{pos}(a,b)) + \\ & \tau_{\varepsilon}; (\text{tr}(b) \mid \text{pos}(a,b)) \end{aligned}$$

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

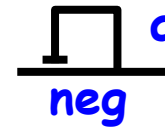
(Can overwhelm degradation, depending on parameters)

$r=1.0, \varepsilon=0.1, \delta=0.01$



pos(a,a)

neg(a,a)



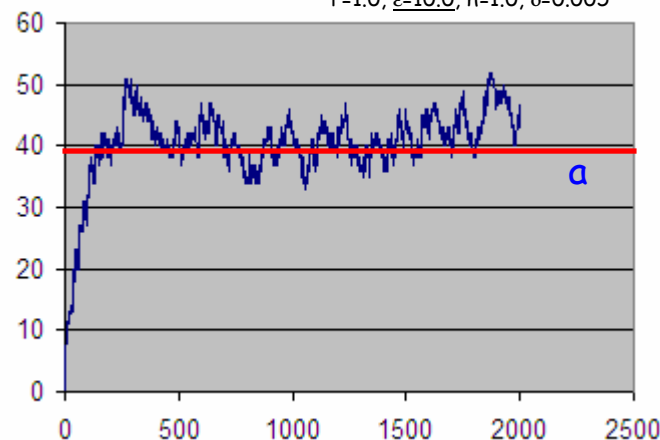
$$\text{neg}(a,b) \triangleq$$

$$\begin{aligned} & \tau_{a_r}; \tau_{h_i}; \text{neg}(a,b) + \\ & \tau_{\varepsilon}; (\text{tr}(b) \mid \text{neg}(a,b)) \end{aligned}$$

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

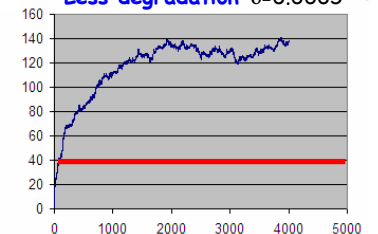
high, to raise the signal

$r=1.0, \varepsilon=10.0, h=1.0, \delta=0.005$

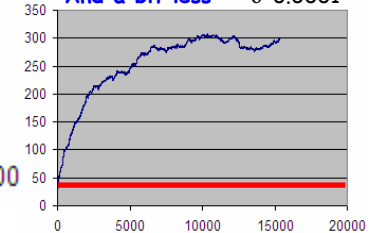


neg(a,a)

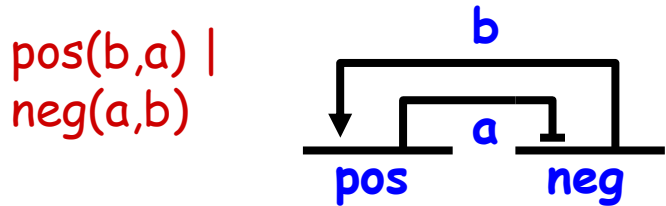
Less degradation $\delta=0.0005$



And a bit less $\delta=0.0001$

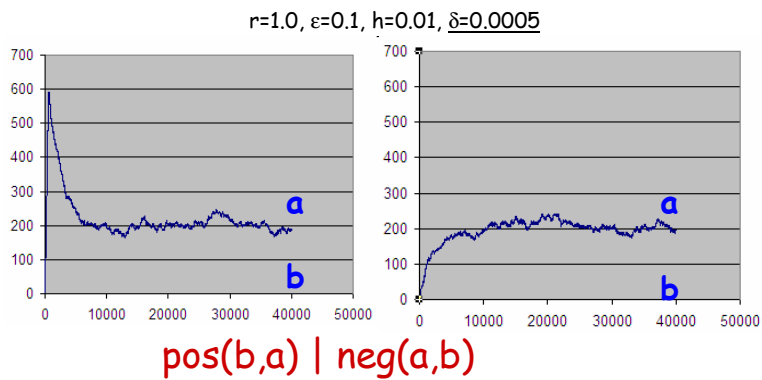


Two-gate Feedback Circuits

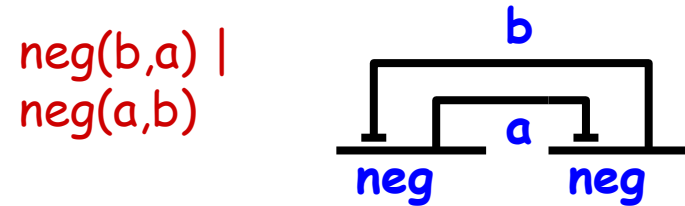
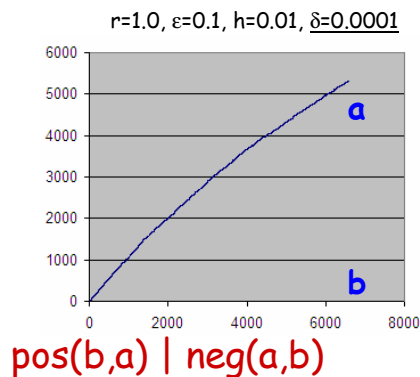


Monostable:

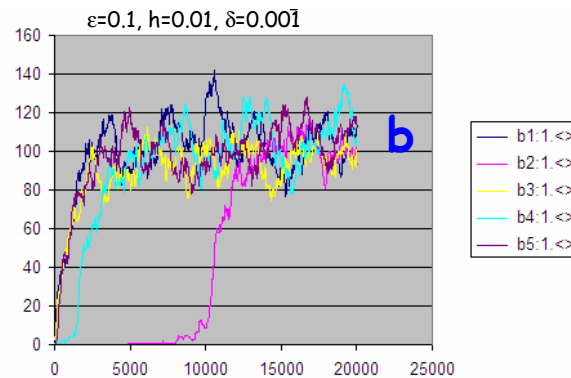
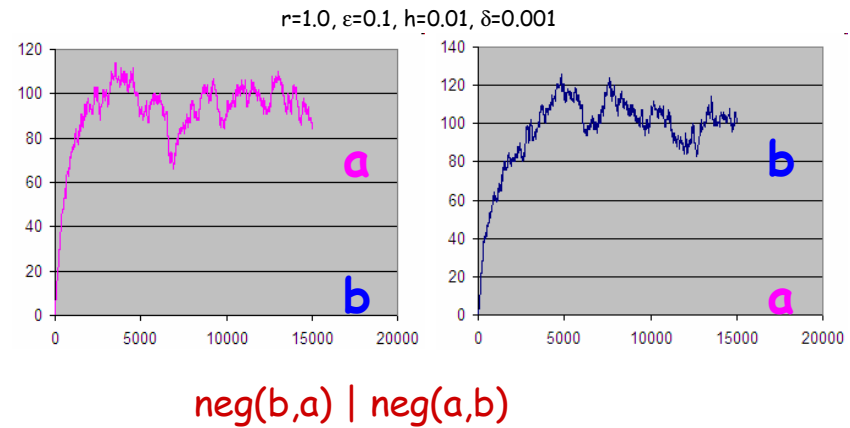
For some degradation rates is quite stable:



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



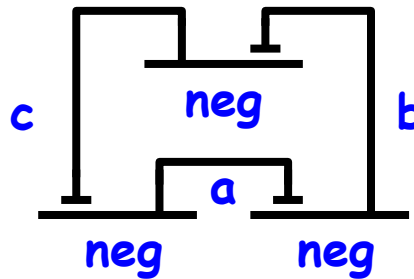
Bistable:



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

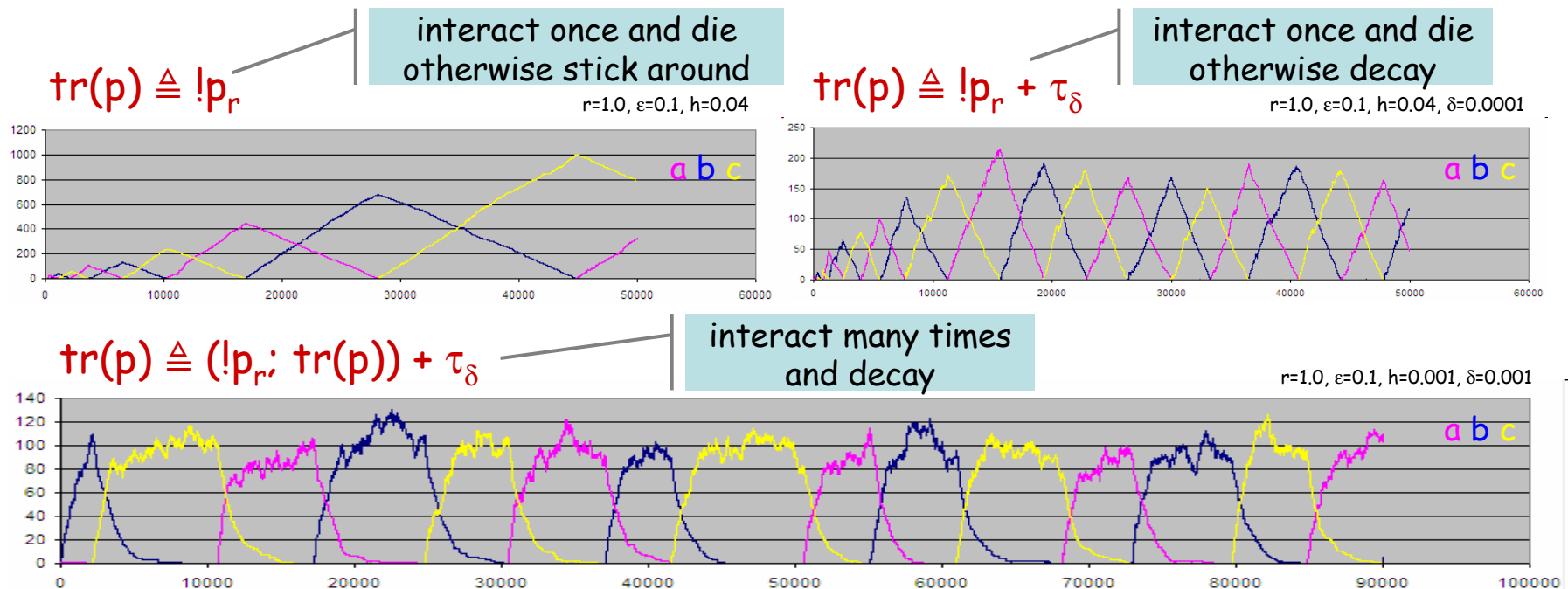
Repressilator

$neg(a,b) \mid$
 $neg(b,c) \mid$
 $neg(c,a)$



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

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



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

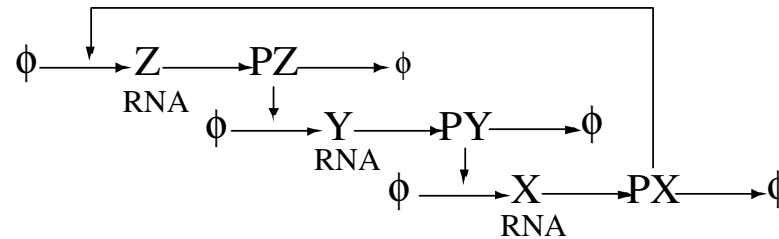
let neg(a:chan(), b:chan()) =
  do ?a; delay@eta; neg(a,b)
  or delay@cst; (tr(b) | neg(a,b))

(* 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

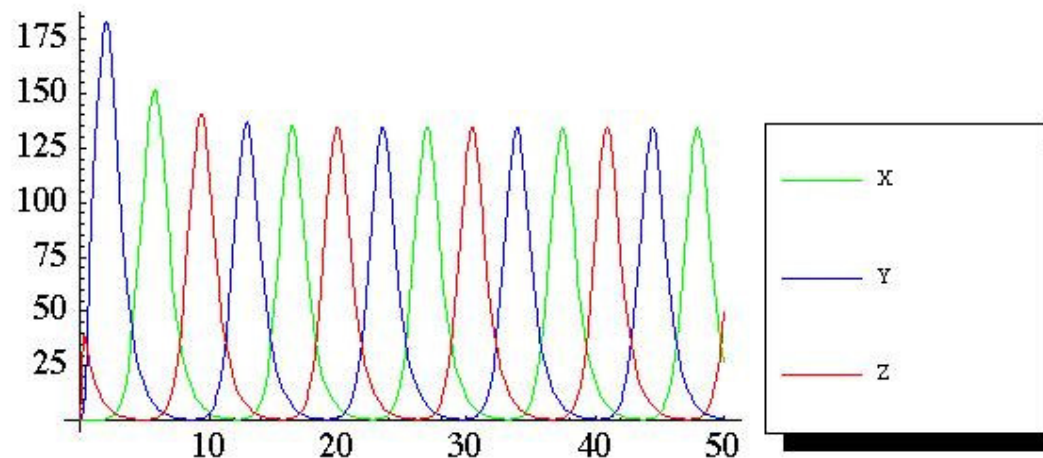
*Bruce E Shapiro
Cellerator*



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

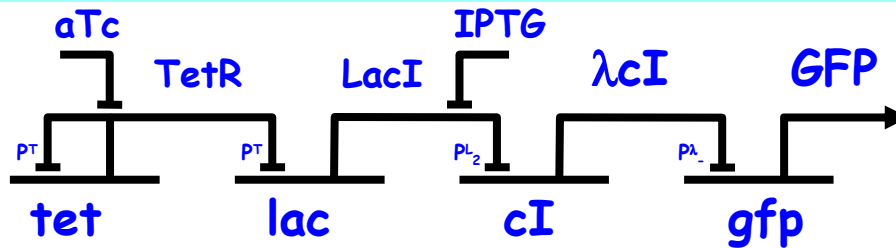
$$\frac{d[Y]}{dt} = \alpha_0 + \frac{\alpha + \alpha_1 [PZ]^n}{K^n + [PZ]^n} - k[Y], \quad \frac{d[PY]}{dt} = \beta\{[Y] - [PY]\}$$

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



Guet et al.: D038/lac⁻

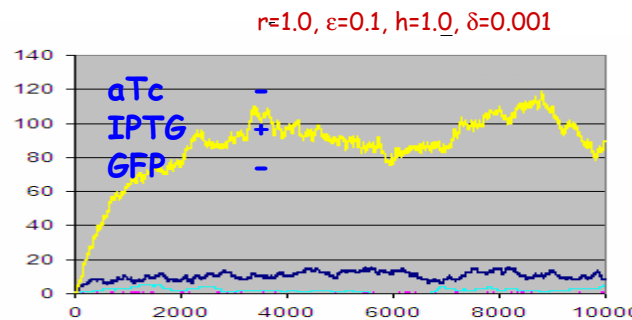
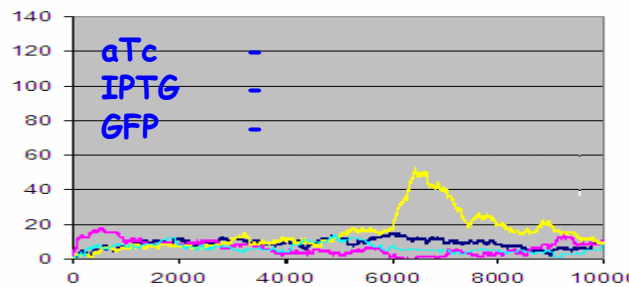
Combinatorial Synthesis of Genetic Networks, Guet, Elowitz, Hsing, Leibler, 1996, *Science*, May 2002, 1466-1470.



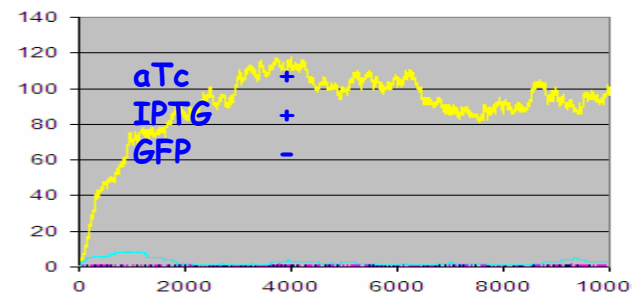
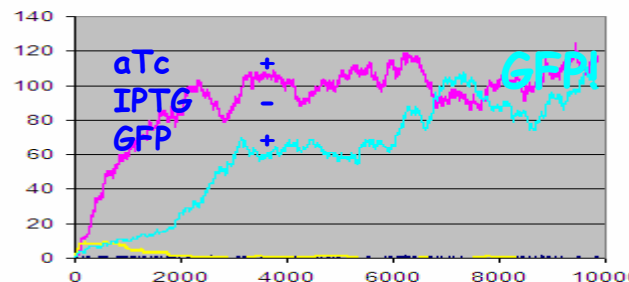
experiment:

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

$neg(TetR, TetR) \mid neg(TetR, LacI) \mid neg(LacI, \lambda CI) \mid neg(\lambda CI, GFP)$



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.

Biochemistry: Huang and Ferrell

Proc. Natl. Acad. Sci. USA 93 (1996)

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

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

The assumed K_m values for each reaction were individually varied over the ranges shown, with the assumed K_m values for the other nine reactions held constant. The effective Hill coefficients were calculated from the steepness of the predicted stimulus/response curves, as described in the text.

[†]The K_m value for reaction 7 has been measured to be 300 nM for the phosphorylation of a mammalian MAPK by a MAPKK (N. Ahn, personal communication). All of the other K_m values were initially assumed to be 300 nM as well.

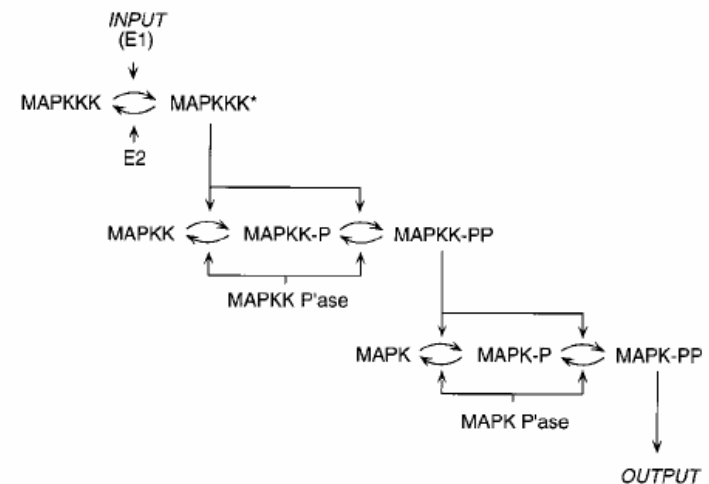
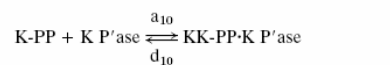
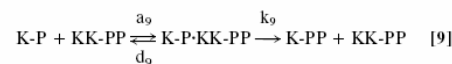
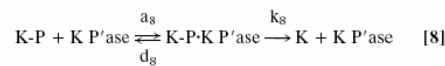
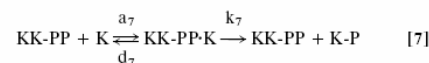
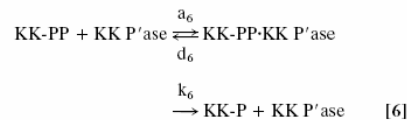
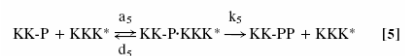
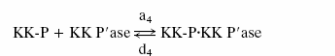
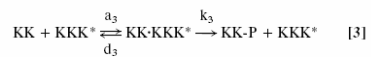
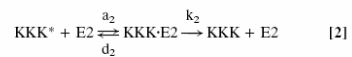
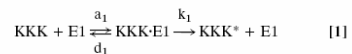


FIG. 1. Schematic view of the MAPK cascade. Activation of MAPK depends upon the phosphorylation of two conserved sites [Thr-183 and Tyr-185 in rat p42 MAPK/Erk2 (4, 5)]. Full activation of MAPKK also requires phosphorylation of two sites [Ser-218 and Ser-222 in mouse Mek-1/MKK1 (6–10)]. Detailed mechanisms for the activation of various MAPKKs (e.g., Raf-1, B-Raf, Mos) are not yet established; here we assume that MAPKKs 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.



As 18 Ordinary Differential Equations Plus 7 conservation equations

$$\frac{d}{dt}[\text{KKK}] = -a_1[\text{KKK}][\text{E1}] + d_1[\text{KKK}\cdot\text{E1}] + k_2[\text{KKK}^*\cdot\text{E2}] \quad [111]$$

$$\frac{d}{dt}[\text{KKK}\cdot\text{E1}] = a_1[\text{KKK}][\text{E1}] - (d_1 + k_1)[\text{KKK}\cdot\text{E1}] \quad [112]$$

$$\frac{d}{dt}[\text{KKK}^*] = -a_2[\text{KKK}^*][\text{E2}] + d_2[\text{KKK}^*\cdot\text{E2}] + k_1[\text{KKK}\cdot\text{E1}] + (k_3 + d_3)[\text{KK}\cdot\text{KKK}^*] - a_3[\text{KKK}^*][\text{KK}] + (k_5 + d_5)[\text{KK}\cdot\text{P}\cdot\text{KKK}^*] - a_4[\text{KK}\cdot\text{P}][\text{KKK}^*] \quad [113]$$

$$\frac{d}{dt}[\text{KKK}^*\cdot\text{E2}] = a_2[\text{KKK}^*][\text{E2}] - (d_2 + k_2)[\text{KKK}^*\cdot\text{E2}] \quad [114]$$

$$\frac{d}{dt}[\text{KK}] = -a_3[\text{KK}][\text{KKK}^*] + d_3[\text{KK}\cdot\text{KKK}^*] + k_4[\text{KK}\cdot\text{P}\cdot\text{KK}\text{P}'ase] \quad [115]$$

$$\frac{d}{dt}[\text{KK}\cdot\text{KKK}^*] = a_3[\text{KK}][\text{KKK}^*] - (d_3 + k_3)[\text{KK}\cdot\text{KKK}^*] \quad [116]$$

$$\frac{d}{dt}[\text{KK}\cdot\text{P}] = -a_4[\text{KK}\cdot\text{P}][\text{KK}\text{P}'ase] + d_4[\text{KK}\cdot\text{P}\cdot\text{KK}\text{P}'ase] + k_5[\text{KK}\cdot\text{KKK}^*] + k_6[\text{KK}\cdot\text{PP}\cdot\text{KK}\text{P}'ase] + d_5[\text{KK}\cdot\text{P}\cdot\text{KKK}^*] - a_5[\text{KK}\cdot\text{P}][\text{KKK}^*] \quad [117]$$

$$+ d_6[\text{KK}\cdot\text{P}\cdot\text{KKK}^*] - a_6[\text{KK}\cdot\text{P}][\text{KKK}^*] \quad [117]$$

$$\frac{d}{dt}[\text{KK}\cdot\text{P}\cdot\text{KK}\text{P}'ase] = a_4[\text{KK}\cdot\text{P}][\text{KK}\text{P}'ase] - (d_4 + k_3)[\text{KK}\cdot\text{P}\cdot\text{KK}\text{P}'ase] \quad [118]$$

$$\frac{d}{dt}[\text{KK}\cdot\text{P}\cdot\text{KKK}^*] = a_5[\text{KK}\cdot\text{P}][\text{KKK}^*] - (d_5 + k_5)[\text{KK}\cdot\text{P}\cdot\text{KKK}^*] \quad [119]$$

$$\frac{d}{dt}[\text{KK}\cdot\text{PP}] = k_7[\text{KK}\cdot\text{P}\cdot\text{KKK}^*] - a_6[\text{KK}\cdot\text{PP}][\text{KK}\text{P}'ase] + d_6[\text{KK}\cdot\text{PP}\cdot\text{KK}\text{P}'ase] - a_7[\text{KK}\cdot\text{PP}][\text{K}] + (d_7 + k_7)[\text{K}\cdot\text{KK}\cdot\text{PP}] + (d_9 + k_9)[\text{K}\cdot\text{P}\cdot\text{KK}\cdot\text{PP}] - a_8[\text{K}\cdot\text{P}][\text{KK}\cdot\text{PP}] \quad [120]$$

$$\frac{d}{dt}[\text{KK}\cdot\text{PP}\cdot\text{KK}\text{P}'ase] = a_6[\text{KK}\cdot\text{PP}][\text{KK}\text{P}'ase] - (d_6 + k_6)[\text{KK}\cdot\text{PP}\cdot\text{KK}\text{P}'ase] \quad [121]$$

$$\frac{d}{dt}[\text{K}] = -a_7[\text{K}][\text{KK}\cdot\text{PP}] + d_7[\text{K}\cdot\text{KK}\cdot\text{PP}] + k_8[\text{K}\cdot\text{P}\cdot\text{K}\text{P}'ase] \quad [122]$$

$$\frac{d}{dt}[\text{K}\cdot\text{KK}\cdot\text{PP}] = a_7[\text{K}][\text{KK}\cdot\text{PP}] - (d_7 + k_7)[\text{K}\cdot\text{KK}\cdot\text{PP}] \quad [123]$$

$$\frac{d}{dt}[\text{K}\cdot\text{P}] = k_7[\text{K}\cdot\text{KK}\cdot\text{PP}] - a_8[\text{K}\cdot\text{P}][\text{K}\text{P}'ase] + d_8[\text{K}\cdot\text{P}\cdot\text{K}\text{P}'ase] - a_9[\text{K}\cdot\text{P}][\text{KK}\cdot\text{PP}] + d_9[\text{K}\cdot\text{P}\cdot\text{KK}\cdot\text{PP}] + k_{10}[\text{K}\cdot\text{PP}\cdot\text{K}\text{P}'ase] \quad [124]$$

$$\frac{d}{dt}[\text{K}\cdot\text{P}\cdot\text{K}\text{P}'ase] = a_8[\text{K}\cdot\text{P}][\text{K}\text{P}'ase] - (d_8 + k_8)[\text{K}\cdot\text{P}\cdot\text{K}\text{P}'ase] \quad [125]$$

$$\frac{d}{dt}[\text{K}\cdot\text{P}\cdot\text{KK}\cdot\text{PP}] = a_9[\text{K}\cdot\text{P}][\text{KK}\cdot\text{PP}] - (d_9 + k_9)[\text{K}\cdot\text{P}\cdot\text{KK}\cdot\text{PP}] \quad [126]$$

$$\frac{d}{dt}[\text{K}\cdot\text{PP}] = -a_{10}[\text{K}\cdot\text{PP}][\text{K}\text{P}'ase] + d_{10}[\text{K}\cdot\text{PP}\cdot\text{K}\text{P}'ase] + k_9[\text{K}\cdot\text{P}\cdot\text{KK}\cdot\text{PP}] \quad [127]$$

$$\frac{d}{dt}[\text{K}\cdot\text{PP}\cdot\text{K}\text{P}'ase] = a_{10}[\text{K}\cdot\text{PP}][\text{K}\text{P}'ase] - (d_{10} + k_{10})[\text{K}\cdot\text{PP}\cdot\text{K}\text{P}'ase] \quad [128]$$

$$[\text{E1}_{\text{tot}}] = [\text{E1}] + [\text{KKK}\cdot\text{E1}] \quad [130]$$

$$[\text{E2}_{\text{tot}}] = [\text{E2}] + [\text{KKK}^*\cdot\text{E2}] \quad [131]$$

$$[\text{KK}_{\text{tot}}] = [\text{KK}] + [\text{KK}\cdot\text{P}] + [\text{KK}\cdot\text{PP}] + [\text{KK}\cdot\text{KKK}^*] + [\text{KK}\cdot\text{P}\cdot\text{KKK}^*] + [\text{KK}\cdot\text{P}\cdot\text{KK}\text{P}'ase] + [\text{KK}\cdot\text{PP}\cdot\text{K}] + [\text{KK}\cdot\text{PP}\cdot\text{K}\cdot\text{P}] \quad [132]$$

$$[\text{KK}\text{P}'ase_{\text{tot}}] = [\text{KK}\text{P}'ase] + [\text{KK}\text{P}'ase\cdot\text{KK}\cdot\text{P}] + [\text{KK}\text{P}'ase\cdot\text{KK}\cdot\text{PP}] \quad [133]$$

$$[\text{K}_{\text{tot}}] = [\text{K}] + [\text{K}\cdot\text{P}] + [\text{K}\cdot\text{PP}] + [\text{K}\cdot\text{KK}\cdot\text{PP}] + [\text{K}\cdot\text{PP}\cdot\text{K}\cdot\text{P}] + [\text{K}\cdot\text{P}\cdot\text{K}\text{P}'ase] + [\text{K}\cdot\text{PP}\cdot\text{K}\text{P}'ase] \quad [134]$$

$$[\text{K}\text{P}'ase_{\text{tot}}] = [\text{K}\text{P}'ase] + [\text{K}\cdot\text{P}\cdot\text{K}\text{P}'ase] + [\text{K}\cdot\text{PP}\cdot\text{K}\text{P}'ase] \quad [135]$$

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

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

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

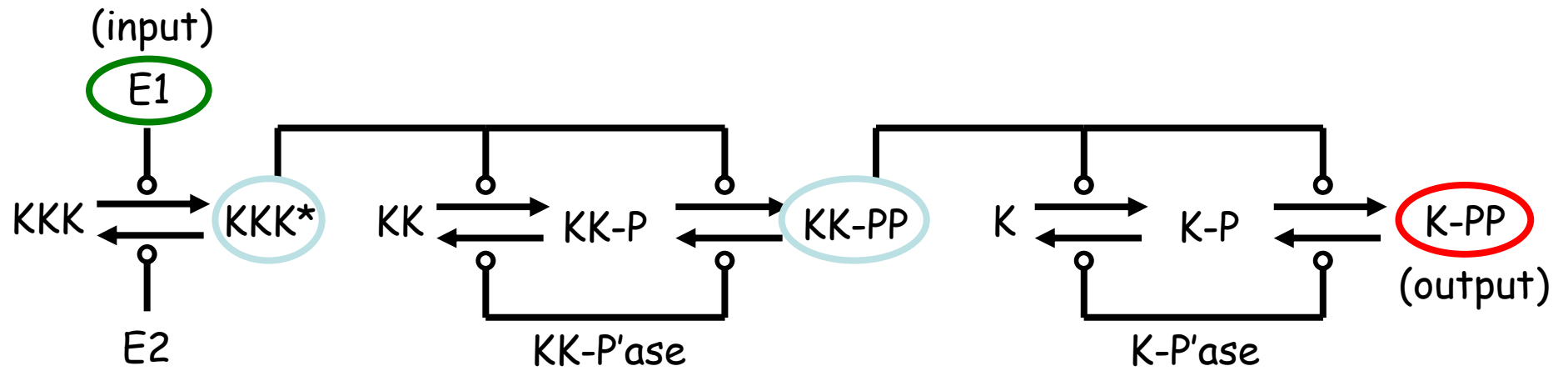
In addition, there are seven conservation equations (Eqs. 29-35).

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

in exactly one state

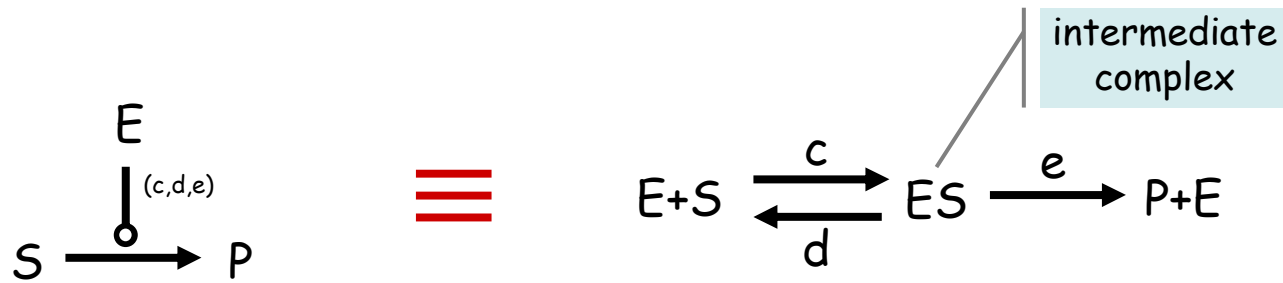
Each molecule

The Circuit

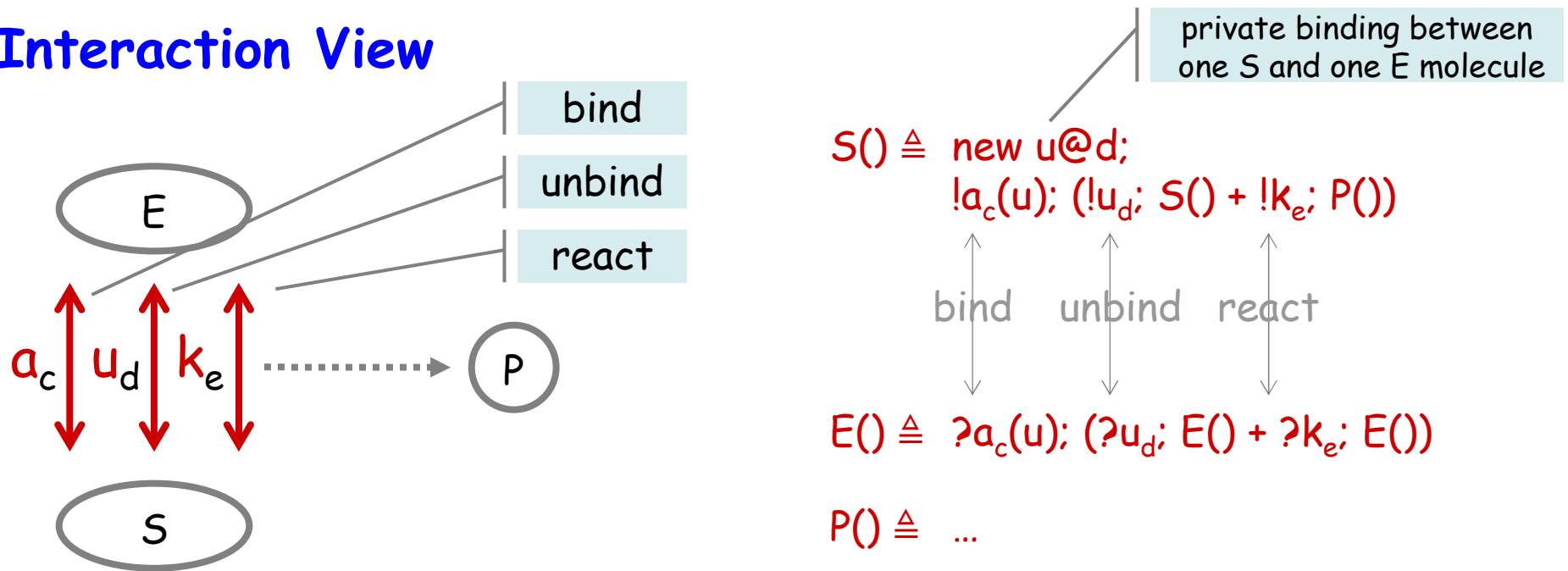


Enzymatic Reactions

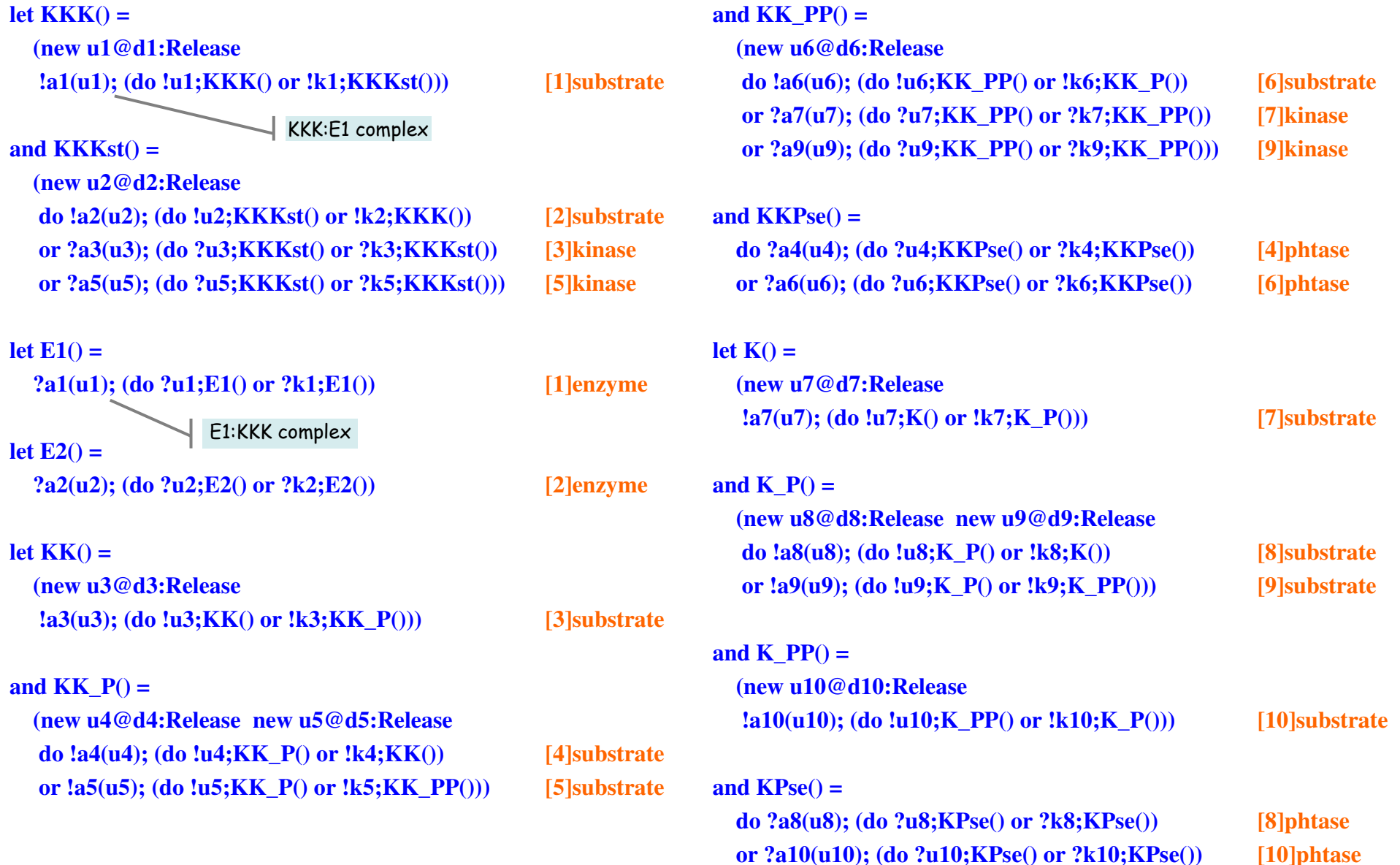
Reaction View



Interaction View



MAPK Cascade in SPiM



... globals

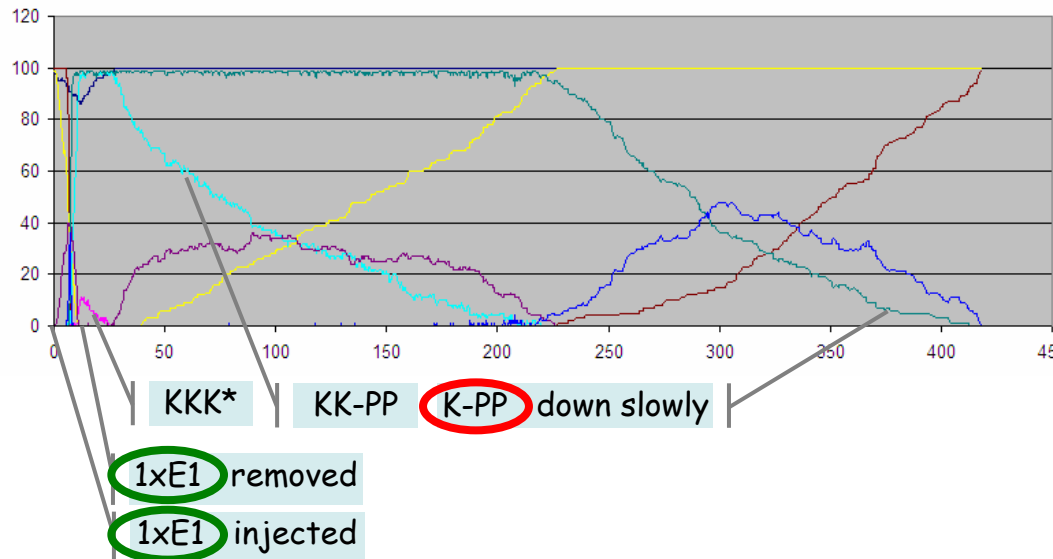
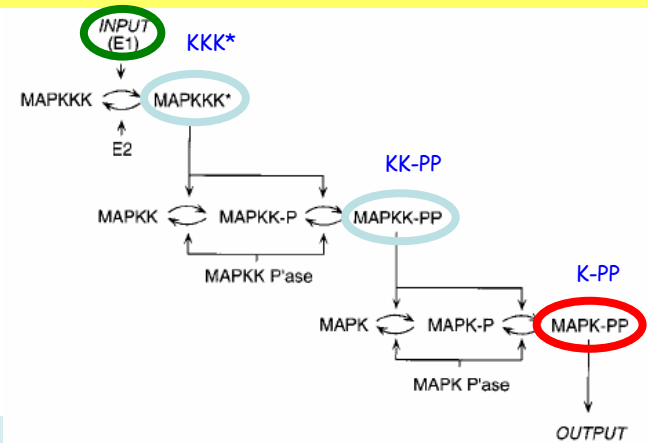
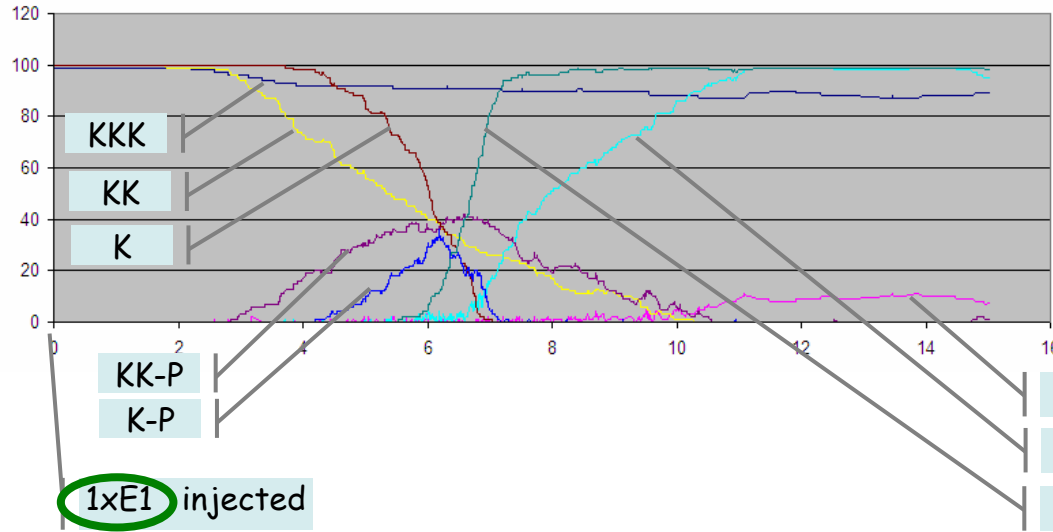
```
type Release = chan()
type Bond = chan(Release)
type React = chan()
```

```
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 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()
```

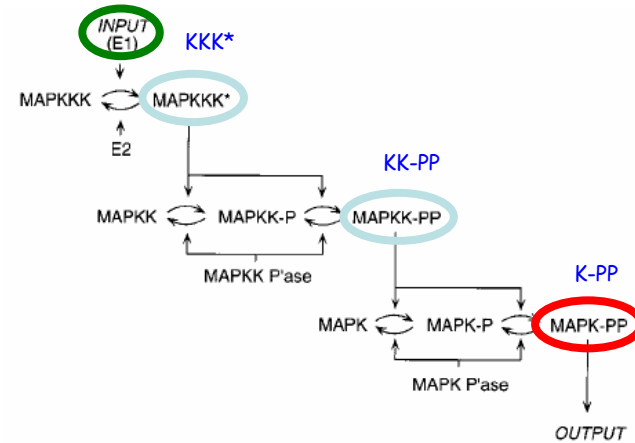
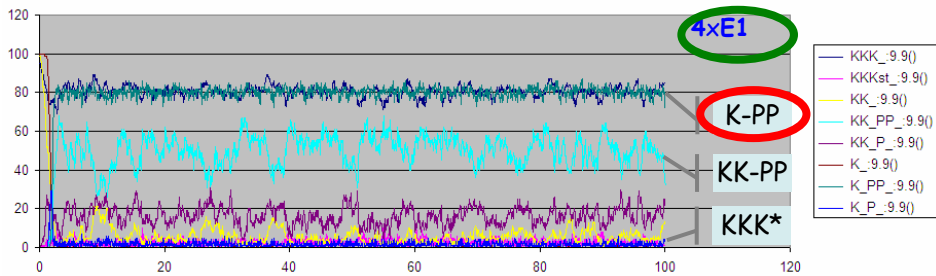
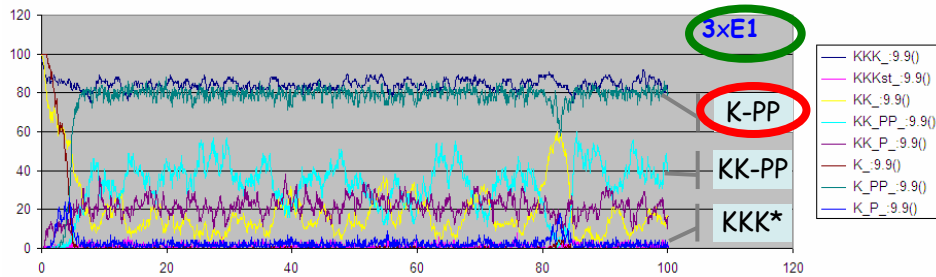
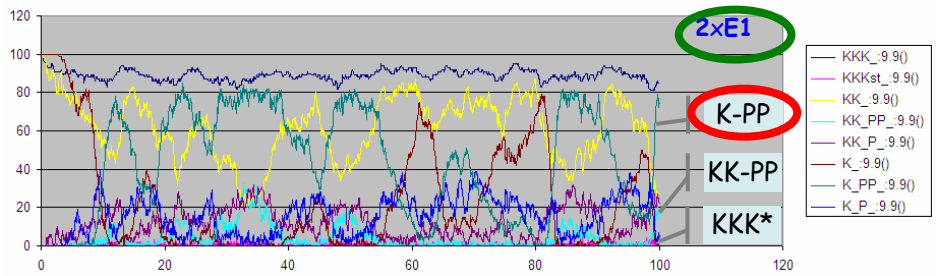
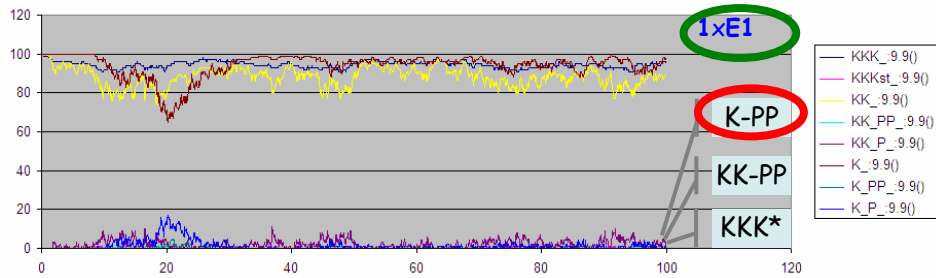
MAPK Cascade Simulation in SPiM



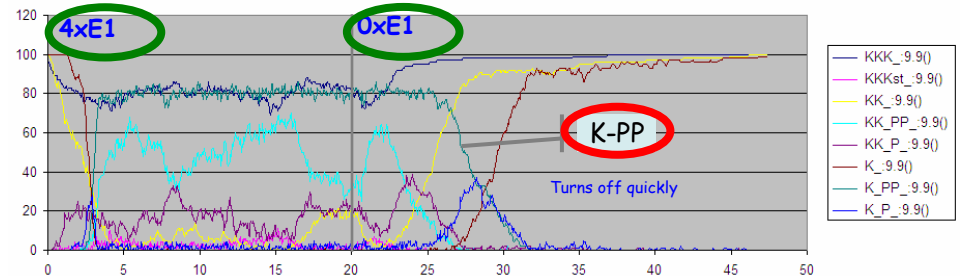
All coefficients 1.0 !!!
 100xKKK, 100xKK, 100xK,
 1xE2, 1xKKPse, 1xKPse.

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)



Model Validation

Model Validation: Simulation

- **Basic stochastic algorithm: Gillespie**
 - Exact (i.e. based on physics) stochastic simulation of chemical kinetics.
 - Can compute concentrations and reaction times for biochemical networks.
- **Stochastic Process Calculi**
 - **BioSpi** [Shapiro, Regev, Priami, et. al.]
 - Stochastic process calculus based on Gillespie.
 - **BioAmbients** [Regev, Panina, Silverma, Cardelli, Shapiro]
 - Extension of BioSpi for membranes.
 - **Case study: Lymphocytes in Inflamed Blood Vessels** [Lecaa, Priami, Quaglia]
 - Original analysis of lymphocyte rolling in blood vessels of different diameters.
 - **Case study: Lambda Switch** [Celine Kuttler, IRI Lille]
 - Model of phage lambda genome (well-studied system).
 - **Case study: VICE** [U. Pisa]
 - Minimal prokaryote genome (180 genes) and metabolism of *whole* VIRTUAL CELL, in stochastic π -calculus, simulated under stable conditions for 40K transitions.
- **Hybrid approaches**
 - **Charon language** [UPenn]
 - Hybrid systems: continuous differential equations + discrete/stochastic mode switching.
 - Etc.

Model Validation: "Program" Analysis

- **Causality Analysis**

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

- **Control Flow Analysis**

- Flow analysis techniques applied to process calculi.
- Overapproximation of behavior used to answer questions about what "cannot happen".
- Analysis of positive feedback transcription regulation in BioAmbients [Flemming Nielson].

- **Probabilistic Abstract Interpretation**

- [DiPierro Wicklicky].

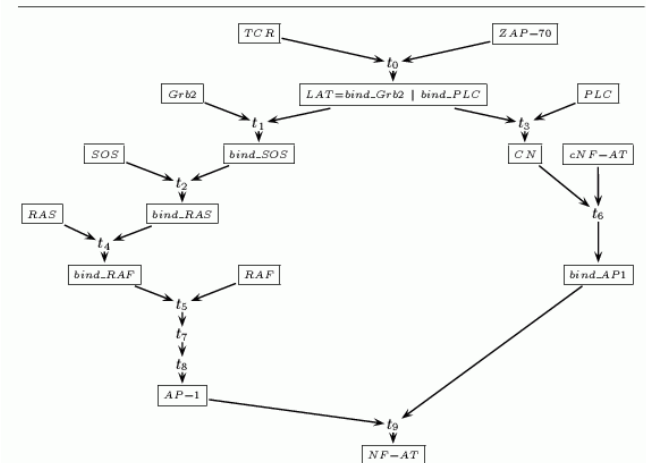


Fig.2. A computation of *Sys*. For readability, the processes, enclosed in boxes, have no address. Causality (both on transitions and processes) is represented by the (Hasse diagram resulting from the) arrows; their absence makes it explicit concurrent activities.

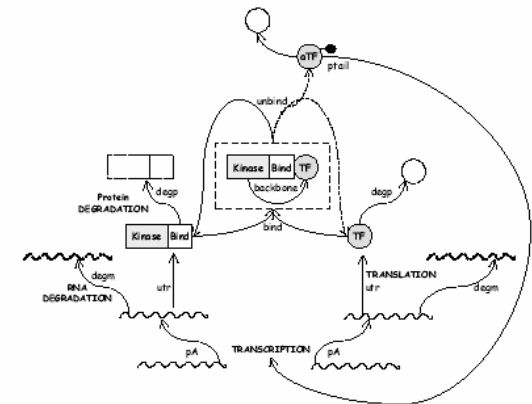


Fig. 1. Graphical presentation of Transcriptional Regulation by Positive Feedback [25].

Model Validation: Modelchecking

- **Temporal**

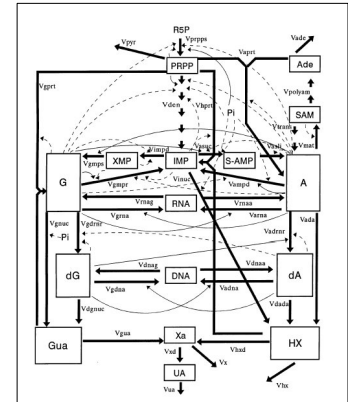
- Software verification of biomolecular systems (NA pump)
[Ciobanu]
- Analysis of mammalian cell cycle (after Kohn) in CTL.
[Chabrier-Rivier Chiaverini Danos Fages Schachter]
 - E.g. is state S_1 a necessary checkpoint for reaching state S_2 ?

- **Quantitative: Simpathica/xssys**

[Antioniotti Park Policriti Ugel Mishra]

- Quantitative temporal logic queries of human Purine metabolism model.

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



- **Stochastic: Spring**

[Parker Normal Kwiatkowska]

- Designed for stochastic (computer) network analysis
 - Discrete and Continuous Markov Processes.
 - Process input language.
 - Modelchecking of probabilistic queries.

What Reactive Systems Do For Us

We can write things down precisely

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

We can calculate and analyze

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

We can visualize

- Automata-like presentations.
- Petri-Net-like presentations.
- State Charts, Live Sequence Charts [Harel]
 - Hierarchical automata.
 - Scenario composition.

We can reason

- Suitable equivalences on processes induce algebraic laws.
- We can relate different systems (e.g. equivalent behaviors).
- We can relate different abstraction levels.
- We can use equivalences for state minimization (symmetries).

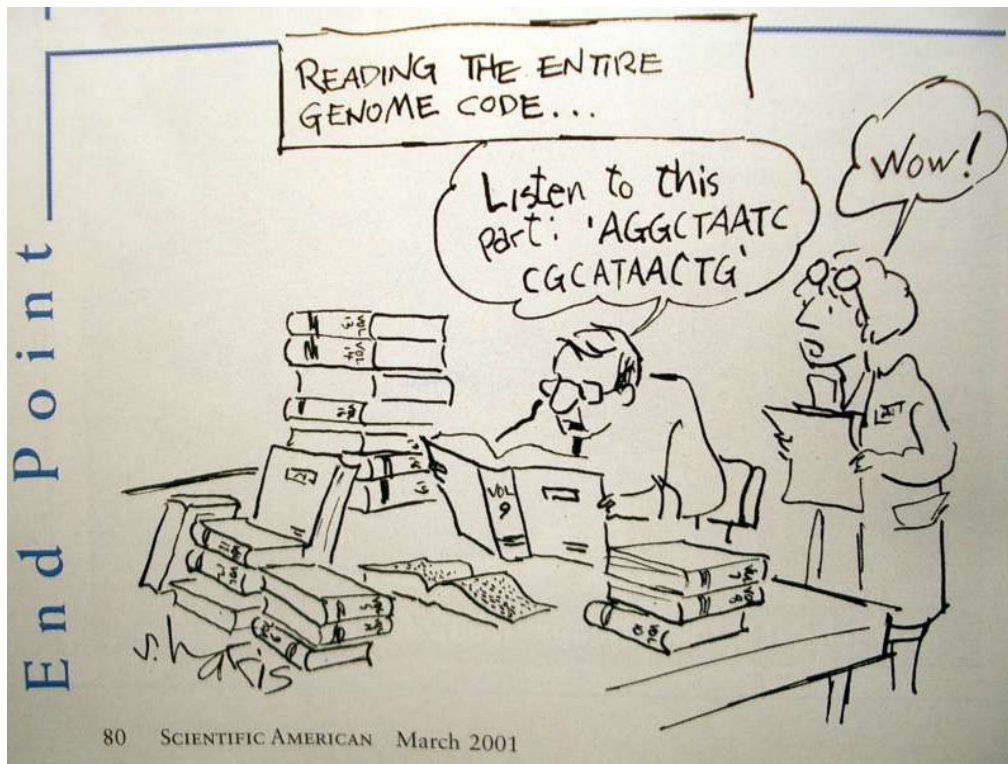
Disclaimers

- Some of these technologies are basically ready (medium-scale stochastic simulation and analysis, medium-scale nondeterministic and stochastic modelchecking).
- Others need to scale up significantly to be really useful. This is (has been) the challenge for computer scientists.

Many approaches, same basic philosophy, tools being built:

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

Conclusions



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