Biological Systems as Reactive Systems

Luca Cardelli

Microsoft Research
Cambridge UK

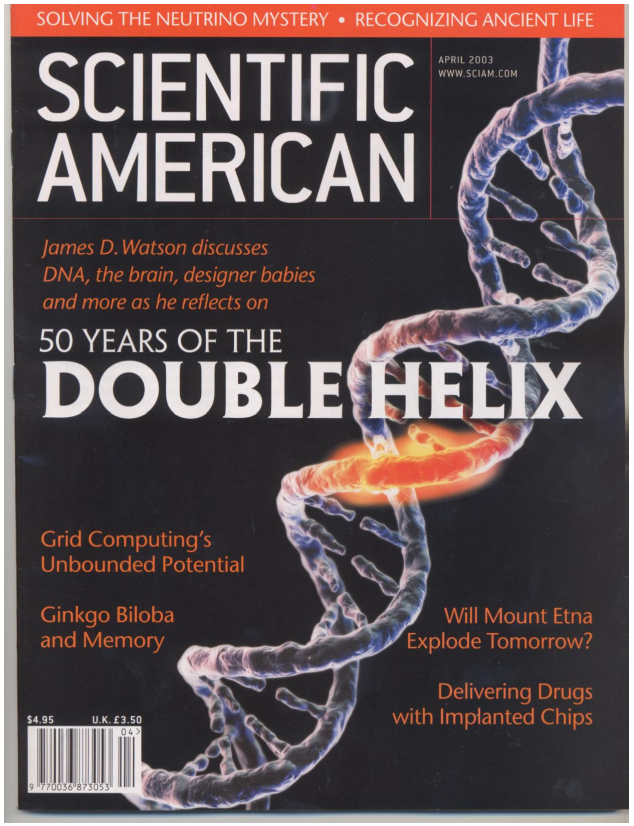
Munich 2005-04-06

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
  - …

- 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.

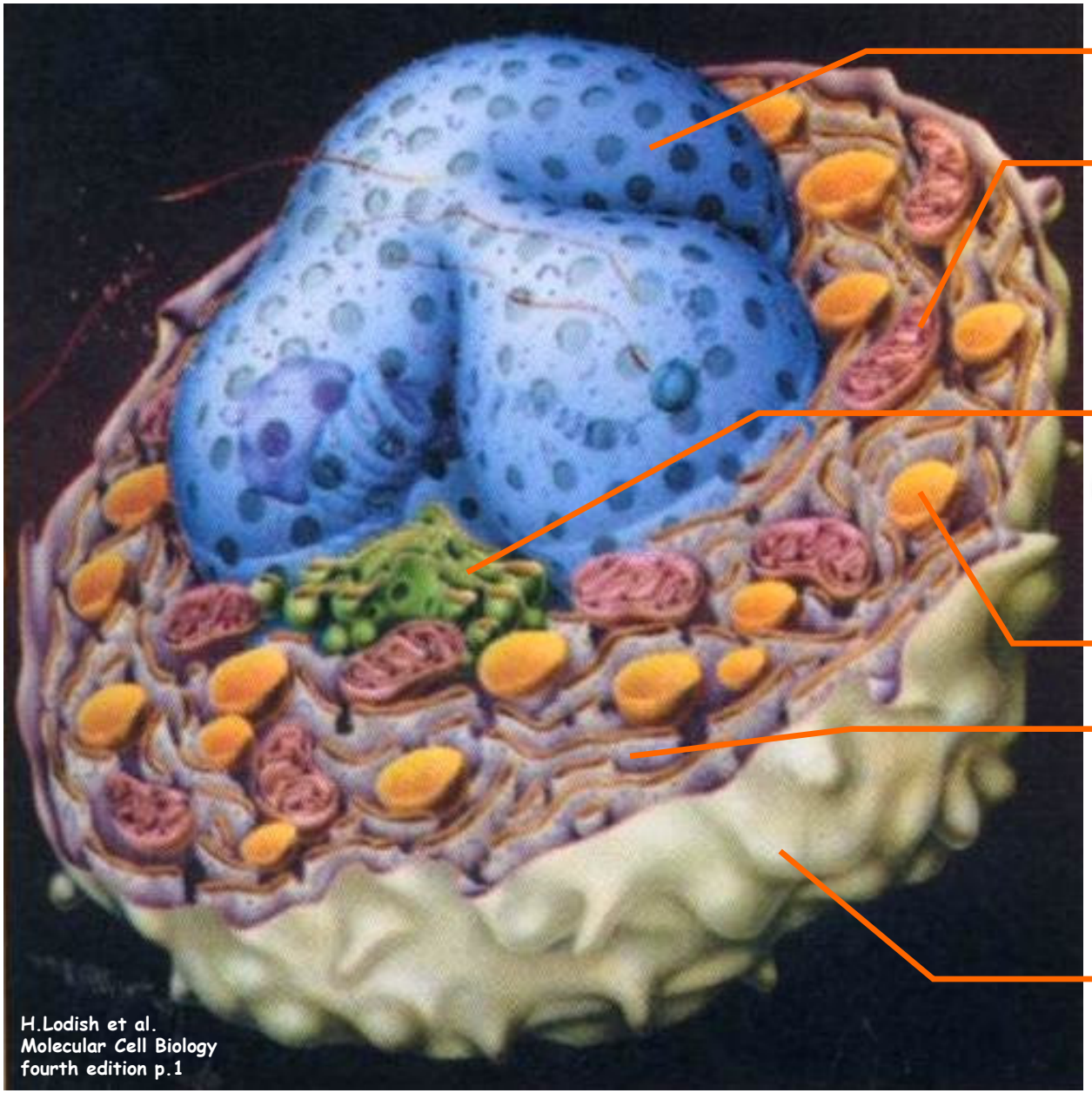
- In computing…
  - There are well-established scalable representations of dynamic reactive processes.
  - They look more or less like little, mathematically based, programming languages.

Structural Architecture

Eukaryotic Cell

(10~100 trillion in human body)

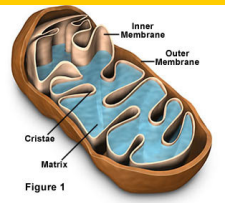
Membranes everywhere



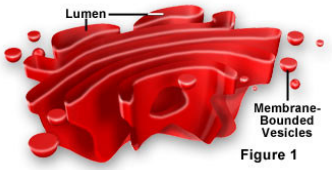
H.Lodish et al.
Molecular Cell Biology
fourth edition p.1

Nuclear membrane

Mitochondria

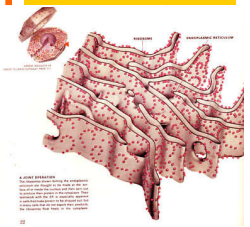


Golgi



Vesicles

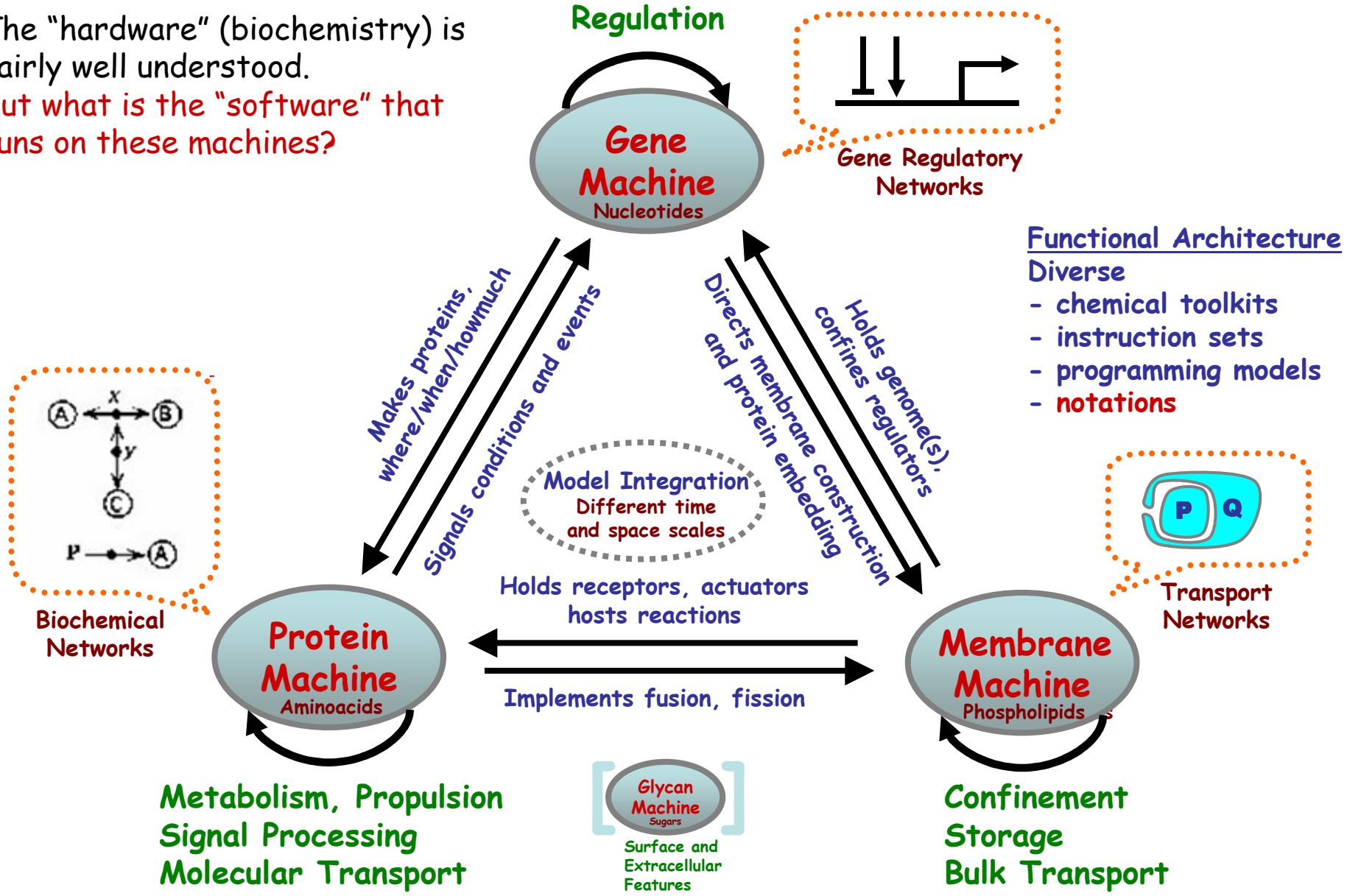
E.R.



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

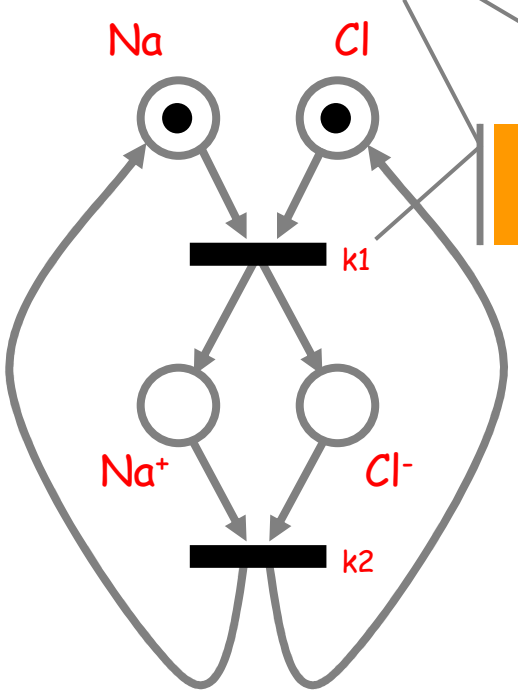
- Modeling biological systems
  - Not as continuous systems (often highly nonlinear)
  - But as discrete reactive systems; abstract machines with:
    - States represent situations
    - Event-driven transitions between states represent dynamics
  - The adequacy of describing (discrete) complex systems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

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

Chemistry vs. π-calculus

A process calculus (chemistry, or SBML)

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

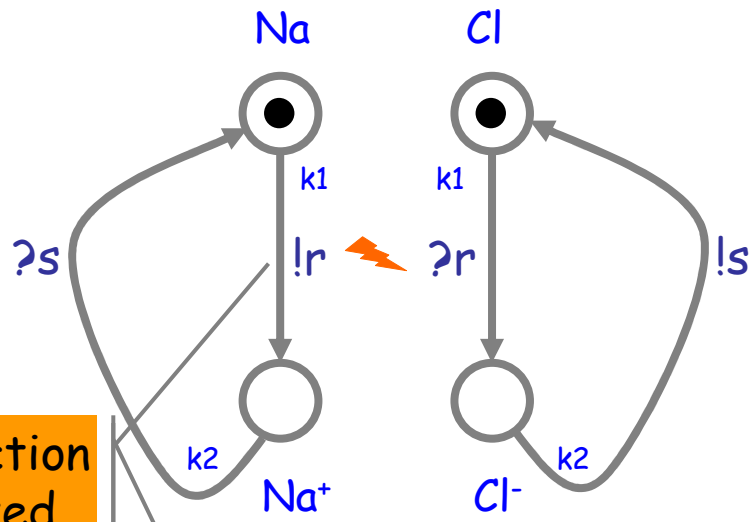


1 line per reaction

Reaction oriented

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

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

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 Computation":

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

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

- 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

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

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

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

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

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).

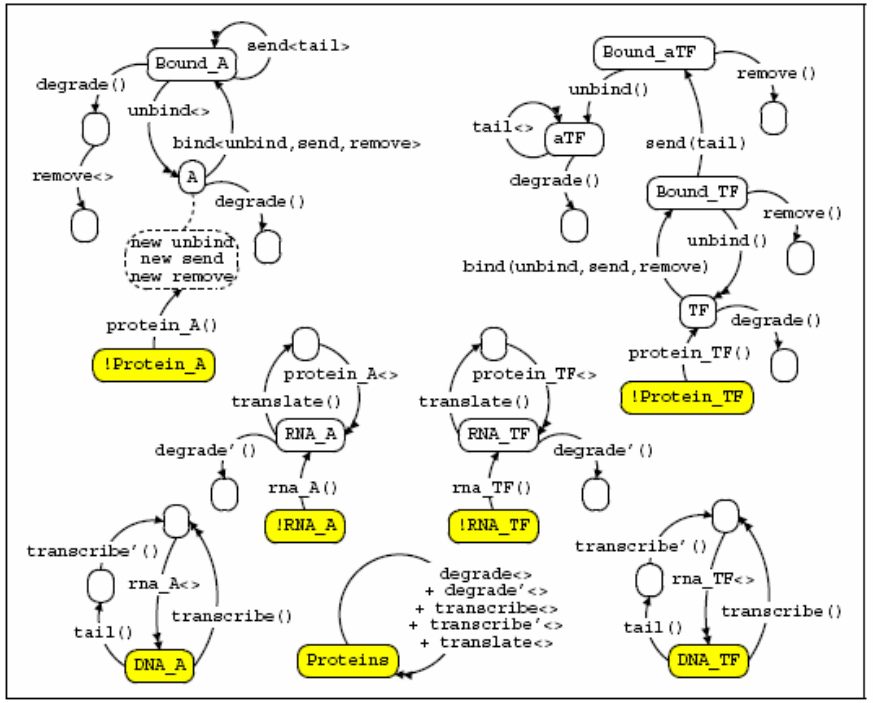


Figure 2. Regulating Gene Expression by Positive Feedback [9]

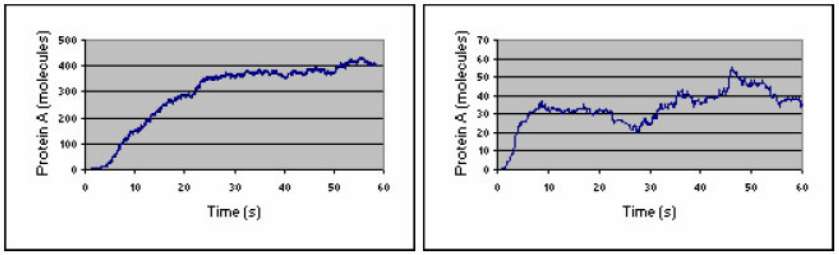


Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of TF

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

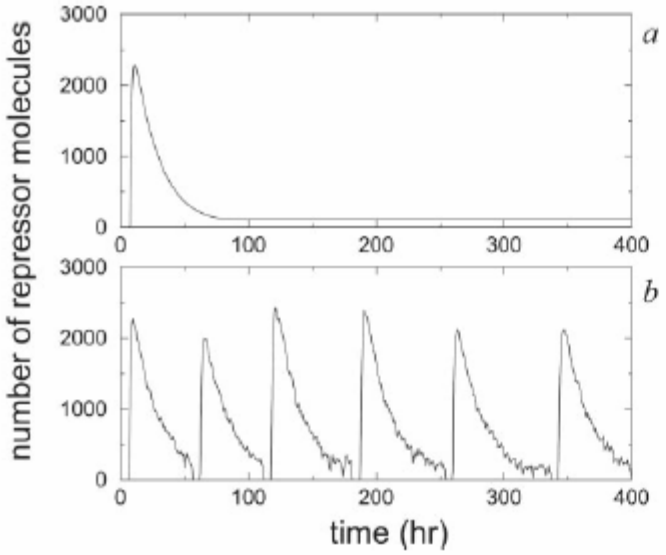


Fig. 5. Time evolution of R for the deterministic Eq. [1] (*a*) and stochastic (*b*) versions of the model. The values of the parameters are as in the caption of Fig. 1, except that now we set $\delta_R = 0.05\ h^{-1}$. For these parameter values, $\tau < 0$, so that the fixed point is stable.

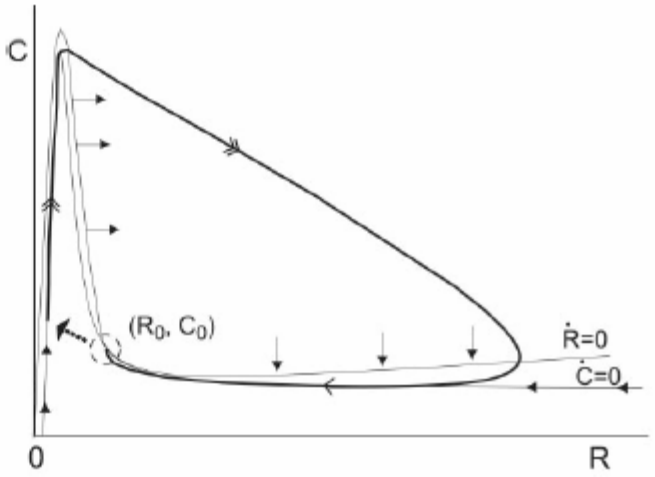
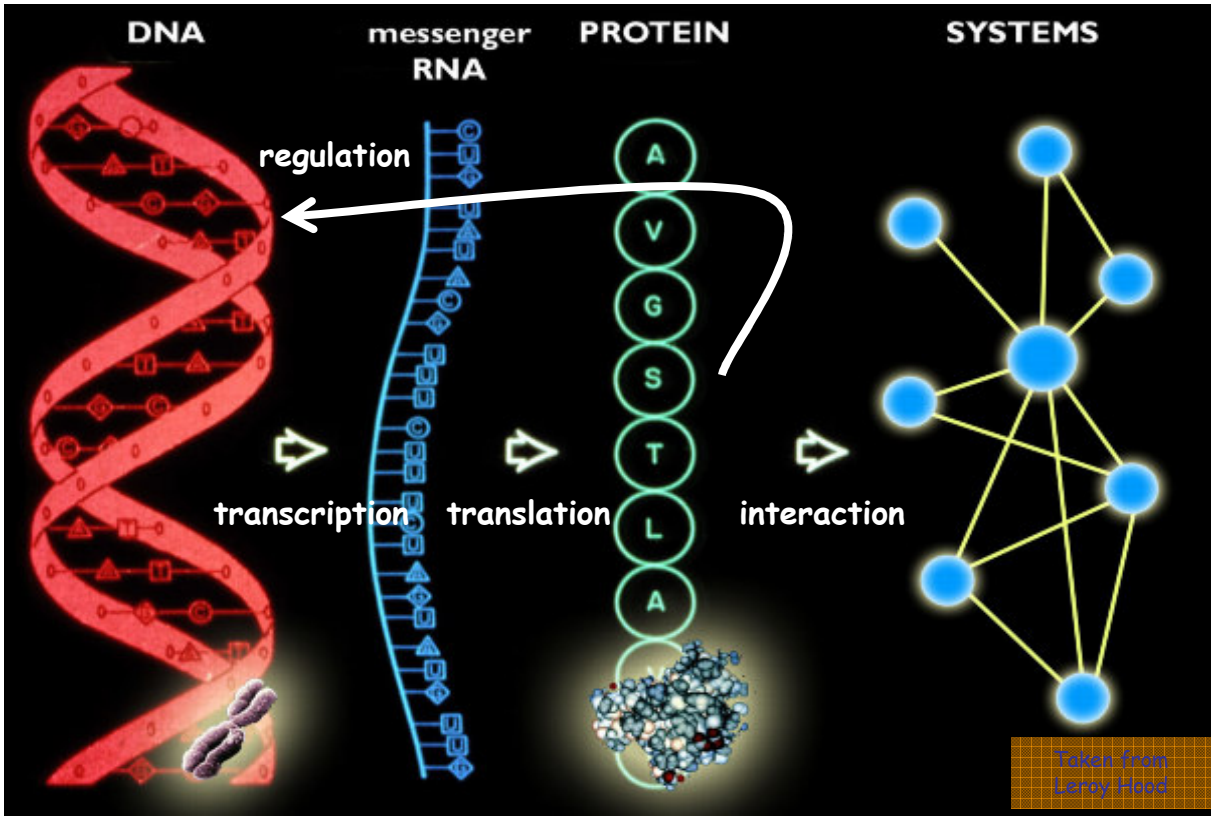


Fig. 6. Phase portrait as in Fig. 4 but for a situation in which the system falls into the stable fixed point (R_0,C_0). The dotted arrow to the left of the fixed point illustrates a perturbation that would initiate a single sweep of the (former) oscillatory trajectory.

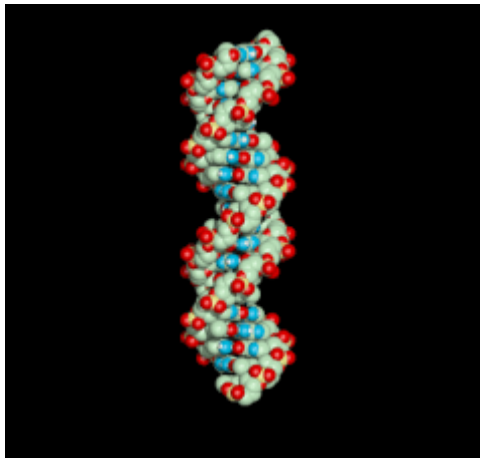
Gene Networks

The Gene Machine

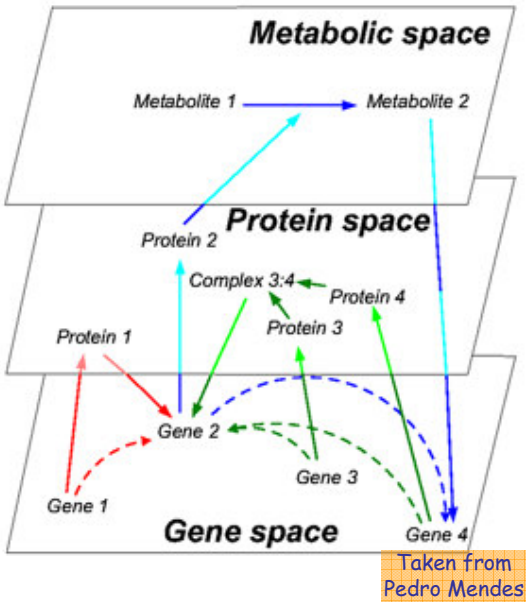
The "Central Dogma" of Molecular Biology



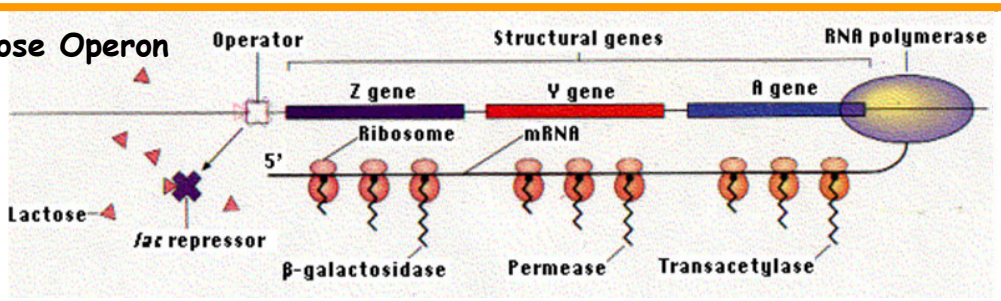
4-letter digital code | 4-letter digital code | 20-letter digital code | 50.000(?) shapes



DNA Tutorial

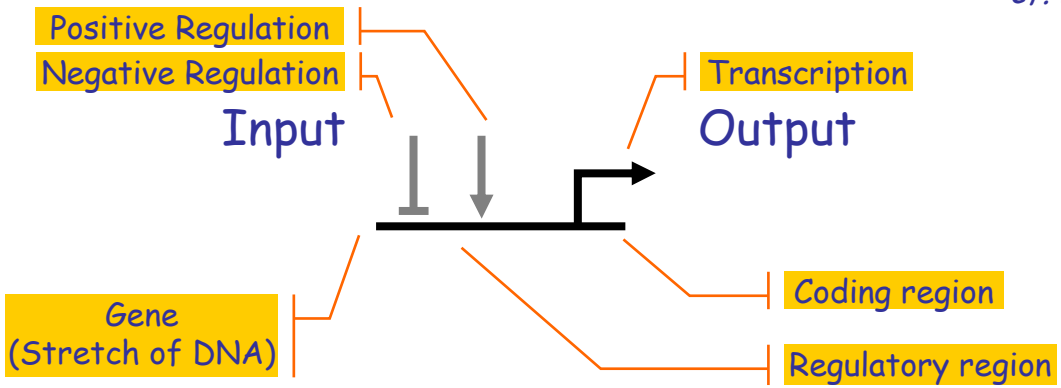


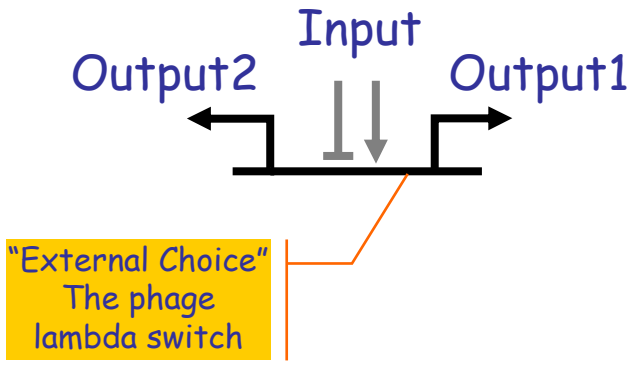
Lactose Operon



The Gene Machine "Instruction Set"

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





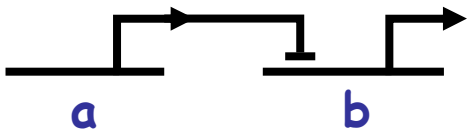
Regulation of a gene (positive and negative) influences transcription. The regulatory region has precise DNA sequences, but not meant for coding proteins: meant for binding regulators.

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

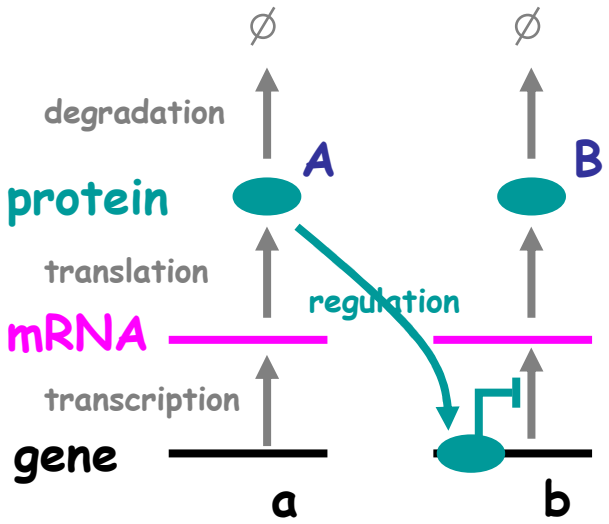
Human (and mammalian) Genome Size
3Gbp (Giga base pairs) 750MB @ 4bp/Byte (CD)
Non-repetitive: 1Gbp 250MB
In genes: 320Mbp 80MB
Coding: 160Mbp 40MB
Protein-coding genes: 30,000-40,000

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



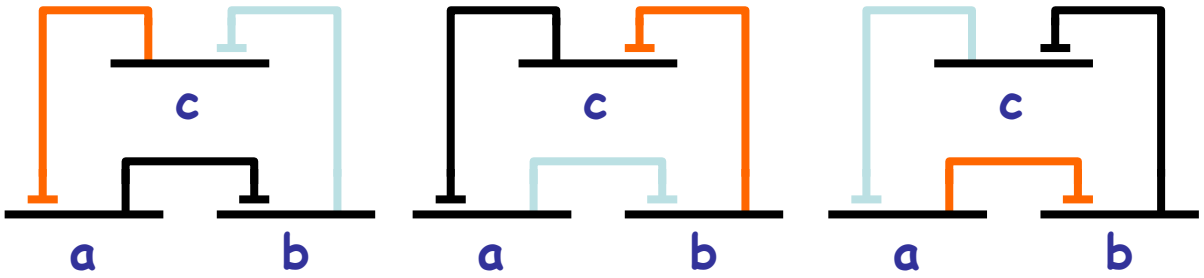
Is a shorthand for:



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

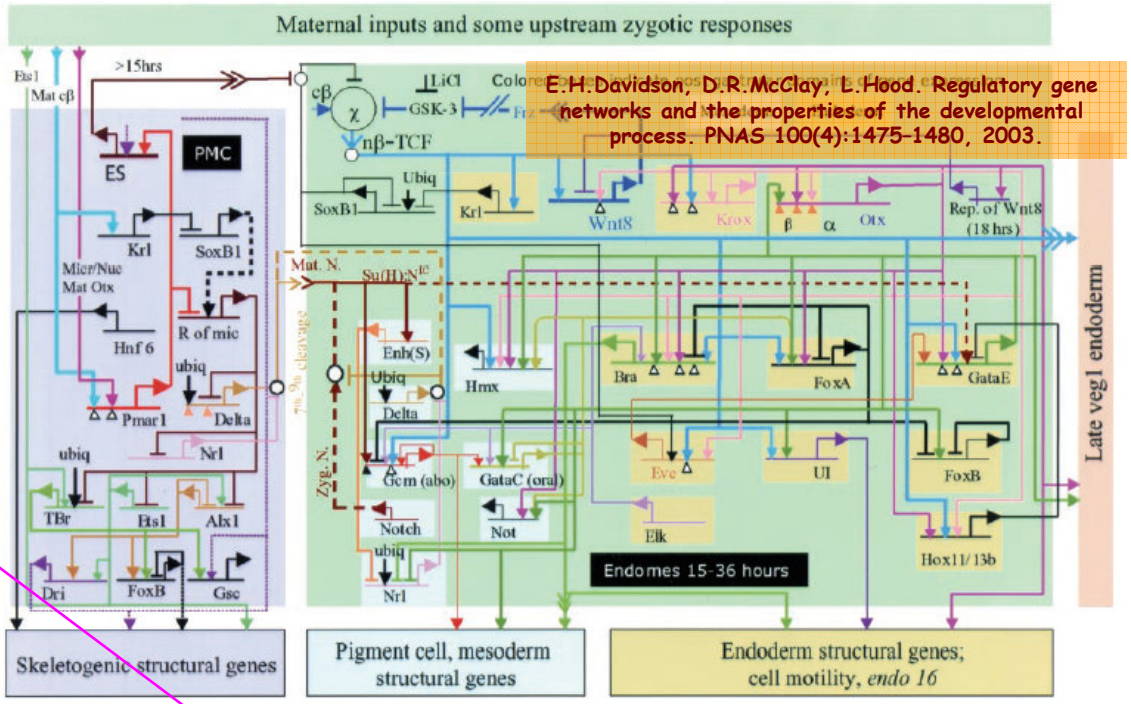


Expressed
Repressed
Expressing

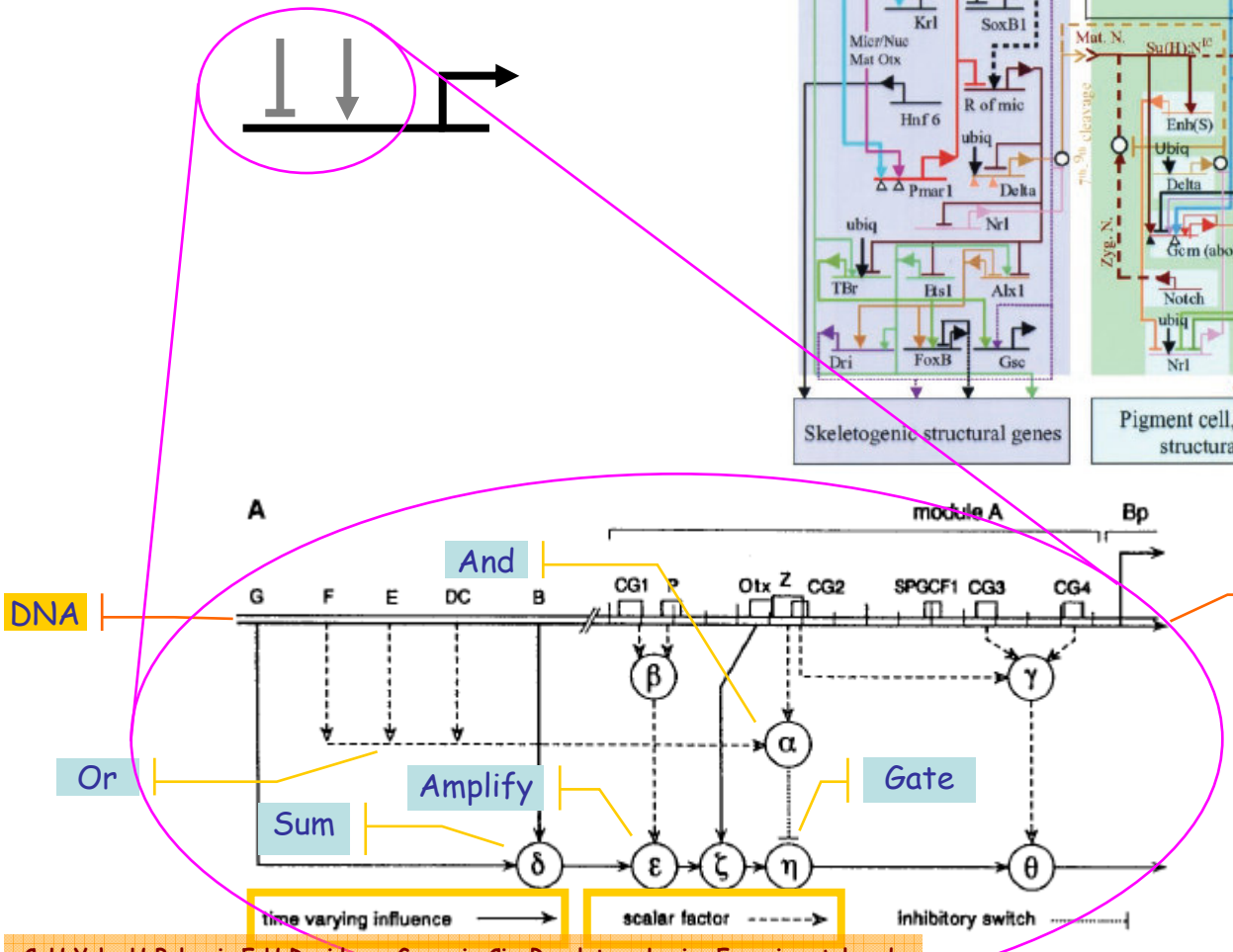
Gene Regulatory Networks

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

NetBuilder



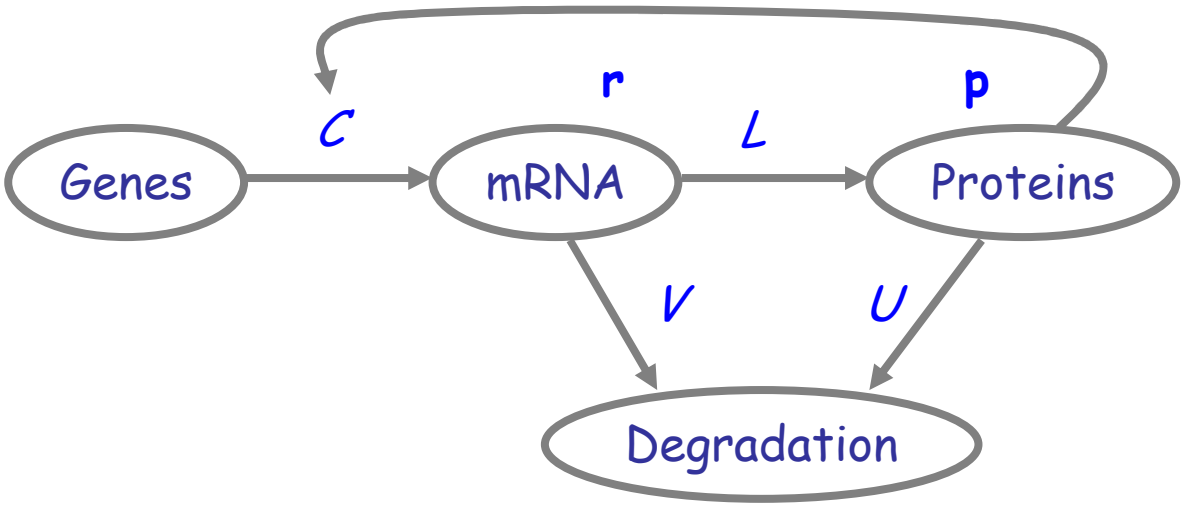
E.H.Davidson, D.R.McClay, L.Hood. Regulatory gene networks and the properties of the developmental process. PNAS 100(4):1475–1480, 2003.



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]



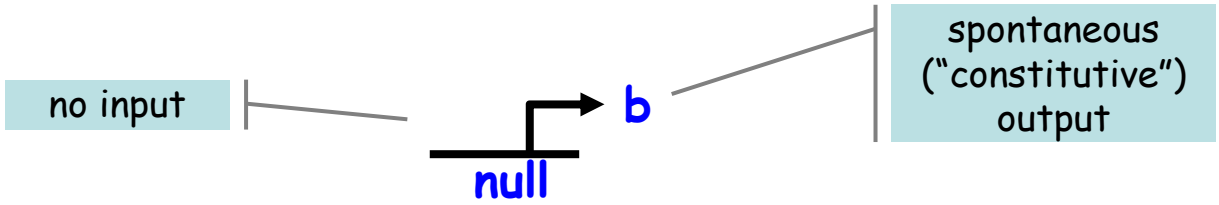
$$\frac{d\mathbf{r}}{dt} = f(\mathbf{p}) - V\mathbf{r}$$

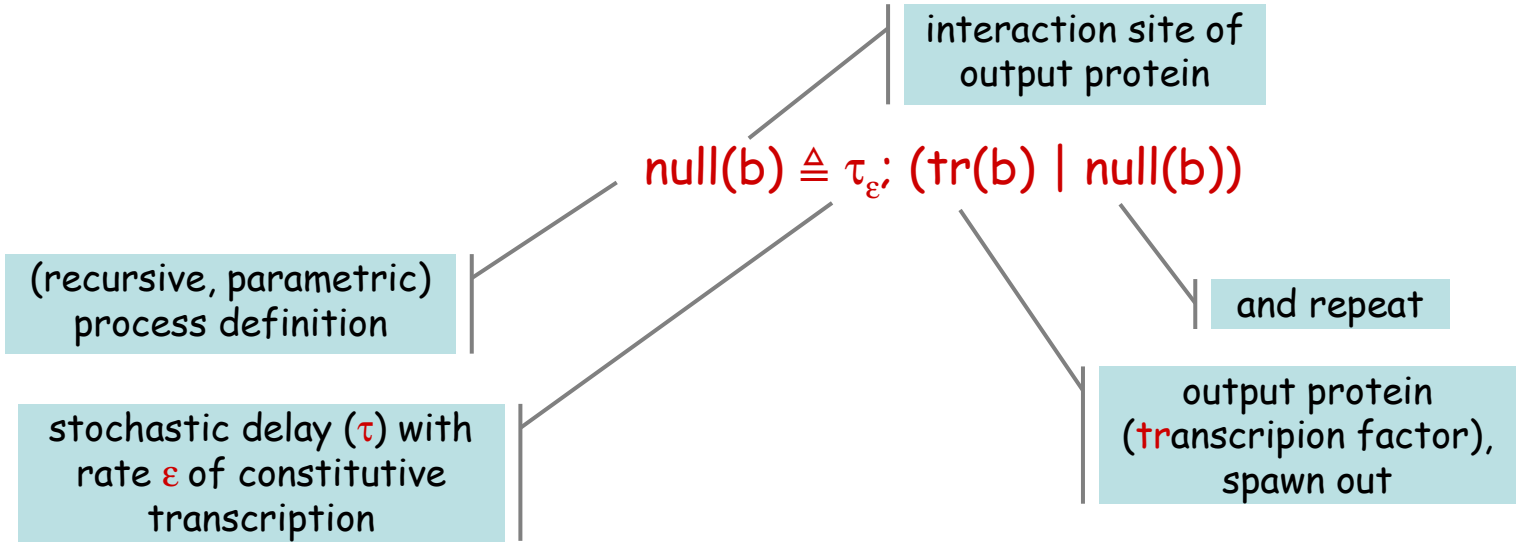
$$\frac{d\mathbf{p}}{dt} = L\mathbf{r} - U\mathbf{r}$$

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

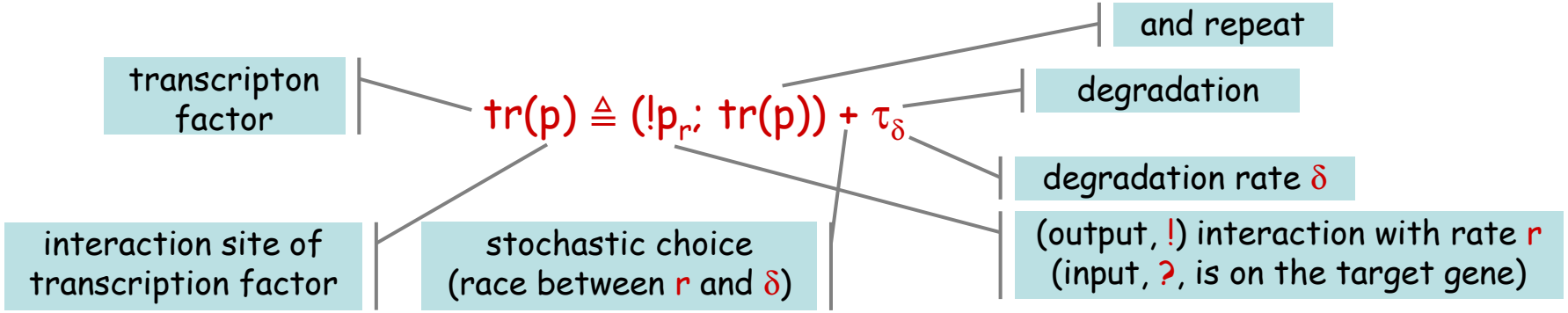




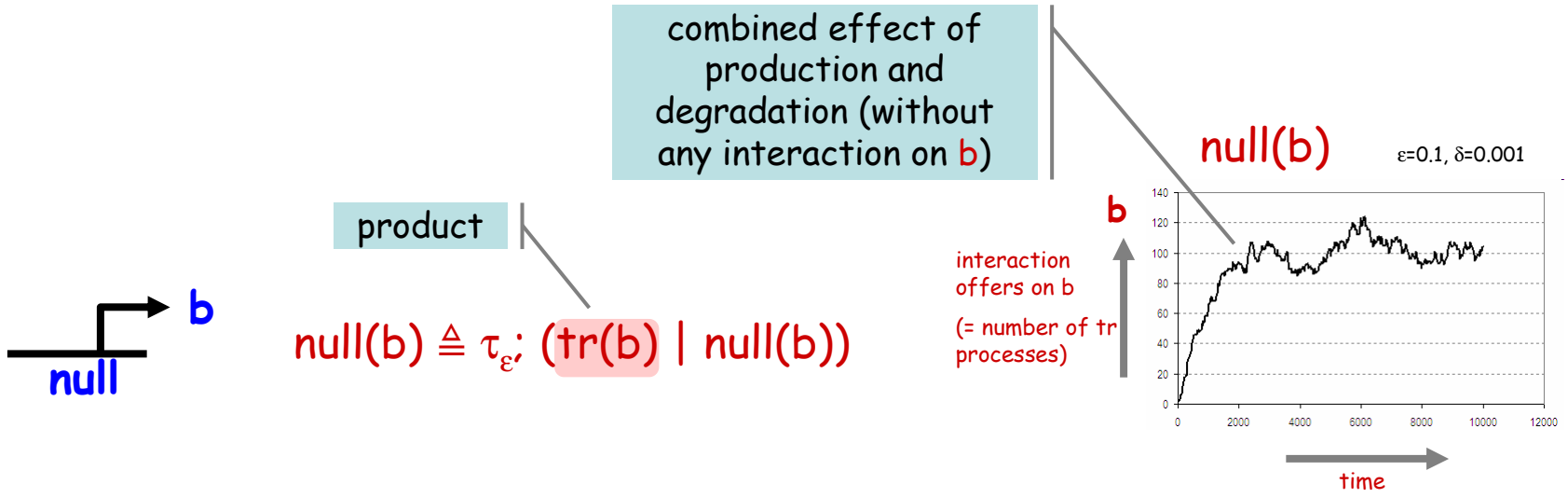
A stochastic rate r is always associated with each channel a_r (at channel creation time) and delay τ_r, but is often omitted when unambiguous.

Production and Degradation

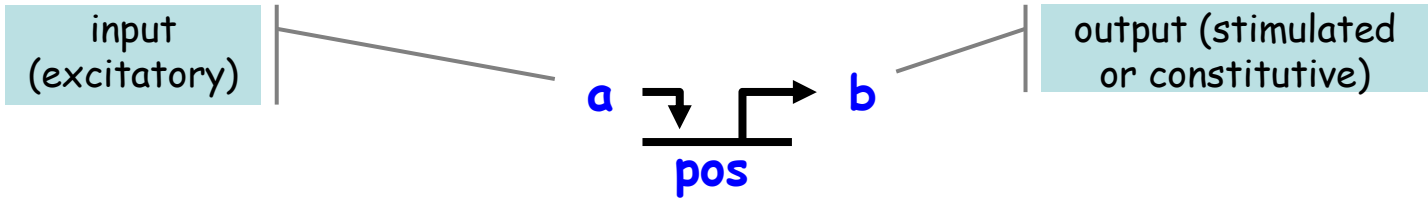
Degradation is extremely important and often deliberate; it changes unbounded growth into (roughly) stable signals.

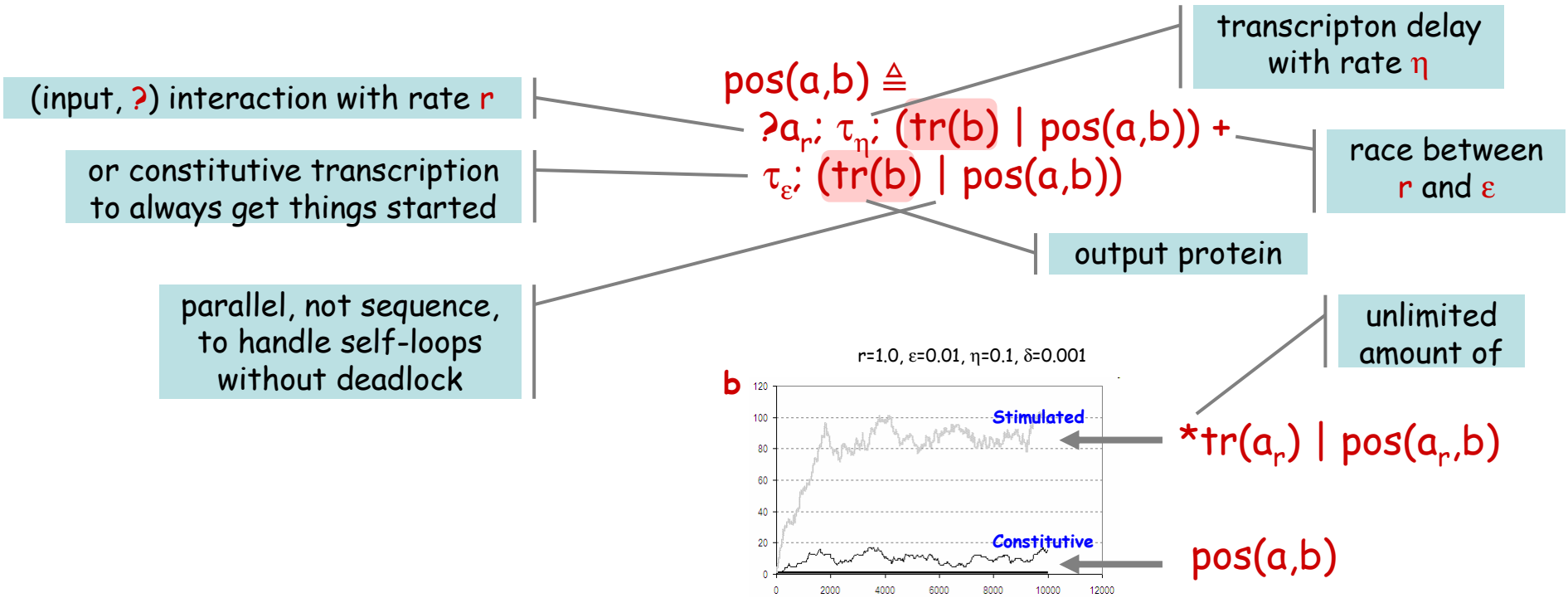


A transcription factor is a process (not a message or a channel): it has behavior such as interaction on p and degradation.

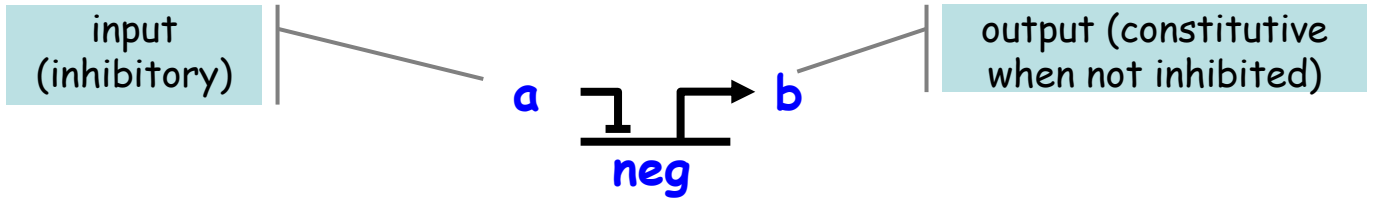


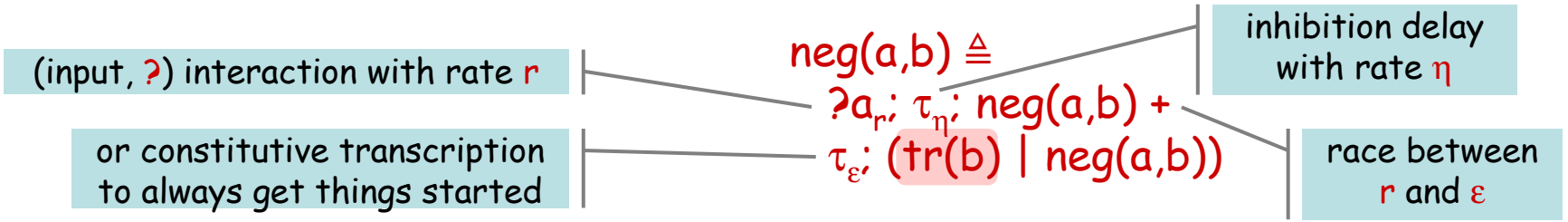
Unary Pos Gate

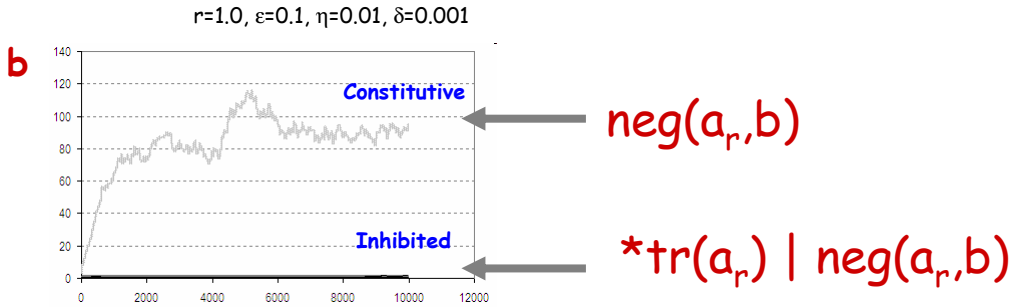




Unary Neg Gate

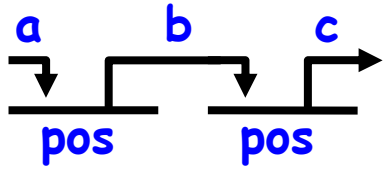






Signal Amplification

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



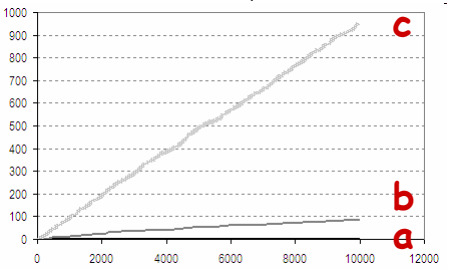
$pos(a,b) \triangleq$
$?a_r; \tau_\eta; (tr(b) \mid pos(a,b)) +$
$\tau_\varepsilon; (tr(b) \mid pos(a,b))$

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

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

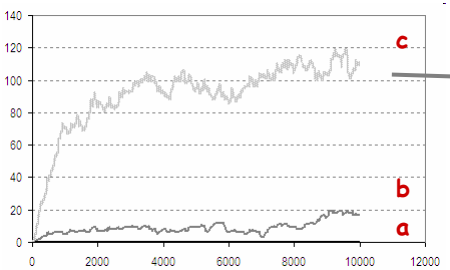
With little degradation

r=1.0, ε=0.01, η=0.1, δ=0.00001



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

r=1.0, ε=0.01, η=0.1, δ=0.001

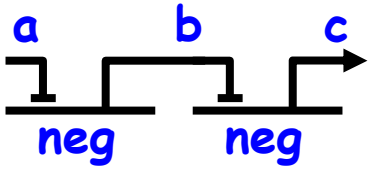


pos(a,b) | pos(b,c)

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

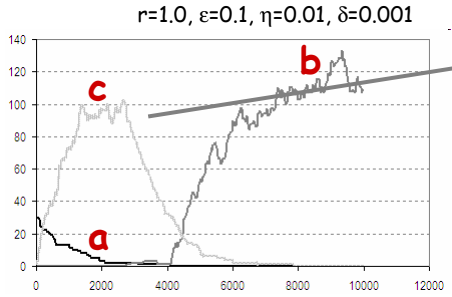
Signal Normalization

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



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

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

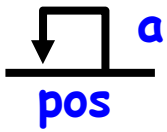


a non-zero input level, a, whether weak or strong, is renormalized to a standard level, c.

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

Self Feedback Circuits

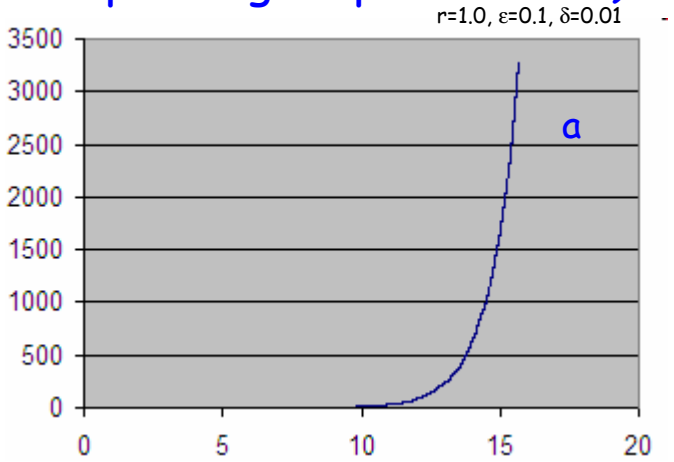
pos(a,a)



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

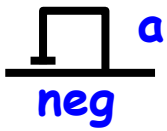
$tr(p) \triangleq (!p_r;\ tr(p)) + \tau_\delta$

(Can overwhelm degradation, depending on parameters)



pos(a,a)

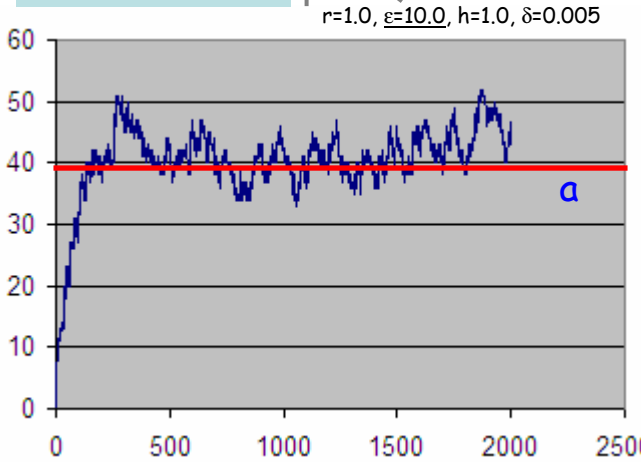
neg(a,a)



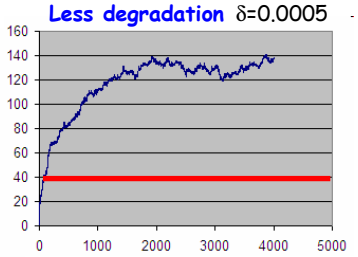
$neg(a,b) \triangleq$
$?a_r;\ \tau_h;\ neg(a,b) +$
$\tau_\varepsilon;\ (tr(b) \mid neg(a,b))$

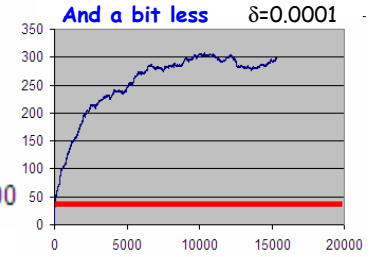
$tr(p) \triangleq (!p_r;\ tr(p)) + \tau_\delta$

high, to raise the signal



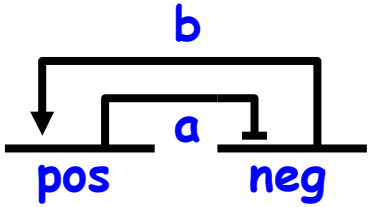
neg(a,a)



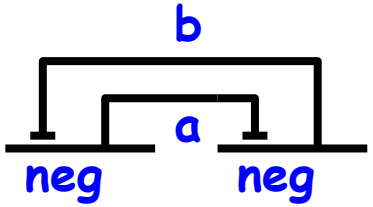


Two-gate Feedback Circuits

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

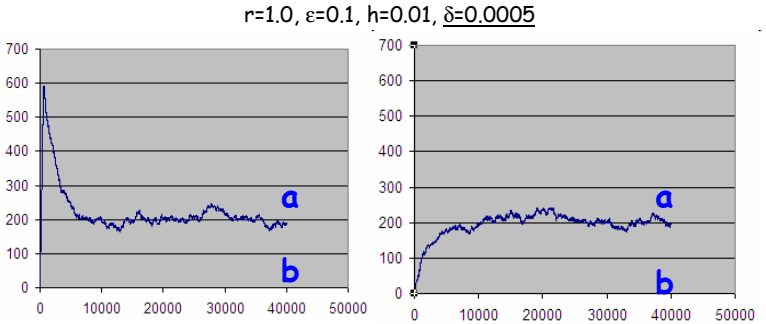


neg(b,a) |
neg(a,b)



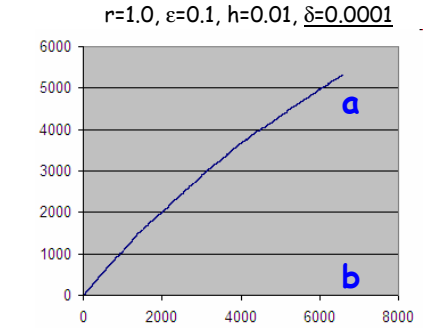
Monostable:

For some degradation rates is quite stable:



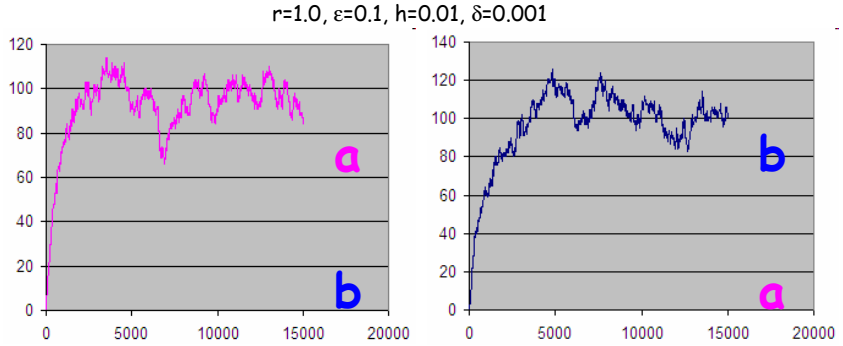
pos(b,a) | neg(a,b)

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

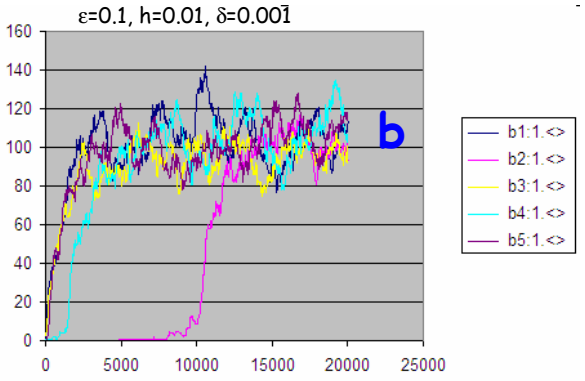


pos(b,a) | neg(a,b)

Bistable:



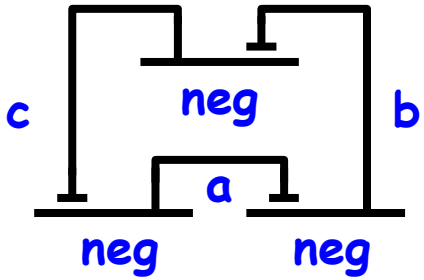
neg(b,a) | neg(a,b)



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)

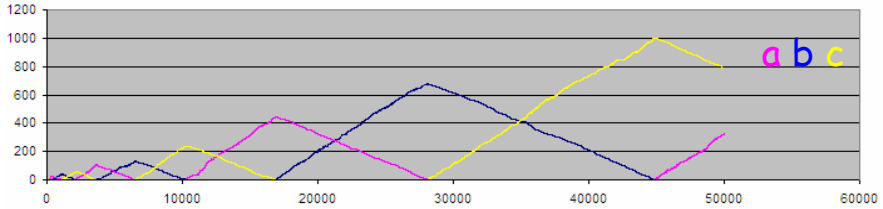


neg(a,b) ≜
  ?a_r; τ_h; neg(a,b) +
  τ_ε; (tr(b) | neg(a,b))

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

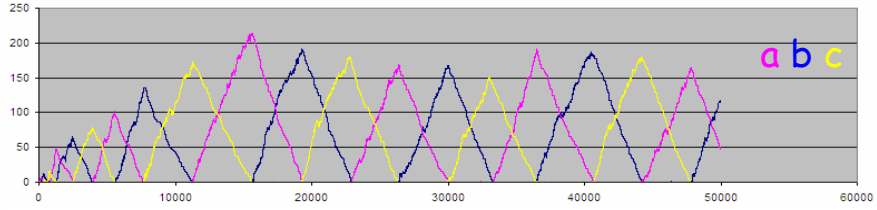
tr(p) ≜ !p_r — interact once and die otherwise stick around

r=1.0, ε=0.1, h=0.04



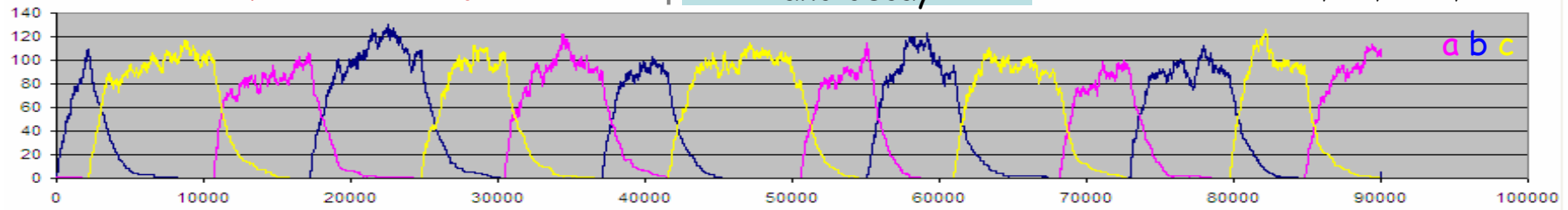
tr(p) ≜ !p_r + τ_δ — interact once and die otherwise decay

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



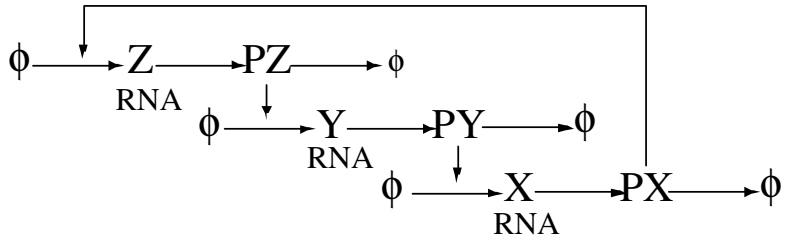
Subtle… at any point one gate is inhibited and the other two can fire constitutively. If one of them fires first, nothing really changes, but if the other one fires first, then the cycle progresses.

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

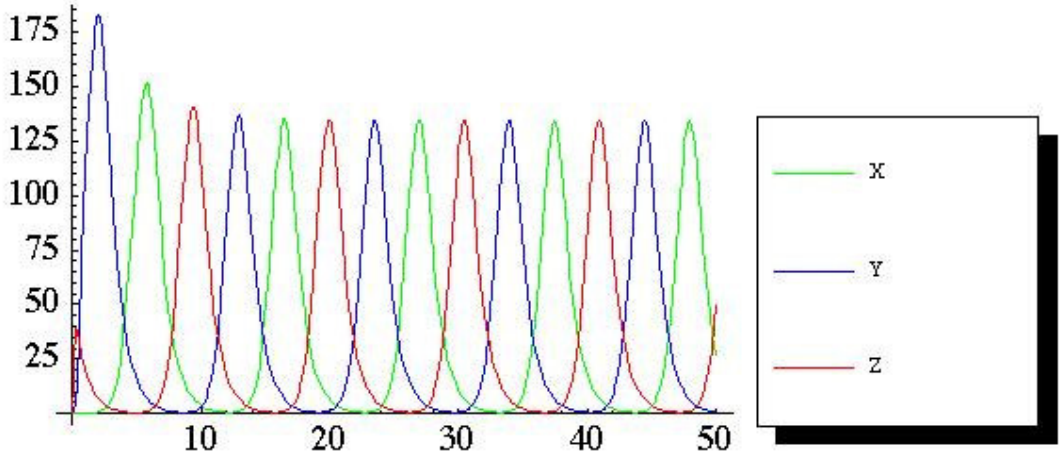
Bruce E Shapiro
Cellerator



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

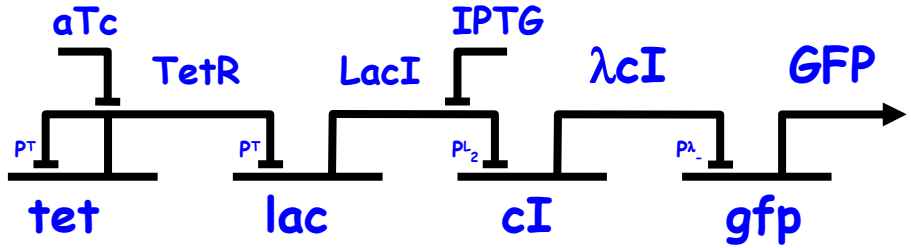
$$\frac{d[\mathrm{Y}]}{dt} = \alpha_0 + \frac{\alpha + \alpha_1 [\mathrm{PZ}]^n}{K^n + [\mathrm{PZ}]^n} - k[\mathrm{Y}], \quad \frac{d[\mathrm{PY}]}{dt} = \beta\{[\mathrm{Y}] - [\mathrm{PY}]\}$$

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



Guet et al.: D038/lac⁻

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

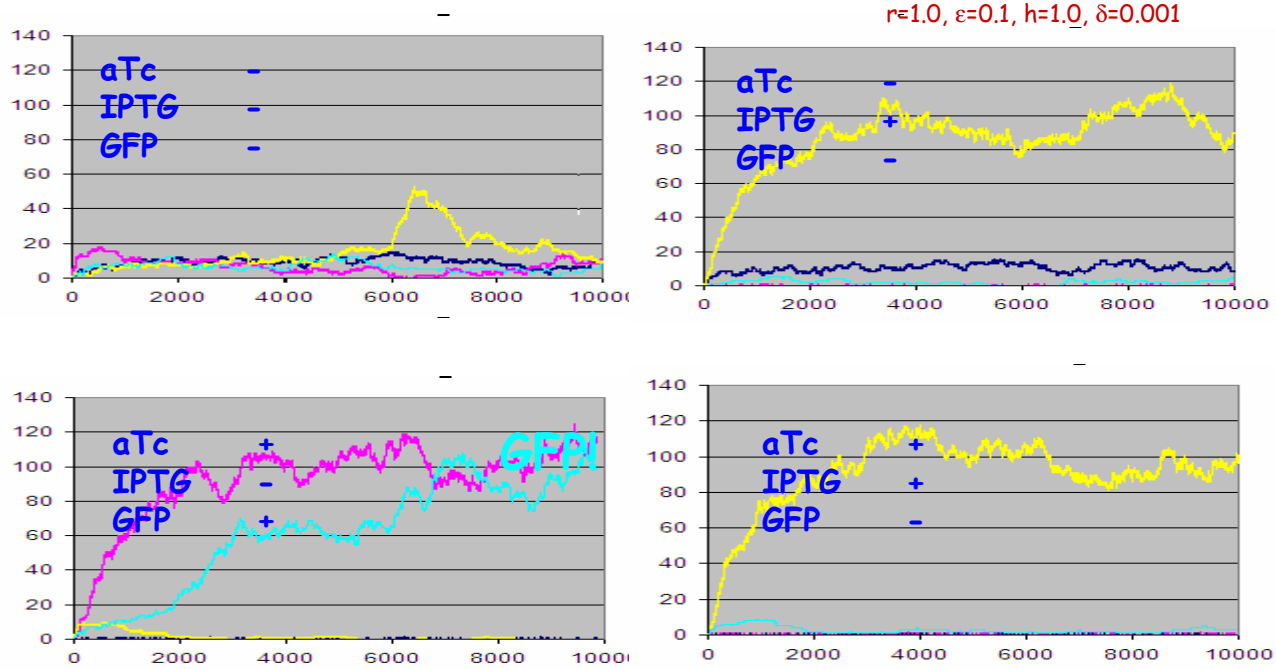


experiment:

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



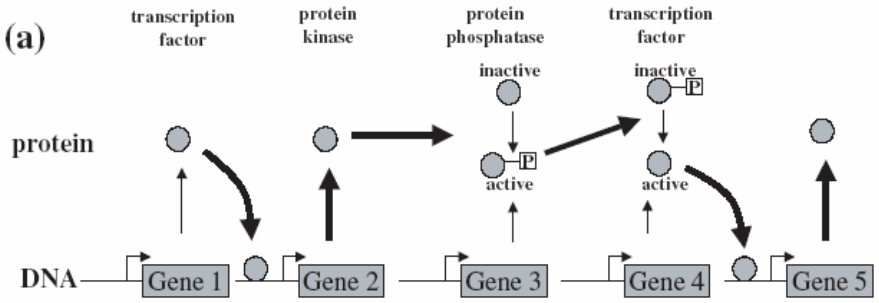
We can model an inducer like aTc as something that competes for the transcription factor.

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

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:	

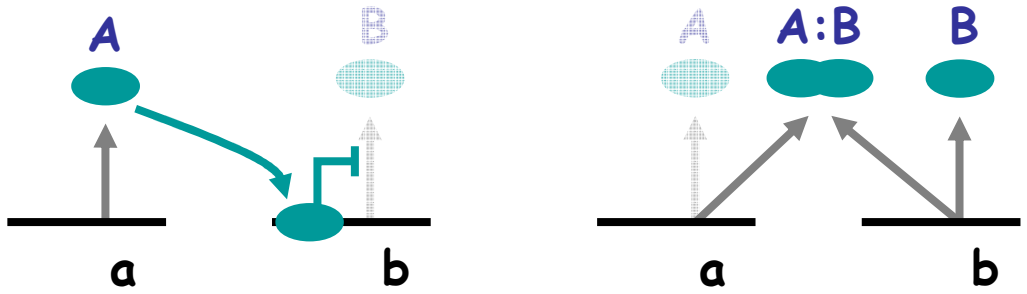
(c)
Aspect of gene activity: phosphorylation state
Genetic perturbation: gene deletion

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

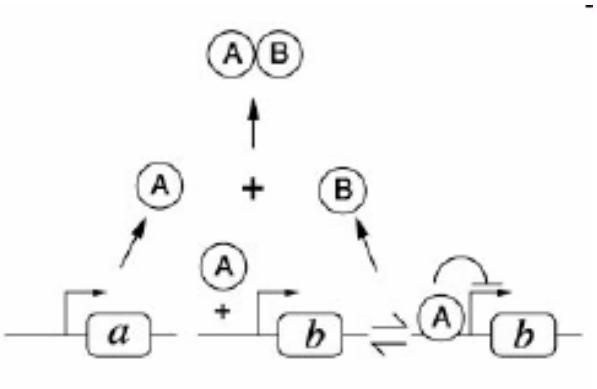
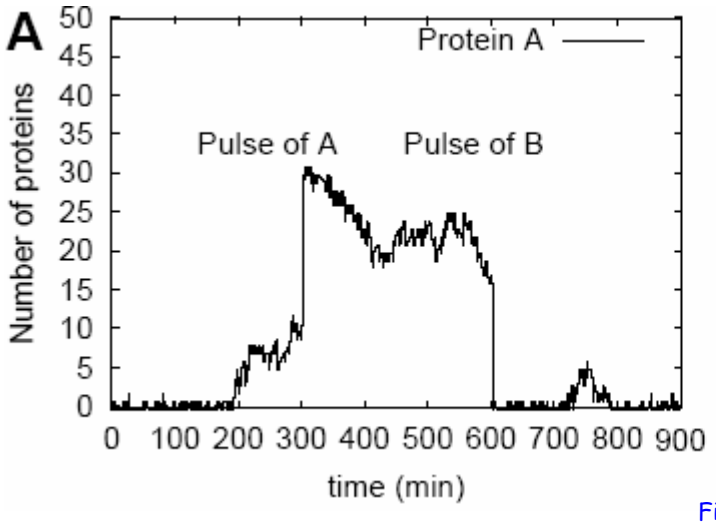


Fig 3A

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



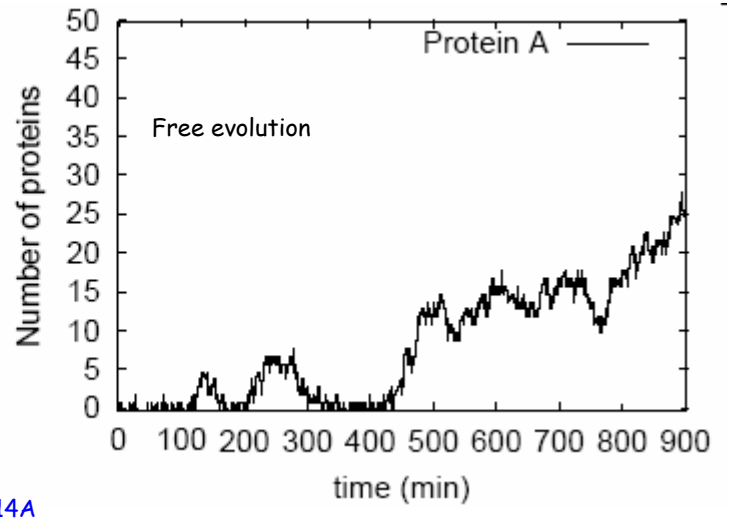
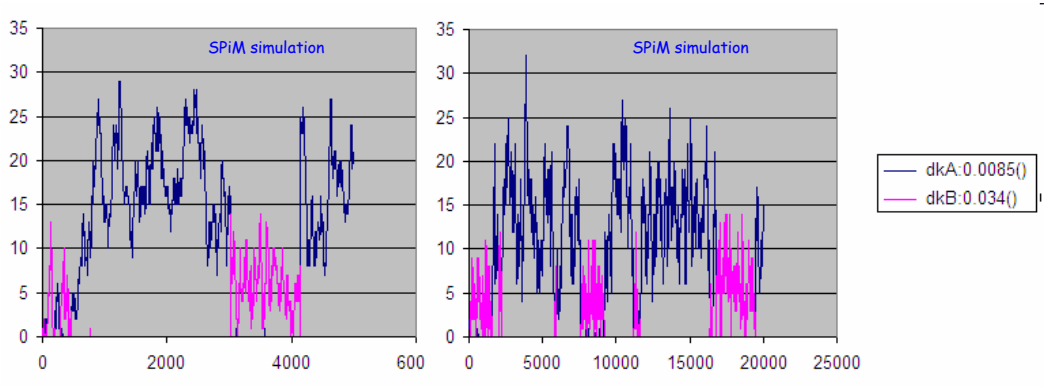


Fig 14A

François & Hakim Fig3A, SPiM simulation

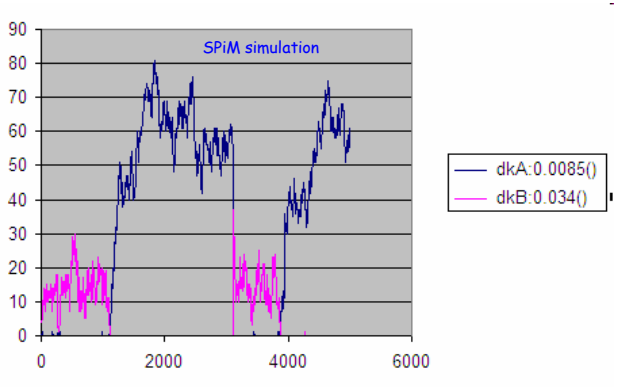
Parameters as in paper



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

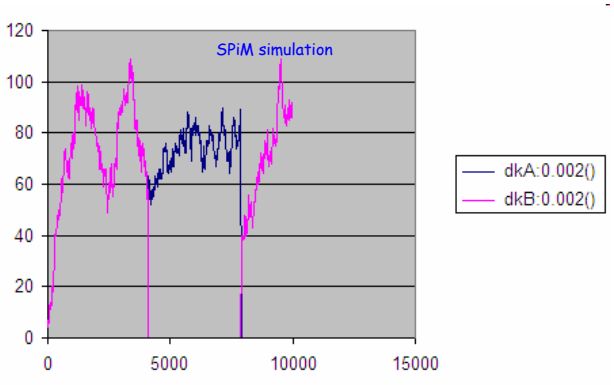
Free evolution

3 copies of each gene.

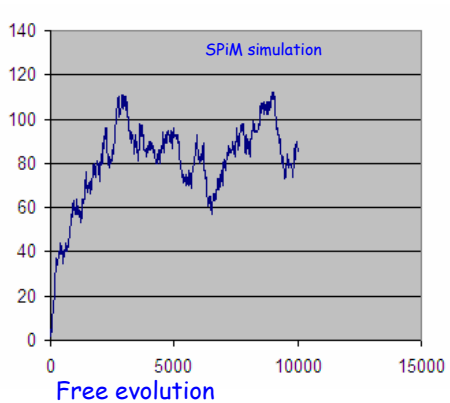


Spontaneous switch at ~1100
100xB injected at ~3000
30xA injected at ~4000

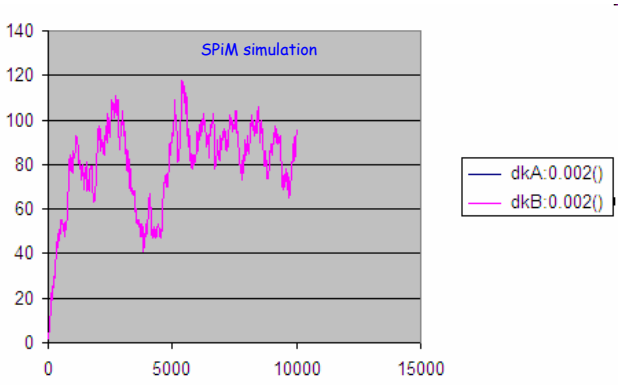
Modified for stability: dkA = 0.02, dkB = 0.02



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

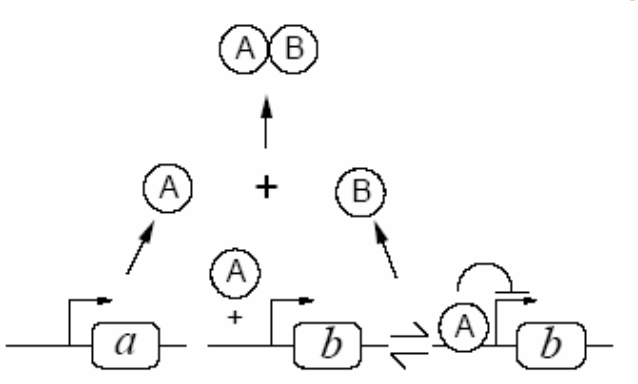


Fig 8

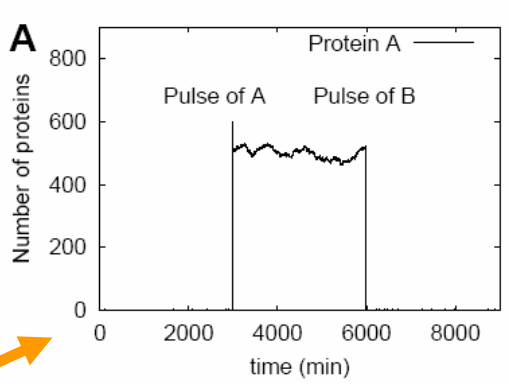
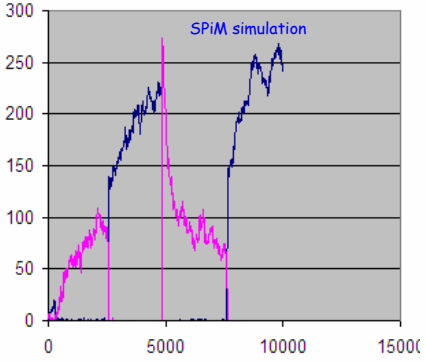
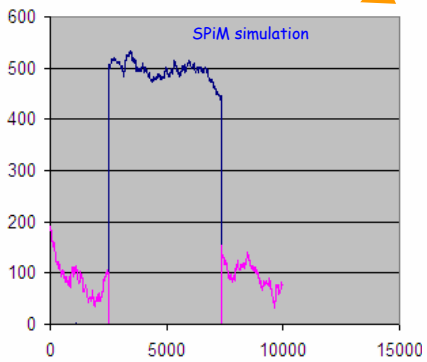


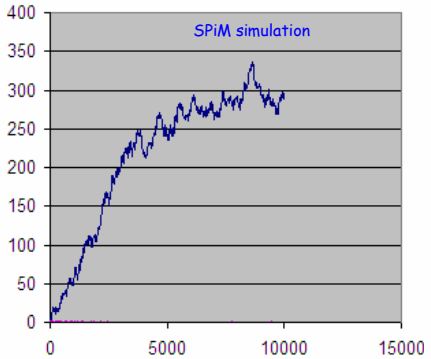
Fig 13A

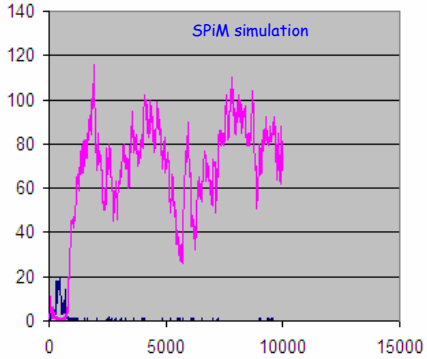


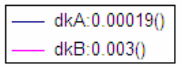
200xA injected at ~2500
500xB injected at ~5000
200xA injected at ~7500



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

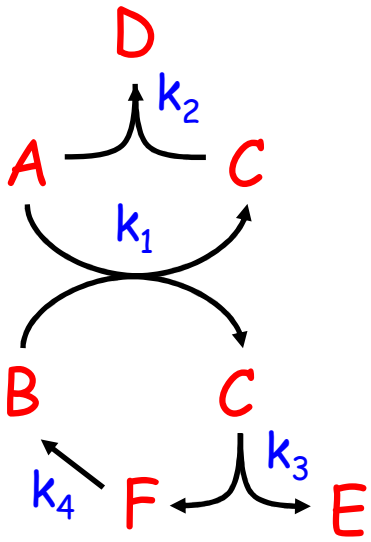
reactions

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: $F \rightarrow_{k_4} B$

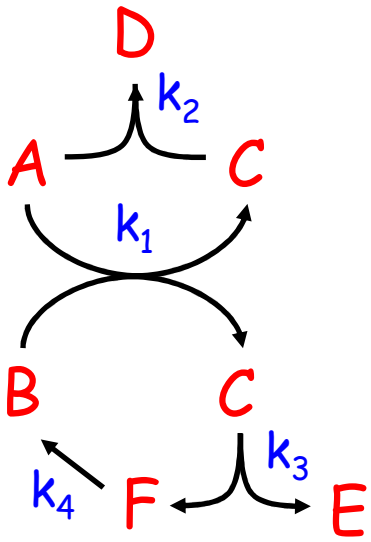
Write the coefficients by columns

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.).

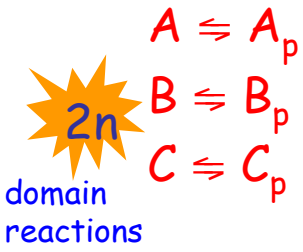
System description is exponential in the number of basic components.

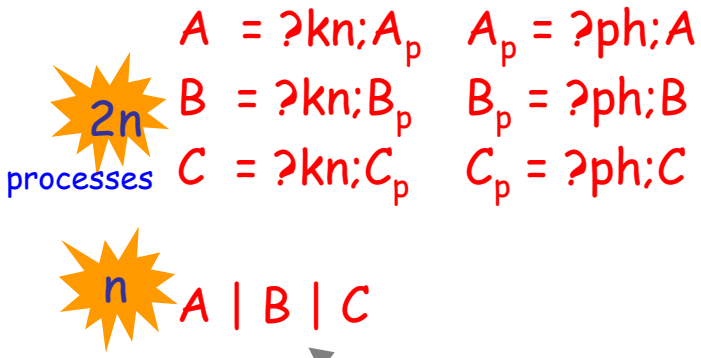
Stoichiometric Matrix

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.

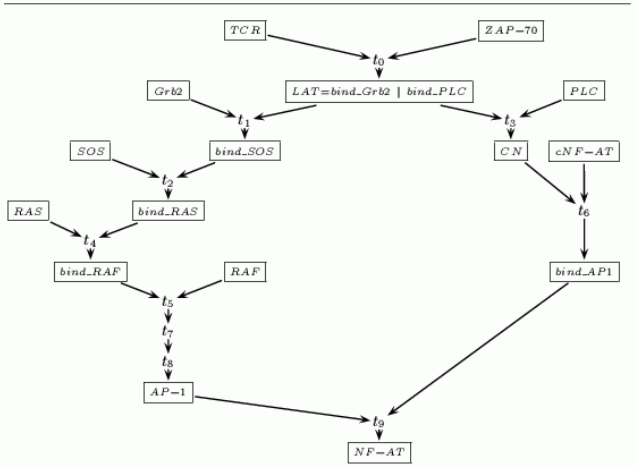


Fig. 2. A computation of *Sys*. For readability, the processes, enclosed in boxes, have no address. Causality (both on transitions and processes) is represented by the (Hasse diagram resulting from the) arrows; their absence makes it explicit concurrent activities.

- 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].

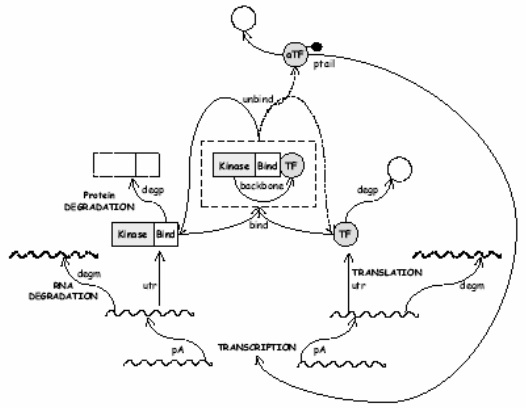


Fig. 1. Graphical presentation of Transcriptional Regulation by Positive Feedback [25].

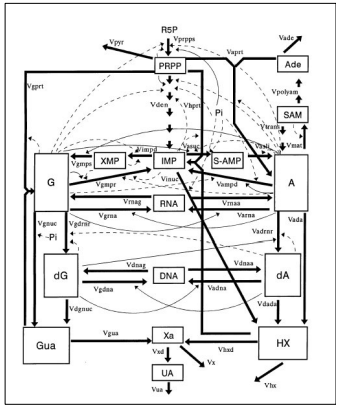
- Probabilistic Abstract Interpretation
  - [DiPierro Wicklicky].

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).

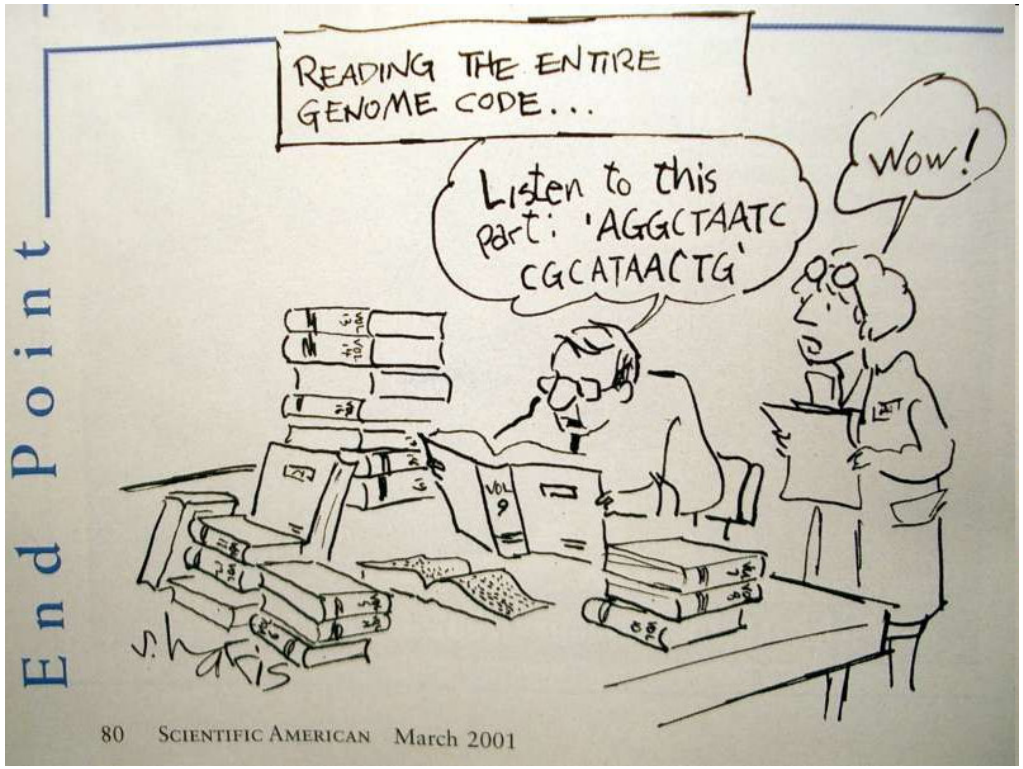
Disclaimers

- 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