A Compositional Approach to the Stochastic Dynamics of Gene Networks

Luca Cardelli

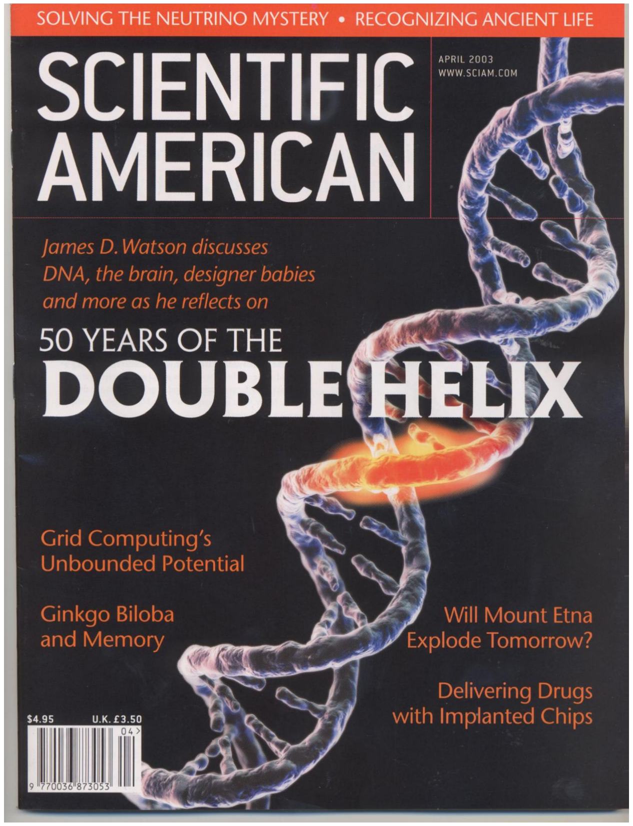
Microsoft Research

with Ralf Blossey and Andrew Phillips

Coquelles 2005-09-04

50 Years of <u>Molecular Cell Biology</u>

- 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


SOLVING THE NEUTRINO MYSTERY • RECOGNIZING ANCIENT LIFE
SCIENTIFIC AMERICAN
APRIL 2003
WWW.SCIAM.COM
James D. Watson discusses DNA, the brain, designer babies and more as he reflects on
50 YEARS OF THE
DOUBLE HELIX
Grid Computing's Unbounded Potential
Ginkgo Biloba and Memory
Will Mount Etna Explode Tomorrow?
Delivering Drugs with Implanted Chips
$4.95
U.K. £3.50

Towards Systems Biology

- Biologists now understand many of the cellular components
  - A whole team of biologists will typically study a single protein for years
  - Reductionism: understand the components in order to understand the system
- But this has not led to understand how "the system" works
  - Behavior comes from complex patterns of interactions between components
  - Predictive biology and pharmacology still rare
  - Synthetic biology still unreliable
- New approach: try to understand "the system"
  - Experimentally: massive data gathering and data mining (e.g. Genome projects)
  - Conceptually: modeling and analyzing networks (i.e. interactions) of components
- What kind of a system?
  - Just beyond the basic chemistry of energy and materials processing…
  - Built right out of digital information (DNA)
  - Based on information processing for both survival and evolution
  - *Highly* concurrent, nondeterministic, stochastic.
- Can we fix it when it breaks?
  - Really becomes: How is information structured and processed?

Storing *Processes*

- Today we represent, store, search, and analyze:
  - Gene sequence data
  - Protein structure data
  - Metabolic network data
  - Signaling pathway data
  - …

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

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

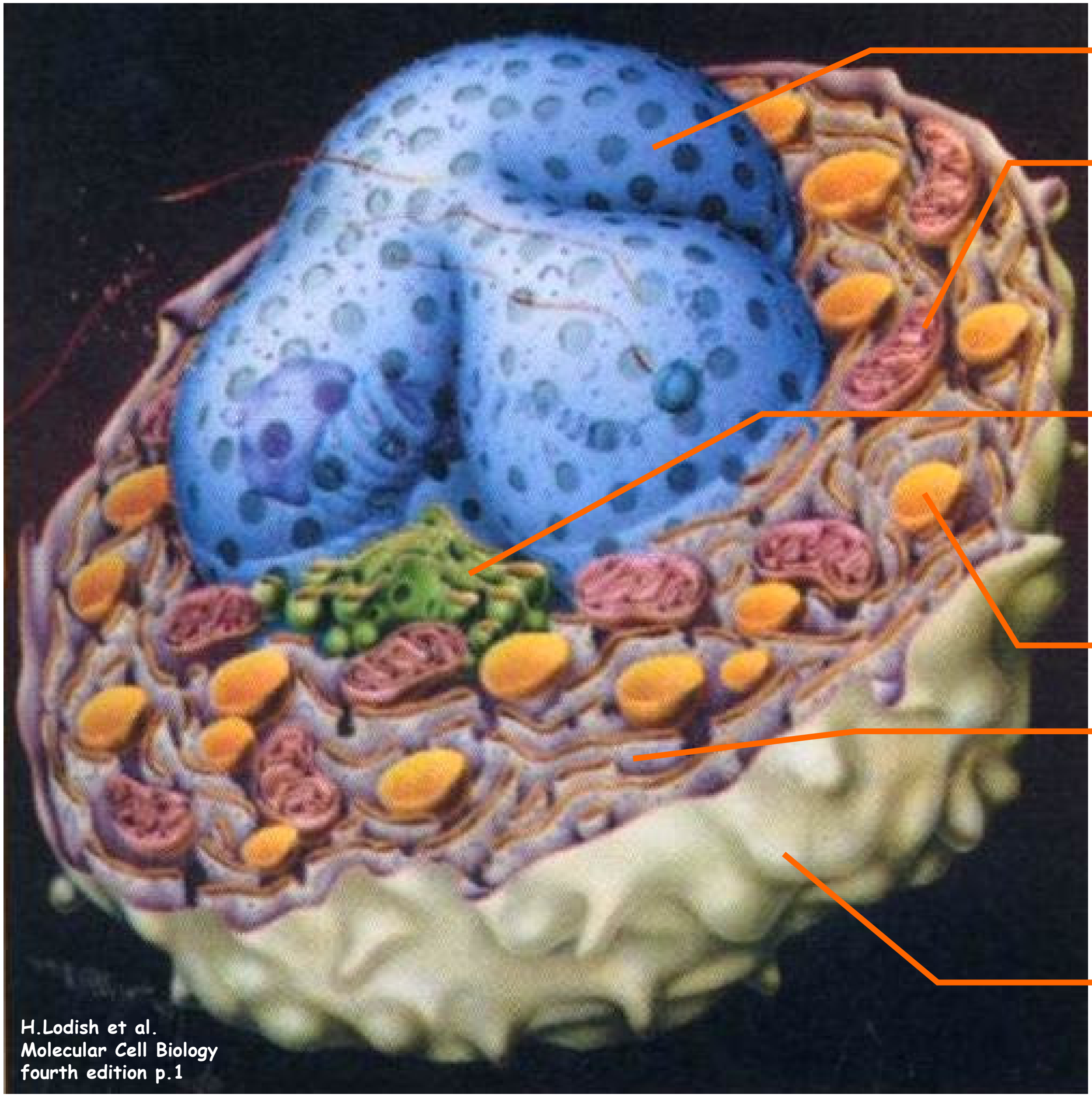
- In computing…
  - There are well-established scalable representations of dynamic reactive processes.
  - They look more or less like little, mathematically based, programming languages.

Structural Architecture

Eukaryotic Cell

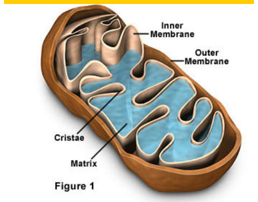
(10~100 trillion in human body)

Membranes everywhere

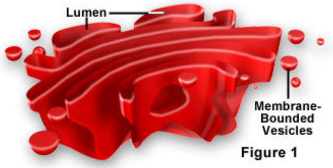

H.Lodish et al.
Molecular Cell Biology
fourth edition p.1

Nuclear membrane

Mitochondria

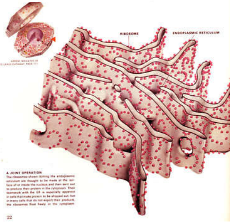


Golgi



Vesicles

E.R.



Plasma membrane (<10% of all membranes)

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

π-calculus (a Process Algebra)

- Processes P,Q,… - components of a system
- Channels a,b,… - interactions between components

0 the process that does nothing

!a(b); P the process that outputs b on channel a (and then does P)

?a(x); P the process that inputs b on channel a (and then does P{x})

P | Q the process made of subprocesses P and Q running concurrently

P + Q the process that behaves like either P or Q nondeterministically

*P the process that behaves like unboundedly many copies of P

=> recursive processes

=> unbounded number and species of processes

new x; P the process that creates a new channel x (and then does P{x})

=> private interactions

=> unbounded number and species of interactions

π-calculus (a Process Algebra)

- Dynamics

$$(!a(b);\ P) + P' \quad | \quad (?a(x);\ Q\{x\}) + Q' \quad \rightarrow \quad P \quad | \quad Q\{b\}$$

- "Compositional" descriptions
  - Describe how the individual components behave
    - i.e. how they interact with any environment they may be placed in
  - Build systems by combining components
    - each components is part of the environment for the other components
  - Behavior (and its analysis) arises from the combinatorics of interactions
    - state space can be arbitrarily larger than its compositional description

- For concurrent, nondeterministic, unbounded-state systems
  - Dynamic creation of new channels (e.g. binding sites)
  - Dynamic creation of new processes (e.g. proteins)

Stochastic π-calculus

- A stochastic variant of π-calculus:
  - Each channels has a stochastic "firing" rate 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 with Andrew Phillips).
  - Is directly executable (via the Gillespie algorithm from physical chemistry).
  - Is analyzable (large body of literature, at least in the non-stochastic case).

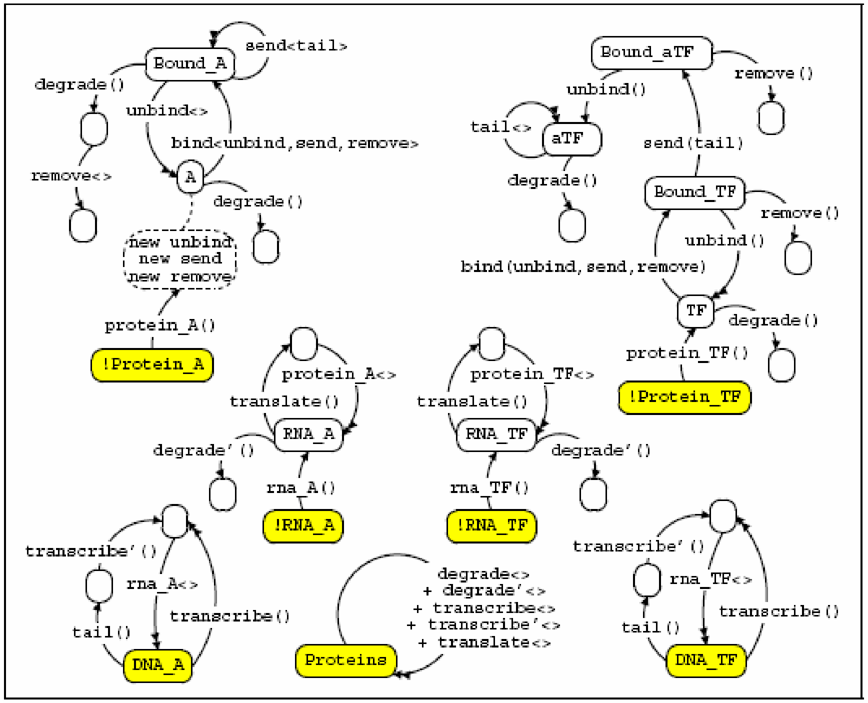


Figure 2. Regulating Gene Expression by Positive Feedback [9]

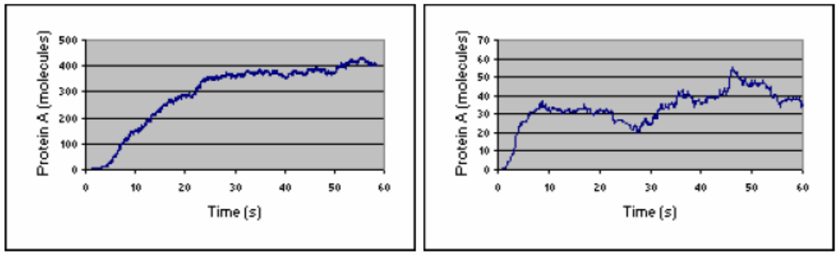


Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of TF

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

Stochastic π-calculus

- A stochastic variant of π-calculus:
  - Each channels has a stochastic "firing" rate 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 with Andrew Phillips).
  - Is directly executable (via the Gillespie algorithm from physical chemistry).
  - Is analyzable (large body of literature, at least in the non-stochastic case).

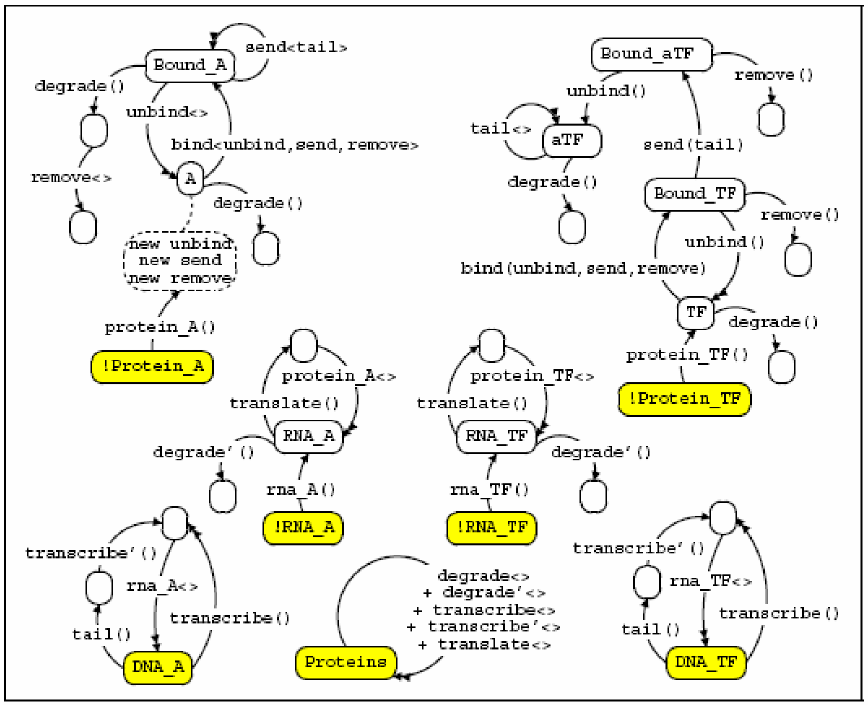


Figure 2. Regulating Gene Expression by Positive Feedback [9]

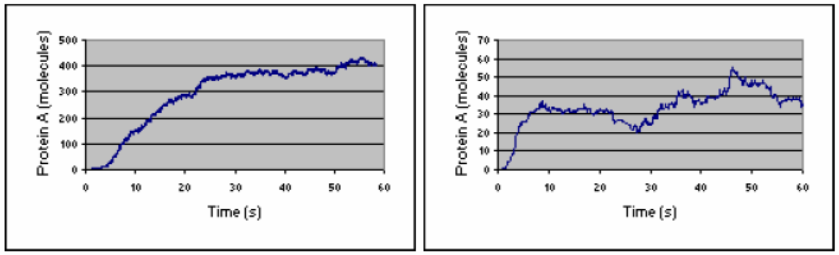


Figure 3. Protein A molecules v.s. time in presence (left) and absence (right) of TF

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

Chemistry vs. π-calculus

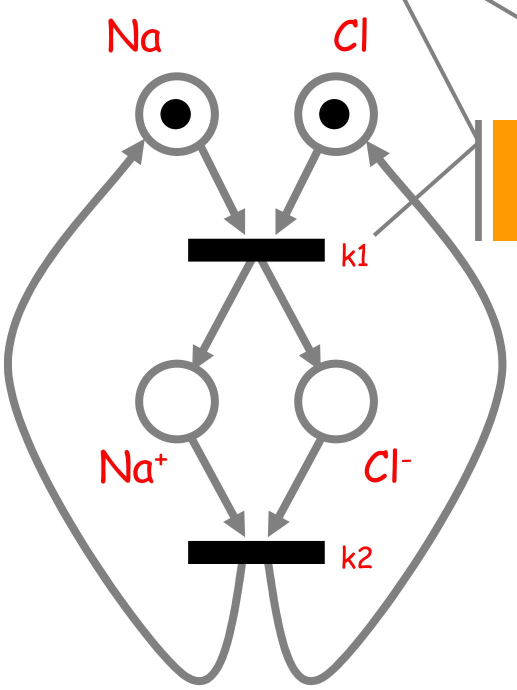
A process calculus (chemistry, or SBML)

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

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

1 line per reaction

Reaction oriented



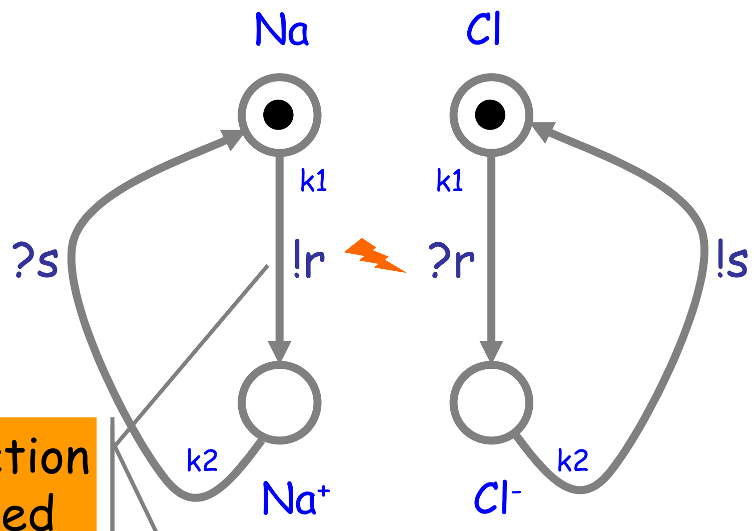
This Petri-Net-like graphical representation degenerates into spaghetti diagrams: precise and dynamic, but not scalable, structured, or maintainable.

The same "model"

Maps to a CTMC

Maps to a CTMC

A compositional graphical representation, and the corresponding calculus.



Interaction oriented

1 line per component

$$Na = !r_{k1};\ \overbrace{?s_{k2};\ Na}^{Na^+}$$

$$Cl = ?r_{k1};\ \underbrace{!s_{k2};\ Cl}_{Cl^-}$$

A different process calculus (π)

Modeling Biological Systems in Process Algebras

Suitable for multiple levels of abstraction:

- Chemistry and Biochemistry
  - Pioneering work by Ehud Shapiro and Aviv Regev (stochastic π-calculus)
  - "low level" modeling: close to the atoms and the proteins (if desired)

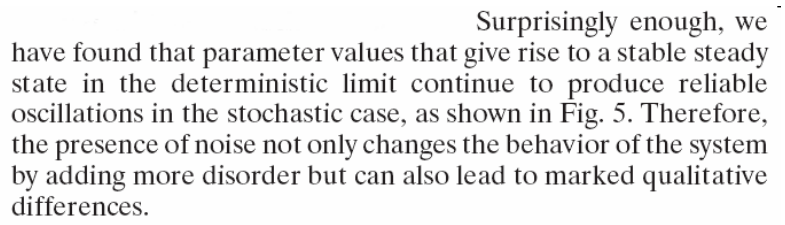
- Dynamic Compartments and Organelles
  - Myself, with above authors
  - "high level" modeling of compartments as a dynamic topology

- Gene Networks
  - This talk: myself with Ralf Blossey and Andrew Phillips
  - "high level" modeling of genes as stochastic *gates*

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.

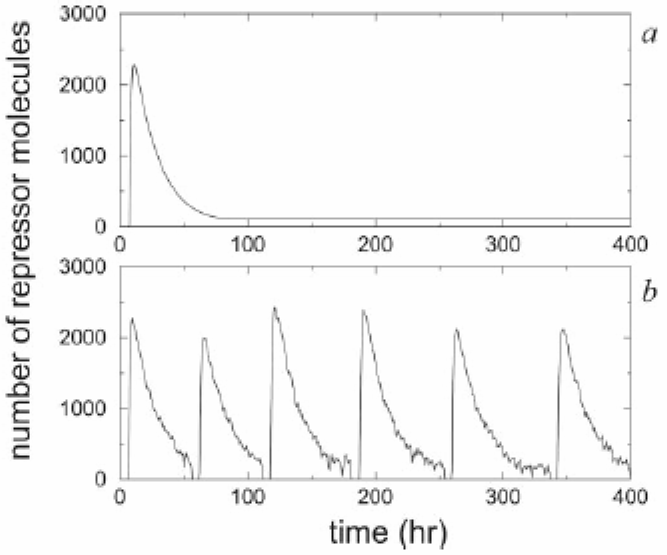


Fig. 5. Time evolution of R for the deterministic Eq. [1] (a) and stochastic (b) versions of the model. The values of the parameters are as in the caption of Fig. 1, except that now we set $\delta_R = 0.05\ h^{-1}$. For these parameter values, $\tau < 0$, so that the fixed point is stable.

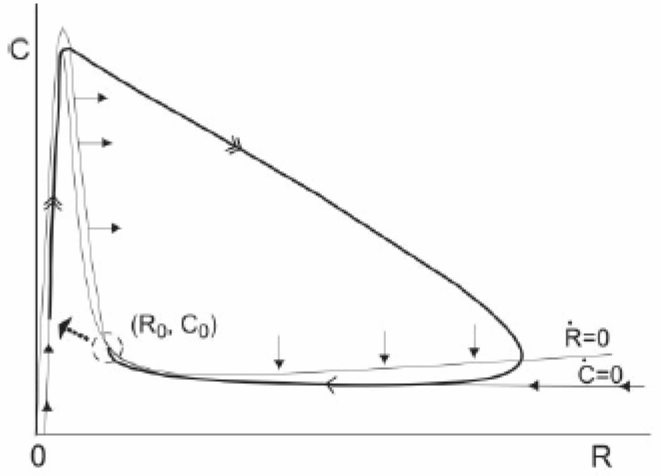


Fig. 6. Phase portrait as in Fig. 4 but for a situation in which the system falls into the stable fixed point (R_0, C_0). The dotted arrow to the left of the fixed point illustrates a perturbation that would initiate a single sweep of the (former) oscillatory trajectory.

Mechanisms of noise-resistance in genetic oscillators

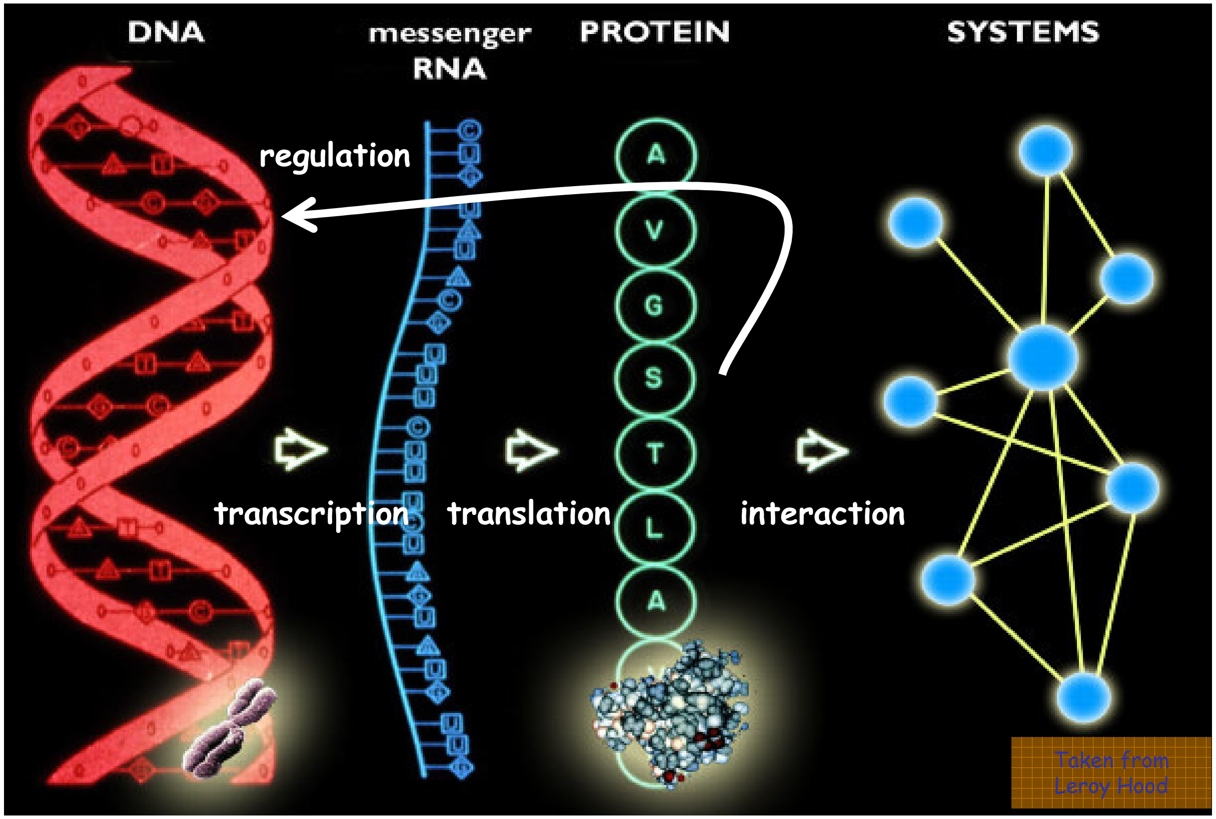
José M. G. Vilar, Hao Yuan Kueh, Naama Barkai, Stanislas Leibler

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

Gene Networks

The Gene Machine

The "Central Dogma" of Molecular Biology



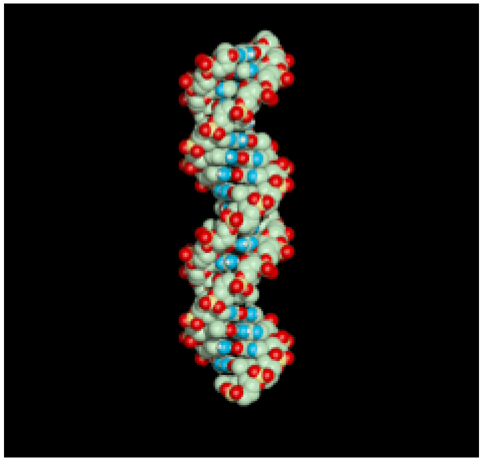
Taken from Leroy Hood

4-letter digital code

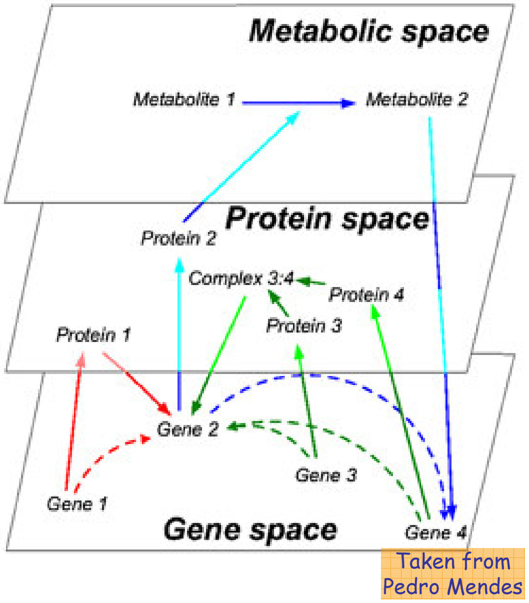
4-letter digital code

20-letter digital code

50.000(?) shapes

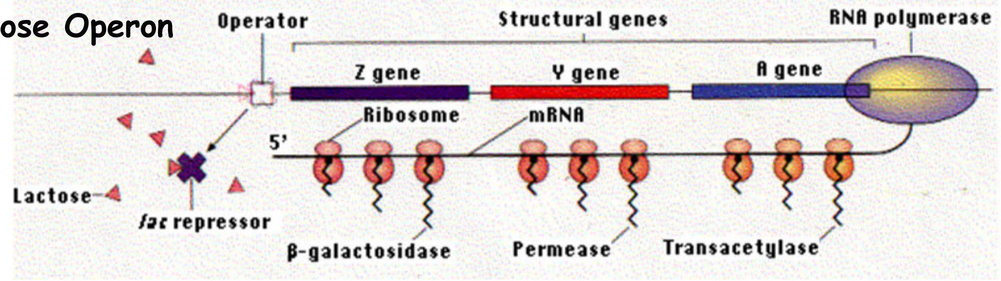


DNA Tutorial

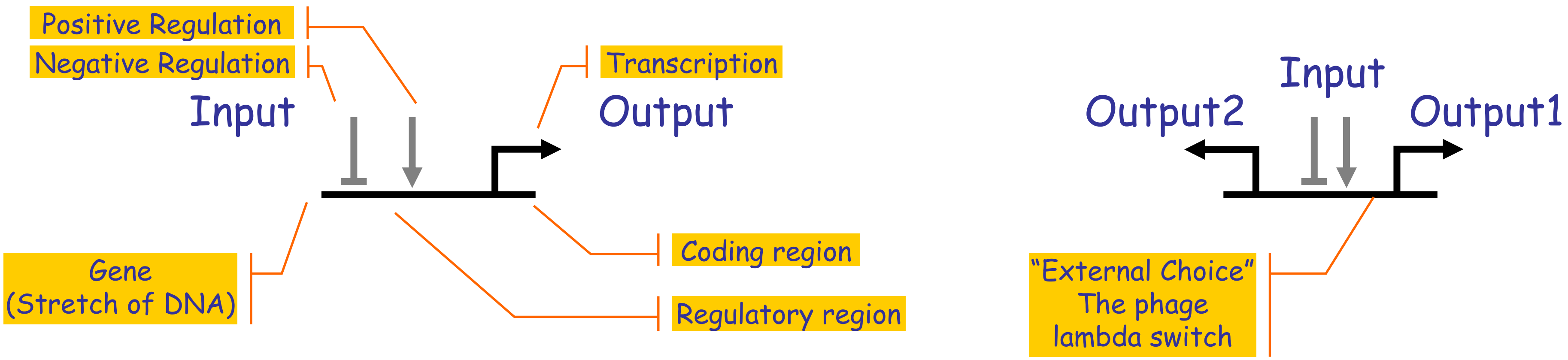


Taken from Pedro Mendes

Lactose Operon



The Gene Machine "Instruction Set"



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)

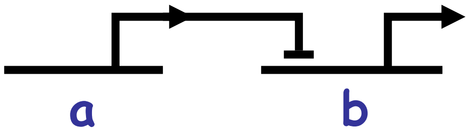
580,073bp 145KB (eBook)

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

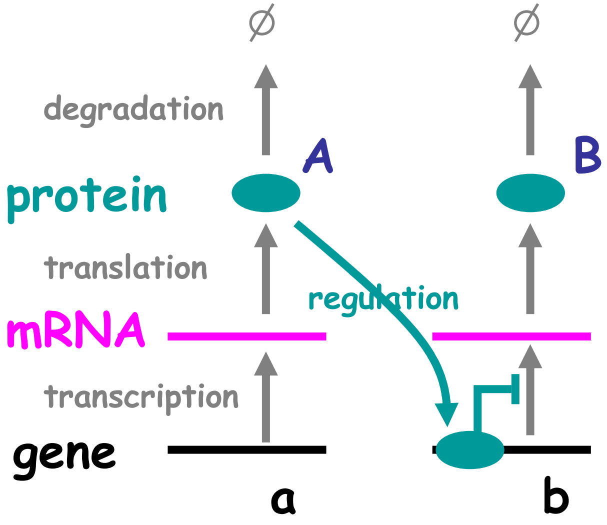
Yeast (eukarya): 12Mbp 3MB (MP3 song)

Wheat 17Gbp 4.25GB (DVD)

Gene Composition



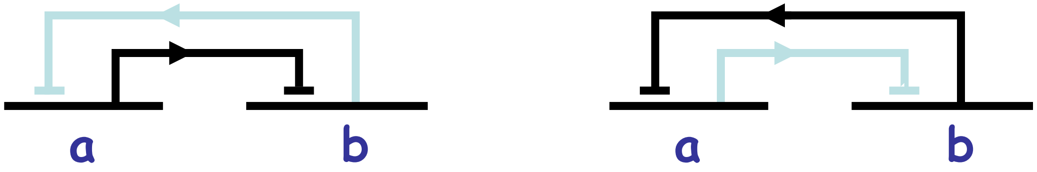
Is a shorthand for:



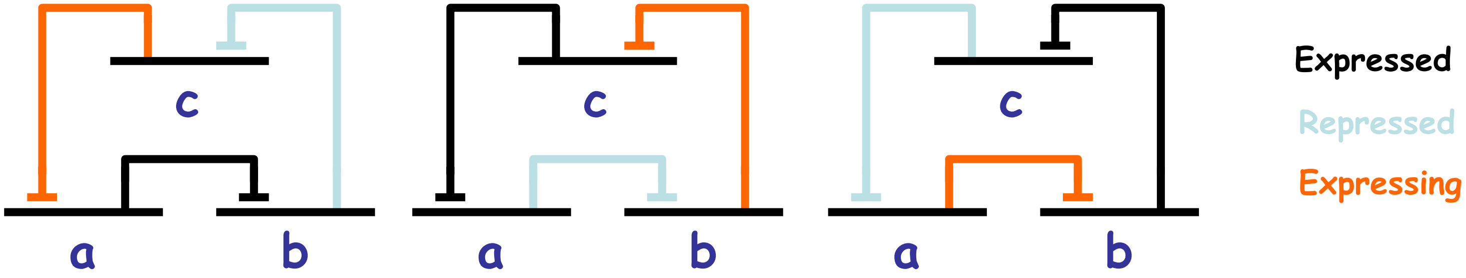
Under the assumptions [Kim & Tidor]

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



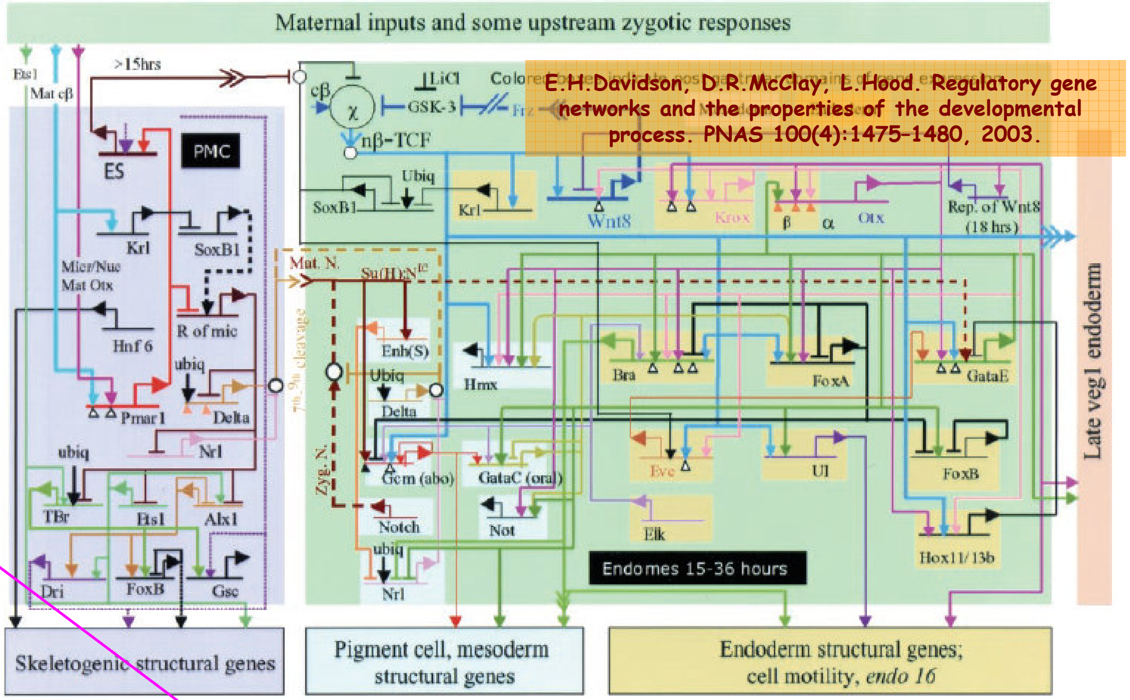
Ex: Oscillator



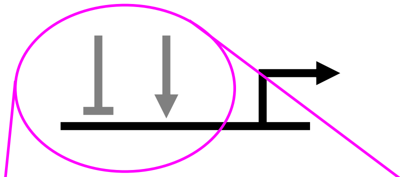
Gene Regulatory Networks

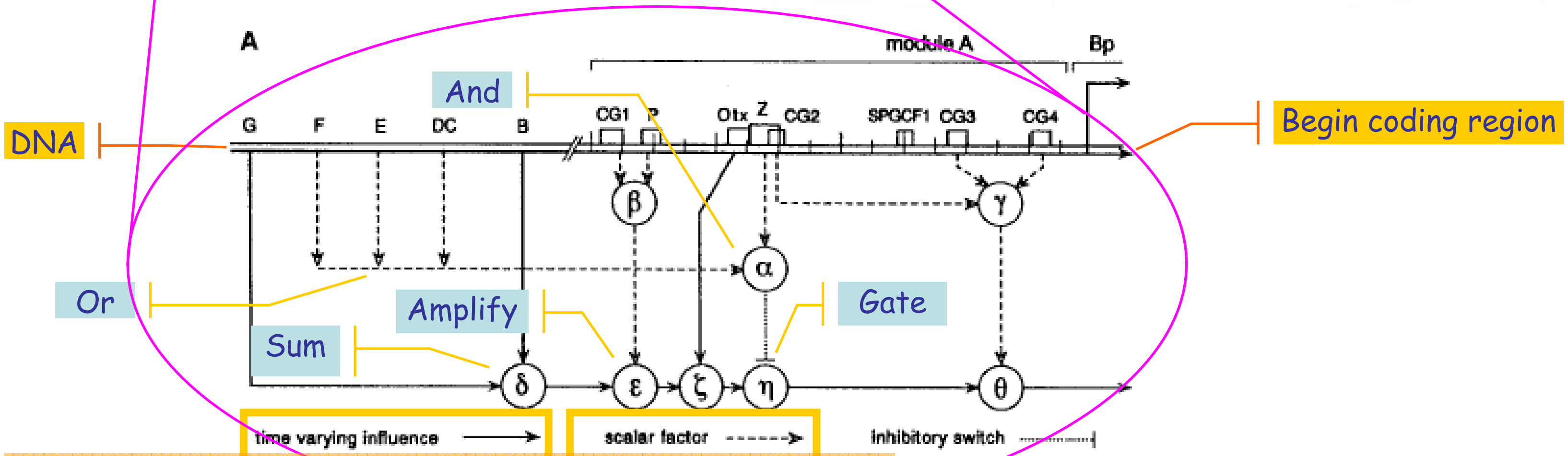
http://strc.herts.ac.uk/bio/maria/NetBuilder/

NetBuilder



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

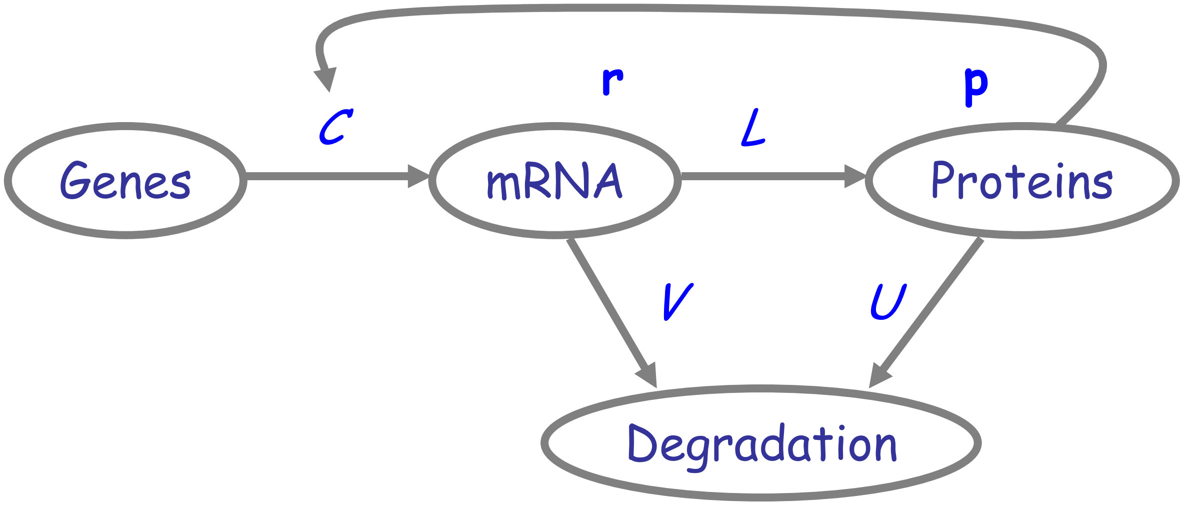




C-H.Yuh, H.Bolouri, E.H.Davidson. Genomic Cis-Regulatory Logic: Experimental and Computational Analysis of a Sea Urchin Gene. Science 279:1896-1902, 1998

(The Classical ODE Approach)

[Chen, He, Church]



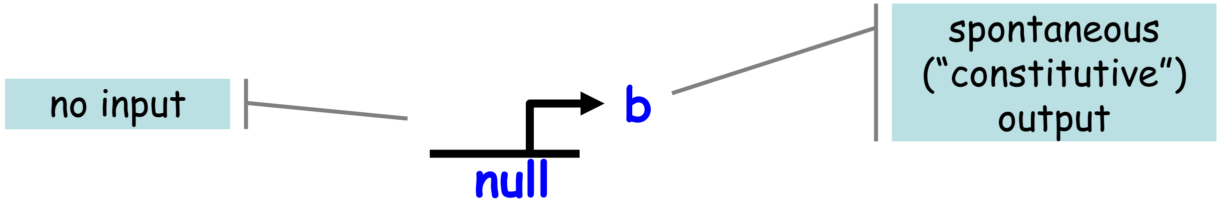
$$\frac{d\mathbf{r}}{dt} = f(\mathbf{p}) - V\mathbf{r}$$

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

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



interaction site of output protein

$$\mathrm{null}(b) \triangleq \tau_{\varepsilon};\ (\mathrm{tr}(b) \mid \mathrm{null}(b))$$

(recursive, parametric) process definition

stochastic delay (τ) with rate ε of constitutive transcription

output protein (transcription factor), spawn out

and repeat

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.

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

transcripton factor

and repeat

degradation

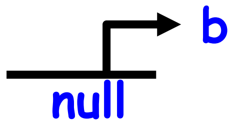
degradation rate δ

(output, !) interaction with rate r (input, ?, is on the target gene)

interaction site of transcription factor

stochastic choice (race between r and δ)

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



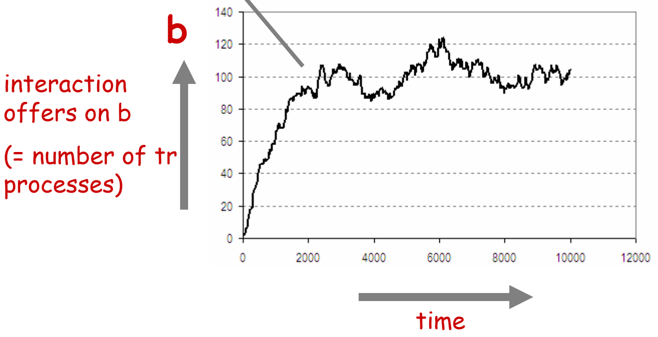
product

$$null(b) \triangleq \tau_\varepsilon;\ (tr(b)\ |\ null(b))$$

combined effect of production and degradation (without any interaction on b)

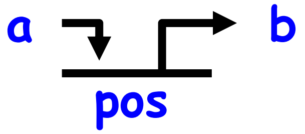
null(b)

ε=0.1, δ=0.001



Unary Pos Gate

input (excitatory)



output (stimulated or constitutive)

$pos(a,b) \triangleq$

$?a_r;\ \tau_\eta;\ (tr(b)\ |\ pos(a,b))\ +$

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

(input, ?) interaction with rate r

or constitutive transcription to always get things started

parallel, not sequence, to handle self-loops without deadlock

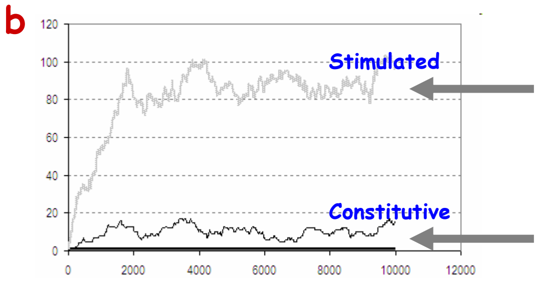
transcripton delay with rate η

race between r and ε

output protein

unlimited amount of

r=1.0, ε=0.01, η=0.1, δ=0.001

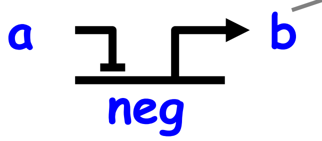


$*tr(a_r)\ |\ pos(a_r,b)$

$pos(a,b)$

Unary Neg Gate

input (inhibitory)



output (constitutive when not inhibited)

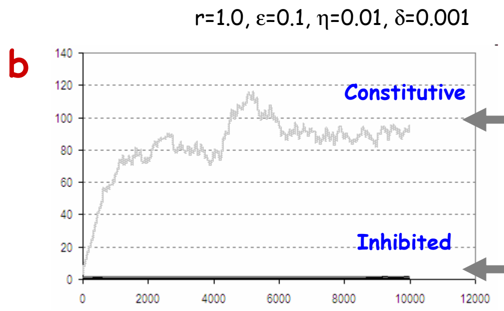
$$neg(a,b) \triangleq$$
$$?a_r;\ \tau_\eta;\ neg(a,b)\ +$$
$$\tau_\varepsilon;\ (tr(b)\ |\ neg(a,b))$$

(input, ?) interaction with rate r

or constitutive transcription to always get things started

inhibition delay with rate η

race between r and ε

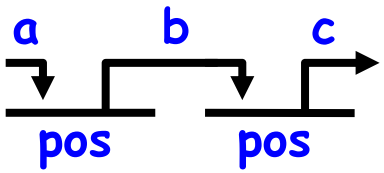


$neg(a_r,b)$

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

Signal Amplification

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

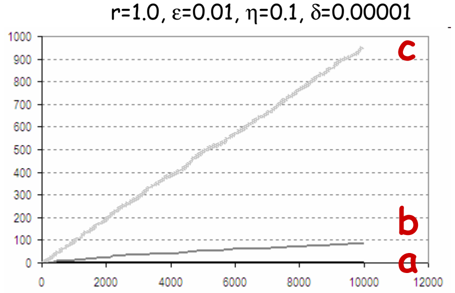


pos(a,b) ≜
$?a_r;\ \tau_\eta;\ (\mathrm{tr}(b) \mid \mathrm{pos}(a,b))\ +$
$\tau_\varepsilon;\ (\mathrm{tr}(b) \mid \mathrm{pos}(a,b))$

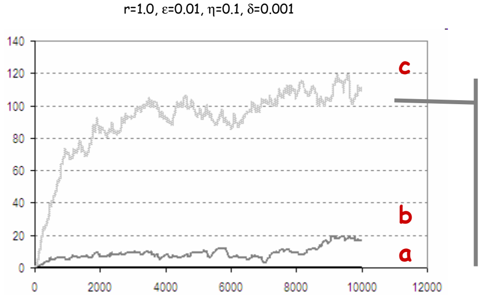
$\mathrm{tr}(p) \triangleq (!p_r;\ \mathrm{tr}(p)) + \tau_\delta$

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

With little degradation



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

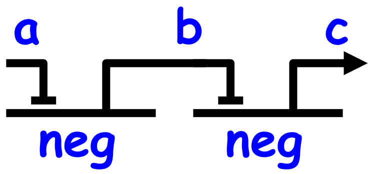


pos(a,b) | pos(b,c)

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

Signal Normalization

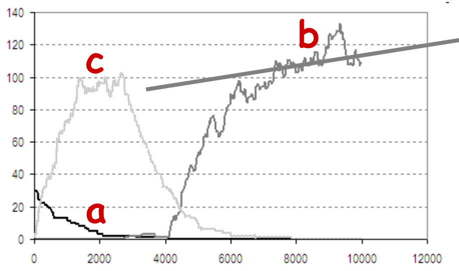
neg(a,b) |
neg(b,c)



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

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

r=1.0, ε=0.1, η=0.01, δ=0.001

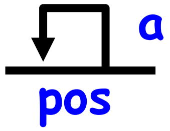


^{30*}tr(a) | neg(a,b) | neg(b,c)

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

Self Feedback Circuits

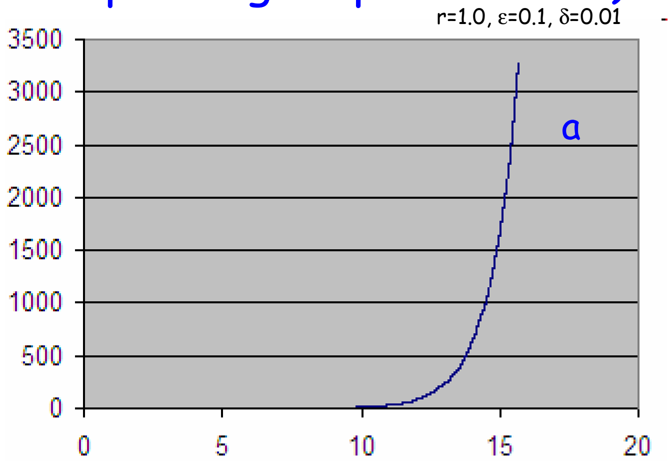
pos(a,a)



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

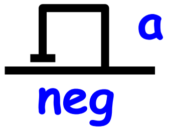
$$tr(p) \triangleq (!p_r;\ tr(p)) + \tau_\delta$$

(Can overwhelm degradation, depending on parameters)



pos(a,a)

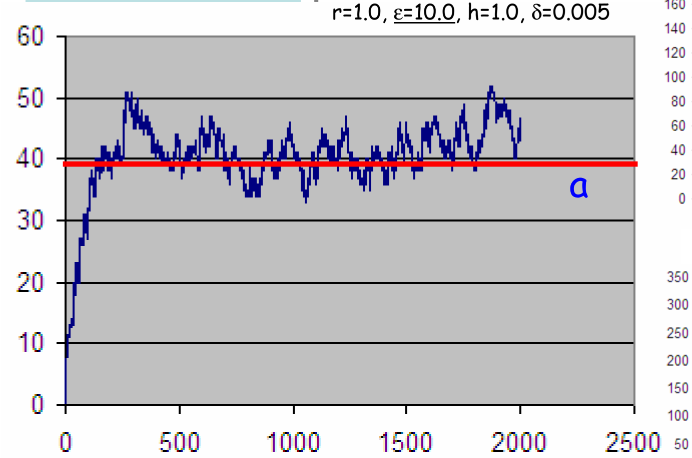
neg(a,a)



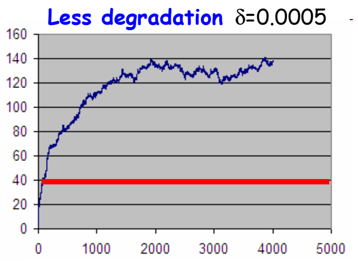
$$\begin{aligned}&neg(a,b) \triangleq \\ &\quad ?a_r;\ \tau_\eta;\ neg(a,b)\ + \\ &\quad \tau_\varepsilon;\ (tr(b) \mid neg(a,b))\end{aligned}$$

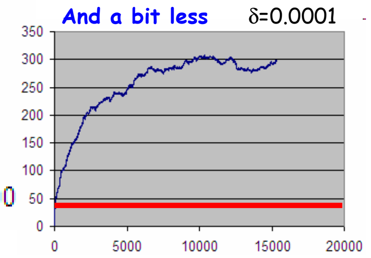
$$tr(p) \triangleq (!p_r;\ tr(p)) + \tau_\delta$$

high, to raise the signal



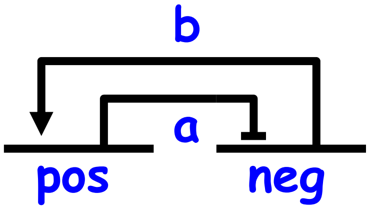
neg(a,a)



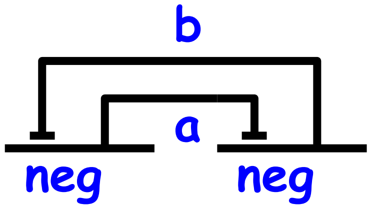


Two-gate Feedback Circuits

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



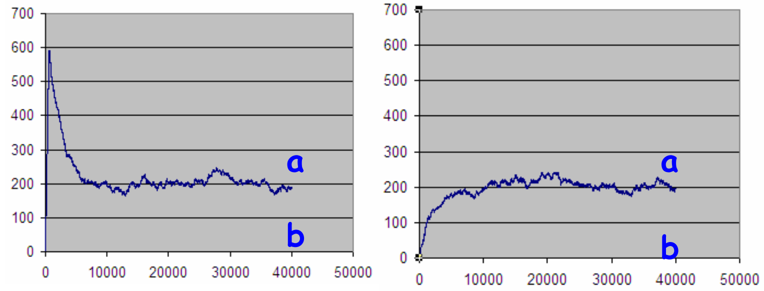
neg(b,a) | neg(a,b)



Monostable:

For some degradation rates is quite stable:

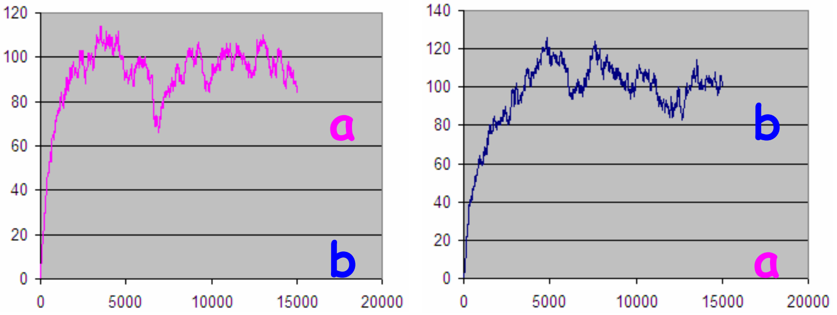
r=1.0, ε=0.1, h=0.01, δ=0.0005



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

Bistable:

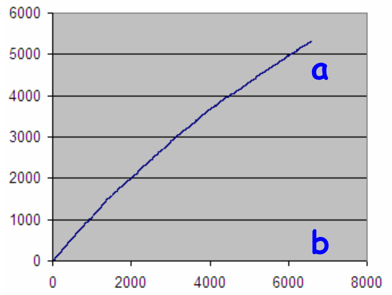
r=1.0, ε=0.1, h=0.01, δ=0.001



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

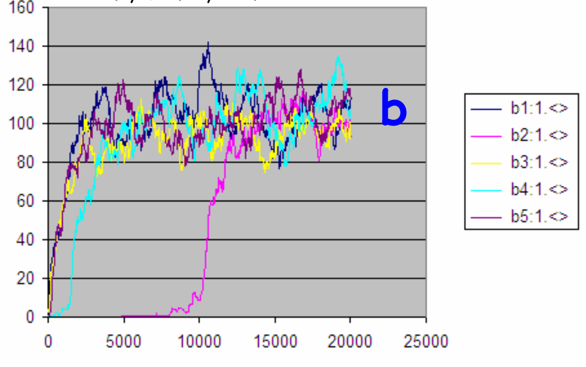
But with a small change in degradation, it goes wild:

r=1.0, ε=0.1, h=0.01, δ=0.0001



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

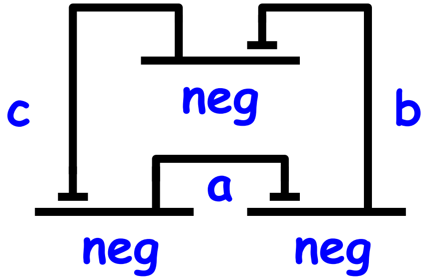
ε=0.1, h=0.01, δ=0.001



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

Repressilator

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



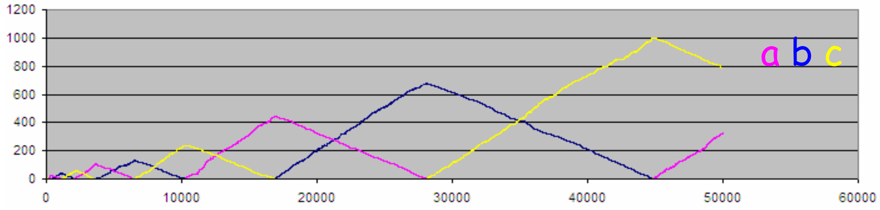
$neg(a,b) \triangleq$
$?a_r; \tau_\eta; neg(a,b) +$
$\tau_\varepsilon; (tr(b) \mid neg(a,b))$

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

$tr(p) \triangleq\ !p_r$

interact once and die
otherwise stick around

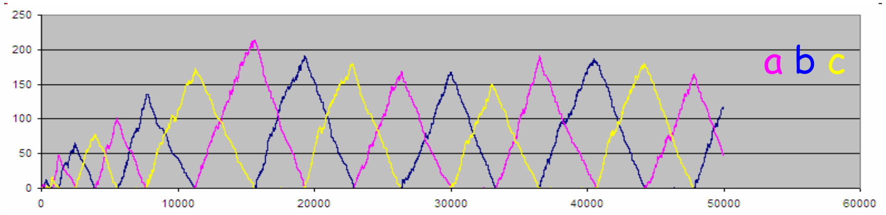
r=1.0, ε=0.1, h=0.04



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

interact once and die
otherwise decay

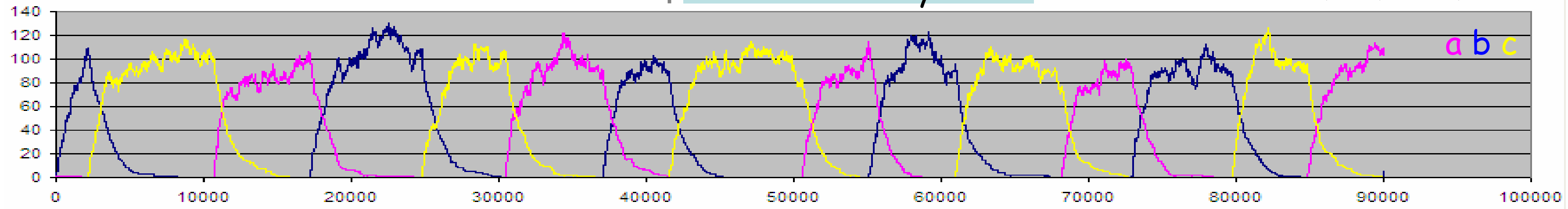
r=1.0, ε=0.1, h=0.04, δ=0.0001



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

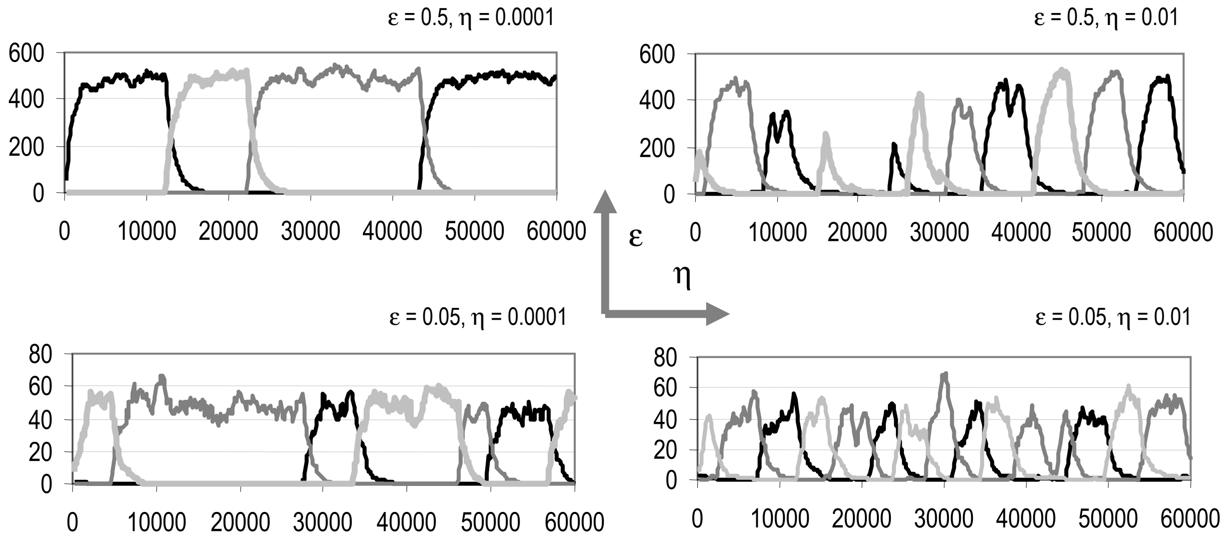
interact many times
and decay

r=1.0, ε=0.1, h=0.001, δ=0.001

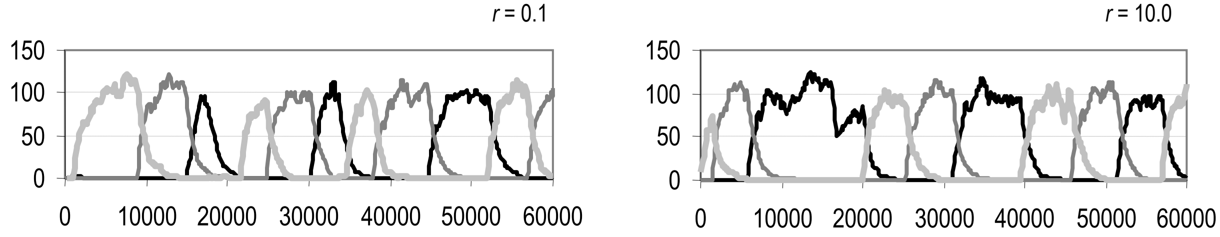


Subtle… at any point one gate is inhibited and the other two can fire constitutively. If one of them fires first, nothing really changes, but if the other one fires first, then the cycle progresses.

System Properties: Oscillation Parameters



The constitutive rate ε (together with the degradation rate) determines oscillation amplitude, while the inhibition rate η determines oscillation frequency.

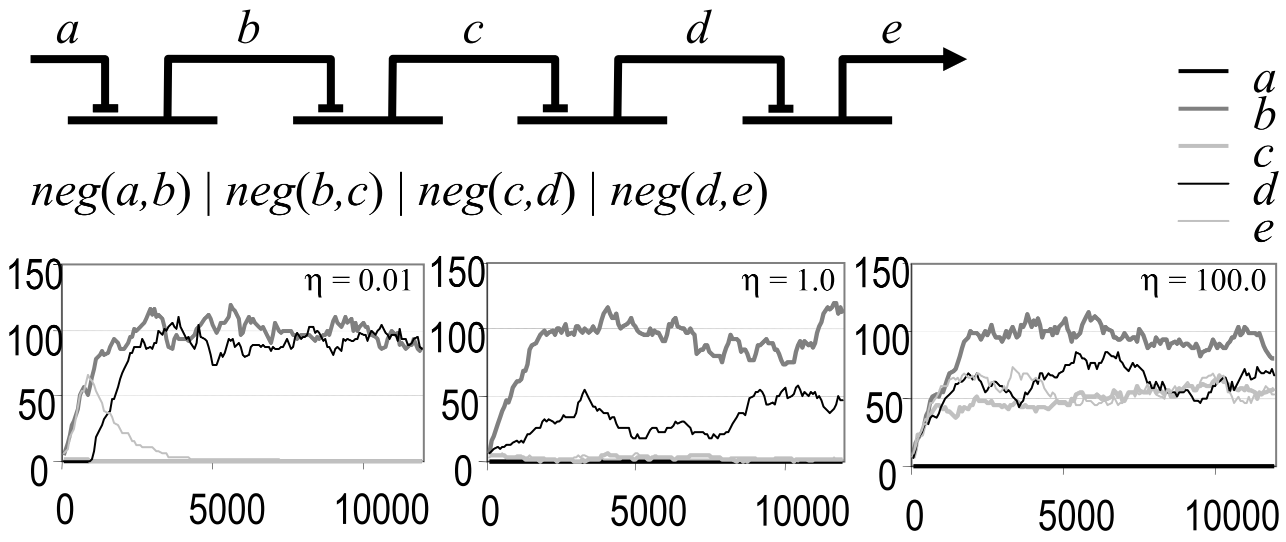


We can view the interaction rate *r* as a measure of the volume (or temperature) of the solution; that is, of how often transcription factors bump into gates. Oscillation frequency and amplitude remain unaffected in a large range of variation of *r*.

Repressilator in SPiM

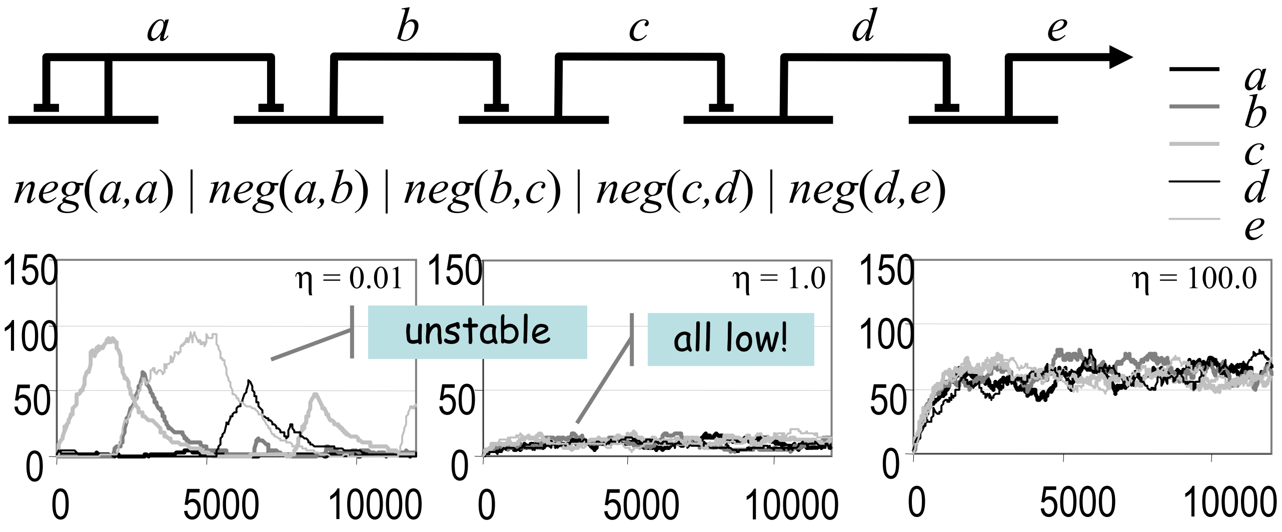
```
val dk = 0.001     (* Decay rate *)
val eta = 0.001    (* Inhibition rate *)
val cst = 0.1      (* Constitutive rate *)

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))
```

System Properties: Fixpoints



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

A sequence of neg gates behaves as expected, with alternating signals, (less "Booleanly" depending on attenuation).



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

unstable

all low!

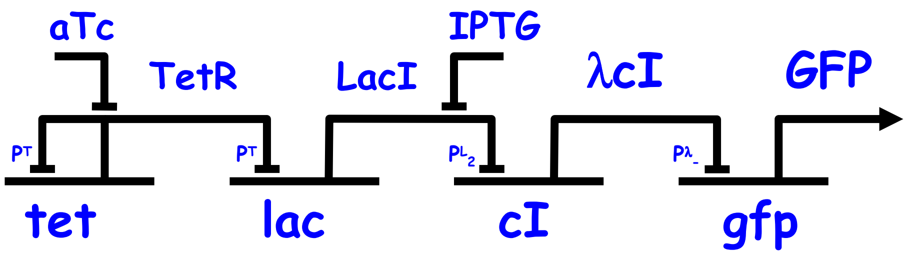
Now add a self-loop at the head. Not a Boolean circuit!

No more alternations, because... each gate is at its fixpoint.

Guet et al.

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

They engineered in E.Coli all genetic circuits with four single-input gates; such as this one:



Then they measured the GFP output (a fluorescent protein) in presence or absence of each of two inhibitors (aTc and IPTG).

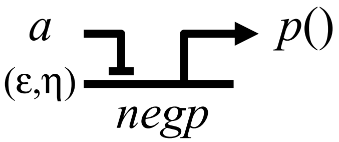
Experiment:	
aTc	0101
IPTG	0011
GFP	0100

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

Here "1" means "high brightness" and "0" means "low brightness" on a population of bacteria after some time. (I.e. integrated in space and time.)

Further Building Blocks

Negp Gate



regulatory input

rates

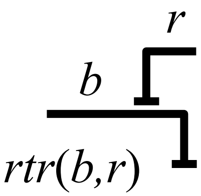
product

$negp(a,(\varepsilon,\eta),p) =$

product generation

$?a.\ \tau_{\eta}.\ negp(a,(\varepsilon,\eta),p)\ +$

$\tau_{\varepsilon}.\ (p()\ |\ negp(a,(\varepsilon,\eta),p))$



repressible factor

$rtr(b,r) =$

interaction | $!b.\ rtr(b,r)\ +$ | binding

interaction | $!r.\ 0\ +$ | repression

delay | $\tau_{\delta}.\ 0$ | degradation

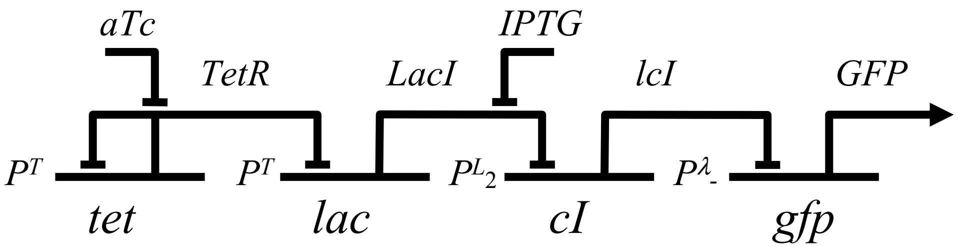
repressor

$rep(r) = ?r.\ rep(r)$

arbitray amounts of..

D038/lac⁻

D038/lac⁻



Experiment:

aTc	0101
IPTG	0011
GFP	0100

Naïve "Boolean" analysis would suggest GFP=0.5 (oscillation) because of self-loop.

GFP=1 there is consistent only with (somehow) the head loop setting TetR=LacI=0. But in that case, aTc should have no effect (it can only subtract from those signals) but instead it affects GFP.

channels $\boxed{TetR{:}\mathbf{r}_1,\ LacI{:}\mathbf{r}_2,\ lcI{:}\mathbf{r}_3,\ GFP{:}\mathbf{r}_4,\ aTc{:}\mathbf{r}_5,\ IPTG{:}\mathbf{r}_6}$

molecules

$$\boxed{P^T = (\varepsilon_1, \eta_1) \qquad P^{L}{}_2 = (\varepsilon_2, \eta_2) \qquad P^{\lambda}{}_{-} = (\varepsilon_3, \eta_3)}$$

promoters

$$tet = negp(TetR,\ P^T,\ rtr(TetR,aTc))$$
$$lac = negp(TetR,\ P^T,\ rtr(LacI,IPTG))$$
$$cI = negp(LacI,\ P^{L}{}_2,\ tr(lcI))$$
$$gfp = negp(lcI,\ P^{\lambda}{}_{-},\ tr(GFP))$$

genes

repressors (when present)

$$D038lac^{-} = \boxed{tet \mid lac \mid cI \mid gfp} \ \mid\ \boxed{rep(aTc) \mid rep(IPTG)}$$

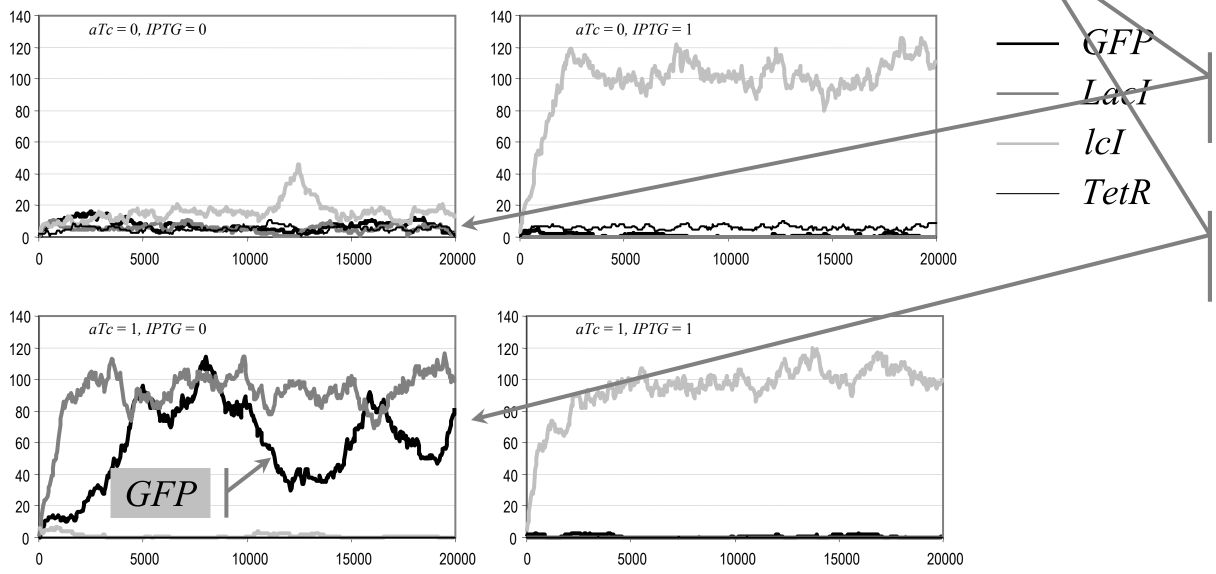
Hence we need to understand better the "dynamics" of this network.

Simulation results for D038/lac$^-$

Experiment:

aTc	0101
IPTG	0011
GFP	0100

r=1.0, ε=0.1, h=1.0, δ=0.001

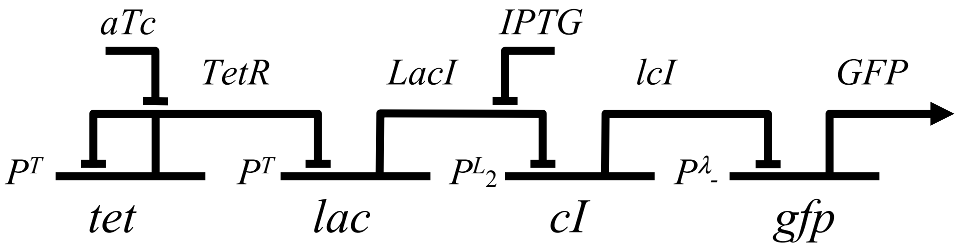


$r = 1.0$, $\varepsilon = 0.1$, $\eta = 0.25$ (P^T), $\eta = 1.0$ ($P^L{}_2$, $P^\lambda{}_-$), $\delta = 0.001$

The fixpoint effect can explain this (all signals set very low).

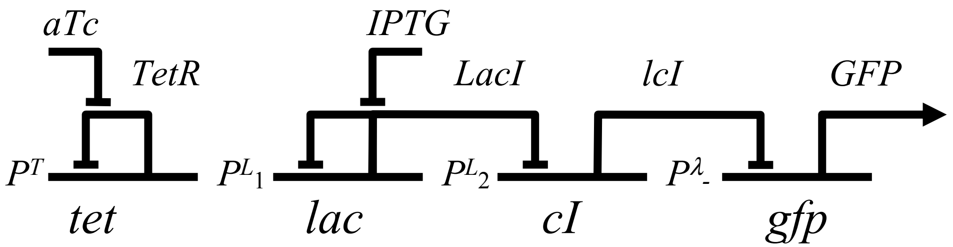
Then, aTc can destabilize the fixpoint, explaining GFP high (oscillating)

D038/lac$^-$



D016/lac$^-$

D016/lac$^-$



Experiment:

aTc	0101
IPTG	0011
GFP	1000

How can aTc affect the result??

One theory: aTc prevents the self-inhibition of tet, so that a very large quantity of TetR is produced. That then overloads the overall degradation machinery of the cell, affecting the rest of the circuit.

Even so, how can GFP be high here?

Even the fixpoint explanation fails here, unless we assume that the lac gate is operating in its instability region.

$\underline{channels}$ $\boxed{TetR{:}\mathbf{r_1},\ LacI{:}\mathbf{r_2},\ lcI{:}\mathbf{r_3},\ GFP{:}\mathbf{r_4},\ aTc{:}\mathbf{r_5},\ IPTG{:}\mathbf{r_6}}$

$$\boxed{P^T = [\varepsilon_1, \eta_1] \quad P^L{}_2 = [\varepsilon_2, \eta_2] \quad P^\lambda{}_- = [\varepsilon_3, \eta_3] \quad P^L{}_1 = [\varepsilon_4, \eta_4]}$$

promoters

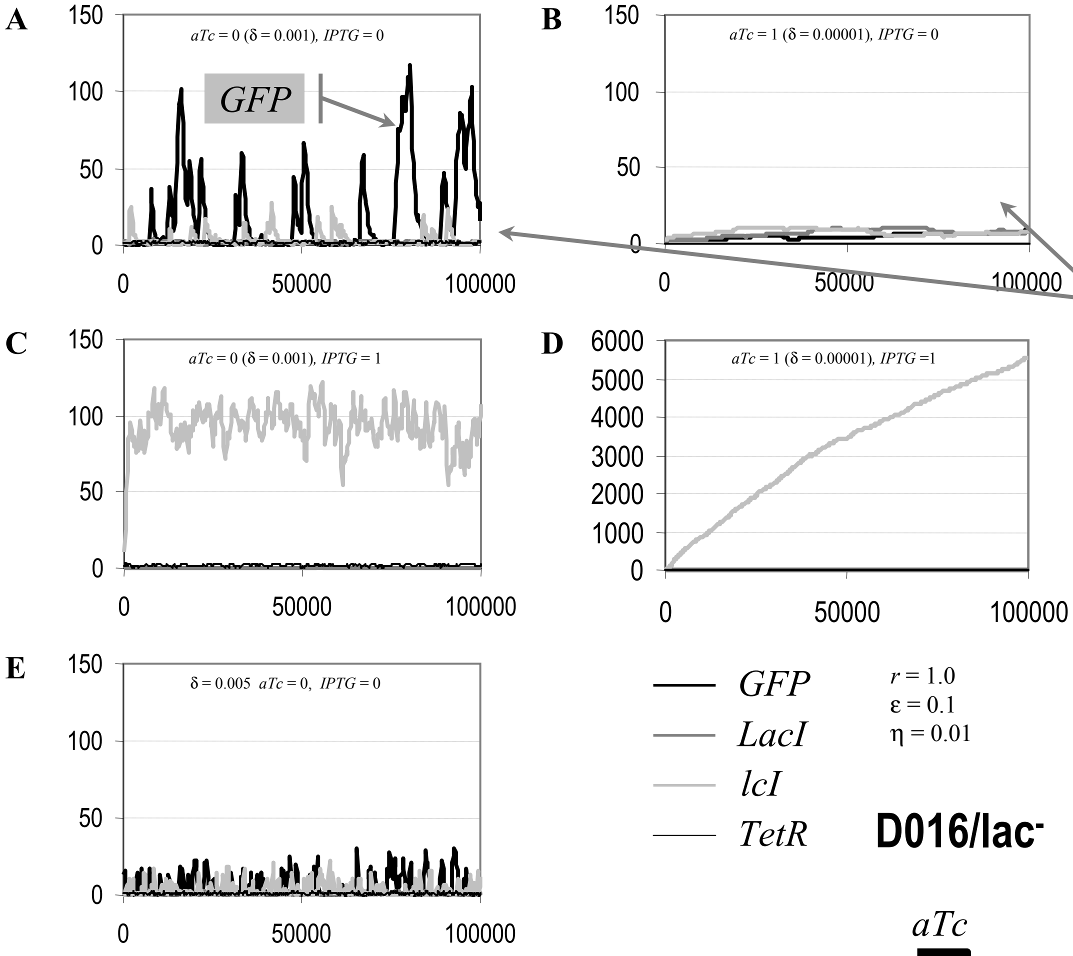
$tet = negp[TetR,\ P^T,\ rtr[TetR,aTc]]$
$lac = negp[LacI,\ P^L{}_1,\ rtr[LacI,IPTG]]$
$cI = negp[LacI,\ P^L{}_2,\ tr[lcI]]$
$gfp = negp[lcI,\ P^\lambda{}_-,\ tr[GFP]]$

genes

repressors

$$D016lac^- = \boxed{tet \mid lac \mid cI \mid gfp} \mid \boxed{rep[aTc] \mid rep[IPTG]}$$

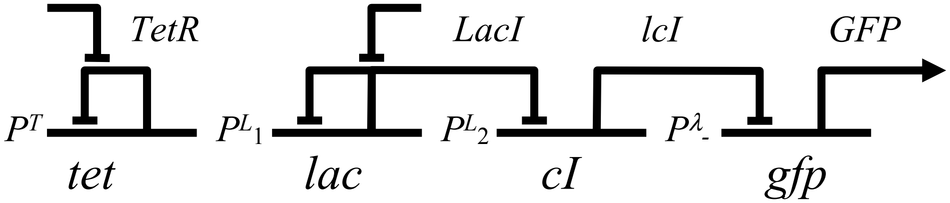
Simulation results for D016/lac$^-$



Experiment:	
aTc	0101
IPTG	0011
GFP	1000

The fixpoint effect, in instability region, explains this: GFP high because wildly oscillating.

Overloading of degradation machinery, induced by aTc, can reinstate the fixpoint regime.



What was the point?

- Deliberately pick a controversial/unsettled example to test the methodology.
- Show that we can easily "play with the model" and run simulations.
- Get a feeling for the kind of subtle effects that may play a role.
  - In particular, stochastic effects (wild oscillations) seem essential to some explanations.
- Get a feeling for kind of analysis that is required to understand the behavior of these systems.
- In the end, we are never "understanding" anything; we are just building theories/models that support of contradict experiments (and that suggest further experiments).

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.

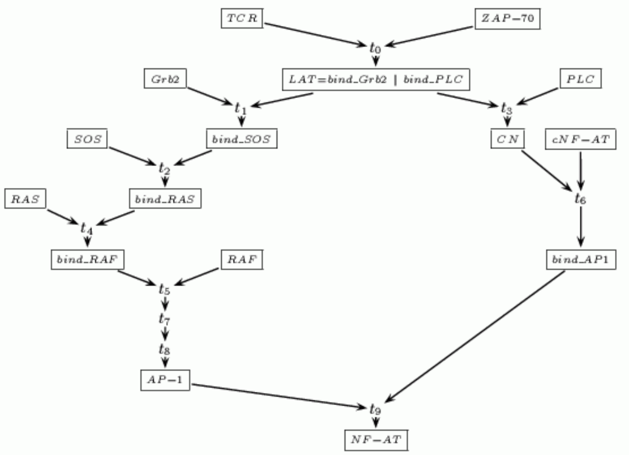


Fig. 2. A computation of *Sys*. For readability, the processes, enclosed in boxes, have no address. Causality (both on transitions and processes) is represented by the (Hasse diagram resulting from the) arrows; their absence makes it explicit concurrent activities.

- 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].

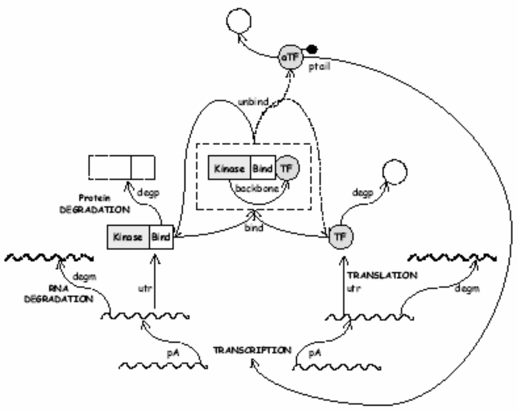


Fig. 1. Graphical presentation of Transcriptional Regulation by Positive Feedback [25].

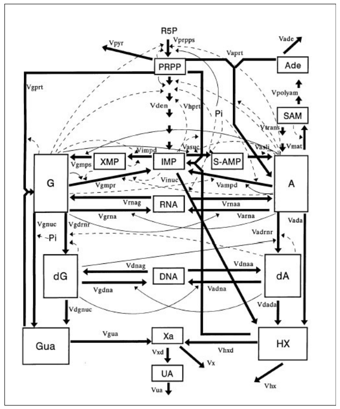
- Probabilistic Abstract Interpretation
  - [DiPierro Wicklicky].

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 Process Algebras Can Do For Us

Formalize mechanistic modeling

- Directly: one process for each "gear in the machine"; one process for each "blob on a biologist's cartoon".

Codify complex systems concisely

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

Calculate and analyze

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

Visualize

- Automata-like presentations.
- State Charts, Live Sequence Charts [Harel]

Reason

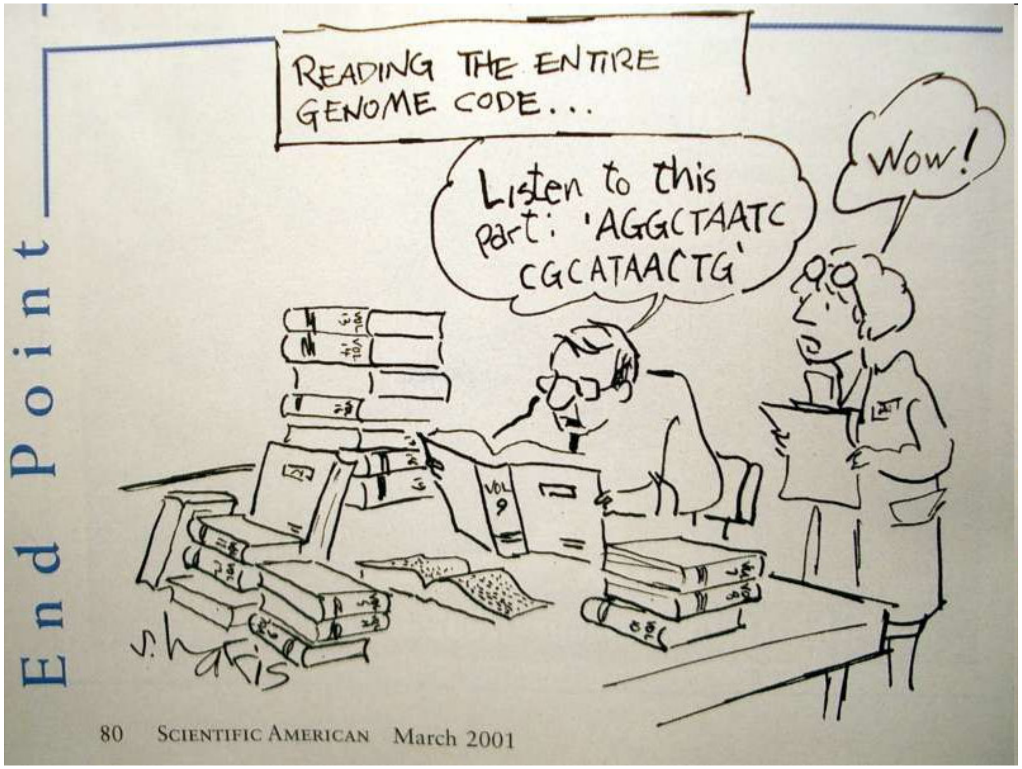
- Suitable equivalences on processes induce algebraic laws.
- We can relate different systems (e.g. equivalent behaviors).
- We can relate different abstraction levels.
- We can use equivalences for state minimization (symmetries).

Disclaimers

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

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

Conclusions


End Point
READING THE ENTIRE GENOME CODE...
Listen to this part: 'AGGCTAATC CGCATAACTG'
Wow!
VOL 9
S. Harris
80 SCIENTIFIC AMERICAN March 2001

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