Theory is closer to practice in theory than in practice.

Case Studies

Luca Cardelli
Microsoft Research
The Microsoft Research - University of Trento Centre for Computational and Systems Biology
Trento, 2006-05-22..26

www.luca.demon.co.uk/ArtificialBiochemistry.htm
Developing the Use of Process Algebra in the Derivation and Analysis of Mathematical Models of Infectious Disease

R. Norman and C. Shankland

Department of Computing Science and Mathematics, University of Stirling, UK.

\{cs,ran\}@cs.stir.ac.uk

Abstract. We introduce a series of descriptions of disease spread using the process algebra WS 


## Epidemics

2006-06-04

Luca Cardelli
Epidemics

As opposed to the way it is normally done.

The model consists of a system of three coupled nonlinear ordinary differential equations,

\[\begin{align*}
\frac{dS}{dt} &= -\beta SI \\
\frac{dI}{dt} &= \beta SI - \gamma I \\
\frac{dR}{dt} &= \gamma I
\end{align*}\]

where \(t\) is time, \(S(t)\) is the number of susceptible people, \(I(t)\) is the number of people infected, \(R(t)\) is the number of people who have recovered and developed immunity to the infection, \(\beta\) is the infection rate, and \(\gamma\) is the recovery rate.

val recover_rate = 0.01
val infect_rate = 0.0001
new infect @infect_rate:chan()

let Recovered() =
  ?infect; Recovered()
and Susceptible() =
  ?infect; Infected()
and Infected() =
  do !infect; Infected() or ?infect; Infected() or delay@recover_rate; Recovered()
run (500 of Susceptible() | 1 of Infected())
Exercise: Epidemic Simulations

Knowing that
\[ \beta = \text{infect\_rate} \]
\[ \gamma = \text{recover\_rate} \]
try various values to see how the infection progresses.

In the previous example, \( R_0 = 5 \) (everybody gets infected).

You can get \( R_0 = 1 \) (infection dies out) by reducing the S population to 100.

But stochastic effects (initial infected population = 1!) play a major role between \( R_0 = 1 \) and \( R_0 = 5 \).

The key value governing the time evolution of these equations is the so-called epidemiological threshold,

\[ R_0 = \frac{\beta S}{\gamma} \]  \hspace{1cm} (4)

Note that the choice of the notation \( R_0 \) is a bit unfortunate, since it has nothing to do with \( R \). \( R_0 \) is defined as the number of secondary infections caused by a single primary infection; in other words, it determines the number of people infected by contact with a single infected person before his death or recovery.

When \( R_0 < 1 \), each person who contracts the disease will infect fewer than one person before dying or recovering, so the outbreak will peter out (\( dI/dt < 0 \)). When \( R_0 > 1 \), each person who gets the disease will infect more than one person, so the epidemic will spread (\( dI/dt > 0 \)). \( R_0 \) is probably the single most important
MHC Class I Flytrap
MHC Class I Antigen Presentation

- part of the cellular immune response
- MHC class I complexes present self and foreign peptide at the cell surface
- recognized by T lymphocytes and natural killer cells
- also required for development of self tolerant T cells in thymus

MHC Class I Peptide Binding
We want to model the situation where the trap is shutting at a constant rate, but different kinds of flies are escaping at different "dissociation" rates. Hence we cannot model this simply as a channel of given rate where trap and flies synchronize.

We need to model a race between two delays in two independent processes. But in the end, both the trap and the fly must agree on whether the fly was captured or not. (With no deadlock.)
Different Flytraps?

The two definitions are alpha-convertible!
Fly $\equiv$ Trap
escaping $\leftrightarrow$ closing
escaped $\leftrightarrow$ captured
Who is the fly and who is the trap?

It’s a race, first, between `@closing` and `@escaping`.

**Finite Rates**
- If `@escaping` wins the race, the fly has escaped.
- If `@closing` wins the race, there is a second race between `@escaping` and `?!captured`.

**Infinite Rates**
- If `@escaping` wins the race, the fly has escaped.
- If `@closing` wins the race, the fly is captured.
  (Race between finite `@closing` and infinite `?!escaped`)

Equivalent flytraps if transitions have infinite rate?
Flytrap Product Automata

Fly|Trap

@escaping ➔ @closing ➔ ?!escaped ➔

@escaping ➔ ?!escaped ➔

@closing ➔

?!captured ➔

Tralp|Fly

@closing ➔ @escaping ➔ ?!captured ➔

@escaping ➔ ?!escaped ➔

@closing ➔

?!captured ➔

@escaping ➔ ?!escaped ➔

?!captured ➔
Exercise (Open)

- Prove or disprove that the two flytraps are equivalent
  - Not necessarily for all intermediate states or quantities but, e.g.,
    - Do Escaped-Flies have the same distribution in both version?
  - What about the infinite-rate version?
A synthetic oscillatory network of transcriptional regulators

Michael B. Elowitz & Stanislas Leibler

Departments of Molecular Biology and Physics, Princeton University, Princeton, New Jersey 08544, USA

Networks of interacting biomolecules carry out many essential functions in living cells, but the ‘design principles’ underlying the functioning of such intracellular networks remain poorly understood, despite intensive efforts including quantitative analysis of relatively simple systems. Here we present a complementary approach to this problem: the design and construction of a synthetic network to implement a particular function. We used three transcriptional repressor systems that are not part of any natural biological clock to build an oscillating network, termed
Gene Gates and Circuits

A gene gate

\[
\text{neg}(a,b) \triangleq \neg a \lor \tau_{\eta} \lor \neg(a,b) + \tau_{\epsilon} \lor (\text{tr}(b) \lor \neg(a,b))
\]

\[
\text{tr}(p) \triangleq (\neg p \lor \text{tr}(p)) + \tau_{\delta}
\]

A genetic circuit (engineered in E.Coli)

\[
\text{neg}(a,b) \lor \neg(b,c) \lor \neg(c,a)
\]

The stochastic-\(\pi\) program

\[
\text{val } dk = 0.001 \quad (* \text{Decay rate }*)
\]

\[
\text{val } inh = 0.001 \quad (* \text{Inhibition rate }*)
\]

\[
\text{val } cst = 0.1 \quad (* \text{Constitutive rate }*)
\]

\[
\text{let } \text{tr}(p: \text{chan}()) =
\]

\[
\text{do } !p; \text{tr}(p) \text{ or delay@dk}
\]

\[
\text{let } \text{neg}(a: \text{chan}(), b: \text{chan}()) =
\]

\[
\text{do } ?a; \text{delay@inh; neg}(a,b) \text{ or delay@cst; (tr}(b) \lor \neg(a,b))
\]

(* The circuit *)

\[
\text{val } bnd = 1.0 \quad (* \text{Protein binding rate }*)
\]

\[
\text{new } a@bnd: \text{chan}() \text{ new } b@bnd: \text{chan}() \text{ new } c@bnd: \text{chan}()
\]

\[
\text{run } (\neg(c,a) \lor \neg(a,b) \lor \neg(b,c))
\]

A stochastic simulation (in SPiM)

\[
r=1.0, \epsilon=0.1, h=0.001, \delta=0.001
\]
Scaling Reactions
Scaling Moles and Rates

• Say the original reaction specifies a quantity of 1mol for a given species A, with reaction rates expressed in mol/Ls (rate of change in concentration mol/L of reactants). (Or maybe it specifies a concentration of 1µM = one millionth of a mole per liter, with rates expressed in µM/s; it does not matter here.) How many processes should we use for the simulation? Well, we can’t use quantity 1; that’s too few!

• So, let’s multiply all mol quantities by say, 10, and use that many processes. What effect does that have on the reactions? Unary (decay) reactions now operate on an initial quantity 10*bigger. However those are exponential decays, which means that the half-life (and the general shape) of the reaction is independent of the initial quantity. So the time it takes for such reactions to operate does not change, and we do not have to scale the time axis (although our vertical axis is now off by a factor of 10).

• Binary reactions, however, now operate on 10*bigger quantities and by the mass action law run 100 times faster (-k(10*[A] 10*[B])), which means they are 10 times too fast with respect to the degradations. We can compensate by dividing the rates of those reactions by 10, the scaling factor.

• That way, in the end, our plots have the same curves as for any other scaling factor, and an accurate timeline, but the vertical axis numbers must be divided by 10 to compare with the original 1mol quantity.
Scaling Quantities and Rates

For example:

A1 + A2 \rightarrow_{c} 0
A2 \rightarrow_{\text{deg}} 0

Molecules halved
1000 molecules each
c=1.0 deg=1000.0
y^2 (plot rescaled)

Binary rate doubled
1000 molecules
c=2.0 deg=1000.0
y^2

Both rates doubled
1000 molecules
c@2.0 deg=2000.0
y^2

Degradation rate doubled
1000 molecules each
c=1.0 deg=1000.0
y^2

Degradation rate halved
1000 molecules each
c=1.0 deg=500.0
y^2, x/2

To get the same curves (up to rescaling of the y axis) we need to scale up the rate of binary reactions (only) by the same factor.

Original reaction
2000 molecules each
c=1.0 deg=1000.0
does not match
does not match
does not match

Scalidown the degradation rates by the same factor works too, but then we have to rescale the x axis as well.
ERK Pathway

Mathematical modeling of the influence of RKIP on the ERK signaling pathway

Kwang-Hyun Cho¹, Sung-Young Shin¹, Hyun-Woo Kim¹, Olaf Wolkenhauer²*, Brian McFerran³,⁴ and Walter Kolch³,⁵
Fig. 1. Graphical representation of the ERK signaling pathway regulated by RKIP: a circle \(\bigcirc\) represents a state for the concentration of a protein and a bar \(\mid\) a kinetic parameter of reaction to be estimated. The directed arc (arrows) connecting a circle and a bar represents a direction of a signal flow. The bi-directional thick arrows represent an association and a dissociation rate at same time. The thin unidirectional arrows represent a production rate of products.

Mathematical modeling of the influence of RKIP on the ERK signaling pathway

Kwang-Hyun Cho\textsuperscript{1\*}, Sung-Young Shin\textsuperscript{1}, Hyun-Woo Kim\textsuperscript{1}, Olaf Wolkenhauer\textsuperscript{2,3}, Brian McFerran\textsuperscript{4} and Walter Kolch\textsuperscript{3,5}
let Raf1s() =
  !k1; Raf1s_RKIP() (* ![01] *)
and RKIP() =
  ?k1; ()                         (* ?[01] *)
and Raf1s_RKIP() =
  do delay@k2; (Raf1s() | RKIP()) (* [02] *)
or !k3; Raf1s_RKIP_ERKPP() (* ![03] *)
and ERKPP() =
  ?k3; () (* ?[03] *)
and Raf1s_RKIP_ERKPP() =
  do delay@k4; (Raf1s_RKIP() | ERKPP()) (* [04] *)
or delay@k5; (Raf1s() | RKIPP() | ERK()) (* [05] *)
or ?b1
and MEKPP() =
  !k6; MEKPP_ERK() (* ![06] *)
and ERK() =
  ?k6; () (* ?[06] *)
and MEKPP_ERK() =
  do delay@k7; (MEKPP() | ERK()) (* [07] *)
or delay@k8; (MEKPP() | ERKPP()) (* [08] *)
or ?b2
and RKIPP() =
  !k9; RKIPP_RP() (* ![09] *)
and RP() =
  ?k9; () (* ?[09] *)
and RKIPP_RP() =
  do delay@k10; (RKIPP() | RP()) (* [10] *)
or delay@k11; (RKIPP() | RP()) (* [11] *)
or ?b3
let many(n:float, p:proc()) = if n<=0.0 then () else p(); many(n-1.0, p)
run many(2.5*quantity*concentration, Raf1s)
run many(2.5*quantity*concentration, RKIP)
run many(2.5*quantity*concentration, ERKPP)
run many(2.5*quantity*concentration, MEKPP)
run many(3.0*quantity*concentration, RP)
SPiM Simulation
Mathematical modeling of the influence of RKIP on the ERK signaling pathway

Kwang-Hyeon Che1, Sung-Young Shin1, Hyun-Woo Koo1, Olaf Wieland2,3, Brigitte M. Ferrier1 and Werner Goldblatt1,3

Fig. 5. Simulation results of the mathematical modeling for fixed initial condition: the upper left shows the dynamics for Raf-1*, RKIP, and their complex Raf-1*/RKIP, the upper right shows the activity of MEK-PP which phosphorylates and activates ERK, the lower left shows the activity of ERK-PP, and the lower right shows the activity of RP.
Epidemics ODE


Epidemics

directive sample 500.0 1000
directive plot Recovered(); Susceptible(); Infected()

new infect @0.001:chan()
val recover = 0.03

let Recovered() =
  ?infect; Recovered()
and Susceptible() =
  ?infect; Infected()
and Infected() =
  do !infect: Infected()
  or ?infect; Infected()
  or delay@recover; Recovered()
run (200 of Susceptible() | 2 of Infected())
\[
\begin{align*}
S &= ?i_{(t)}; I \\
I &= !i_{(t)}; I \oplus ?i_{(t)}; I \oplus \tau_r; R \\
R &= ?i_{(t)}; R \\
\end{align*}
\]

\[
S + I \rightarrow^t I + I \\
I + I \rightarrow^t I + I \\
I \rightarrow^r R \\
R + I \rightarrow^t R + I \\
\]

“useless” reactions

\[
[S]^* = -t[S][I] \\
[I]^* = t[S][I] - r[I] \\
[R]^* = r[I] \\
\]

Automata match the standard ODE model!

\[
\begin{align*}
\frac{dS}{dt} &= -a IS \\
\frac{dI}{dt} &= a IS - bI \\
\frac{dR}{dt} &= bI \\
\end{align*}
\]

(the Kermack-McKendrick, or SIR model)
**Simplified Model**

- $S = \text{?}i(t) ; I$
- $I = \text{!}i(t) ; I \oplus \tau_r ; R$
- $R = 0$

$S \rightarrow^r I$  
$I \rightarrow^r R$

$[S]^* = -t[S][I]$  
$[I]^* = t[S][I] - r[I]$  
$[R]^* = r[I]$  

Same ODE, hence equivalent automata models.

Not totally obvious that one could have simplified the automata model.

---

directive sample 500.0 1000
directive plot Recovered(); Susceptible(); Infected()

new infect @0.001:chan()
val recover = 0.03
let Recovered() = ()
and Susceptible() = ?infect; Infected()
and Infected() = do !infect; Infected() or delay@recover; Recovered()
run (200 of Susceptible() | 2 of Infected())
Groupies ODEs
**Groupies ODE**

\[ A = !a(r); A \oplus ?b(r); B \]
\[ B = !b(r); B \oplus ?a(r); A \]

\[ A + B \rightarrow^r A + A \]
\[ B + A \rightarrow^r B + B \]

\[ [A]^* = r[A][B] - r[B][A] \]
\[ [B]^* = r[B][A] - r[A][B] \]

\[ [A]^* = 0 \]
\[ [B]^* = 0 \]

Wrong Answer?

ODE predicts stability \([A]^* = 0\) for any value of \([A]\), while the stochastic system is stable only when \([A]\) is either 0 or Max.
Doped Groupies ODE

\[ \begin{align*}
A &= \text{!}a(r) \cdot A \oplus \text{?}b(r) \cdot B \\
B &= \text{!}b(r) \cdot B \oplus \text{?}a(r) \cdot A \\
A_d &= \text{!}a(r) \cdot A_d \\
B_d &= \text{!}b(r) \cdot B_d
\end{align*} \]

\[ \begin{align*}
A + B \rightarrow^r A + A \\
B + A \rightarrow^r B + B \\
A + B_d \rightarrow^r B + B_d \\
B + A_d \rightarrow^r A + A_d
\end{align*} \]

\[ \begin{align*}
\end{align*} \]

\[ \begin{align*}
[A]^* &= -rk([A] - [B]) \\
[B]^* &= rk([A] - [B])
\end{align*} \]

At \([B]=0\):
\[\begin{align*}
[A]^* &= -rk[A], \\
[B]^* &= rk[A]
\end{align*} \]

At \([A] \approx [B] \approx 0\):
\[\begin{align*}
[A]^* &= -rk[A] \\
[B]^* &= rk[A]
\end{align*} \]

At \([A]=[B]:\)
\[\begin{align*}
[A]^* &= 0 \\
[B]^* &= 0
\end{align*} \]

\([A_d],[B_d]\) are constant; assume them both = \(k\)

**Wrong Answer?**

ODE predicts converging stable equilibrium at \([A]=[B]\) instead of the total chaos observed in the stochastic system!

For \(k=0\) (no dope), predicts deadlock \([A]^*=[B]^*=0\) but at any value of \([A]\), which is definitely not true in the stochastic system.
### Hysteric Groupies ODE

**A = !a_{(r)}; A ⊕ ?b; A' A' = ?b; B B = !b_{(r)}; B ⊕ ?a; B' B' = ?a; A**

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A+B</td>
<td>A+B'</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B+A</td>
<td>B+A'</td>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>A+B</td>
<td>A+B'</td>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B+A</td>
<td>B+A'</td>
<td>B</td>
<td>B</td>
</tr>
</tbody>
</table>

**A = !a_{(r)}; A_d A_d = !a_{(r)}; A_d**

\[ \begin{align*}
\end{align*} \]

\[ \begin{align*}
[A_d] & = 0 \\
[B_d] & = 0
\end{align*} \]

\[ (A_d), (B_d) \text{ are constant; assume them both } k \]

**Doping**

**A_d**

**B_d**

**Wrong Answer?**

**ODE predicts dampened oscillation, while the stochasic system keeps oscillating at max level.**
Modeling the cell division cycle: cdc2 and cyclin interactions

(maturation promoting factor/metaphase arrest/wee1/cdc25)

JOHN J. TYSON

Department of Biology, Virginia Polytechnic Institute and State University, Blacksburg, VA 24061

Communicated by David M. Prescott, May 20, 1991 (received for review January 23, 1991)

ABSTRACT The proteins cdc2 and cyclin form a heterodimer (maturation promoting factor) that controls the major events of the cell cycle. A mathematical model for the interactions of cdc2 and cyclin is constructed. Simulation and analysis of the model show that the control system can operate in three modes: as a steady state with high maturation promoting factor activity, as a spontaneous oscillator, or as an excitable switch. We associate the steady state with metaphase arrest in unfertilized eggs, the spontaneous oscillations with rapid division cycles in early embryos, and the excitable switch with growth-controlled division cycles typical of nonembryonic cells.

Passage through the cell cycle is marked by a temporally organized sequence of events including DNA replication, mitosis, and the appearance of certain cell-cycle specific proteins and cyclin activity (2). We model proliferation of...
Fig. 1. The relationship between cyclin and cdc2 in the cell cycle. In step 1, cyclin is synthesized de novo. Newly synthesized cyclin may be unstable (step 2). Cyclin combines with cdc2-P (step 3) to form "preMPF." At some point after heterodimer formation, the cyclin subunit is phosphorylated. (Assuming phosphorylation is faster than dimerization, I write the two-step process as a single step, rate-limited by dimerization.) The cdc2 subunit is then dephosphorylated (step 4) to form "active MPF." In principle, the activation of MPF may be opposed by a protein kinase (step 5). Assuming that active MPF enhances the catalytic activity of the phosphatase (as indicated by the dashed arrow), I arrange that MPF activation is switched on in an autocatalytic fashion. Nuclear division is triggered when a sufficient quantity of MPF has been activated, but concurrently active MPF is destroyed by step 6. Breakdown of the MPF complex releases phosphorylated cyclin, which is subject to rapid proteolysis (step 7). Finally, the cdc2 subunit is phosphorylated (step 8, possibly reversed by step 9), and the cycle repeats itself. aa, amino acids; ~P, ATP; P, inorganic phosphate.

Table 1. Kinetic equations governing the cyclin–cdc2 cycle in Fig. 1

<table>
<thead>
<tr>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{d\text{C2}}{dt} = k_d[M] - k_d[\sim P][\text{C2}] + k_d[\text{CP}] )</td>
<td>Cyclin synthesis, cyclin dephosphorylation, cyclin activation by MPF</td>
</tr>
<tr>
<td>( \frac{d\text{CP}}{dt} = -k_d[\text{CP}][Y] + k_d[\sim P][\text{C2}] - k_{d}[\text{CP}] )</td>
<td>Cyclin activation by MPF, cyclin dephosphorylation</td>
</tr>
<tr>
<td>( \frac{d\text{pM}}{dt} = k_p[\text{CP}][Y] - [\text{pM}] + k_d[\sim P][\text{M}] )</td>
<td>Protein synthesis, protein dephosphorylation</td>
</tr>
<tr>
<td>( \frac{d[M]}{dt} = [\text{pM}][\text{P}][\text{M}] - k_d[\sim P][\text{M}] - k_d[M] )</td>
<td>Protein activation by MPF</td>
</tr>
<tr>
<td>( \frac{d[Y]}{dt} = k_d[\text{aa}] - k_d[Y] - k_d[\text{CP}][Y] )</td>
<td>Protein degradation</td>
</tr>
<tr>
<td>( \frac{d[\text{YP}]}{dt} = k_d[M] - k_d[\text{YP}] )</td>
<td>Cyclin degradation</td>
</tr>
</tbody>
</table>

\( t \), time; \( k_d \), rate constant for step \( i \) (\( i = 1, \ldots, 9 \)); aa, amino acids. The concentrations [aa] and [P] are assumed to be constant. There are six time-dependent variables: the concentrations of cyclin ([C2]), cdc2-P ([CP]), preMPF = P-cyclin-cdc2-P ([pM]), active MPF = P-cyclin-cdc2 ([M]), cyclin (YP), and cyclin-P ([YP]). The activation of step 4 by active MPF is described by the function \( F([\text{M}]) = k_f + k_d[M]/[\text{CT}]^2 \), where \( k_f \) is the rate constant for step 4 when [active MPF] = 0 and \( k_d \) is the rate constant when [active MPF] = [CT], where [CT] = total cdc2. I assume \( k_d >> k_f \). This form of \( F([\text{M}]) \) is only one of many possible ways to describe the autocatalytic feedback of active MPF on its own production.

Table 2. Parameter values used in the numerical solution of the model equations

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_d[\text{aa}]/[\text{CT}] )</td>
<td>0.015 min⁻¹</td>
<td>*</td>
</tr>
<tr>
<td>( k_3 )</td>
<td>0</td>
<td>†</td>
</tr>
<tr>
<td>( k_3[\text{CT}] )</td>
<td>200 min⁻¹</td>
<td>*</td>
</tr>
<tr>
<td>( k_4 )</td>
<td>10–1000 min⁻¹</td>
<td>(adjustable)</td>
</tr>
<tr>
<td>( k_5 )</td>
<td>0.018 min⁻¹</td>
<td>†</td>
</tr>
<tr>
<td>( k_5[\sim P] )</td>
<td>0</td>
<td>†</td>
</tr>
<tr>
<td>( k_6 )</td>
<td>0.1–10 min⁻¹</td>
<td>(adjustable)</td>
</tr>
<tr>
<td>( k_7 )</td>
<td>0.6 min⁻¹</td>
<td>†</td>
</tr>
<tr>
<td>( k_8[\sim P] )</td>
<td>&gt;&gt; k_8</td>
<td>†</td>
</tr>
<tr>
<td>( k_9 )</td>
<td>&gt;&gt; k_d</td>
<td>†</td>
</tr>
</tbody>
</table>

*It is assumed that [CT] = [C2] + [CP] + [pM] + [M] = constant. For growing cells, this implies that cdc2 protein is continuously synthesized to maintain a constant concentration of cdc2 subunits (31).†In the absence of evidence to the contrary, it is assumed that newly synthesized cyclin is stable (\( k_5 = 0 \)). If \( k_5 \neq 0 \), the behavior of the model is basically unchanged, as long as \( k_5 << k_5[\text{CT}] \). In accord with experimental evidence, I assume that cyclin-P subunits released from MPF complexes are quickly degraded (half-life = 1 min)."In all calculations reported here, I ignore rephosphorylation of the cdc2 subunit of active MPF (step 5). Similar results can be obtained with \( k_8 \neq 0 \).§I assume that cdc2 protein is phosphorylated as soon as it dissociates from the active MPF complex—i.e., \( k_5[\sim P] >> k_d >> [CP] \). This allows us to neglect the first differential equation in Table 1 (i.e., \( d[C2]/dt = 0 \)) and \( [C2] = (k_d/k_5[\sim P])[CP] << [CP] \).
The Tyson Cell Cycle in SPiM

directive sample 10.0 1000
directive plot Cyclin(): Cdc2P1(): Cdc2();
   Cdc2P1_CyclinP1(): Cdc2_CyclinP1(): CyclinP1()

val factor = 200.0 (* Scaling Factor *)
val k1 = 5.0 (* 0.015 cyclin production cranked up *)
val k2 = 0.0
val k3 = 200.0/factor
val k4 = 0.018
val k4p = 0.018
val k5 = 0.0
val k6 = 0.0
val k7 = 0.6
val k8 = 1000.0
val k9 = 10.0

(* THE REACTIONS
k1  0 -> Cyclin
k2  Cyclin -> 0
k3  Cyclin + Cdc2P1 -> Cdc2P1_CyclinP1

k4p Cdc2P1_CyclinP1 -> Cdc2_CyclinP1
k4  Cdc2P1_CyclinP1 + Cdc2_CyclinP1 -> 2* Cdc2_CyclinP1
k5  Cdc2_CyclinP1 -> Cdc2P1_CyclinP1

k6  Cdc2_CyclinP1 -> CyclinP1 + Cdc2
k7  CyclinP1 -> 0

k8  Cdc2 -> Cdc2P1
k9  Cdc2P1 -> Cdc2
*)

new c3@k3:chan
new c4@k4:chan
let genCyclin() = delay@k1; (Cyclin() | genCyclin())
and Cyclin() =
do delay@k2; ()
or ?c3; Cdc2P1_CyclinP1()

and Cdc2P1() =
do !c3; ()
or delay@k9; Cdc2()

and Cdc2() =
delay@k8; Cdc2P1()

and Cdc2P1_CyclinP1() =
do delay@k4p; Cdc2_CyclinP1()
or ?c4; Cdc2_CyclinP1()

and Cdc2_CyclinP1() =
do !c4; Cdc2_CyclinP1()
or delay@k5; Cdc2P1_CyclinP1()
or delay@k6; (CyclinP1() | Cdc2())

and CyclinP1() =
delay@k7; ()
run genCyclin()
run 200 of Cdc2P1()
SPiM Simulation

Very high relative level of Cdc2P1, in fast reaction (bad for simulation)

\[ k_8 \text{ Cdc2} \rightarrow \text{Cdc2P1} \]
\[ k_9 \text{ Cdc2P1} \rightarrow \text{Cdc2} \]
val \( k_8 = 1000.0 \)
val \( k_9 = 10.0 \)

Cell Divisions

Simulation: Time = 60.520033 (6052 points at 1.4183 simTime/sysTime and running)
The Tyson Cell Cycle in BIOCHAM

%Description
%A model of the cell cycle based on the interactions between cdc2 and cyclin.

%present(Cdc2,0.39).
%present(Cdc2~{p1},0.0001).
%present(Cyclin,0.0001).
%present(Cdc2~{p1}-Cyclin~{p1},0.0001).
%present(Cdc2-Cyclin~{p1},0.0001).
%present(Cyclin~{p1},0.0001).

present(Cdc2,1).
absent(Cdc2~{p1}).
absent(Cyclin).
absent(Cdc2~{p1}-Cyclin~{p1}).
absent(Cdc2-Cyclin~{p1}).
absent(Cyclin~{p1}).

k1 for _=>Cyclin.
k2*[Cyclin] for Cyclin=>_.

k3*[Cyclin]*[Cdc2~{p1}] for Cyclin+Cdc2~{p1} => Cdc2~{p1}-Cyclin~{p1}.

k4*[Cdc2~{p1}-Cyclin~{p1}] for Cdc2~{p1}-Cyclin~{p1} => Cdc2-Cyclin~{p1}.

k5*[Cdc2-Cyclin~{p1}] for Cdc2-Cyclin~{p1} => Cdc2~{p1}-Cyclin~{p1}.

k6*[Cdc2-Cyclin~{p1}] for Cdc2-Cyclin~{p1} => Cyclin~{p1}+Cdc2.

k7*[Cyclin~{p1}] for Cyclin~{p1} =>_.
k8*[Cdc2] for Cdc2 => Cdc2~{p1}.
k9*[Cdc2~{p1}] for Cdc2~{p1} => Cdc2.

%Cdc2-Cyclin~{p1}=>Cdc2~{p1}.

macro(YT,[Cyclin]+[Cdc2~{p1}]-[Cdc2~{p1}-Cyclin~{p1}]+[Cdc2-Cyclin~{p1}]).
macro(CT,[Cdc2]+[Cdc2~{p1}]-[Cdc2~{p1}-Cyclin~{p1}]+[Cdc2-Cyclin~{p1}]).
macro(ratio,YT/CT).

parameter(k1,0.015).
parameter(k2,0.015).
parameter(k3,200).
parameter(k4,0.018).
parameter(k4,180).
parameter(k5,5).
parameter(k6,1).
parameter(k7,0.6).
parameter(k8,100).
parameter(k9,100).
# The Tyson Cell Cycle in Cellerator

## Cell Cycle Model; Tyson (1991, 6 variables)

**Citation**

**Description**
A model of the cell cycle based on the interactions between cdc2 and cyclin. The model has six dynamic variables: C2 (cdc2), CP (cdc2-P complex), pM (P- cyclin-cdc2-P complex), M (active MPP, P- cyclin- cdc2 complex), Y (cyclin), and YP (cyclin-P). Total cyclin concentration (YT) is the sum Y+YP+pM+M4

<table>
<thead>
<tr>
<th>Rate constant</th>
<th>Reaction</th>
</tr>
</thead>
<tbody>
<tr>
<td>k1aa = 0.015</td>
<td>EmptySet -&gt; Y</td>
</tr>
<tr>
<td>k2 = 0</td>
<td>Y -&gt; EmptySet</td>
</tr>
<tr>
<td>k3 = 300</td>
<td>CP + Y -&gt; pM</td>
</tr>
<tr>
<td>k4prime + k4*M[t]^2 pM -&gt; M</td>
<td></td>
</tr>
<tr>
<td>k5notP = 0</td>
<td>M -&gt; pM</td>
</tr>
<tr>
<td>k6 = 1</td>
<td>M -&gt; C2 + YP</td>
</tr>
<tr>
<td>k7 = 0.6</td>
<td>YP -&gt; EmptySet</td>
</tr>
<tr>
<td>k8notP = 10000000</td>
<td>C2 -&gt; CP</td>
</tr>
<tr>
<td>k9 = 100000</td>
<td>CP -&gt; C2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variable IC ODE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C2</td>
</tr>
<tr>
<td>CP</td>
</tr>
<tr>
<td>M[t]</td>
</tr>
<tr>
<td>pM</td>
</tr>
<tr>
<td>Y[t]</td>
</tr>
<tr>
<td>YP[t]</td>
</tr>
</tbody>
</table>

Generated by Cellerator Version 1.0 update 2.1125 using Mathematica 4.2 for Mac OS X (June 4, 2002), November 25, 2002 14:40:26
Ultrasensitivity in the mitogen-activated protein kinase cascade

Chi-Ying F. Huang and James E. Ferrell, Jr.

Department of Molecular Pharmacology, Stanford University School of Medicine, Stanford, CA 94305-5332

Communicated by Daniel E. Koshland, Jr., University of California, Berkeley, CA, May 16, 1996 (received for review January 22, 1996)

ABSTRACT The mitogen-activated protein kinase (MAPK) cascade is a highly conserved series of three protein kinases implicated in diverse biological processes. Here we demonstrate that the cascade arrangement has unexpected consequences for the dynamics of MAPK signaling. We solved the rate equations for the cascade numerically and found that MAPK is predicted to behave like a highly cooperative enzyme, even though it was not assumed that any of the enzymes in the cascade were regulated cooperatively. Measurements of MAPK activation in Xenopus oocyte extracts confirmed this prediction. The stimulus/response curve of the MAPK was found to be as steep as that of a cooperative enzyme with a Hill coefficient of 4–5, well in excess of that of the classical allosteric protein hemoglobin. The shape of the MAPK stimulus/response curve may make the cascade particularly appropriate for mediating processes like mitogenesis, cell fate induction, and oocyte maturation, where a cell switches from one discrete state to another.

Fig. 1. Schematic view of the MAPK cascade. Activation of MAPK depends upon the phosphorylation of two conserved sites.

Biochemistry: Huang and Ferrell

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

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Range of assumed $K_m$ values</th>
<th>Range of effective Hill coefficients (nM) predicted for</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAPKKK → MAPKK*</td>
<td>60–1500 nM</td>
<td>1.0 1.7 4.9</td>
</tr>
<tr>
<td>MAPKK → MAPKK-P</td>
<td>60–1500 nM</td>
<td>1.0 1.3–2.3 4.0–5.1</td>
</tr>
<tr>
<td>MAPKK-P → MAPKK</td>
<td>60–1500 nM</td>
<td>1.0 1.5–1.9 3.6–6.7</td>
</tr>
<tr>
<td>MAPKK-P → MAPKK-PP</td>
<td>60–1500 nM</td>
<td>1.0 1.2–2.4 3.8–5.2</td>
</tr>
<tr>
<td>MAPKK-P → MAPKK-P</td>
<td>60–1500 nM (300 nM)</td>
<td>1.0 1.7 4.1–6.4</td>
</tr>
<tr>
<td>MAPKK → MAPK</td>
<td>60–1500 nM</td>
<td>1.0 1.7 4.3–5.2</td>
</tr>
<tr>
<td>MAPKK-P → MAPK-P</td>
<td>60–1500 nM</td>
<td>1.0 1.7 4.7–5.1</td>
</tr>
<tr>
<td>MAPKK-P → MAPK</td>
<td>60–1500 nM</td>
<td>1.0 1.7 4.7–5.1</td>
</tr>
</tbody>
</table>

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

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

Calculations. Eqs. 1–10 represent the reactions of the MAPK cascade, which are shown schematically in Fig. 1. We have used Goldbeter and Koshland’s nomenclature for the rate constants—the latter a denotes association, d denotes dissociation without catalysis, and t denotes product formation (11). KKK denotes MAPKK, KK denotes MAPKK and K denotes MAPK.

\[ \begin{align*}
K_{KK} + E_1 & \rightleftharpoons K_{KK}E_1 \rightleftharpoons K_{KK} + E_1 & [1] \\
K_{KK} + E_2 & \rightleftharpoons K_{KK}E_2 \rightleftharpoons K_{KK} + E_2 & [2] \\
K + K_{KK}^* & \rightleftharpoons K_{KK} + K & [3] \\
K_{KK} + K_{KK}^* & \rightleftharpoons K_{KK} + K_{KK}^* & [4] \\
K_{KK} + K_{KK}^* & \rightleftharpoons K_{KK} + K_{KK}^* & [5] \\
\end{align*} \]

\[ \begin{align*}
K_{P} + K_{P}P & \rightleftharpoons K_{P}P + K_{P}P & [6] \\
K_{P} + K_{P}P & \rightleftharpoons K_{P} + K_{P}P & [7] \\
K + K_{P} & \rightleftharpoons K + K_{P} & [8] \\
K_{P} + K_{P}P & \rightleftharpoons K_{P} + K_{P}P & [9] \\
K_{P} + K_{P}P & \rightleftharpoons K_{P} + K_{P}P & [10] \\
\end{align*} \]

Fig. 1. Schematic view of the MAPK cascade. Activation of MAPK depends upon the phosphorylation of two conserved sites [Thr-183 and Tyr-185 in rat p42 MAPK/Erk2 (4, 5)]. Full activation of MAPK also requires phosphorylation of two sites [Ser-218 and Ser-222 in mouse Mek-1/MKK1 (6–10)]. Detailed mechanisms for the activation of various MAPKKs (e.g., Raf-1, B-Raf, Mos) are not yet established; here we assume that MAPKKs are activated and inactivated by enzymes we denote E1 and E2. MAPKK* denotes activated MAPKK, MAPKK-P and MAPKK-PP denote singly and doubly phosphorylated MAPKK, respectively. MAPK-P and MAPK-PP denote singly and doubly phosphorylated MAPK. Pase denotes phosphatase.
As 18 Ordinary Differential Equations

Plus 7 conservation equations

\[
\frac{d}{dt} [KKK] = -a_{KK}[KK][E1] + \delta_{KK}[KK-E1] \\
+ k_{KK}[KK^* - E2] \tag{11}
\]

\[
\frac{d}{dt} [KKK-E1] = a_{KK}[KK][E1] - (d_t + k_3)[KKK-E1] \tag{12}
\]

\[
\frac{d}{dt} [KK]^* - E1 = -a_{KK}[KK][E1] + \delta_{KK}[KK^* - E2] \\
+ k_{KK}[KK^* - E1] + (d_t + k_3)[KK^* - KK-E1] - a_{KK}[KK]^* \tag{13}
\]

\[
\frac{d}{dt} [KK-E2] = a_{KK}[KK][E2] - (d_t + k_3)[KK-E2] \tag{14}
\]

\[
\frac{d}{dt} [KK] = -a_{KK}[KK]^* + \delta_{KK} [KK-KK^*] \\
+ k_{KK}[KK-P-KK^*] \tag{15}
\]

\[
\frac{d}{dt} [KK-P-KK^*] = a_{KK}[KK][KK]^* \\
- (d_t + k_3)[KK-KK^*] \tag{16}
\]

\[
\frac{d}{dt} [KK-P] = -a_{KK}[KK-P][KK^*] + \delta_{KK} [KK-P-KK^*] \\
+ k_{KK}[KK-KK^*] + \delta_{KK}[KK-PP-KK^*] \tag{17}
\]

\[
\frac{d}{dt} [KK-PP-KK^*] = a_{KK}[KK-P][KK^*] - (d_t + k_3)[KK-P-KK^*] \tag{18}
\]

\[
\frac{d}{dt} [KK-PP-KP] = a_{KK}[KK-P][KK^*] \\
- (d_t + k_3)[KK-P-KP] \tag{19}
\]

\[
\frac{d}{dt} [KK-PP-KK] = k_{KK}[KK-P-KK] + \delta_{KK}[KK-PP-KK-KK^*] \\
+ d_{KK}[KK-PP-KK-KP] - a_{KK}[KK-PP][KK^*] \tag{20}
\]

\[
\frac{d}{dt} [KK-PP-KP] = \delta_{KK}[KK-PP] + k_{KK}[KK-PP-KK] - a_{KK}[KK-PP] \tag{21}
\]

\[
\frac{d}{dt} [KK-PP-KP-ase] = a_{KK}[KK-PP][KK^*] \\
- (d_t + k_3)[KK-PP-KP] \tag{22}
\]

\[
\frac{d}{dt} [K] = -a_{KK}[KK-P] + \delta_{KK}[KK-PP] + k_{KK}[KK-KP-KP-ase] \tag{23}
\]

\[
\frac{d}{dt} [K-P] = k_{KK}[KK-PP] - a_{KK}[K-P][K-P^*] + \delta_{KK}[K-P-KP] - a_{KK}[K-P][KK] + d_{KK}[K-P-KP-KP-ase] - a_{KK}[K-P][K-P^*] \tag{24}
\]

\[
\frac{d}{dt} [K-P-KP-PASE] = a_{KK}[K-P][K-P^*] + (d_t + k_3)[K-P-KP-ase] \tag{25}
\]

\[
\frac{d}{dt} [K-P-KP-KP-KP-ase] = a_{KK}[K-P][K-P^*] + (d_t + k_3)[K-P-KP-ase] \tag{26}
\]

\[
\frac{d}{dt} [K-P-KP-KP-ase] = -a_{KK}[K-P][K-P^*] \tag{27}
\]

\[
\frac{d}{dt} [K-P-ase] = a_{KK}[K-P][K-P^*] \tag{28}
\]

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

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

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

\[
[KKK_{tot}] = [KKK] + [KKK^*] + [KKK-E1] + [KKK^* - E2] + [KKK^* - K] + [KKK^* - K-P] \tag{29}
\]

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

Reaction View

\[ S \xrightarrow{c,d,e} E \xrightarrow{c} E+S \xrightarrow{e} P+E \]

Intermediate complex

Interaction View

Private bindings between one S and one E molecule

\[ S() \triangleq \text{new } u@d \text{ new } k@e \]
\[ a_c(u,k); (\!u_d; S() + !k_e; P()) \]

Bind unbind react

\[ E() \triangleq \ ?a_c(u,k); (?u_d; E() + ?k_e; E()) \]

Bind unbind react

\[ P() \triangleq \ldots \]
As 12 processes (in SPiM)

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

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

let E1() =
?a1(u1,k1); (do ?u1;E1() or ?k1;E1())

let E2() =
?a2(u2,k2); (do ?u2;E2() or ?k2;E2())

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

and KK_PP() =
(new u6@d6:Release new k6@r6:React
  do !a6(u6,k6); (do !u6;KK_PP() or !k6;KK_P())
  or ?a7(u7,k7); (do ?u7;KK_PP() or ?k7;KK_PP())
  or ?a9(u9,k9); (do ?u9;KK_PP() or ?k9;KK_PP()))

and KKPse() =
do ?a4(u4,k4); (do ?u4;KKPse() or ?k4;KKPse())

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

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

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

and KPse() =
do ?a8(u8,k8); (do ?u8;KPse() or ?k8;KPse())

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

No need for conservation equations: implicit in “choice” operator in the calculus.
... and 30 Interaction Channels

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

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

... 

run 100 of KKK() run 100 of KK() run 100 of K()
run 1 of E2() run 1 of KKPse() run 1 of KPse()
run 1 of E1()
**MAPK Cascade Simulation in SPiM**

1st stage:
- KKK* barely rises

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

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

Rates and concentrations from paper:

- $1 \times E_2 (0.3 \text{ nM})$
- $1 \times \text{KKP'ase} (0.3 \text{ nM})$
- $120 \times \text{KKP'ase} (120 \text{ nM})$
- $3 \times \text{KKK} (3 \text{ nM})$
- $1200 \times \text{KK} (1.2 \text{ uM})$
- $1200 \times \text{K} (1.2 \text{ uM})$

$dx = rx = 150, \ ax = 1$

$(K_{mx} = (dx + rx) / ax, \ Km = 300 \text{ nM})$

$1 \times E_1$
MAPK Cascade Simulation in SPiM

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

1\textsuperscript{st} stage:
KKK\(^*\) barely rises

2\textsuperscript{nd} stage:
KK-PP rises, but is not stable

3\textsuperscript{rd} stage:
K-PP flips up to max
   even anticipating 2\textsuperscript{nd} stage

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

Input is 1xE1.
Output is 90xK-PP (ultrasensitivity).
Parameters from paper
(wide rate range: 1-150, wide concentration range: 3nm – 1200nm)

Artificial parameters
(all rates 1.0, all concentrations 1000)
Inverter ODE
Inverter ODE

\[
\text{Not}_{hi}(a,b) = !b;\text{Not}_{hi}(a,b) \oplus ?a;\text{Not}_{lo}(a,b)
\]

\[
\text{Not}_{lo}(a,b) = \tau_{del};\text{Not}_{hi}(a,b)
\]

\[
A(a) = !a;(A(a)|A(a)) \text{ linearly increasing input}
\]

\[
A(x_{(r)}) \mid n \text{ of Not}_{hi}(x_{(r)},y_{(s)})
\]

\[
\text{Not}_{hi}/x,y + A/x \rightarrow r \text{ Not}_{lo}/x,y + A/x + A/x
\]

\[
\text{Not}_{lo}/x,y \rightarrow \text{del} \text{ Not}_{hi}/x,y
\]

\[
A/x + n \text{ of Not}_{hi}/x,y
\]

\[
[A/x]^* = r[\text{Not}_{hi}/x,y][A/x]
\]

\[
[\text{Not}_{hi}/x,y]^* = -r[\text{Not}_{hi}/x,y][A/x]+\text{del}[\text{Not}_{lo}/x,y]
\]

\[
[\text{Not}_{lo}/x,y]^* = r[\text{Not}_{hi}/x,y][A/x]-\text{del}[\text{Not}_{lo}/x,y]
\]
Francois and Hakim

Design of genetic networks with specified functions by evolution in silico

Paul Francois and Vincent Hakim*

Laboratoire de Physique Statistique, Centre National de la Recherche Scientifique–Unité Mixte de Recherche 8550, Ecole Normale Supérieure, 24, Rue Lhomond, 75231 Paris, France

Edited by Nancy J. Kopell, Boston University, Boston, MA, and approved November 19, 2003 (received for review July 19, 2003)

Recent studies have provided insights into the modular structure of genetic regulatory networks and emphasized the interest of quantitative functional descriptions. Here, to provide a priori knowledge of the structure of functional modules, we describe an evolutionary procedure in silico that creates small gene networks performing basic tasks. We used it to create networks functioning as bistable switches or oscillators. The obtained circuits provide a variety of functional designs, demonstrate the crucial role of posttranscriptional interactions, and highlight design principles also found in known biological networks. The procedure should prove helpful as a way to understand and create small functional modules with diverse functions as well as to analyze large networks.

Fig. 1. Sketch of a bistable switch with reciprocal transcriptional repression between genes a and b.
Design of genetic networks with specified functions by evolution in silico

Francois & Hakim Fig3A

Fig 3A

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Constants</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \rightarrow a+A$</td>
<td>0.20</td>
<td>0.9 - 1.1</td>
</tr>
<tr>
<td>$A \rightarrow \text{Nothing}$</td>
<td>0.0085</td>
<td>0.6 - 1.5</td>
</tr>
<tr>
<td>$b \rightarrow b+B$</td>
<td>0.37</td>
<td>0.7 - 1.3</td>
</tr>
<tr>
<td>$B \rightarrow \text{Nothing}$</td>
<td>0.034</td>
<td>0.0 - 8.9</td>
</tr>
<tr>
<td>$A+B \rightarrow A:B$</td>
<td>0.72</td>
<td>0.1 - 10</td>
</tr>
<tr>
<td>$A:B \rightarrow \text{Nothing}$</td>
<td>0.53</td>
<td>Irrelevant</td>
</tr>
<tr>
<td>$b+A \rightarrow b:A$</td>
<td>0.19</td>
<td>0.7 - 7.6</td>
</tr>
<tr>
<td>$b:A \rightarrow b+A$</td>
<td>0.42</td>
<td>0.2 - 1.5</td>
</tr>
<tr>
<td>$b:A \rightarrow b:A+B$</td>
<td>0.027</td>
<td>0.0 - 2.3</td>
</tr>
</tbody>
</table>

Fig 14A
François & Hakim Fig3A, SPiM simulation

Parameters as in paper

Free evolution

Spontaneous switch at ~500
30xA injected at ~3000
30xB injected at ~4000

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

3 copies of each gene.

Modified for stability: $d_k A = 0.02$, $d_k B = 0.02$
Circuit of Fig 3A with parameters from Supporting Text Fig 8, plotted in Fig 13A

<table>
<thead>
<tr>
<th>Reactions</th>
<th>Constants</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a \rightarrow a+A$</td>
<td>0.52</td>
</tr>
<tr>
<td>$A \rightarrow$Nothing</td>
<td>0.00019</td>
</tr>
<tr>
<td>$b \rightarrow b+B$</td>
<td>0.79</td>
</tr>
<tr>
<td>$B \rightarrow$Nothing</td>
<td>0.0030</td>
</tr>
<tr>
<td>$A+B \rightarrow A:B$</td>
<td>0.053</td>
</tr>
<tr>
<td>$A:B \rightarrow$Nothing</td>
<td>0.15</td>
</tr>
<tr>
<td>$b+A \rightarrow b:A$</td>
<td>0.22</td>
</tr>
<tr>
<td>$b:A \rightarrow b+A$</td>
<td>0.31</td>
</tr>
<tr>
<td>$b:A \rightarrow b:A+B$</td>
<td>0.43</td>
</tr>
</tbody>
</table>

Fig 8

Fig 13A
(* François and Hakim circuit 3A *)

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())