Calculs Analogiques dans les Programmes Biochimiques Naturels et Synthétiques

François Fages

Project-Team Lifeware

http://lifeware.inria.fr/

Institut National de Recherche en Informatique et Automatique

Inria Saclay – Ile de France

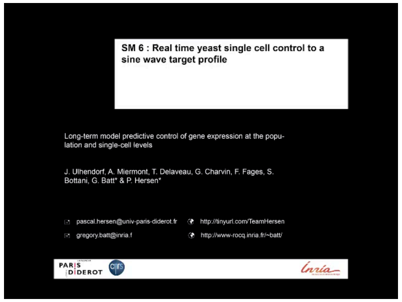
Cells Compute

They process signals

Regulate their metabolism

Take decisions such as

- Replication
- Differentiation
- Migration



SM 6 : Real time yeast single cell control to a sine wave target profile

Long-term model predictive control of gene expression at the population and single-cell levels

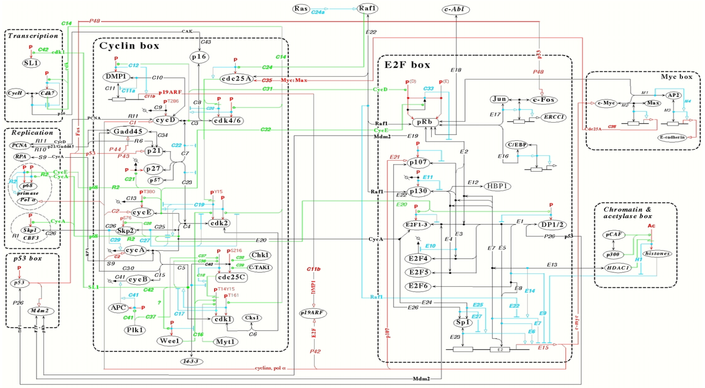
J. Ulhendorf, A. Miermont, T. Delaveau, G. Charvin, F. Fages, S. Bottani, G. Batt* & P. Hersen*

pascal.hersen@univ-paris-diderot.fr http://tinyurl.com/TeamHersen

gregory.batt@inria.f http://www-rocq.inria.fr/~batt/

PARIS DIDEROT

but it's an analog world
gradual activation of proteins
computations with no clock

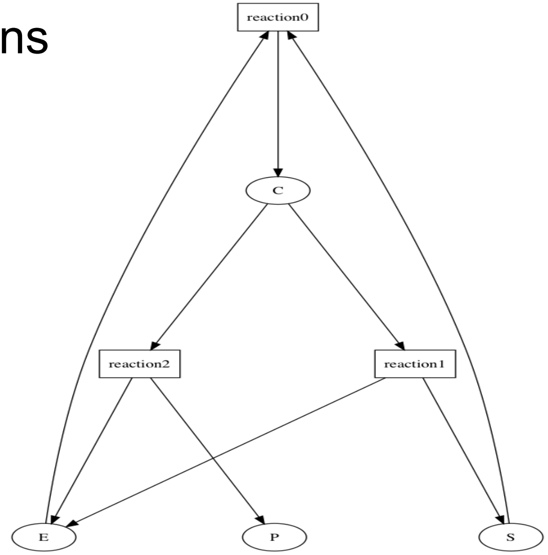


Understanding these processes is a central difficulty in many applications in medicine, health, agriculture and the ultimate goal of molecular cell biology.

Chemical Reaction Networks (CRNs)

CRN structure: network of reactants, products, and reactions

- $2H_2 + O_2 \to 2H_2O$ but not necessarily mass balanced
- synthesis reactions $\_ \to A$ degradation reactions $A \to \_$
- hypergraph of reactions (bipartite species-reaction graph)



CRN dynamics: several interpretations

- differential equations, continuous-time Markov chains,
- Petri net, Boolean transitions

CRN model repositories (Systems Biology Markup Language SBML):

- BioModels.net ≈ 2000 models + 10000 models of metabolism

CRN theory: interplay between CRN structure and CRN dynamics

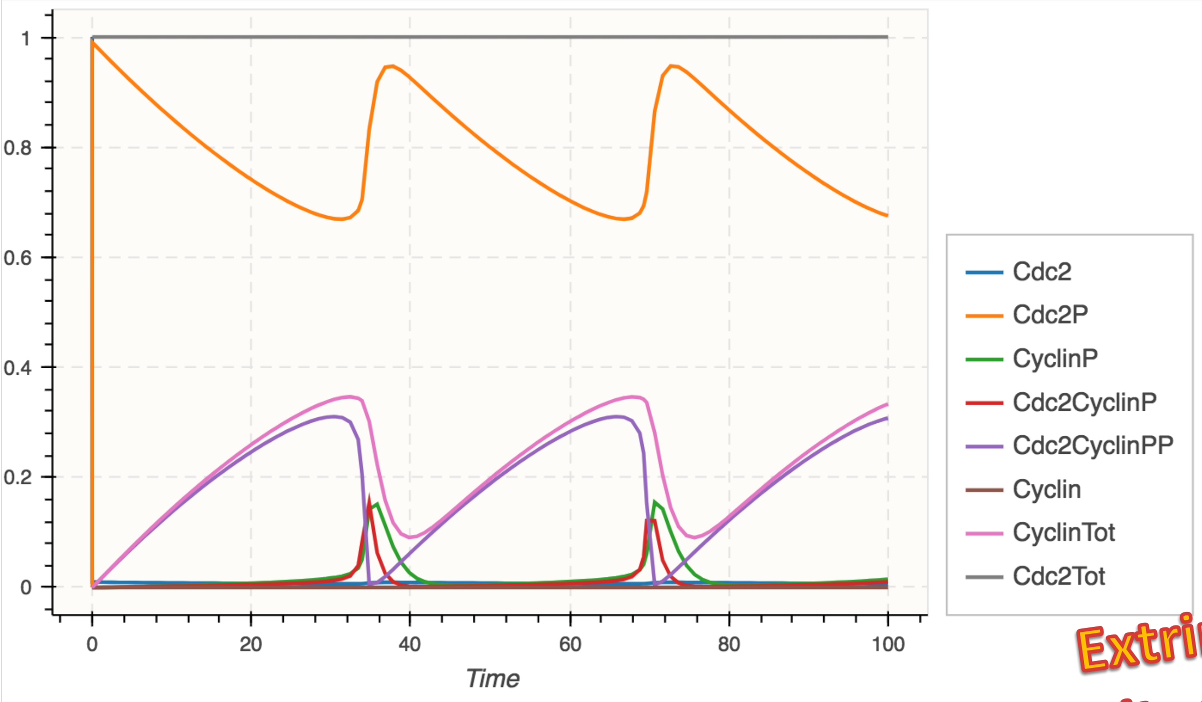
- static analysis of steady states, stable states, oscillations,…
- reductions by quasi-steady state (QSS) and quasi-equilibrium (QE) approximations
- Turing completeness and computational complexity of CRNs

Several Interpretations $A + B \xrightarrow{k.A.B} C$

Continuous semantics: concentrations, continuous time evolution

Ordinary differential equations (ODE)

$$\frac{dA}{dt} = -k.A.B \quad \frac{dB}{dt} = -k.A.B \quad \frac{dC}{dt} = k.A.B$$



Extrinsic variability (sensitivity to parameter change)

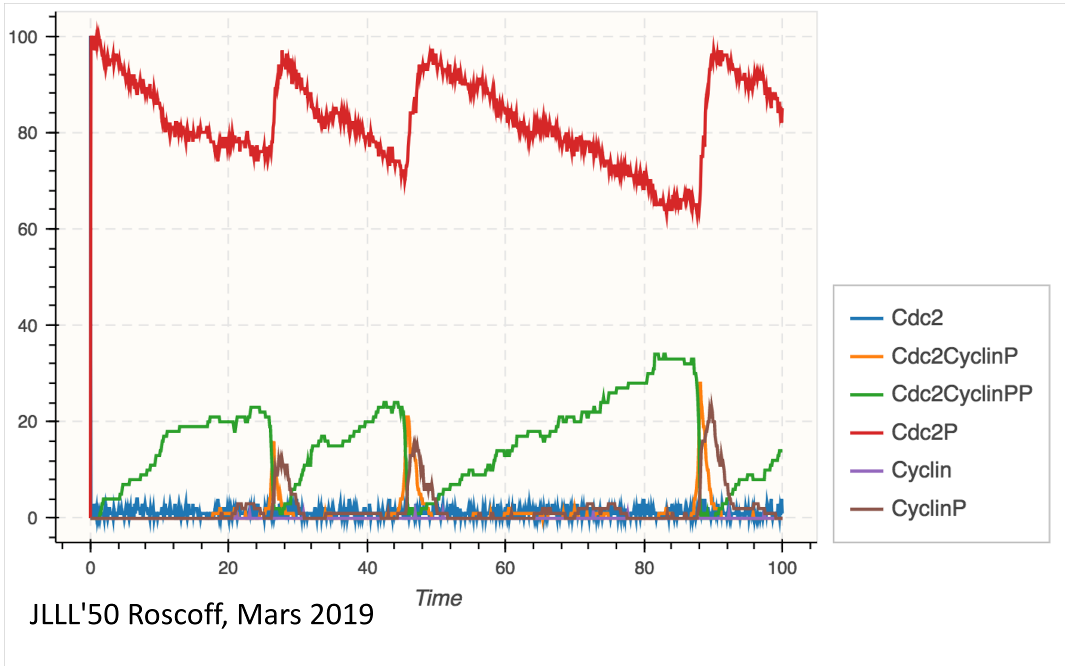
Several Interpretations $A + B \xrightarrow{k.A.B} C$

Continuous semantics: concentrations, continuous time evolution

Ordinary differential equations (ODE) $\frac{dA}{dt} = -k.A.B \quad \frac{dB}{dt} = -k.A.B \quad \frac{dC}{dt} = k.A.B$

Stochastic semantics: numbers of molecules, probability and time of transition

Continuous Time Markov Chain (CTMC) $A\,,B \xrightarrow{p(S_i),\,t(Si)} C{+}{+},\,A{-}{-},\,B{-}{-}$



Intrinsic variability (with same genetic and epigenetic parameters)

Several Interpretations $A + B \xrightarrow{k.A.B} C$

Continuous semantics: concentrations, continuous time evolution

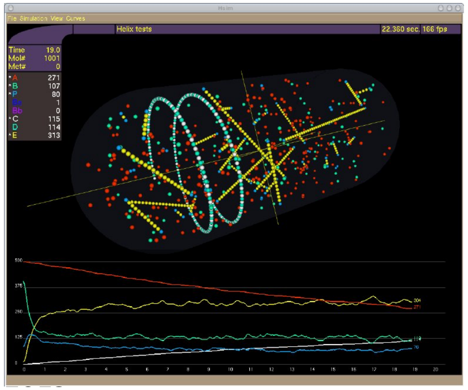
Ordinary differential equations (ODE) $\frac{dA}{dt} = -k.A.B \quad \frac{dB}{dt} = -k.A.B \quad \frac{dC}{dt} = k.A.B$

Stochastic semantics: numbers of molecules, probability and time of transition

Continuous Time Markov Chain (CTMC) $A\,,B \xrightarrow{p(S_i),\,t(Si)} C{+}{+},\, A{-}{-},\, B{-}{-}$

Multi-agent simulation: numbers of molecules, space, diffusion speed, affinity

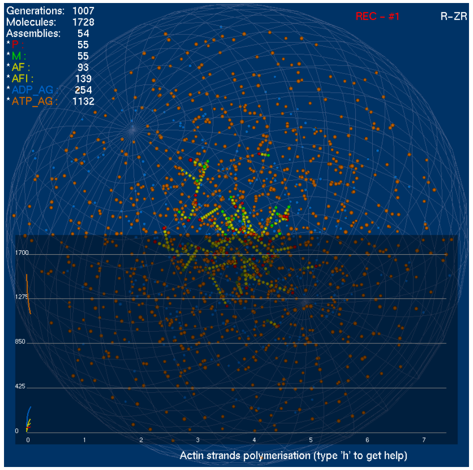
Random walk (ex. Hsim simulator [Amar 04])



```
Helix tests                22.360 sec. 166 fps

Time    19.0
Mol#    1001
Met#       0

* A      271
* B      107
* P       80
  Ba       1
  Bb       0
* C      115
  D      114
* E      313
```



```
Generations:  1007
Molecules:    1728
Assemblies:     54
* P :           55
* M :           55
* AF :          93
* AFI :        139
* ADP_AG :     254
* ATP_AG :    1132
```

Actin strands polymerisation (type 'h' to get help)

Several Interpretations $A + B \xrightarrow{k.A.B} C$

Continuous semantics: concentrations, continuous time evolution

Ordinary differential equations (ODE) $\frac{dA}{dt} = -k.A.B \quad \frac{dB}{dt} = -k.A.B \quad \frac{dC}{dt} = k.A.B$

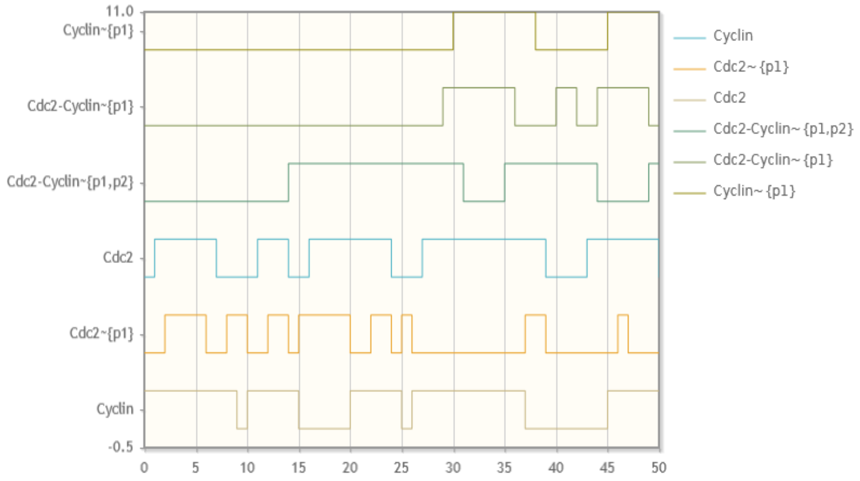
Stochastic semantics: numbers of molecules, probability and time of transition

Continuous Time Markov Chain (CTMC) $A\,,B \xrightarrow{p(S_i),\,t(Si)} C{+}{+},\,A{-}{-},\,B{-}{-}$

Petri net semantics: numbers of molecules $A\,,B \rightarrow C{+}{+},\,A{-}{-},\,B{-}{-}$

Multiset rewriting

CHAM [Berry Boudol 90] [Banatre Le Metayer 86]



Several Interpretations $\mathrm{A} + \mathrm{B} \xrightarrow{\mathrm{k.A.B}} \mathrm{C}$

Continuous semantics: concentrations, continuous time evolution

Ordinary differential equations (ODE) $\frac{dA}{dt} = -k.A.B \quad \frac{dB}{dt} = -k.A.B \quad \frac{dC}{dt} = k.A.B$

Stochastic semantics: numbers of molecules, probability and time of transition

Continuous Time Markov Chain (CTMC) $\mathrm{A}\,,\mathrm{B} \xrightarrow{\mathrm{p(S_i), t(Si)}} \mathrm{C{+}{+}, A{-}{-}, B{-}{-}}$

Petri net semantics: numbers of molecules $\mathrm{A}\,,\mathrm{B} \rightarrow \mathrm{C{+}{+}, A{-}{-}, B{-}{-}}$

Multiset rewriting

CHAM [Berry Boudol 90] [Banatre Le Metayer 86]

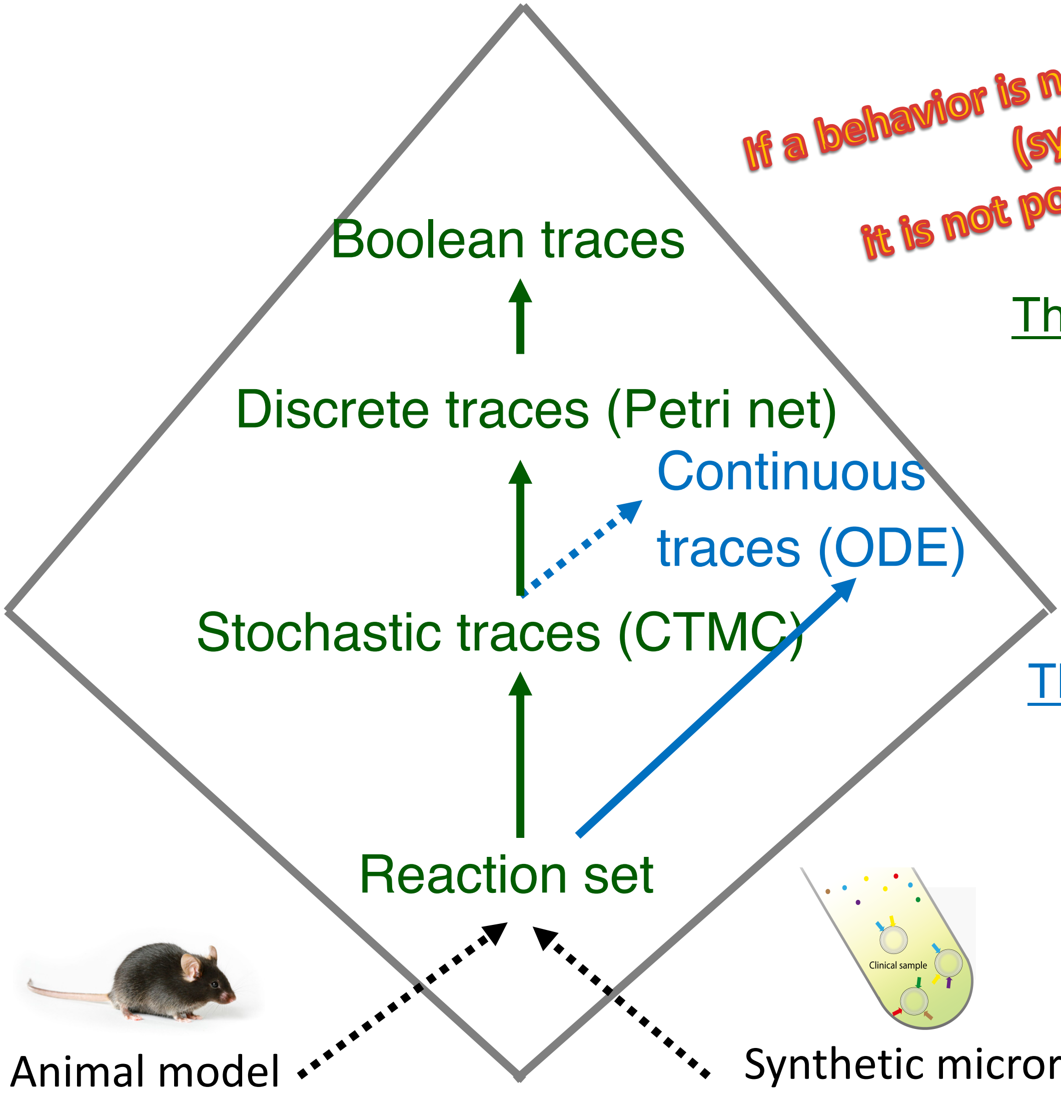
Boolean semantics: presence/absence $\mathrm{A} \wedge \mathrm{B} \rightarrow \mathrm{C} \wedge \neg\mathrm{A} \wedge \neg\mathrm{B}$

Asynchronous transition system $\mathrm{A} \wedge \mathrm{B} \rightarrow \mathrm{C} \wedge \mathrm{A} \wedge \neg\mathrm{B}$

Symbolic model-checking $\mathrm{A} \wedge \mathrm{B} \rightarrow \mathrm{C} \wedge \neg\mathrm{A} \wedge \mathrm{B}$

$\mathrm{A} \wedge \mathrm{B} \rightarrow \mathrm{C} \wedge \mathrm{A} \wedge \mathrm{B}$

Hierarchy of CRN Semantics


Boolean traces

Discrete traces (Petri net)

Continuous traces (ODE)

Stochastic traces (CTMC)

Reaction set

Animal model

Synthetic microreactor

If a behavior is not possible in the Boolean semantics (symbolic model-checking) it is not possible in the stochastic semantics for any reaction rates

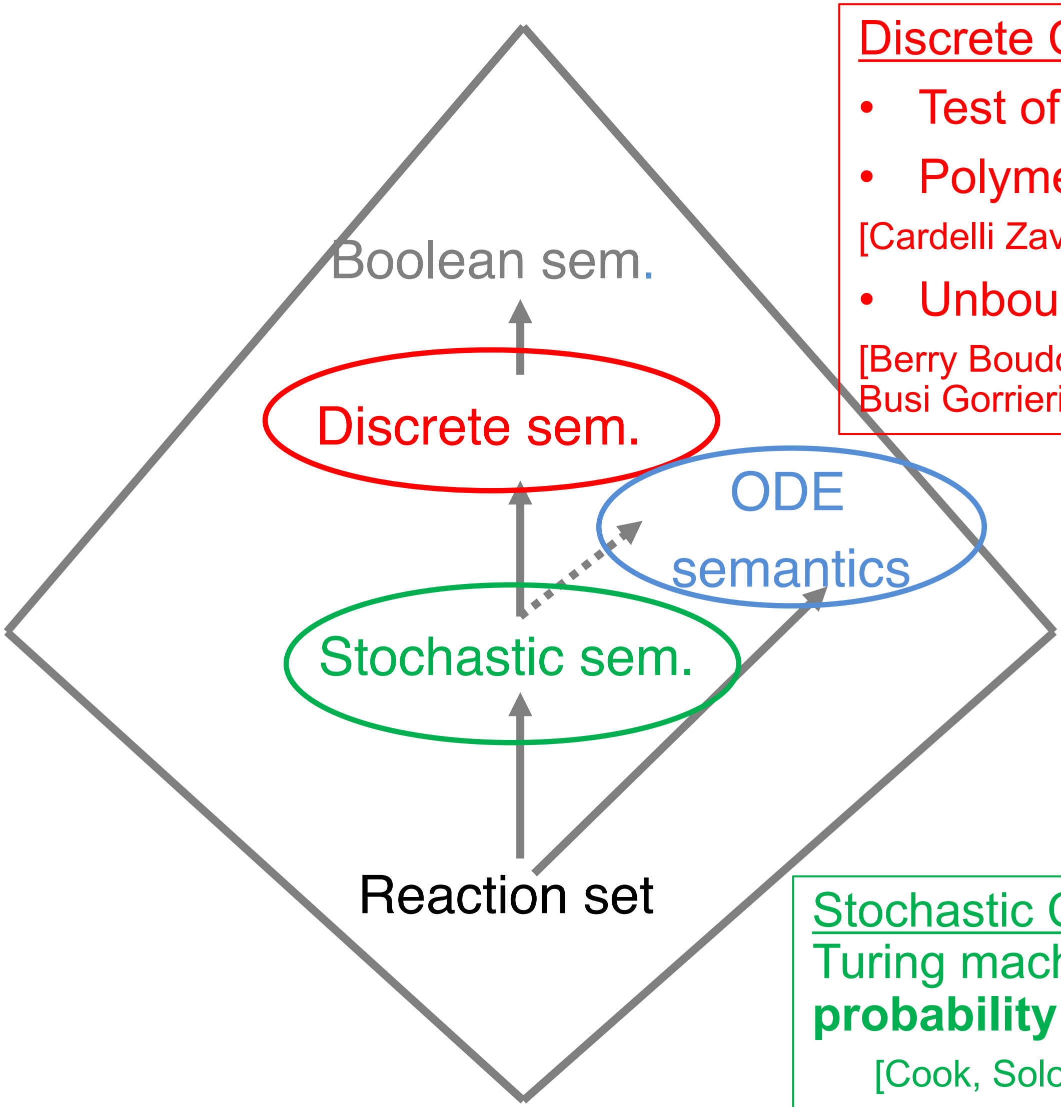
Thm. (abstract interpretation ↑) Galois connections between the syntactical, stochastic, Petri net and Boolean trace semantics

[Fages Soliman Theoretical Computer Science 2008]

Thm. (approximation ⇡) For large numbers of molecules the ODE semantics approximates the *mean* stochastic behavior [Gillespie 1971 Kurtz 1978]

Model cell-to-cell variability Intrinsic and extrinsic variability

Turing Completeness of CRNs ?



Discrete CRN: **Not Turing complete** without

- Test of absence (Petri net inhibitor arc)
- Polymerisation reactions

[Cardelli Zavatero MSCS 2010, Cook et al 2009]

- Unbounded membranes

[Berry Boudol CHAM 1994, Paun Rozenberg TCS 2002, Busi Gorrieri CMSB 2005]

Continuous CRN: **Non uniform computability**: for each function *for each input there exists a circuit* computing the result

[Magnosco 1997 Phys. Rev., Helmfelt Weinberger PNAS 1991]

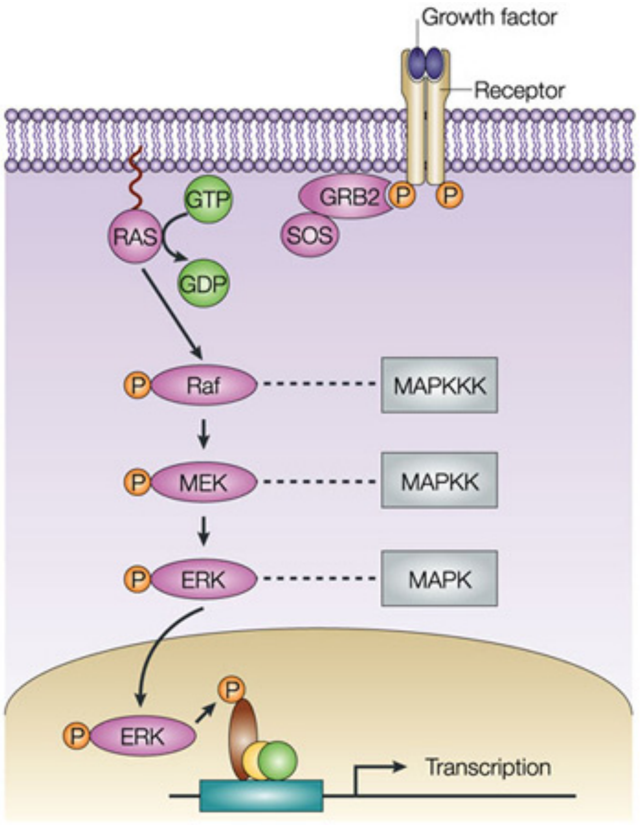
Strong Turing completeness?

Stochastic CRN: Simulation of a Turing machine with a **small probability of error**

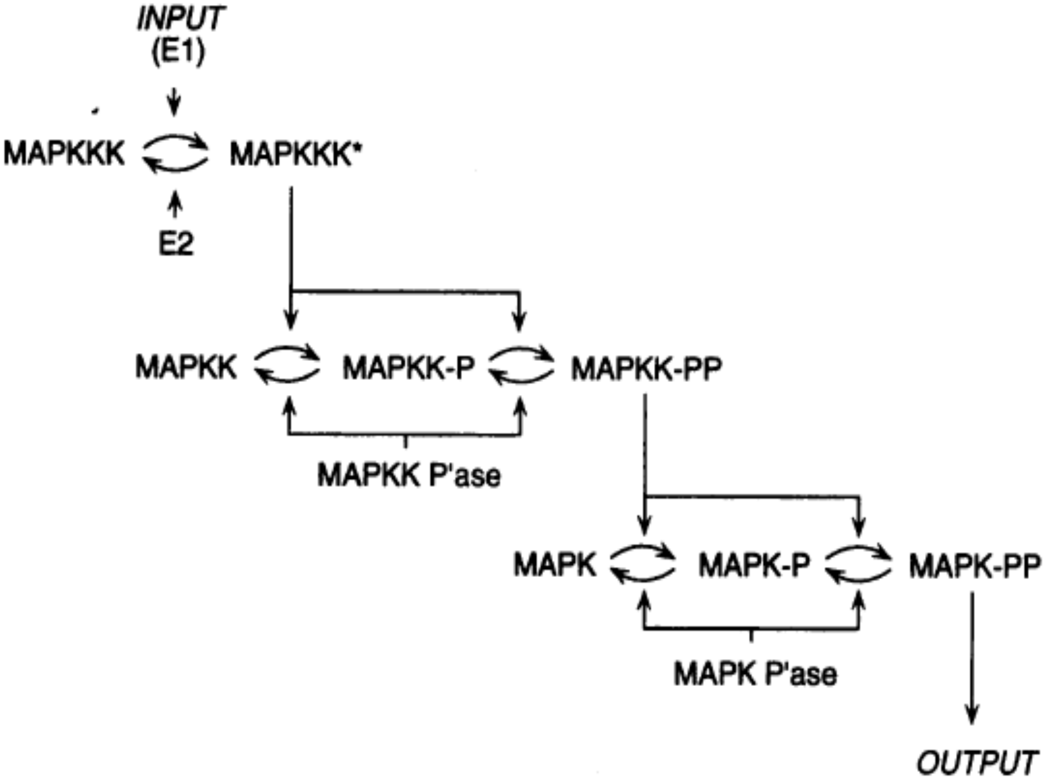
[Cook, Soloveichik, Winfree, Bruck 2009]

MAPK Signalling Cascade

MAPK Signaling Network: 30 reactions 18 species [Huang Ferrel PNAS 1996]



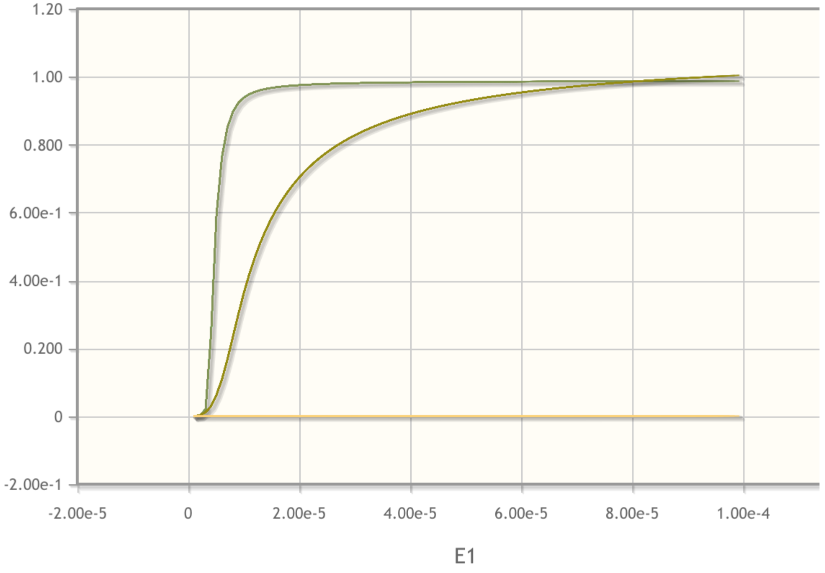
Nature Reviews | Molecular Cell Biology



MAPK Input/Output Function

Dose-response diagrams alias Bifurcation diagrams

```
biocham: load(library:examples/mapk/mapk).
biocham: dose_response('E1',1.0e-6,1e-4,200).
```



MAPK responses as Hill function $\frac{x^n}{c + x^n}$

[Huang Ferrel 96 PNAS]

n ≈ 4.9 at 3rd level

n ≈ 1.7 at 2nd level

n = 1 at 1st level (Michaelis-Menten)

MAPK implements the function of an analog/digital converter in the cell.

How would one program $\frac{x^n}{c + x^n}$ with biochemical reactions ?

What does it mean to compute with real numbers ?

Computable Real Numbers and Functions

Classical definitions of computable analysis based on Turing machines

Definition. A real number *r* is computable if there exists a Turing machine with

Input: precision $p \in N$

Output: rational number $q \in Q$ with $| r-q | < 2^{-p}$

Examples. Rational numbers, limits of computable Cauchy sequences π, e, …

Definition. A real function f:R→R is computable if there exists a Turing machine that computes f(x) with an oracle for x.

Examples. Polynomials, trigonometric functions, …

Counter-examples. x=0, $[x]$ are not computable (undecidable on x=0.000…) discontinuous functions

Analog encoding e(w) of decision problems by f: accept w if $f(e(w)) \geq 1$ reject if ≤ -1

Analog Computer? Differential Analyzer [Bush 1931]

Underlying principles: Lord Kelvin, 1876

First ever built: Vannevar Bush, MIT, 1931

Applications: from gunfire control up to aircraft design

- Intensively used by the U.S. and Japanese armies during world war II
- Electronic versions from late 40s, used until 70s

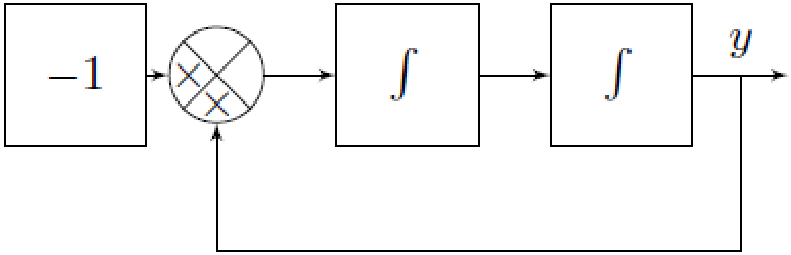
General Purpose Analog Computer [Shannon 1941]

Shannon's formalization of the Differential Analyser by GPAC circuits

A time function if GPAC-generated if it is the output of some unit of a GPAC circuit built from:

1. Constant unit
2. Sum unit
3. Product unit
4. Integral $\int x \, dy$ unit

What does this GPAC circuit compute ?

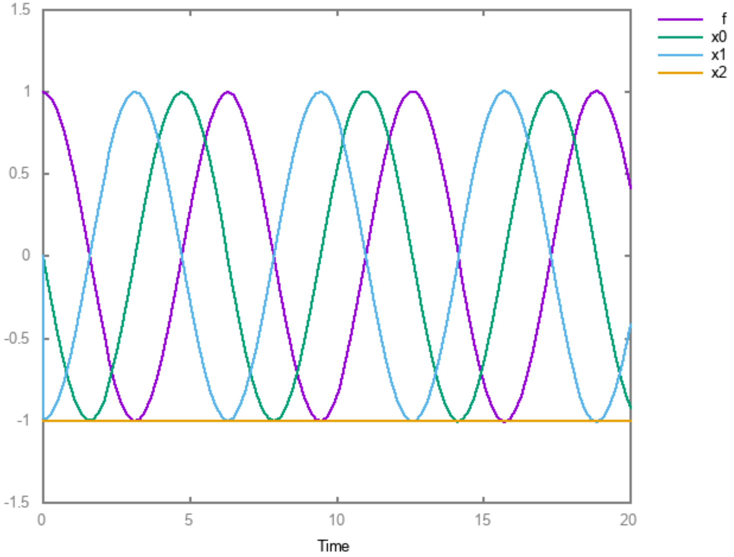


$$y_1 = \frac{dy}{dt}$$

$$\frac{dy_1}{dt} = -y \;=\; y''$$

if $y(0) = 1, y_1(0) = 0$

$y(t) = cos(t) \quad y_1(t) = sin(t)$



CRN Implementation of GPAC Units

Mass action law kinetics reaction network with output concentration stabilizing on the result of the operation applied to the input concentrations

Positive constant units: molecular concentrations

Product unit $z = x.y$

$$x + y \xrightarrow{k.x.y} x + y + z$$

$$\mathrm{z} \xrightarrow{k.z} \_$$

$$\frac{dz}{dt} = k(xy - z)$$

$$= 0 \text{ when } z = x.y$$

Sum unit $z = x + y$

$$\mathrm{x} \xrightarrow{k.x} x + z$$

$$y \xrightarrow{k.y} y + z$$

$$z \xrightarrow{k.z} \_$$

$$\frac{dz}{dt} = k(x + y - z)$$

$$= 0 \text{ when } z = x + y$$

Time integral $\mathrm{z} = \int x\, dt$ unit

$$\mathrm{x} \xrightarrow{x} x + z$$

$$\frac{dz}{dt} = x$$

$$\mathrm{z}=\int_0^T x\, dt$$

Polynomial ODE Initial Value Problems (PIVP)

Graça and Costa 2003's formalization of Shannon's GPAC

Definition. A real time function $f:R_+\rightarrow R$ is GPAC-generable iff there exist a vector of polynomials $p \in R^n[R^n]$ and of initial values $y(0) \in R^n$ and a solution function $y:R_+\rightarrow R^n$ such that $y'(t)=p(y(t))$ and $\mathbf{f(t)=y_1(t)}$

Closure properties:

f+g, f-g, f.g, 1/f, ,f ◦g, y s.t. y' =f(y) are GPAC-generable if f, g are.

A GPAC-generated function must be analytic (i.e. convergent power series)

Famous analytic non-GPAC-generable functions [Shannon 41]

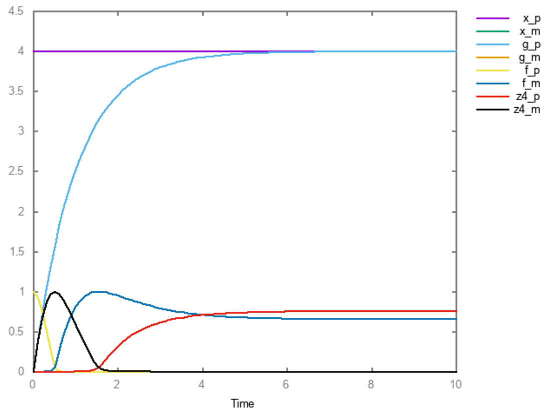
- Euler's Gamma function $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}dt$ [Hölder1887]
- Riemann's Zeta function $\zeta(x) = \sum_{k=0}^{\infty}\frac{1}{k^x}$ [Hilbert]

But analytic functions are computable

PIVP-Computable Functions f(x)

Definition. [Graça Costa 03 J. Complexity] A real function $f:R\to R$ is PIVP-computable if there exists vectors of polynomials $p\in R^n[R^n]$ and $q\in R^n[R]$ and a function y: $R^n \to R^n$ such that $y'(t)=p(y(t))$, $\mathbf{y(0)=q(x)}$ and $\mathbf{|y_1(t)-f(x)|<y_2(t)}$ with $y_2(t) \geq 0$ decreasing for t>1 and $\lim_{t\to\infty} y_2(t) = 0$

Example. cos(4)



Reconciles Digital Computation and Analog Computation !

Theorem (analog characterization of Turing computability).

[Bournez Campagnolo Graça Hainry 07 J. Complex]]

A real function is computable (by Turing machine) iff it is PIVP-computable.

Analog characterization of Ptime

Time in ODE is a bad measure of complexity

- Exponential speedup by changing time variable $t' = e^t$
- But price to pay in the amplitude of t'

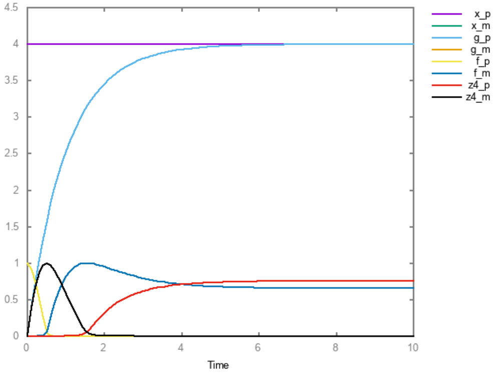
A computational complexity measure should combine time and space-amplitude

- length in the n dimensions of the trajectory to compute the result

Theorem [Pouly PhD thesis 2015, Bournez Graca Pouly 16 ICALP]

A real function is computable in P iff it is PIVP-computable with a trajectory of polynomial length (i.e. polynomial time and polynomial amplitude)

Reconciles Digital and Analog Ptime Complexity !



Turing Completeness of Continuous CRNs 1/3

Lemma (positive systems) Any PIVP-computable function can be encoded by a PIVP of double dimension on R^+, preserving polynomial length complexity.

Proof. Encode $y_i \in R$ by y^-_i $y^+_i \in R^+$ such that $\mathbf{y_i = y^+_i - y^-_i}$ at each time

(encoding used in [Oishi Klavins 2011] for linear I/O systems)

Let $\underline{p}_i(y^+_1, y^-_1, \ldots, y^+_n, y^-_n) = p_i[y = y^+_i - y^-_i]$ and $\underline{p}_i = \underline{p}^+_i - \underline{p}^-_i$

$y^+_i{}' = \underline{q}^+_i - f_i\, y^+_i\, y^-_i$ $\qquad y^+_i(0) = \max(0, y_i(0))$

$y^-_i{}' = \underline{q}^-_i - f_i\, y^+_i\, y^-_i$ $\qquad y^-_i(0) = \max(0, -y_i(0))$

Where $f_i = q^+_i + q^-_i$ are positive coefficient polynomials $f_i \geqq \max(q^+_i, q^-_i)$

- Fast annihilation reactions: $y^+_i + y^-_i \xrightarrow{f_i} \_$
- n-ary catalytic synthesis reactions for each monomial $\underline{m}^+_{i,j}$ in $\underline{p}^+_i$, $\underline{m}^-_{i,j}$ in $\underline{p}^-_i$:

$$M_{i,j}{}^{+} \xrightarrow{\underline{m}^+_{i,j}} y^+_i + M_{i,j}{}^{+}$$

$$M_{i,j}{}^{-} \xrightarrow{\underline{m}^-_{i,j}} y^+_i + Mi_{,j}{}^{-}$$

Turing Completeness of Continuous CRNs 2/3

Lemma (quadratic systems) [Carothers Parker Sochacki Warne 2005]
Any PIVP can be encoded by a PIVP of degree ≤ 2.

Proof. Introduce variable $v_{i1,\ldots,in}$ for each possible monomial $y_1^{i1}\ldots y_n^{in}$

We have $y_1 = v_{1,0\ldots,0}$, $y_2 = v_{0,1,0\ldots,0}$,…

y'_i is of degree one in $v_{i1,\ldots,in}$

$v'_{i1,\ldots,in} = \sum_{k=1}^{n} i_k \, v_{i_1,\ldots,i_{k\_1},\ldots,i_n} \; y'_k$ is of degree at most 2.

i.*e. trade high dimension for low degrees.*

(yet naïve algorithm of exponential complexity)

Turing Completeness of Continuous CRNs 3/3

Theorem (Turing completeness of continuous CRNs) [F Le Guludec Bournez Pouly CMSB 2017]

Any computable function over the reals can be computed by a continuous CRN over a finite set of molecular species (no polymerization, no locations)

Proof: By previous lemmas, any PIVP-computable function can be encoded by a PIVP of degree at most 2 with positive variables. A positive PIVP of degree at most 2 can be represented by an elementary CRN with at most 2 reactants per reaction.

In this view, the (protein) concentrations are the information carriers.

The programs of a cell are implicitly defined by the set of all possible reactions with the proteins encoded in its genome and the chemicals of the environment.

Program change is determined by gene expression (= metaprogram).

In this view, programming becomes a natural science

From Abstract to Concrete Implementation

Theorem (abstract CRN)
A real function is computable (respectively in polynomial time)
if and only if it is computable by a system of elementary reactions of the form

_ => z   or   x => x+z   or   x+y => x+y+z

plus annihilation reactions x+y => _   with mass action law kinetics
(respectively with trajectories of polynomial length as a function of both the unary precision and the argument values).

Proof Close analysis of the encoding used in the lemmas (positive monomials)

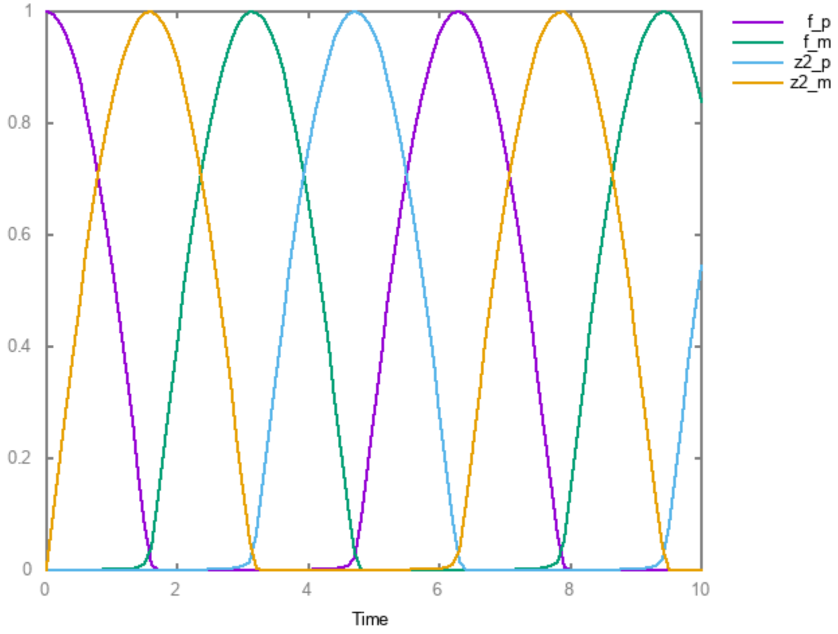
Intermediate CRN: Instead of formal synthesis and degradation reactions, activation and deactivation reactions (e.g. phosphorylation, complexation)

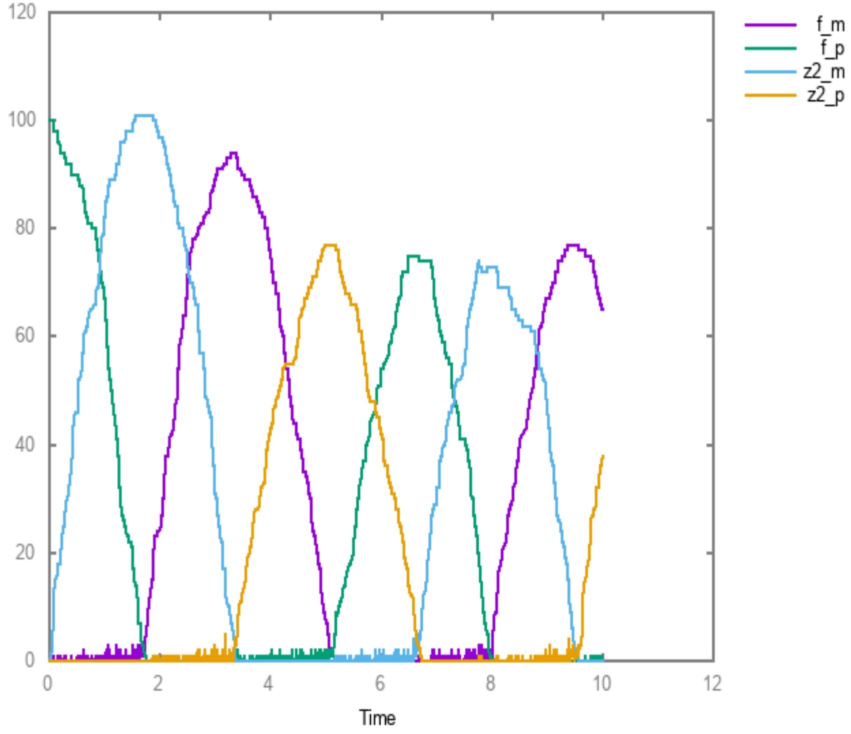
Concrete CRN: catalogue of real enzymes [F. Molina's Lab Sys2Diag, Montpellier]
microreactors in DNA-free vesicles created by microfluidic device

Compilation of the Cosine(t) function

```
biocham: compile_from_expression(cos,time,f).
 _ =[z_p]=> f_p.   z_m+z_p => _.
 _ =[z_m]=> f_m.   f_m+f_p => _.
 _ =[f_m]=> z_p.
 _ =[f_p]=> z_m.
present(f_p,1).
```

$$d(f_p)/dt = z_p-k*f_m*f_p$$
$$d(f_m)/dt = z_m-k*f_m*f_p$$
$$d(z_p)/dt = f_m-k*z_m*z_p$$
$$d(z_m)/dt = f_p-k*z_m*z_p$$
$$f_p(0)=1$$



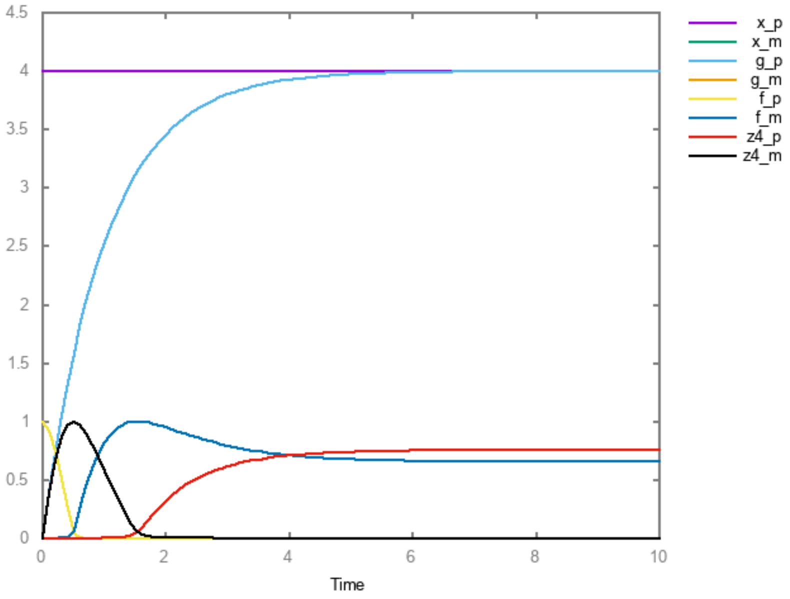


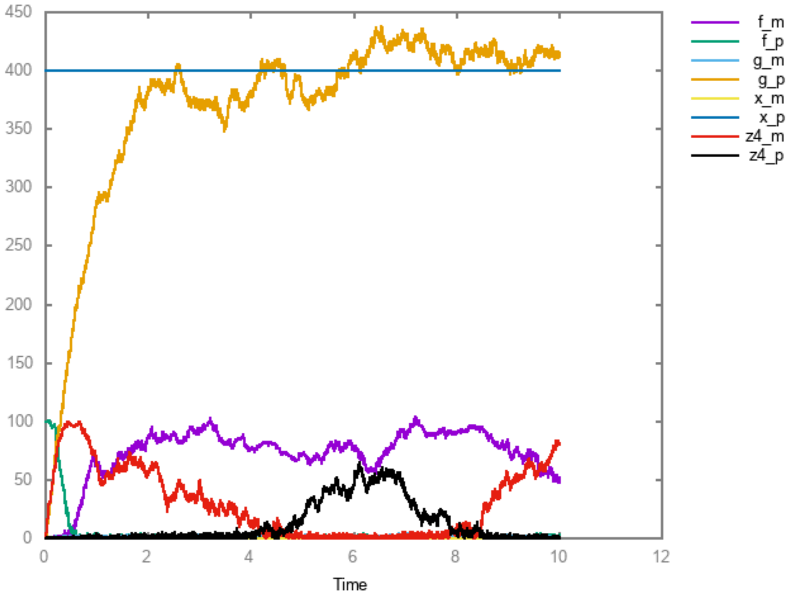
Compilation of the Cosine(x) Function

```
biocham: present(x_p, 4).
biocham: compile_from_expression(cos,x,f).
 present(f_p, 1).
_=[g_m]=>g_p.                _=[g_m+f_m]=>z_p.
_=[x_p]=>g_p.                _=[g_p+f_p]=>z_p.
_=[g_p]=>g_m.                _=[x_p+f_m]=>z_p.
_=[x_m]=>g_m.                _=[x_m+f_p]=>z_p.
_=[g_m+z_p]=>f_p.          _=[g_m+f_p]=>z_m.
_=[g_p+z_m]=>f_p.          _=[g_p+f_m]=>z_m.
_=[x_m+z_m]=>f_p.          _=[x_m+f_m]=>z_m.
_=[x_p+z_p]=>f_p.          _=[x_p+f_p]=>z_m.
_=[g_m+z_m]=>f_m.          _=[x_p+f_p]=>z_m.
_=[g_p+z_p]=>f_m.          _=[x_m+z_p]=>f_m.
```

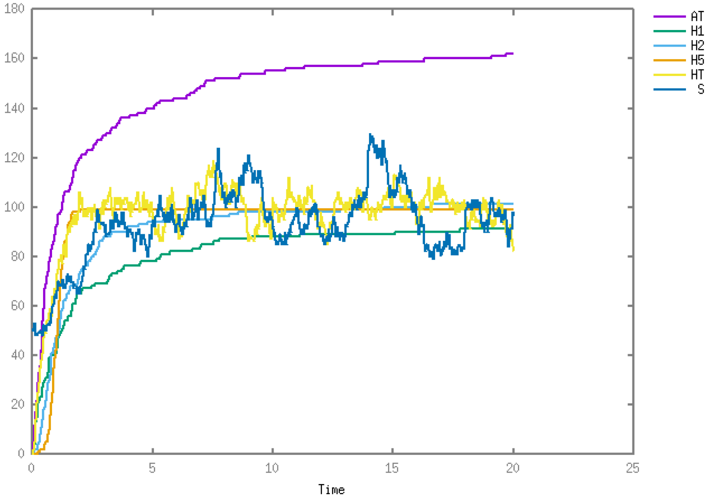
PIVP that generates f(g(t))
with $\lim_{t-\infty} g(t) = x$

$$g'(t) = x - g(t)$$
$$g(t) = x + (x0 - x)e^{-t}$$





Sigmoid Functions



Hyperbolic tangent

```
d(HT)/dt=1-HT^2
```

```
_=>HT.
HT=[HT]=>_.
```

Logistic

```
d(S)/dt=S-S^2
```

```
_=[S]=>S.
S=[S]=>_.
present(S,0.001).
```

Arc tangent

```
d(T)/dt=1
d(AT)/dt=1/ (1+T^2)
```

```
_=>T.
1/ (1+T^2) for _/T=>AT
```

Hill functions order 1,2,5

```
d(H1)/dt=NH1^2
d(NH1)/dt= -NH1^2

d(H2)/dt=2*T*NH2^2
d(NH2)/dt= - (2*T*NH2^2)

d(H5)/dt=5*T^4*NH5^2
d(NH5)/dt= - (5*T^4*NH5^2)
```

```
NH1=[NH1]=>_.
_=[2*NH1]=>H1.
present(NH1,1).
MA(2)for NH2=[T+NH2]=>_.
MA(2)for _=[T+2*NH2]=>H2.
present(NH2,1).
MA(5)for NH5=[4*T+NH5]=>_.
MA(5)for _=[4*T+2*NH5]=>H5.
present(NH5,1).
```

Logical Gates

And C = A /\ B	A+B => C	[C] = min([A],[B])

Or C = A \/ B	A => C
	B => C
		[C] = [A]+[B]

Not C = ¬ A	C+A => _	[C] = max([C_0]-[A], 0)

Computer-Aided Biochemical Programming of Synthetic Micro-reactors as Diagnostic Devices

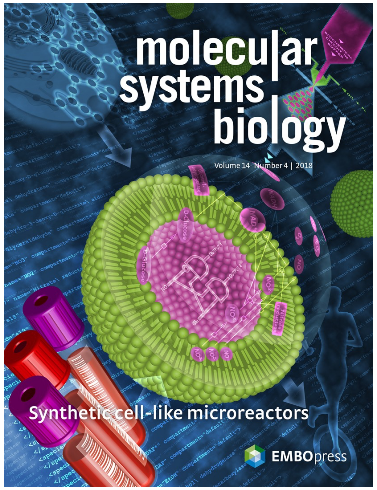
Alexis Courbet [1], Patrick Amar [2], François Fages [3],
Eric Renard [4], Franck Molina [1]

[1] *Sys2diag UMR9005 CNRS/ALCEDIAG, Montpellier*

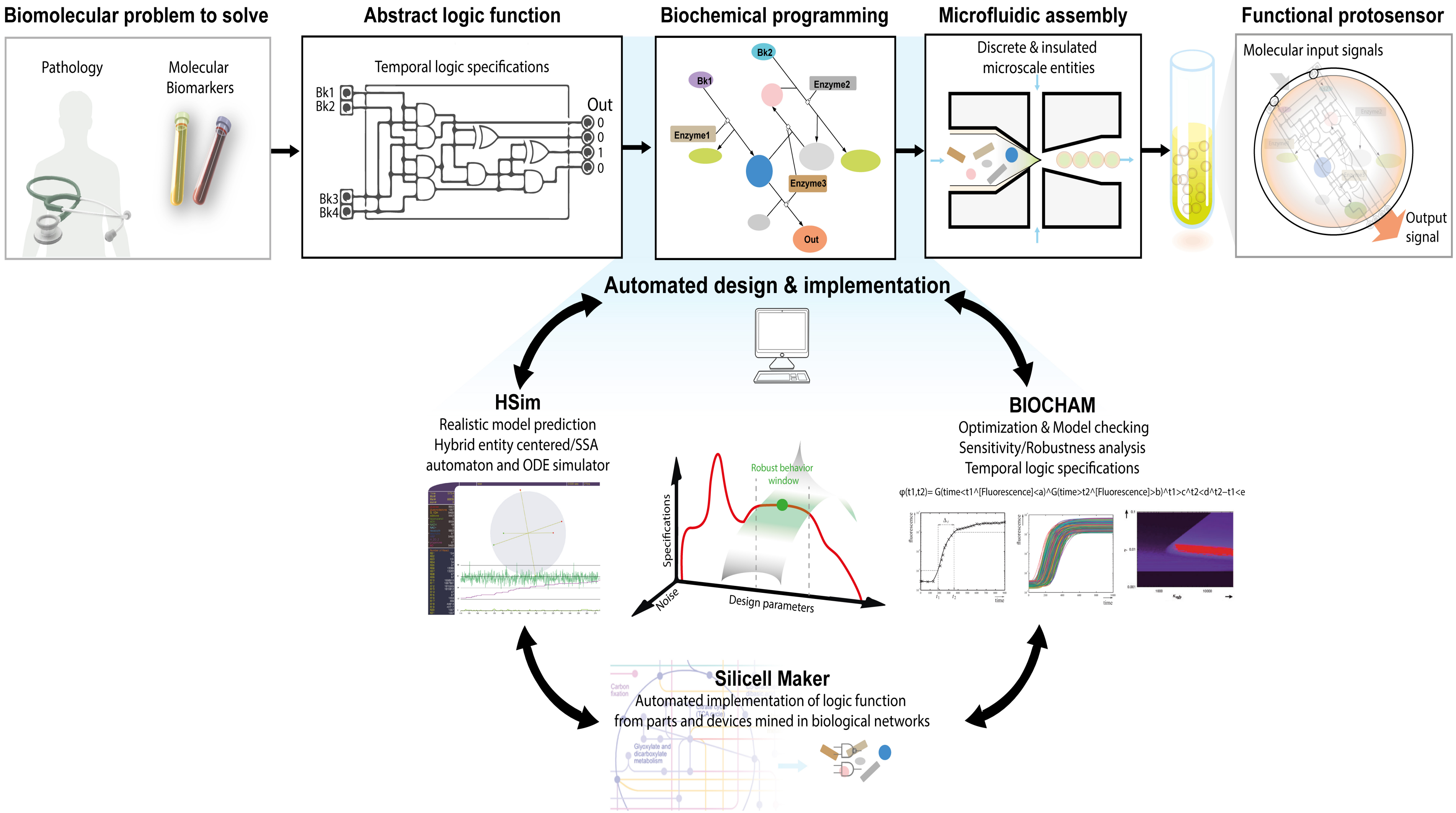
[2] *LRI, Université Paris Sud - UMR CNRS 8623, Orsay*

[3] *http://lifeware.inria.fr, Inria Saclay IdF, Palaiseau*

[4] *INSERM 1411, Montpellier University Hospital*


molecular systems biology
Volume 14 Number 4 | 2018
Synthetic cell-like microreactors
EMBOpress

Protosensor CRN Design Workflow



Biomolecular problem to solve

Pathology

Molecular Biomarkers

Abstract logic function

Temporal logic specifications

Biochemical programming

Microfluidic assembly

Discrete & insulated microscale entities

Functional protosensor

Molecular input signals

Output signal

Automated design & implementation

HSim

Realistic model prediction

Hybrid entity centered/SSA automaton and ODE simulator

BIOCHAM

Optimization & Model checking

Sensitivity/Robustness analysis

Temporal logic specifications

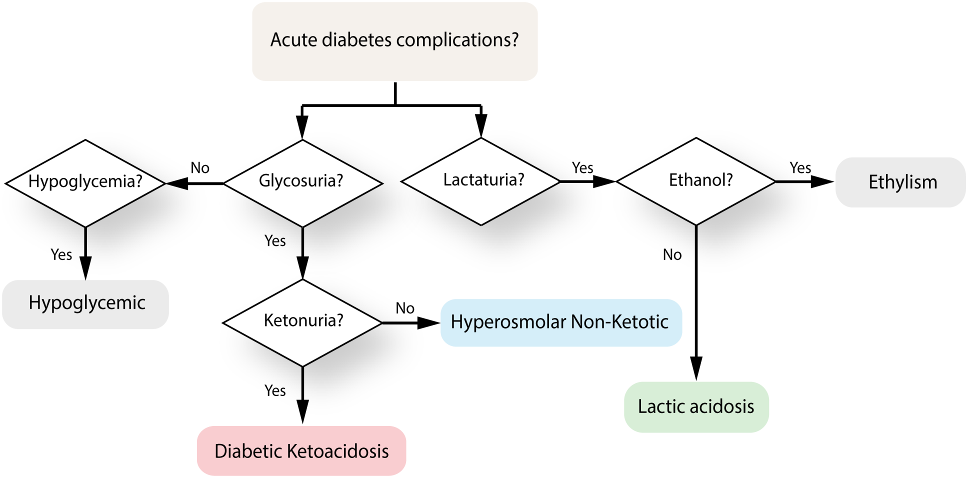
φ(t1,t2)= G(time<t1^[Fluorescence]<a)^G(time>t2^[Fluorescence]>b)^t1>c^t2<d^t2–t1<e

Silicell Maker

Automated implementation of logic function from parts and devices mined in biological networks

Carbon fixation
Citrate cycle (TCA cycle)
Glyoxylate and dicarboxylate metabolism

Diabetes Differential Diagnostic Algorithm



Acute diabetes complications?

Hypoglycemia?

No

Glycosuria?

Lactaturia?

Yes

Ethanol?

Yes

Ethylism

Yes

Hypoglycemic

Yes

No

Ketonuria?

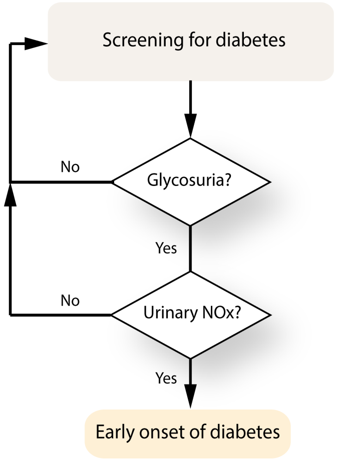
No

Hyperosmolar Non-Ketotic

Yes

Lactic acidosis

Diabetic Ketoacidosis



Screening for diabetes

No

Glycosuria?

Yes

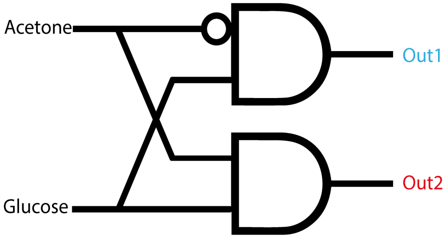
No

Urinary NOx?

Yes

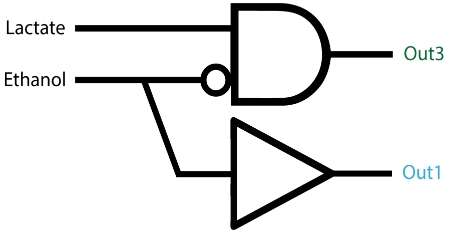
Early onset of diabetes

GluONe



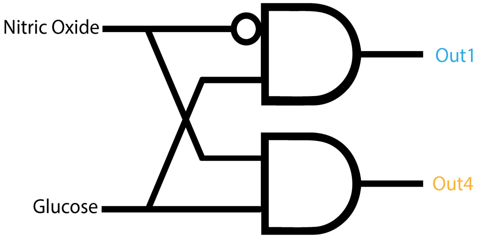
Glucose	Acetone	Out2	Out1
0	0	0	0
1	0	0	1
0	1	0	0
1	1	1	0

LacOH



Lactate	EtOH	Out3	Out1
0	0	0	0
1	0	1	0
0	1	0	1
1	1	0	1

GluNOx



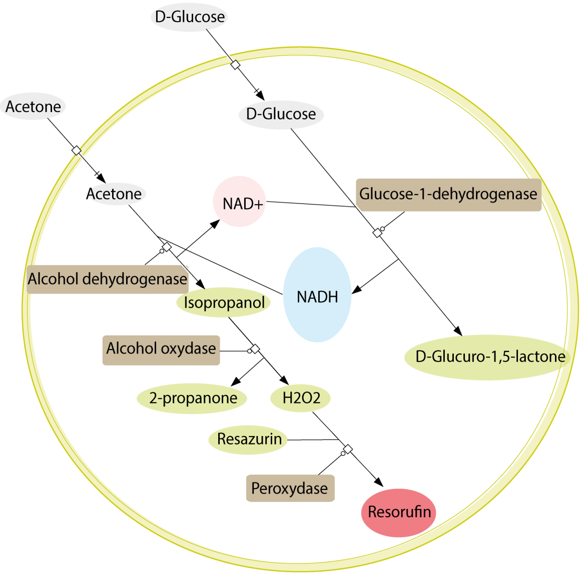
NOx	Glucose	Out4	Out1
0	0	0	0
1	0	0	0
0	1	0	1
1	1	1	0

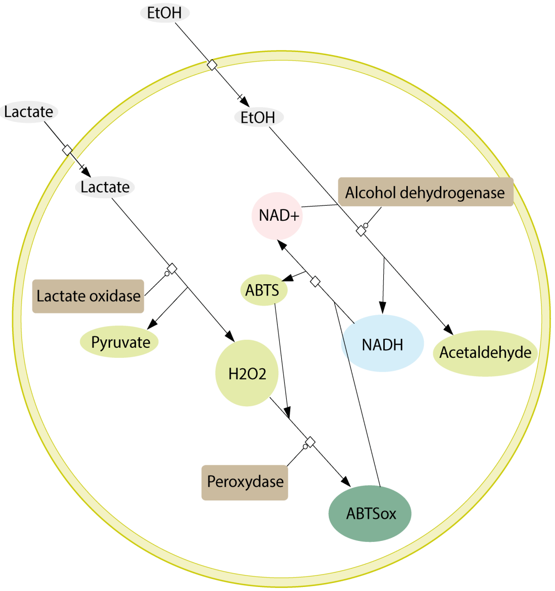
Reactions for Implementing Logical Gates

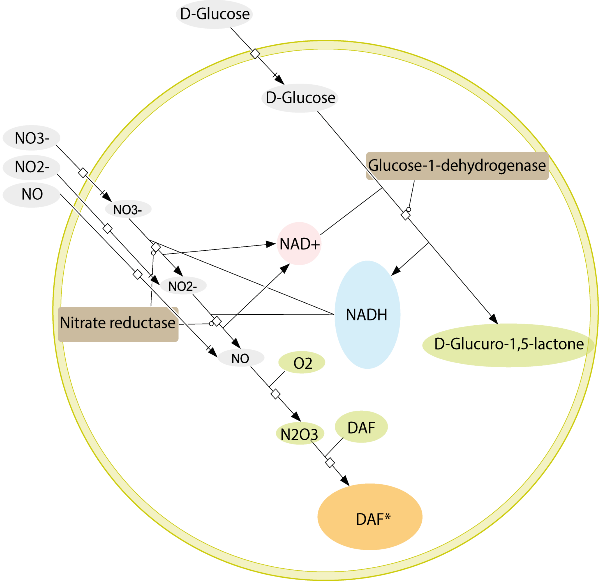
And C = A /\ B	A+B => C	[C] = min([A],[B])

Or C = A \/ B	A => C
	B => C
		[C] = [A]+[B]

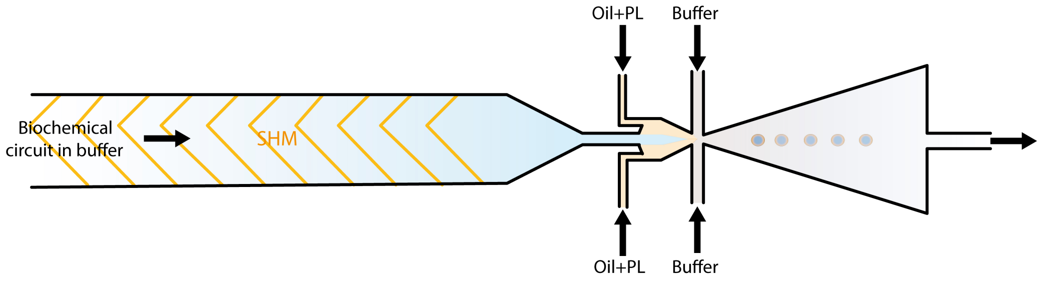
Not C = ¬ A	C+A => _	$[C] = \max([C_0]-[A], 0)$

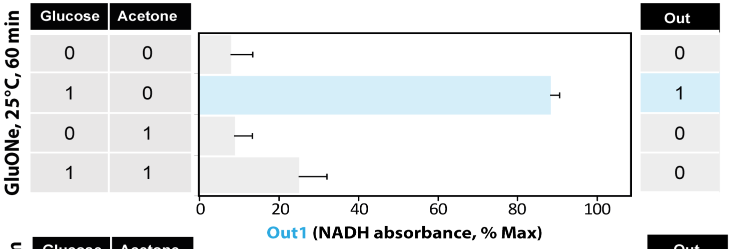




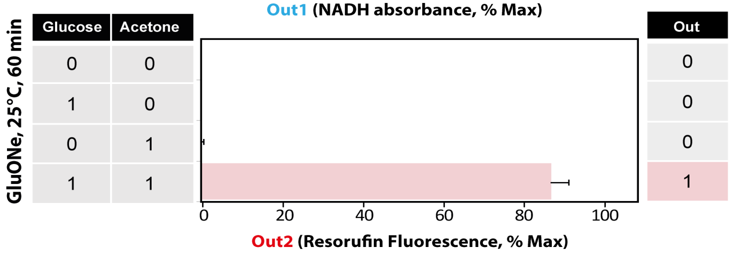


Microfluidic Assembly and Validation in Human Urine

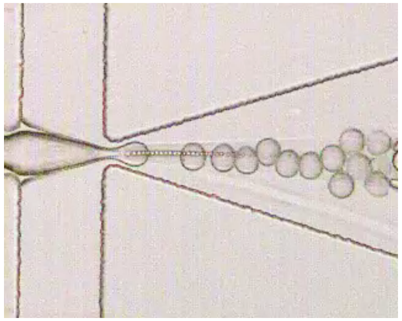


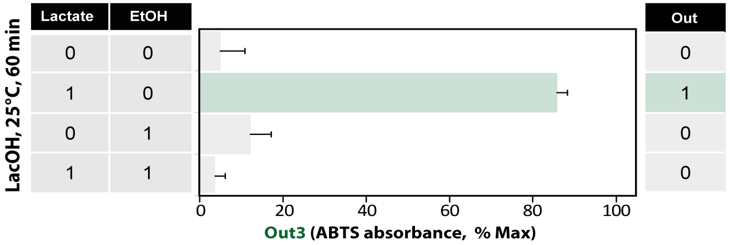


Glucose	Acetone	Out
0	0	0
1	0	1
0	1	0
1	1	0

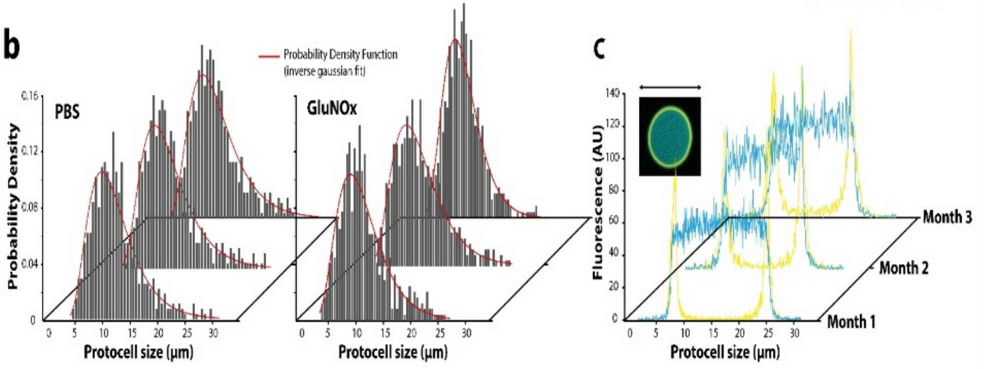


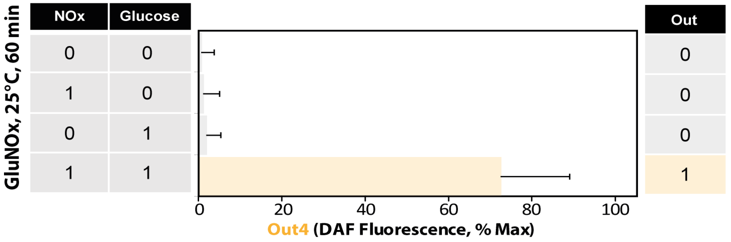
Glucose	Acetone	Out
0	0	0
1	0	0
0	1	0
1	1	1





Lactate	EtOH	Out
0	0	0
1	0	1
0	1	0
1	1	0





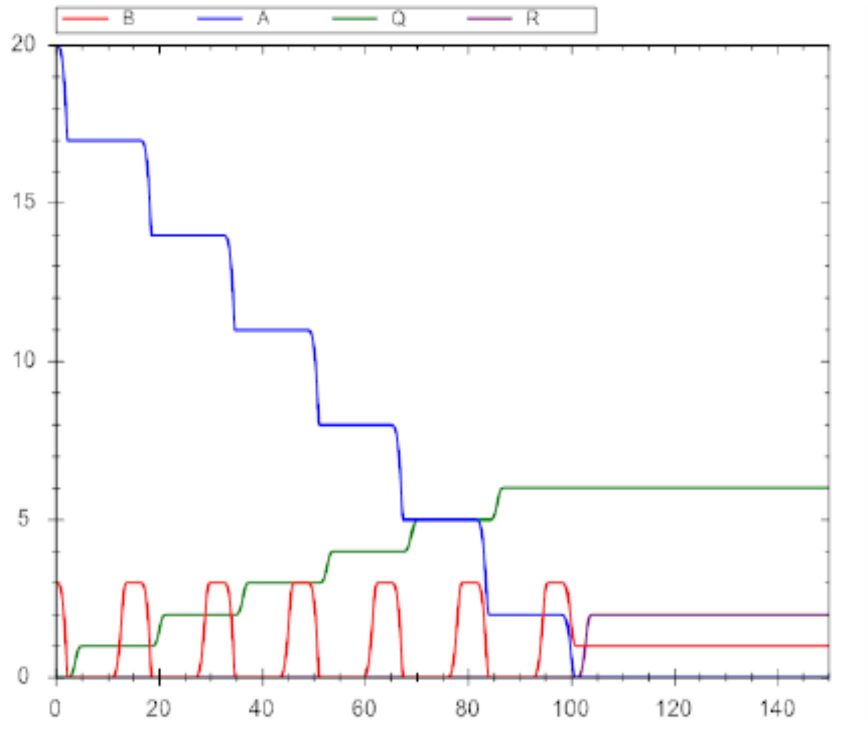
NOx	Glucose	Out
0	0	0
1	0	0
0	1	0
1	1	1

Sequentiality and Iteration

```
Division(A, B)
  begin
  01    while $A \geq B$
  02        $A := A - B$
  03        $Q := Q + 1$
  04    $R := A$
  end
```

Main Reactions		Preconditions
01	while $[A] \geq [B]$	
02	$(A + B \to D)$	$\neg G_\theta$
03	$\quad C \to Q + E$	$A_\theta \wedge \neg B_\theta$
04	$\quad D \to F$	$\neg C_\theta$
05	$\quad E \to G$	$\neg D_\theta$
06	$\quad F \to B$	$\neg E_\theta$
07	$\quad G \to C$	$\neg F_\theta$
08	$D \to R$	$\neg A_\theta$

[Huang Jiang Huang Cheng 2012 ICCAD]
[Huang Huang Chiang Jiang F 2013 IWBDA]



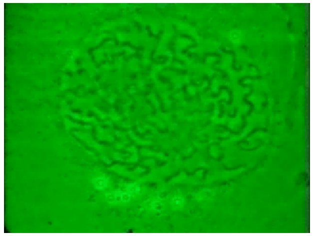
Cell Division Cycle Program

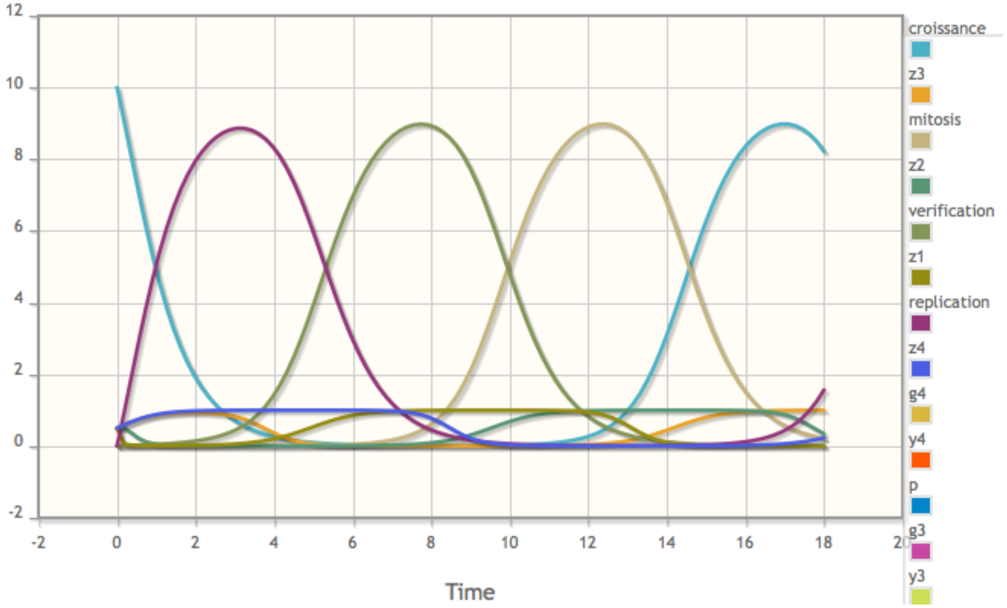
```
while true {growing; replication; verification; mitosis}
```

→ compilation of sequentiality and loops with program control variables

→ 50 reactions

→ 13 variables





Cyclins D, E, A, B appear as necessary markers for implementing sequentiality

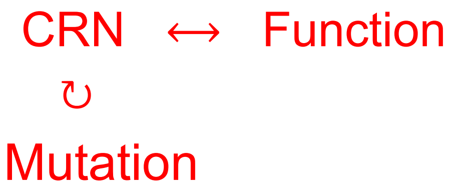
Conclusion

- Binary reaction systems over a finite set of molecules (without polymerization) are Turing-complete under the differential semantics
  - PIVP definition of computable function
  - Notion of computational complexity as trajectory length of stabilizing PIVPs

- Biochemical compiler of real functions (in Biocham modeling software)
  - Input: Function specification by PIVP, mixed digital-analog program
  - Output: system of binary reactions with mass action law kinetics
  - Exact characterization of the result for an ideal fluid implementation

- Comparison to natural circuits
  - The natural MAPK program implements an analog-digital converter (sigmoid Hill5)
  - Different from generated CRN for Hill5 but similar complexity

- Alternative design by artificial evolution:
  - Artificial evolution of CRNs
  - Nature algorithms for learning [Valliant 2013]


CRN ↔ Function
↻
Mutation