Analog computations in the cell with Chemical Reaction Networks: from Turing completeness result to questions of low complexity classes





Cells Compute:

- Cells process signals
- Regulate their metabolism
- Take decisions such as
 - Replication
 - Migration
 - Differentiation
 - Apoptosis (suicide)
- Control those processes





How do cells compute? Chemical Reaction Networks (CRN) Analog computation with proteins: gradual concentration levels, continuous time What are the links to Turing machines and digital computation? Can we understand, beyond describing, natural CRNs? (Systems Biology) Can we synthetize artificial CRN to implement a function? (Synthetic Biology) With which appropriate notions of computational complexity and robustness?

Kickoff ∂ifference 2020

Example: MAPK Signaling Cascade

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



Signal amplification is trivial with one enzymatic reaction $S + E \rightarrow P + E$ Why three levels of double phosphorylation ?



MAPK Input/Output Function

dose-response diagram alias bifurcation diagram alias functional specification biocham: dose response('E1', 1.0e-6, 1e-4, 200).



MAPK implements an analog/digital converter: ultrasensitivity, noise filtering How to program $\frac{x^n}{c+x^n}$ with biochemical reactions ? Can we implement any computable function with a CRN ?

Kickoff ∂ifference 2020

Differential semantics: concentrations, continuous time evolution

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







Differential 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--



Differential 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--



Differential 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 Structural invariants for ODEs

Boolean semantics: presence/absence Asynchronous transition system

$$A \land B \to C \land \neg A \land \neg B$$
$$A \land B \to C \land A \land \neg B$$
$$A \land B \to C \land \neg A \land B$$
$$A \land B \to C \land \neg A \land B$$

 $\begin{array}{ll} \mbox{Differential semantics: concentrations, continuous time evolution} \\ \mbox{Ordinary differential equations (ODE)} & \box{$\frac{dA}{dt}=-k.A.B$} & \box{$\frac{dB}{dt}=-k.A.B$} & \box{$\frac{dC}{dt}=k.A.B$} \\ \mbox{Stochastic semantics: numbers of molecules, probability and time of transition} \\ \mbox{Continuous Time Markov Chain (CTMC)} & A, B^{\rm p(S), t(SB)} & \box{C++, A-, B--} \\ \mbox{Petri net semantics: numbers of molecules A, B \to C++, A-, B--} \\ \mbox{Multiset rewriting} \\ \mbox{Structural invariants for ODEs} \\ \mbox{Boolean semantics: presence/absence} & A \land B \to C \land \neg A \land \neg B \\ \mbox{A synchronous transition system} & A \land B \to C \land \neg A \land B \\ \mbox{A } B \to C \land A \land B \\ \mbox{A } A \to C \land A \land B \\ \mbox{A } B \to C \land A \land B \\ \end{array}$

Kickoff ∂ifference 2020





Turing Completeness Results for CRNs



Computable Real Numbers and Functions

Computable Analysis: definitions based on Turing machines and arbitrary precision

Definition. A real number *r* is computable if there exists a Turing machine with <u>Input</u>: precision $p \in \mathbb{N}$ <u>Output</u>: rational number $q \in \mathbb{Q}$ with $|r-q| < 2^{-p}$

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

Definition. A real function $f:R \rightarrow R$ is computable if there exists a Turing machine that computes f(x) assuming one for x.

Counter-examples. x=0, [X] are not computable (undecidable on x=0.000...) discontinuous functions **Examples.** Polynomials, trigonometric functions, ...

Encoding of discrete decision problem: is input word in language $w \in L$? yes/no Real encoding of input e(w), function f, answer yes if $f(e(w)) \ge 2$, no if $f(e(w)) \le 1$ Kickoff difference 2020

CRN Implementation of GPAC Units

On positive values: molecular concentrations

Product unit z = x. y

Sum unit
$$z = x + y$$

Time integral $z = \int x dt$ unit

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

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

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

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

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

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

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

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

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

$$x \xrightarrow{x} x + z$$
$$\frac{dz}{dt} = x$$
$$z = \int_0^T x \, dt$$

Polynomial ODE Initial Value Problems (PIVP)

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

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





PIVP-Computable Functions f(x)

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



Theorem (analog characterization of computability). [Bournez Campagnolo Graça Hainry 2007]]
A real function is computable (by Turing machine) iff it is PIVP-computable.
Theorem (analog characterization of Ptime). [Bournez Graca Pouly 2016 ICALP]
A real function is in FP iff it is PIVP-computable with a trajectory of polynomial length (i.e. both polynomial time and space-amplitude)

Kickoff ∂ifference 2020

Turing Completeness of Continuous CRN

- Mass action law kinetics (for elementary reactions)
 - polynomial ODEs
 - PIVP computation of input/output function
- Molecular concentration are positive real values
 - Restriction to positive dynamical systems
- Elementary reactions with at most two reactants
 - Restriction to PIVP of degree at most 2



Restriction to Positive Systems

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 $y_i = y_{i}^{+} - y_{i}^{-}$ at each time (encoding used in [Oishi Klavins 2011] for linear I/O systems) Let $\underline{p}_i(y_{1}^{+}, y_{1}^{-}, ..., y_{n}^{+}, y_{n}^{-}) = p_i[y = y_{i}^{+} - y_{i}^{-}]$ and $\underline{p}_i = \underline{p}_{i}^{+} - \underline{p}_{i}^{-}$ $y_{i}^{+} = \underline{q}_{i}^{+} - f_i y_{i}^{+} y_{i}^{-}$ $y_{i}^{+}(0) = \max(0, y_i(0))$ $y_{i}^{-} = \underline{q}_{i}^{-} - f_i y_{i}^{+} y_{i}^{-}$ $y_{i}^{-}(0) = \max(0, -y_i(0))$ where $f_{i}^{-} = a_{i}^{-} - f_{i} y_{i}^{+} y_{i}^{-}$ $a_{i}^{-} = a_{i}^{-} - f_{i}^{-} - f_{i}$

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

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

$$\begin{array}{cccc} M_{i,j} & \stackrel{+}{\longrightarrow} & \stackrel{\mathbf{m}^{+}_{i,j}}{\longrightarrow} & \mathbf{y}^{+}_{i} + M_{i,j} & \stackrel{+}{\longrightarrow} \\ M_{i,j} & \stackrel{-}{\longrightarrow} & \stackrel{\mathbf{m}^{-}_{i,j}}{\longrightarrow} & \mathbf{y}^{+}_{i} + M_{i,j} & \stackrel{-}{\longrightarrow} \end{array}$$



Restriction to at most Binary Reactions

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

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

We have $y_1 = v_{1,0\dots,0}$, $y_2 = v_{0,1,0\dots,0}$,...

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

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

i.e. trade high dimension for low degrees.

> this algorithm may introduce an exponential number of variables in $O(d^n)$

Quadratization Problem

Theorem [Hemery, F, Soliman 2020] In the non-succinct representation (i.e. matrix of monomials as input) the nsQTDP (resp. nsQTP) is NP-complete (resp. NP-hard). Proof: by reduction of vertex set covering.

Conjecture In the succinct representation (symbolic expression as input), the QTDP is NEXP-complete.

Implemented in BIOCHAM using a MAXSAT sover

Example CRN generated by compilation of I/O function $\frac{x^5}{1+x^5}$ from ODE spec.

- MA(5.0) for i+it4x=>h+it4x
 MA(1.0) for x=>_
 MA(1.0) for 2*x=>tx+2*x
 MA(1.0) for tx=>_
 MA(3.0) for 2*tx=>t3x+2*tx
 MA(1.0) for t3x=>
 - MA(1.0) for ix=>_ MA(5.0) for it4x+ix=>it4x MA(4.0) for ix+t3x=>it4x+ix+t3x MA(1.0) for it4x=>_ MA(5.0) for 2*it4x=>it4x

Kickoff ∂ifference 2020



Turing Completeness of Continuous CRNs

Theorem [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)

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 which can be seen as a (digital) metaprogram

- No artificial construct (no polymers)
- Compatible with natural cells



Normal Form Theorem

Theorem (abstract CRN normal form)

A real function is computable if and only if it is computable by a system of elementary reactions of the form

 $_{z} = z$ or x = x+z or x+y = x+y+zplus annihilation reactions x+y = z all with mass action law kinetics

Realistic CRN:

- formal annihilations by complexations (e.g. in a stable inactive complex)
- formal syntheses by modifications (e.g. phosphorylation with kinases)

Concrete CRN: search mapping with real enzymes (e.g. Brenda database)

- Easier for CRN with rate independence property
- Robustness w.r.t. parameter perturbations (extrinsic noise)
- Robustness w.r.t. stochastic simulations (intrinsic noise)





d(fm)/dt = zm-k*fm*fp

d(zp)/dt = fm-k*zm*zp

d(zm)/dt = fp-k*zm*zp

fp(0)=1

Compilation of the Cosine(t) function

biocham: compile_from_expression(cos,time,f). d(fp)/dt = zp-k*fm*fp

- $zm \Rightarrow zm+fm$. $fm+fp \Rightarrow _.$
- fm => fm+zp.
- fp => fp+zm.

```
present(fp,1).
```





Compilation of the Cosine(x) Function

biocham: present(x_p, 4). biocham: compile_from_expression(cos,x,f).

presen	t(fp, 1).		
gm => g	Jm+gp.	gm+fm =>	gm+fm+zp.
xp => x	<p+gp.< td=""><td>gp+fp =></td><td>gp+fp+zp.</td></p+gp.<>	gp+fp =>	gp+fp+zp.
gp => g	gp+gm.	xp+fm =>	xp+fm+zp.
xm => x	cm+gm.	xm+fp =>	xm+fp+zp.
gm+zp =	=> gm+zp+fp.	gm+fp =>	gm+fp+zm.
gp+zm =	=> gp+zm+fp.	gp+fm =>	gp+fm+zm.
xm+zm =	=> xm+zm+fp.	xm+fm =>	<pre>xm+fm+zm.</pre>
xp+zp =	=> xp+z+fp.	xp+fp =>	xp+fp+zm.
gm+zm =	=> gm+zm+fm.	xp+fp =>	xp+fp+zm.
gp+zp =	=> gp+zp+fm.	<pre>xm+zp =></pre>	<pre>xm+zp+fm.</pre>

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

$$g'(t) = x - g(t)$$

$$g(t) = x + (x0 - x)e^{-t}$$



Logical Gates

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

Or $C = A \lor B$ A => C [C] = [A]+[B]B => C

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



Example of Rate-Independent CRN



ODE da/dt=-k1.a db/dt=-k2.b dx/dt=k1.a-k3.x.y dy/dt=k2.b-k3.x.y dz/dt=k3.x.y-k4.c.z dr/dt=k4.c.z dc/dt=k1.a+k2.b-k4.c.z

Initialization: x=y=z=r=c=0

Input: a, b

Output: c(~)= max(a(0),b(0)) independently of the reaction rates



Rate-Independent CRN





The I/O function computed by that CRN structure is independent of the kinetics

a => x+c

b => **y**+**c**

x+y => z

c+z => r

 $c^{*} = \max(a(0), b(0)) = a(0)+b(0)-\min(a(0), b(0))$ $x^{*} = \max(0, a(0)-b(0))$ $r^{*} = \min(a(0), b(0))$ $z^{*} = 0, a^{*} = 0, b^{*} = 0$ Absolute robustness Ideal circuit designs Ideal circuit biology

CMSB 2020



Theorem [Chen-Doty-Soloveichik 2014 ITCS]

A real function is computable by a rate-independent CRN if and only if it is positivecontinuous piecewise linear with rational coefficient.

Theorem [Chalk Kornerup Reeves Soloveichik 2018 CMSB] A real function is computable by a composable CRN if and only if it is superadditive positive-continuous piecewise rational linear.

Theorem [Degrand F Soliman 2020] A funnel CRN (i.e. fork-free on species, *circuit-free and synthesis-free*) is rate-independent for any output species.

Theorem. Any CRN is rate-independent on its structurally persistent products (i.e. $\forall i R_i(x) \leq P_i(x)$, covered by a Petri-net P-invariant S, $\forall i S. R_i = S. P_i$, and not belonging to a critical siphon)



```
Better design with
finite difference
approach?
```

Sequentiality and Iteration

1. Asynchronous (precondition) CRN programming

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

Pb many species and reactions

2. Synchronous (clock) CRN programming

[Vasic, Soloveichik, Khurshid 2018 CRN++]



Cell Division Cycle Program

while true {growing; replication; verification; mitosis}

 \rightarrow compilation of sequentiality and loops with program control variables



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

Plan of the Talk

- 1. Chemical Reaction Networks (CRN)
 - Syntax: formal reactions with rate functions
 - Semantics: differential, stochastic, discrete, boolean hierarchy
 - Computed input/output functions (chemical computer)
- 2. Turing Completeness of finite CRNs with differential semantics
 - analog computation
 - protein analog programs and genetic digital metaprogram
- 3. Abstract CRN compiler: oscillators, switches, logical circuits, sequentiality
- 4. Concrete CRN implementation in DNA-free vesicles
 - study case: design of diagnosis biosensors [Courbet Molina et al 2018]
- 5. Conclusion on CRN design workflows

Kickoff ∂ifference 2020



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

Alexis Courbet¹, Patrick Amar², François Fages³, Eric Renard⁴, Franck Molina¹

- ¹ Sys2diag UMR9005 CNRS/ALCEDIAG, Montpellier
- ² LRI, Université Paris Sud UMR CNRS 8623, Orsay
- ³ <u>http://lifeware.inria.fr</u>, Inria Saclay IdF, Palaiseau
- ⁴ INSERM 1411, Montpellier University Hospital



Biosensor CRN Design Workflow [F. Molina lab Montpellier, Mol. Sys. Biol. 2018]





Differential Diagnosis Algorithm





CRN Schema for Logical Gates



Robustness optimization by parameter search for initial concentrations w.r.t. quantitative temporal logic properties [Rizk et al. 2009, 2011]

Kickoff ∂ifference 2020



Microfluidic Building and Test in Human Urine



Kickoff ∂ifference 2020

Informatics mathematics

CRN Design Workflows

Quantitative Temporal Logic Workflow [Rizk Batt F Soliman, Bioinformatics 2009, TCS 2011]

Input: 1. CRN structure (with or without kinetic parameters)

2. Behavior specification with FO-LTL(\mathbb{R}_{lin}) formulae

- \rightarrow Verification with continuous satisfaction degree in [0,1]
- → Sensivity, robustness wrt parameter perturbations
- → Parameter search by continuous optimization (formula robustness optimization)

Compilation Workflow [F Le Guludec Bournez Pouly CMSB 17, Hemery F Soliman CMSB 2020 Degrand F S 20]

Input: Real valued function specified as solution of a PIVP

- \rightarrow CRN structure with kinetic parameters (quadratized)
- → Rate-independence property as low analog complexity class?

Artificial Evolution Workflow [Degrand Hemery F CMSB 19] Genetic algorithm (GA) with nested parameter optimization (CMA-ES)

 \rightarrow Unexpected designs for oscillators and sigmoids

Kickoff ∂ifference 2020

Questions investigated in ∂ ifference

- Finite difference semantics of CRNs
- Relationship to ODE, stochastic and Petri net semantics
- CRN design methods based on difference semantics
 - Implementation of sequentiality
- Rate independence property
 - Expressive power
 - Sufficient conditions
- Low computational complexity classes
- Examples (repository BioModels of 2.10³ CRN hand made models)

Kickoff ∂ifference 2020



Lifeware team at INRIA Saclay

François Fages (Research Scientist) Sylvain Soliman (Research Scientist) Mathieu Hemery (PostDoc)

Éléonore Bellot (PhD Inria) Marine Collery (IBM) Eléa Greugny (Cifre Johnson&Johnson) Jeremy Grignard (Cifre Servier) Julien Martinelli (PhD INSERM-Inria)

Current collaborations:

- O. Bournez, Ecole Polytechnique
- F. Molina CNRS Sys2diag, Montpellier, J.H. Jiang, NTU, Taiwan
 - ANR-MOST BIOPSY project: Biochemical Programming Systems
- A. Weber, Univ. Bonn, T. Sturm, Nancy, O. Radulescu, Montpellier, S. Walcher
 ANR-DFG SYMBIONT project: Symbolic Methods for Biological Networks
- A. Niarakis, Univ. Evry
- A. Ballesta, INSERM Villejuif

Kickoff ∂ifference 2020



