BIO-INSPIRED COMPUTING:

Achievements and Dreams

Gheorghe Păun
Romanian Academy, Bucharest
Sevilla University, Spain
george.paun@imar.ro, gpaun@us.es

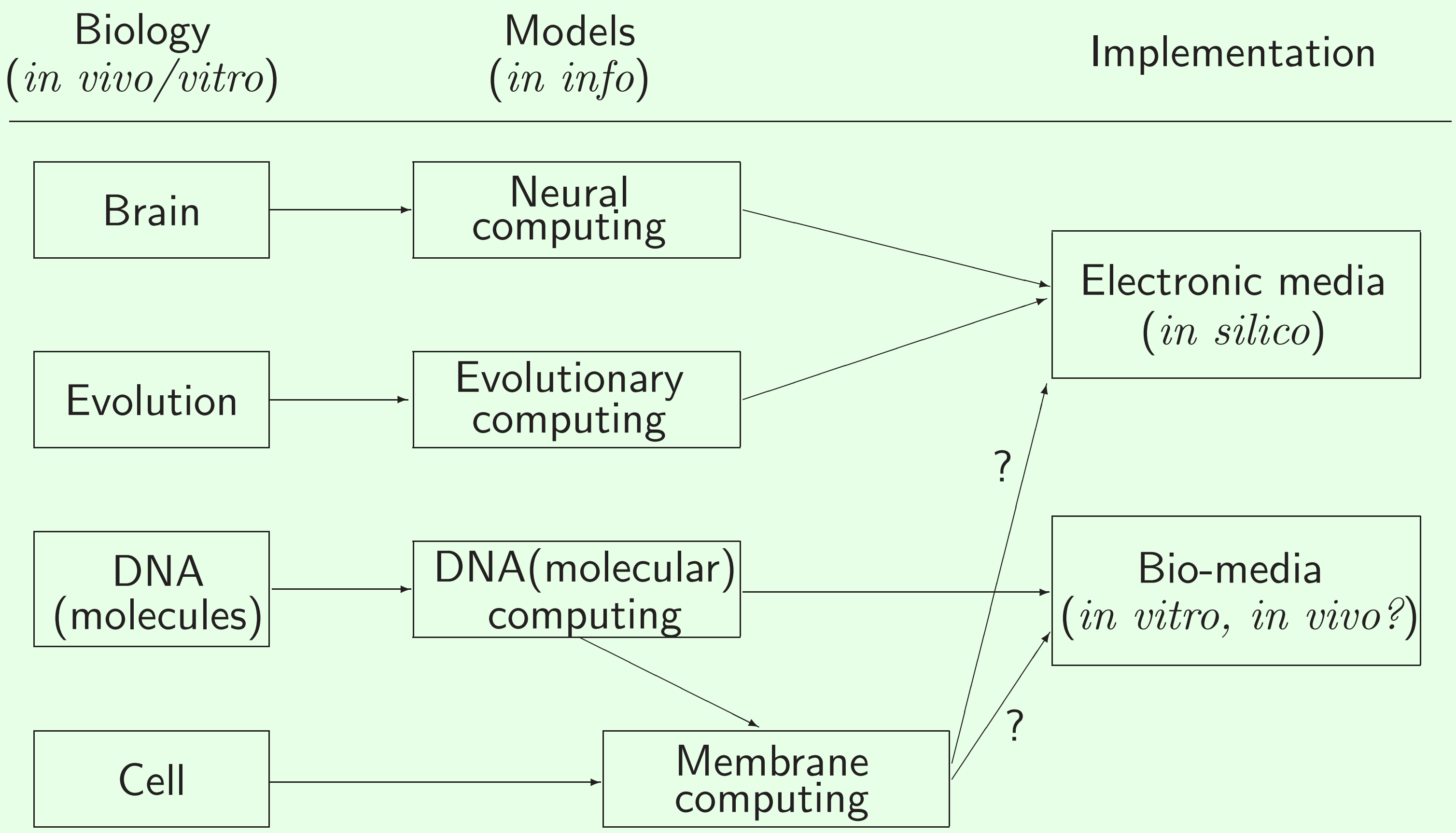
curteadelaarges@gmail.com

http://ppage.psystems.eu

Summary:

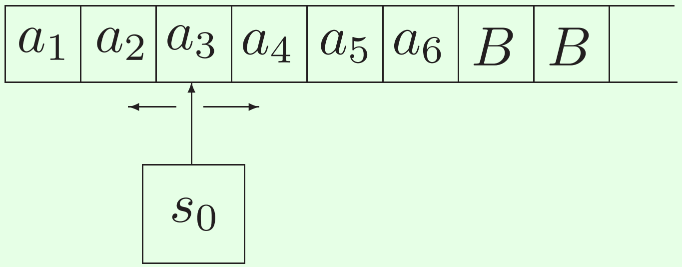
- generalities about natural computing
  1. what is a computation?
  2. does nature compute?
  3. why natural computing? (or, the limits of current computers)
  4. "everything" goes back to Turing
  5. a (simulated) wondering: why GA are so good?
  6. what means to compute "in a natural way"?
- DNA computing:
  1. history (old times, T. Head, L. Adleman)
  2. the marvelous DNA molecule
  3. computing by splicing
  4. DNA computing today
- membrane computing:
  1. the general idea
  2. (types of) results
  3. (types of) applications
- at the edge of science-fiction
- do we dream too much?

(Basic)Bio-inspired areas of natural computing



What is a computation?

Here (math.): Turing computability (input–output, algorithm, universal TM)



If "computation = information processing", then "everything is a computation".

Does a falling drop of water solves differential equations?

Ciliates: list processing before McCarthy...

Computation = a process which is considered computation by an observer

T. Toffoli: Nothing makes sense in computing except in the light of evolution. *Int. J. of Unconventional Computing*, 1 (2005), 3–29.

It is not useful to call 'computation' just any nontrivial yet somewhat disciplined coupling between state variables. We also want this coupling to have been *intentionally* set up for the purpose of predicting or manipulating – in other words, for *knowing* or *doing* something. (...) *The concept of computation must emerge as a natural, well–characterized, objective construct, recognizable by and useful to humans, Martians and robots alike*" (my emphasis, Gh.P.).

Does nature computes?

Here, nature = bio-entities, excluding *homo sapiens* (and Martians...)

YES/NO, depending on the definition – see above

My position: the goal of life is living, not computing (but we can use ideas, data structures, operations, architectures, strategies, paradigms, materials, etc. from biology for computing – improving the existing computers or imagining new ones).

The limits of current computers

Complexity (tractable/**P** versus intractable/**NP**), information storage capacity, energy efficiency, adaptability/learnability, robustness/self-healing, etc.

Beware of exponentials...:
problem Q of complexity 3^n, computer today solving $Q(100)$, computer 1000 times faster $\Longrightarrow$ solving (in the same time) $Q(106)$

P$\neq$**NP** is the first Millennium Prize Problem of Clay Mathematics Institute (`www.claymath.org`)

...and, of course, the computer does not play GO/weichi

Why natural computing?

1. limits of Turing – von Neumann paradigm
2. limits of silicon (Moore law, communication complexity)
3. adaptation, learning, self-healing, robustness, nondeterminism, etc.
4. energy efficiency
5. mathematical/computability challenge
   (what means to compute *in a natural way?*)
6. by-products (biology, medicine)
7. why not?

“Everything” goes back to Turing: http://www.AlanTuring.net

Turing Machine = abstraction of what a bank clerk is doing when computing

A. Turing: *Intelligent machinery*, 1948 (published in 1968: sir Charles Darwin considered it a “schoolboy essay”...) – (randomly connected) nets of neurons, hence neural computing

Also in 1948: “genetical or evolutionary search”, hence evolutionary computing (later, Holland, Bremermann, Fogel, Owens, Walsh – in sixties, Koza, etc., etc.).

Turing also imagined ways to “compute beyond Turing” (hypermachines, oracles) – PhD thesis, Princeton, 1938 (and still, “forgotten”...)

Why genetic algorithms are so good?

"When you do not know the right direction, walk randomly"...
(random here = crossovering, mutation, selection, etc.)

"Bio-mystical" answer: because life has used and improved the respective ingredients for billions of years

Optimistic consequence for all branches of natural computing

What means to compute "in a natural way"?

Which data structures (strings? DNA-like double sequences? multisets?), which operations (local rewriting? splicing? multiset processing?), which types of devices (sequential? parallel? synchronous/asynchronous? distributed? halting?), etc., etc.

Interesting mathematical problems (e.g., are Turing, Chomsky, Lindenmayer, etc. levels of computability *natural*? can computability be reconstructed on other bases?)

DNA computing

$$
\begin{array}{lcr}
5' & & 3' \\
\hline
 & \mathsf{C\ G\ G\ A\ T\ A\ A\ T\ G\ G\ A\ C\ C} & \\
 & \mathsf{G\ C\ C\ T\ A\ T\ T\ A\ C\ C\ T\ G\ G} & \\
\hline
3' & & 5'
\end{array}
$$

- Bennett, Conrad, etc.
- T. Head (1987) – splicing
- L. Adleman (1994) – solving HPP in a lab
- E. Winfree – self-assembly (Wang Hao tiles)
- coding, nano-technology, scaling-up, errors

Intrinsic computational universality – via twin-shuffle languages:

$V, \bar{V}, \bar{x}, \sqcup\!\sqcup$

$$TS_V = \bigcup_{x \in V^*} (x \sqcup\!\sqcup \bar{x})$$

Theorem [Engelfriet, Rozenberg 80]: $L \in RE, L \subseteq T^*$, $L = pr_T(TS_V \cap R)$

Corollary: $L \in RE, L = gsm_L(TS_{\{0,1\}})$

	upper strand	lower strand
A, T	0	$\bar{0}$
C, G	1	$\bar{1}$

$\Longrightarrow$ DNA $\approx TS_{\{0,1\}}$

Bidirectionality? – no problem: reverse twin-shuffle:

$$RTS_V = \bigcup_{x \in V^*} (x \sqcup\!\sqcup \, mi(\bar{x}))$$

Theorem [Engelfriet 96]: $L \in RE, L \subseteq T^*$, $L = pr_T(RTS_V \cap R)$

Corollary: $L \in RE, L = gsm_L(RTS_{\{0,1\}})$

All nucleotides necessary? – not: (reverse) semi-twin-shuffle

$$\text{complementarity}: c(0) = \bar{0}, \quad c(1) = 1$$

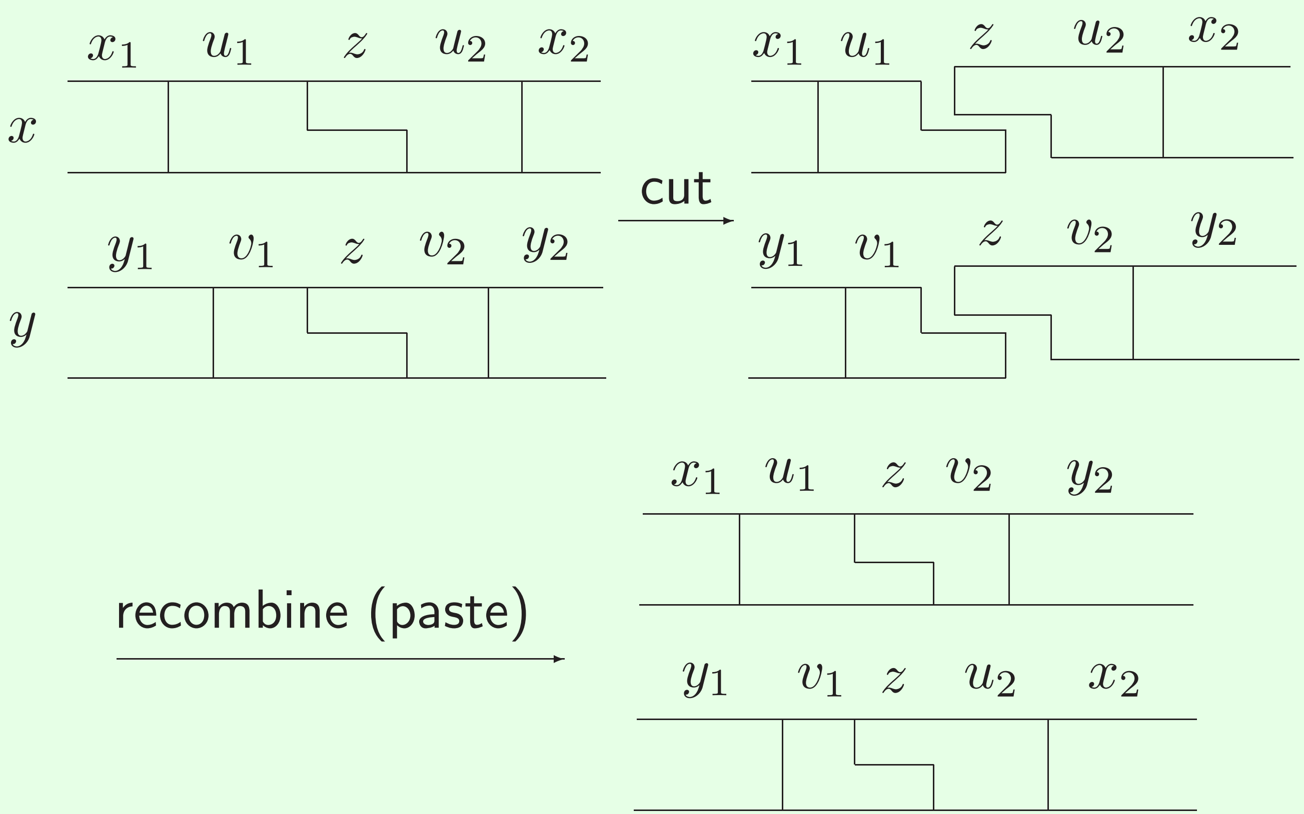
$$STS_V = \bigcup_{x \in V^*} (x \sqcup\!\sqcup c(x))$$

$$RSTS_V = \bigcup_{x \in V^*} (x \sqcup\!\sqcup mi(c(x)))$$

Theorem [DNA book]: $L \in RE, L = gsm_L([R]STS_{\{0,1\}})$

Gh. Păun, G. Rozenberg, A. Salomaa,
DNA Computing. New Computing Paradigms,
Springer 1998; Tokyo 1999; Tsinghua Univ. Press, Beijing, 2004; Mir, Moskow, 2005

Computing by splicing



Splicing rule (over V, $\# \notin V$): $r = u_1 \# u_2 \$ v_1 \# v_2$

Operation:

$$(x, y) \vdash_r (w, z) \quad \text{iff} \quad x = x_1 u_1 u_2 x_2, \; y = y_1 v_1 v_2 y_2,$$
$$w = x_1 u_1 u v_2 y_2, \; z = y_1 v_1 u_2 x_2, \quad x_1, x_2, y_1, y_2 \in V^*.$$

Splicing system/H system: $\gamma = (V, T, A, R)$

Language generated: $L(\gamma) = \{x \in T^* \mid A \Longrightarrow_{\text{iterated splicing}} x\}$

[Culik II, Harju, 1991], [Pixton 1995], [Manca 2000] Only regular languages

Universality, using controls, distribution (membranes!) [Păun, 1995, 96, 97...]
(not very realistic...)

DNA computing today:

- no significant computability application reported (cryptography?), but
- important mathematical developments
- nano-technology
- medicine (speculations: curing robots)
- computing in vivo (ciliates)

Difficulties: trading space for time (Hartmanis), coping with errors (nondeterminism)

Membrane computing

Goal: abstracting computing models/ideas from the structure and functioning of living cells (and from their organization in tissues, organs, organisms)

hence not producing models for biologists (although, this is now a tendency)

result:

- distributed, parallel computing model
- compartmentalization by means of membranes
- basic data structure: multisets (but also strings; recently, numerical variables)

WHY?

– the cell exists! (challenge for mathematics)

– biology needs new models (discrete, algorithmic; system biology, the whole cell modelling/simulating)

– computer science can learn (e.g., parallelism, coordination, data structure, architecture, operations, strategies)

– computing in vitro/in vivo ("the cell is the smallest computer")

– distributed extension of molecular computing

– a posteriori: power, efficiency ("solving" NP-complete problems)

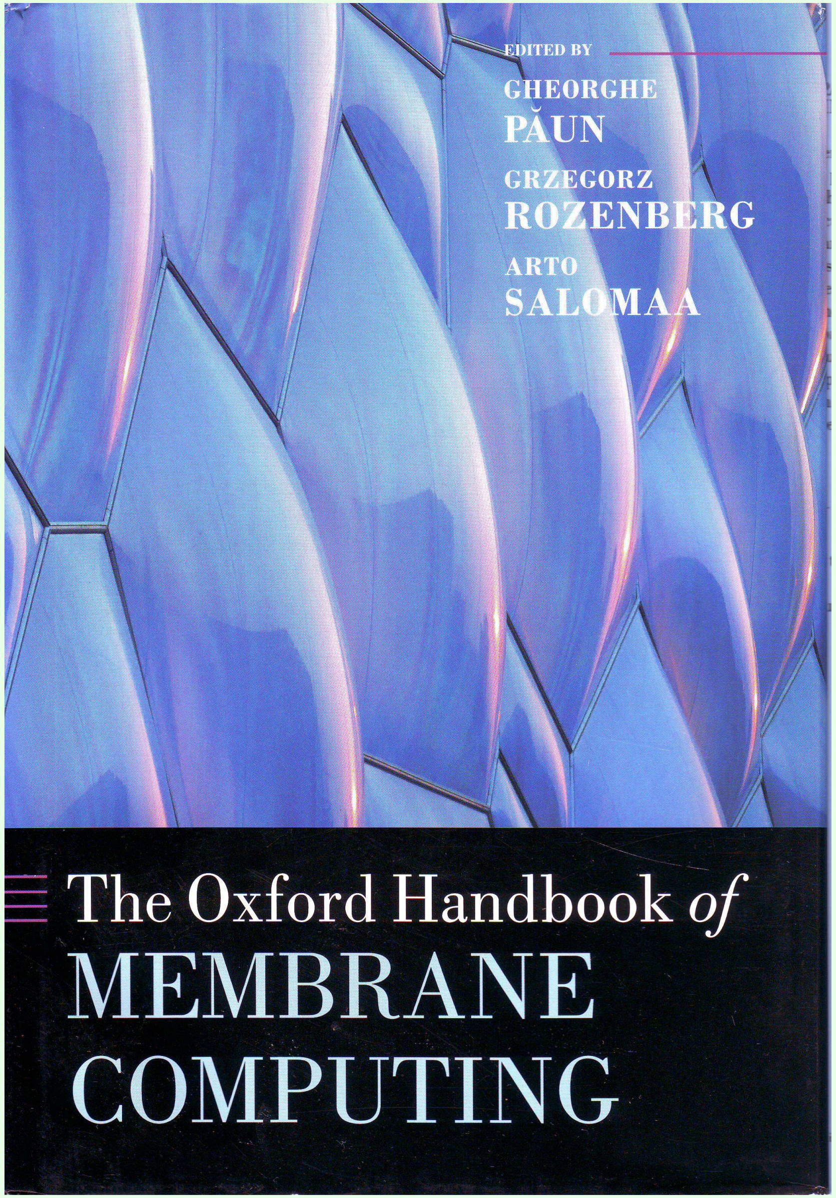
– a posteriori: applications in biology, computer graphics, linguistics, economics, etc.

– nice mathematical/computer science problems

References:

- Gh. Păun, Computing with Membranes. *Journal of Computer and System Sciences*, 61, 1 (2000), 108–143, and *Turku Center for Computer Science-TUCS Report* No 208, 1998 (www.tucs.fi)
  ISI: "fast breaking paper", "emerging research front in CS" (2003) http://esi-topics.com
- Gh. Păun, *Membrane Computing. An Introduction*, Springer, 2002
- G. Ciobanu, Gh. Păun, M.J. Pérez-Jiménez, eds., *Applications of Membrane Computing*, Springer, 2006
- Gh. Păun, G. Roznberg, A. Salomaa, eds., *The Oxford Handbook of Membrane Computing*, 2010
- Website: http://ppage.psystems.eu

(Yearly events: BWMC (February), WMC (summer), TAPS/WAPS (fall))


EDITED BY
GHEORGHE
PĂUN
GRZEGORZ
ROZENBERG
ARTO
SALOMAA
The Oxford Handbook of
MEMBRANE
COMPUTING

SOFTWARE AND APPLICATIONS:

http://ppage.psystems.eu

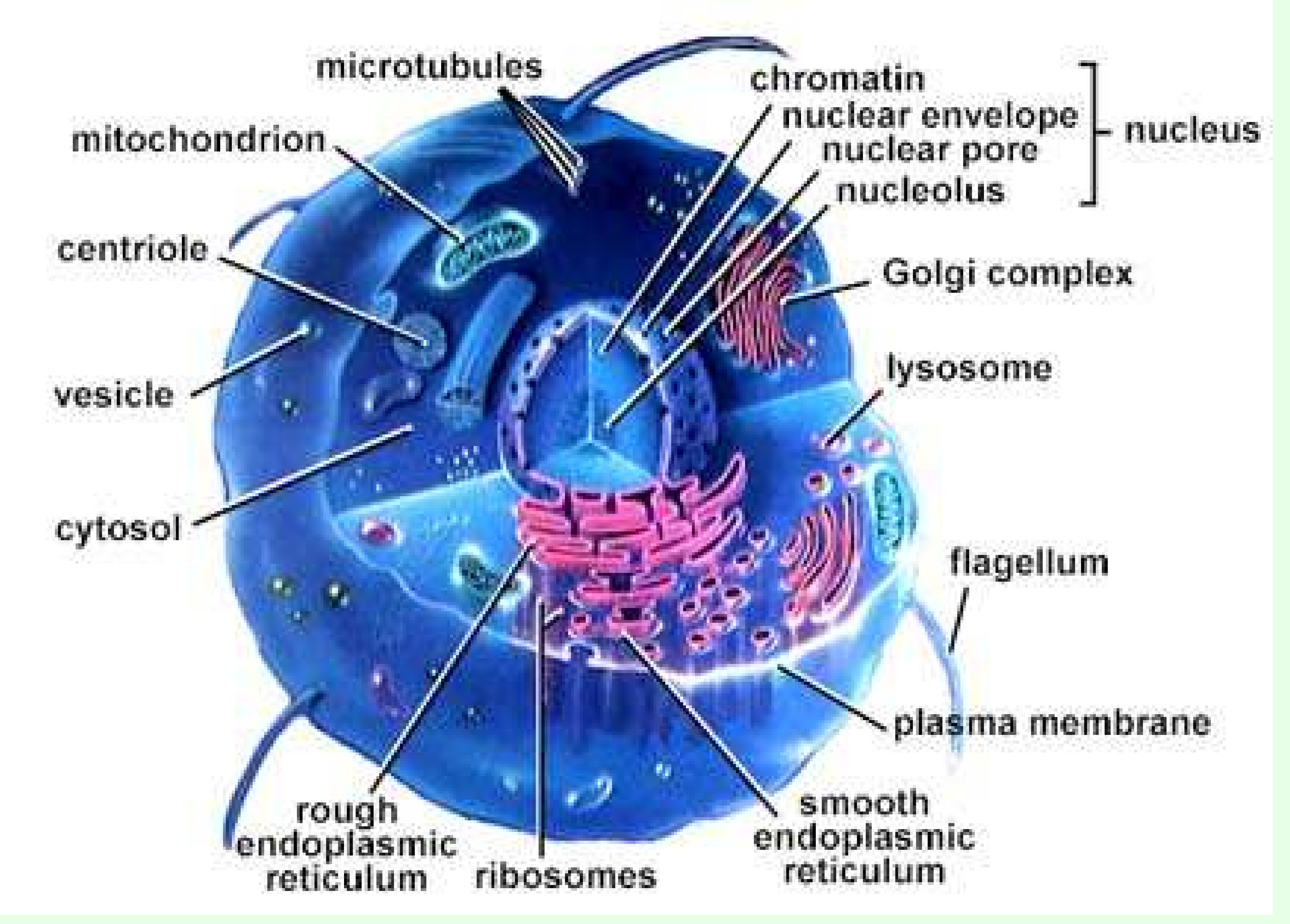
Verona (Vincenzo Manca: vincenzo.manca@univr.it)

Sheffield (Marian Gheorghe: M.Gheorghe@dcs.shef.ac.uk)

Sevilla (Mario Pérez-Jiménez: marper@us.es)

Milano (Giancarlo Mauri: mauri@disco.unimib.it)

Nottingham, Trento, Nagoya, Leiden, Vienna, Evry, Iași, Madrid


microtubules
mitochondrion
chromatin
nuclear envelope
nuclear pore
nucleolus
nucleus
centriole
Golgi complex
vesicle
lysosome
cytosol
flagellum
plasma membrane
rough
endoplasmic
reticulum
ribosomes
smooth
endoplasmic
reticulum

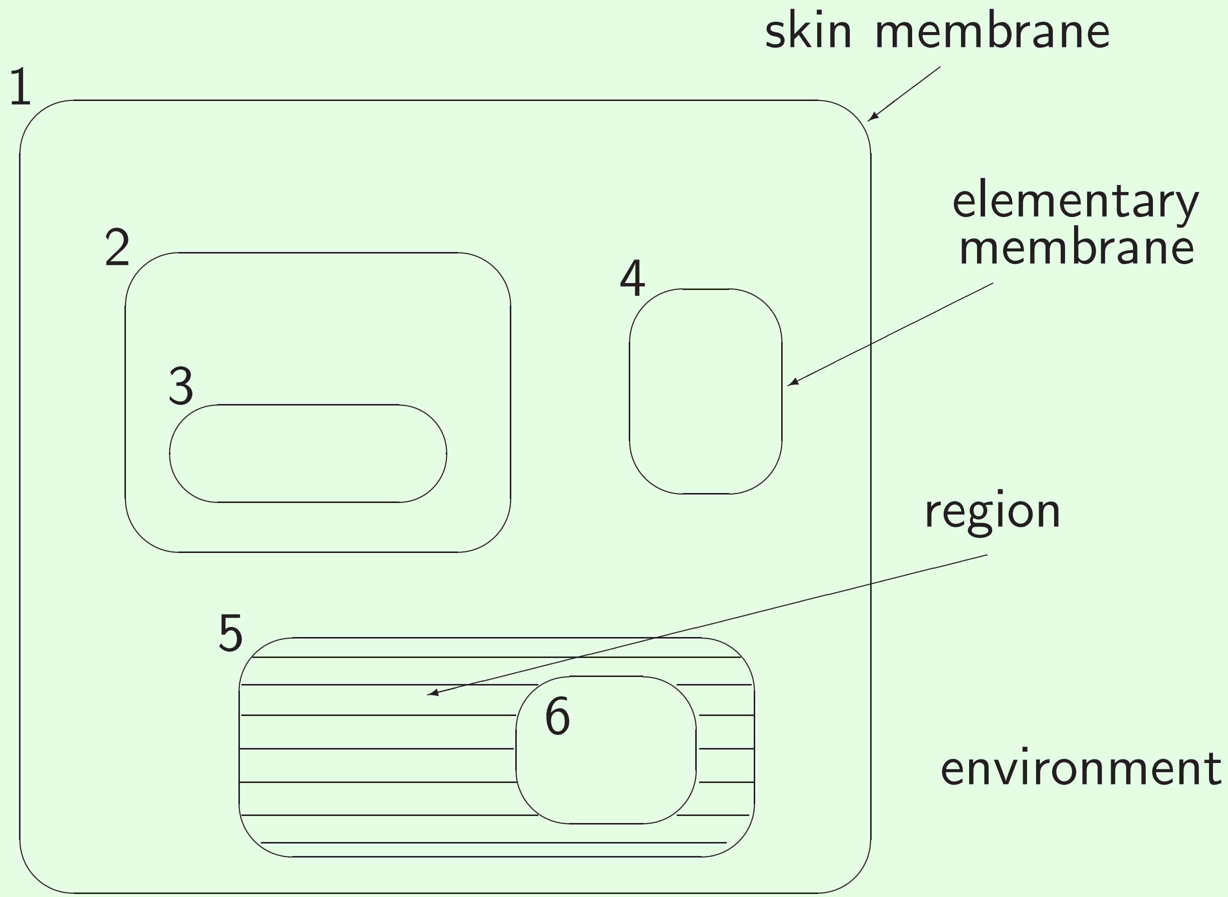
WHAT IS A CELL? (for a mathematician)

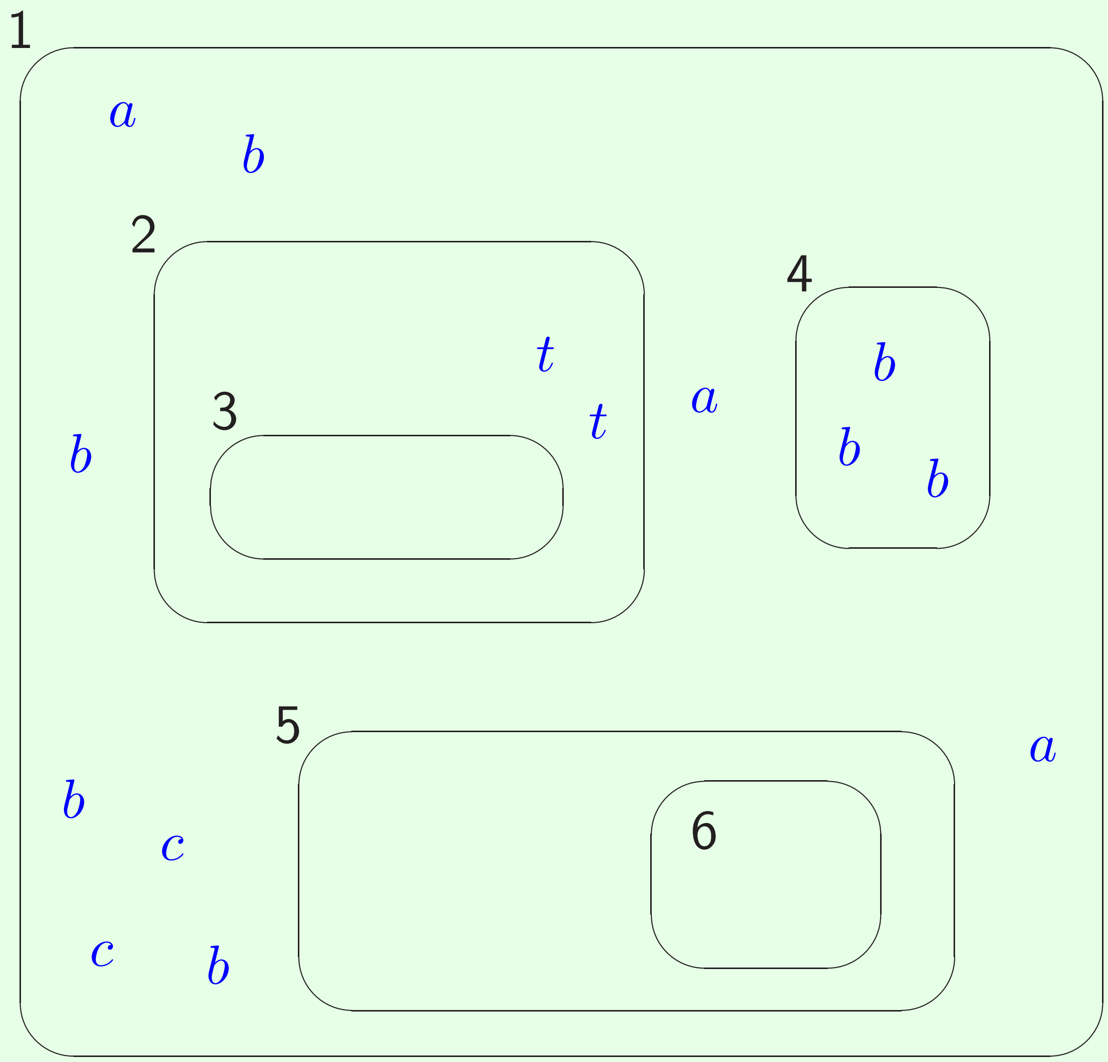
- membranes, separating "inside" from "outside" (hence protected compartments, "reactors")
- chemicals in solution (hence multisets)
- biochemistry (hence parallelism, nondeterminism, decentralization)
- enzymatic activity/control
- selective passage of chemicals across membranes
- etc.

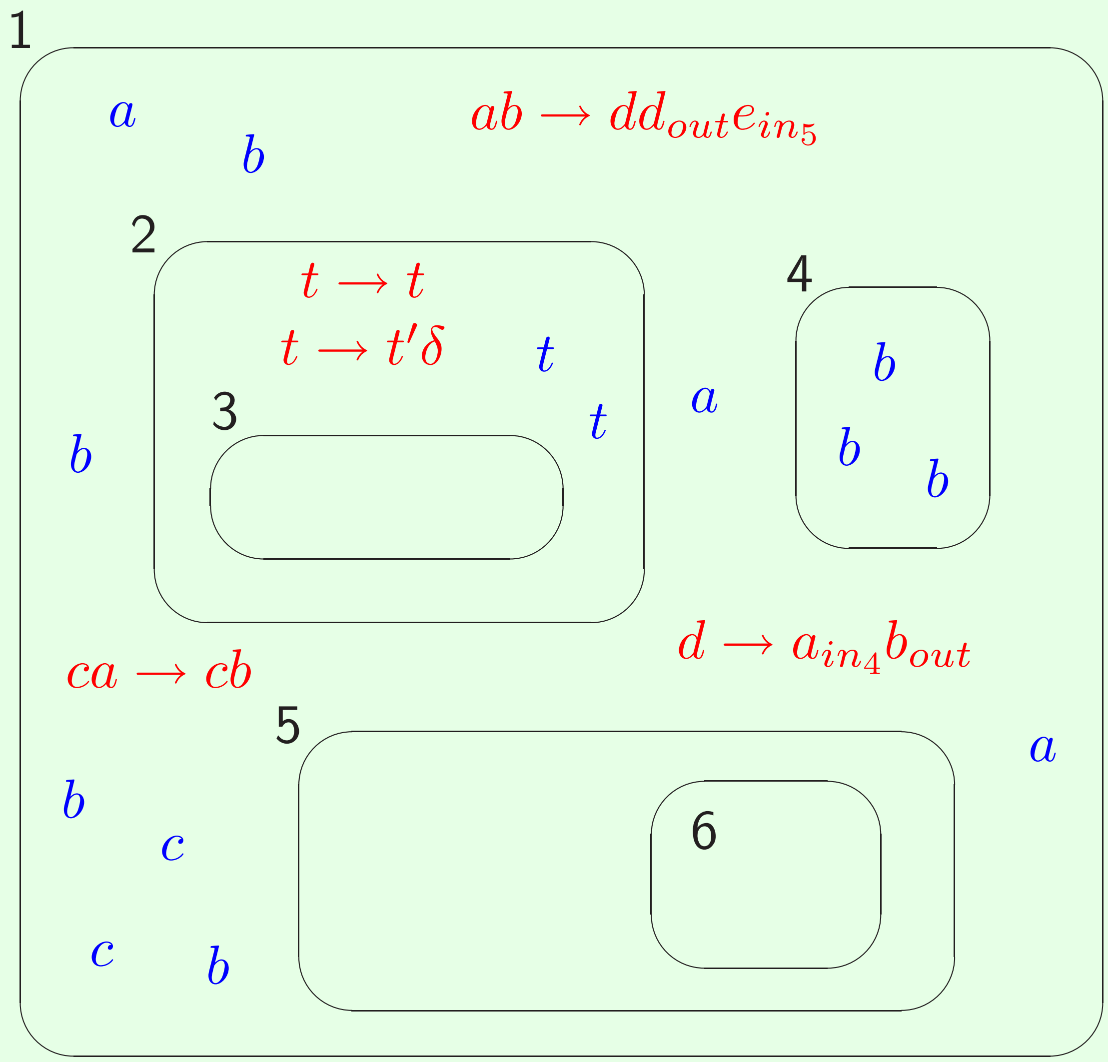
Importance of membranes for biology:. . .

MARCUS: $Life = DNA\ software + membrane\ hardware$

THE BASIC IDEA




1
a
b
2
t
3
t
a
4
b
b
b
b
5
a
b
c
6
c
b


1
a
b
ab → dd_out e_in5
2
t → t
t → t'δ
t
t
3
a
4
b
b
b
b
ca → cb
d → a_in4 b_out
5
a
b
c
6
c
b

A bird eye view to the MC jungle:

- cell-like, tissue-like, neural-like (spiking neural) systems
- symbols, strings, arrays, numerical variables, etc.
- multisets, sets, fuzzy
- multiset rewriting, symport/antiport, membrane evolving, combinations
- controls: priority, promoters, inhibitors, δ, τ, activators,
- maximal, bounded, minimal parallelism, sequential/asynchronous, time-, clock-free
- generating, accepting, computing/translating, dynamical system
- computing power, computing efficiency, others
- colonies
- implementations/simulations
- applications: biology/medicine, economics, optimization, computer graphics, linguistics, computer science, cryptography, etc.
- etc. (e.g., brane-membrane bridge, quantum-like)

A bird eye view to the MC jungle:

- cell-like, tissue-like, neural-like (spiking neural) systems
- symbols, strings, arrays, numerical variables, etc.
- multisets, sets, fuzzy
- multiset rewriting, symport/antiport, membrane evolving, combinations
- controls: priority, promoters, inhibitors, δ, τ, activators, etc.
- maximal, bounded, minimal parallelism, sequential/asynchronous, time-, clock-free
- generating, accepting, computing/translating, dynamical system
- computing power, computing efficiency, others
- colonies
- implementations/simulations
- applications: biology/medicine, economics, optimization, computer graphics, linguistics, computer science, cryptography, etc.
- etc. (e.g., brane-membrane bridge, quantum-like)

Results:

- characterization of Turing computability (RE, NRE, $PsRE$)
  Examples: by catalytic P systems (2 catalysts) [Sosik, Freund, Kari, Oswald]
  by (small) symport/antiport P systems [many]
  by spiking neural P systems [many]
- polynomial solutions to NP-complete problems (by using an exponential workspace created in a "biological way": membrane division, membrane creation, string replication, etc.) [Sevilla team], [Madras team], [Obtulowicz], [Alhazov, Pan] etc. even characterizations of **PSPACE**
- other types of mathematical results (normal forms, hierarchies, determinism versus nondeterminism, complexity) [Ibarra group]
- connections with ambient calculus, Petri nets, X-machines, quantum computing, lambda calculus, brane calculus, etc. [many]
- simulations and implementations
- applications

Open problems, research topics:

Many: see the P page

- borderlines: universality/non-universality, efficiency/non-efficiency (local problems: the power of 1 catalyst, the role of polarizations, dissolution, etc. general problems: uniform versus semi-uniform, deterministic-confluent, pre-computed resources)
- semantics (events, causality, etc.)
- neural-like systems (more biology, complexity, applications, etc.)
- user friendly, flexible, and efficient (!) software for bio-applications
- MC and economics
- implementations (electronics, bio-lab)
- finding a killer-app

SFAQ:

- computing beyond Turing? (no, but ...acceleration)
- what kind of implementation? (none, but ...Adelaide, Madrid, Technion-Haifa)
- why so many variants?
- why so powerful? (RE = CS + erasing)

Applications:

- biology, medicine, ecosystems (continuous versus discrete mathematics) [Sevilla, Verona, Milano, Sheffield, Ruston-Lousiana, Trento, etc.]

- computer science (computer graphics, sorting/ranking, 2D languages, cryptography, general model of distributed-parallel computing) [many]

- linguistics (modeling framework, parsing) [Tarragona]

- optimization (membrane algorithms [Nishida, 2004], [many])

- economics ([Warsaw group], [R. Păun], [Vienna group])

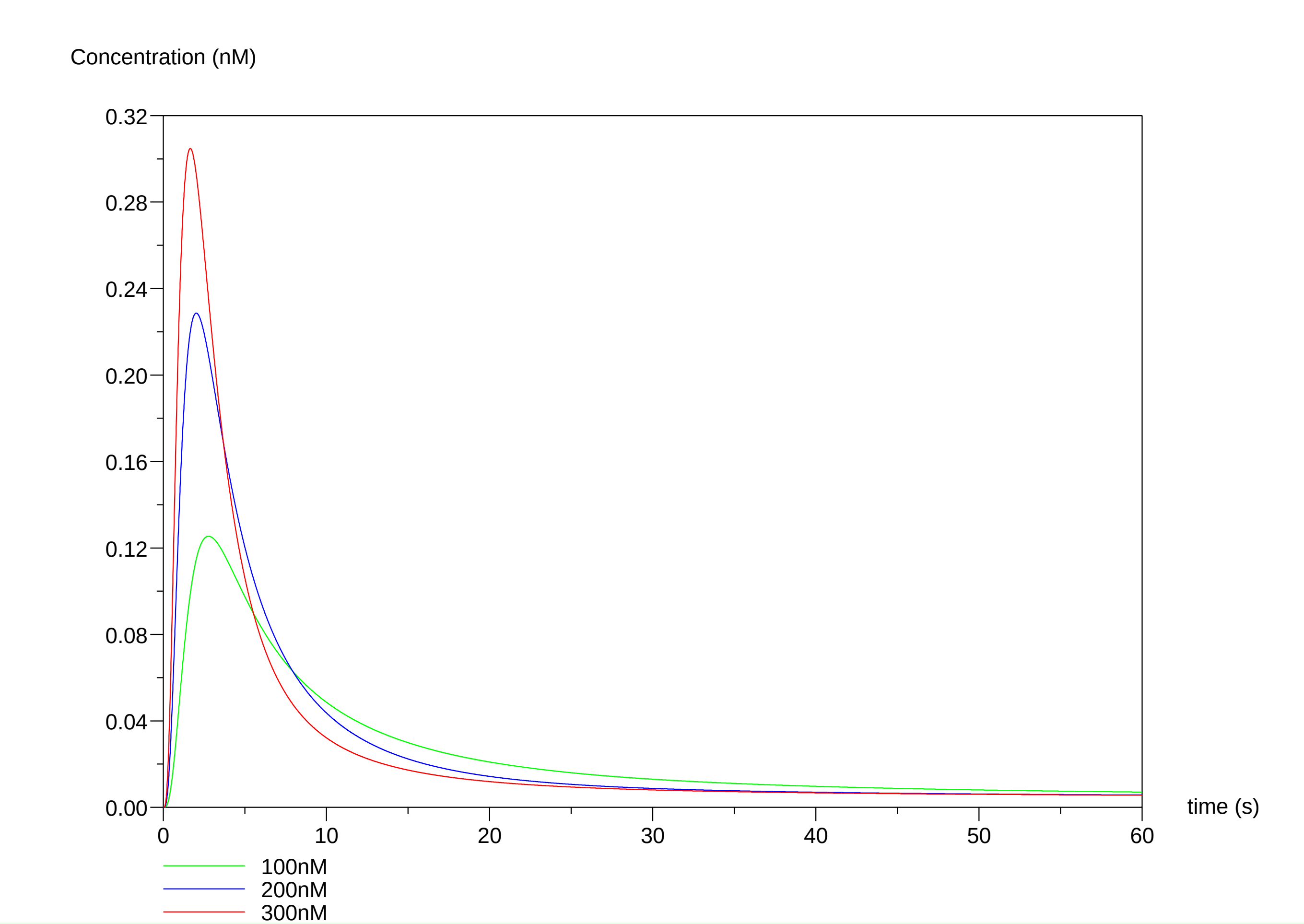
- controlling robots ([Bucharest Polytechnical University])

A typical application in biology/medicine:

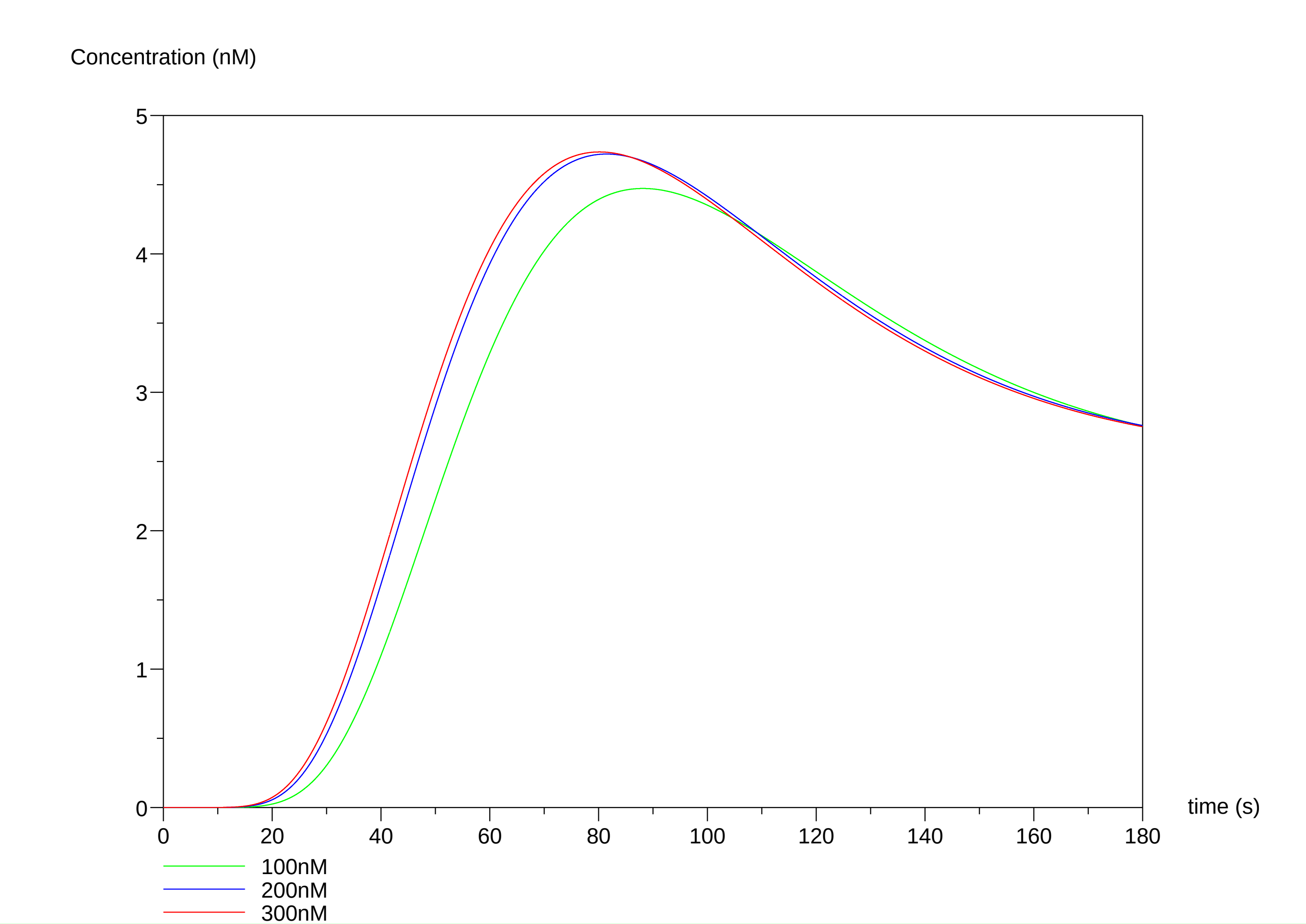
M.J. Pérez–Jiménez, F.J. Romero–Campero:
A Study of the Robustness of the EGFR Signalling Cascade Using Continuous Membrane Systems.
In *Mechanisms, Symbols, and Models Underlying Cognition. First International Work-Conference on the Interplay between Natural and Artificial Computation*, IWINAC 2005 (J. Mira, J.R. Alvarez, eds.), LNCS 3561, Springer, Berlin, 2005, 268–278.

- 60 proteins, 160 reactions/rules
- reaction rates from literature
- results as in experiments

Typical outputs:



The EGF receptor activation by auto-phosphorylation
(with a rapid decay after a high peak in the first 5 seconds)



The evolution of the kinase MEK
(proving a surprising robustness of the signalling cascade)

Other bio-applications:

- photosynthesis [Nishida, 2002]
- Brusselator [Suzuki, Verona, Milano]
- quorum sensing in bacteria [Nottingham, Sheffield, Sevilla]
- circadian cycles [Verona]
- apoptosis [Ruston-Louisiana]
- signaling pathways in yeast [Milano]
- HIV infection [Edinburgh]
- peripheral proteins [Trento]
- others [Milano, Iași, Bucharest, Sevilla, Verona, etc.]

Modeling ecosystems

Y. Suzuki, H. Tanaka, Artificial life and P systems, WMC1, Curtea de Argeș, 2000 (herbivorous, carnivorous, volatiles)

Lotka-Voltera model (predator-prey) [Verona, Milano]

M. Cardona, M.A. Colomer, M.J. Perez-Jimenez, S. Danuy, A. Margalida,
A P system modeling an ecosystem related to the bearded vulture, 6BWMC

Applications in economics:

- J. Bartosik, W. Korczynski, etc. (accounting, human resource management, etc.)
- Vienna group (Freund et colab.)
- Gh. Păun, R. Păun (general interpretation, Numerical P systems, modeling producer-retailer transactions)

Nishida's membrane algorithms:

- candidate solutions in regions, processed locally (local sub-algorithms)
- better solutions go down
- static membrane structure – dynamical membrane structure
- two-phases algorithms

Excellent solutions for Travelling Salesman Problem (benchmark instances)

- rapid convergence
- good average and worst solutions (hence reliable method)
- in most cases, better solutions than simulated annealing

Still, many problems remains: check for other problems, compare with sub-algorithms, more membrane computing features, parallel implementations (no free lunch theorem)

Recent: L. Huang, N. Wang, J. Tao; G. Ciobanu, D. Zaharie; A. Leporati, D. Pagani; M. Gheorghe et al. (quantum-membrane-algorithms)

What about future? (at the edge of science-fiction)

- hard to predict the future...
- ...but the progresses should not be underestimated
- natural computing will pay-off (directly, or through by-products)
- e.g., through nano-technology

Dreams:

- efficiency (through massive parallelism, nondeterminism)
- robust computers/algorithms
- adaptable, evolvable, learning, self-healing hardware/software
- nano-robots (for medicine)
- computing beyond Turing (stronger consequences than $\mathbf{P} = \mathbf{NP}$)

Do we dream too much?

- nature has different goals (and resources: time, materials, energy), is redundant, cruel
- theoretical limits:
  - Conrad theorems (programmability/universality, efficiency, learnability are contradictory)
  - Gandy principles for computing mechanisms (preventing the possibility to go beyond Turing)
- for modeling/simulating intelligence and life, maybe something essentially new is necessary (Mc Carthy, Brooks, etc.)

Every attempt to employ mathematical methods in the study of biological questions must be considered

Every attempt to employ mathematical methods in the study of biological questions must be considered profoundly irrational and contrary to the spirit of biology.

If mathematical analysis should ever hold a prominent place in biology - an aberration which is happily almost impossible - it would occasion a rapid and widespread degeneration of that science.

Auguste Comte (full name: Isidore Marie Auguste François Xavier Comte; January 17, 1798 - September 5, 1857): Pilosophie Positive, 1830

Thank you!

...and please do not forget: http://ppage.psystems.eu

(with mirrors in China: http://bmc.hust.edu.cn/psystems,
http://bmchust.3322.org/psystems*)*