# Membrane Computing:
# Power, Efficiency, Applications

## (A Quick Introduction)

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

## Summary:

- generalities
- the basic idea
- examples
- classes of P systems
- types of results
- types of applications
  - applications in biology
  - modeling/simulating ecosystems
  - Nishida's membrane algorithms
  - MC and economics; numerical P systems

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
- forthcoming *Handbook of Membrane Computing*, OUP
- Website: `http://ppage.psystems.eu`

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

SOFTWARE AND APPLICATIONS:

`http://www.dcs.shef.ac.uk/~marian/PSimulatorWeb/P_Systems_applications.htm`

`www.cbmc.it` – PSim2.X simulator

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

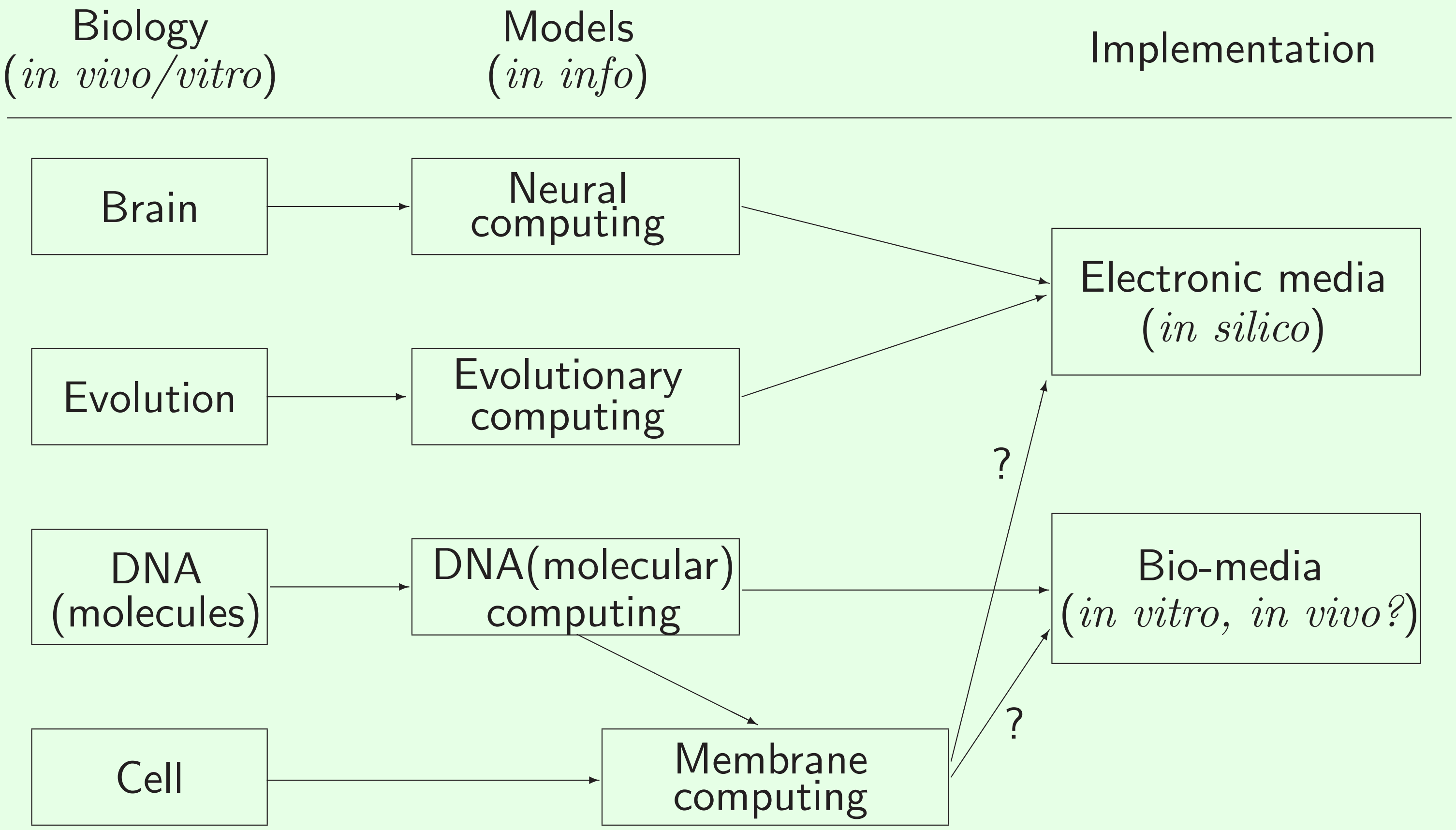
Sheffield (Marian Gheorghe: `M.Gheorghe@dcs.shef.ac.uk`)

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

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

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

# FRAMEWORK: Natural computing



Biology
(*in vivo/vitro*)

Models
(*in info*)

Implementation

Brain

Neural computing

Evolution

Evolutionary computing

DNA (molecules)

DNA(molecular) computing

Cell

Membrane computing

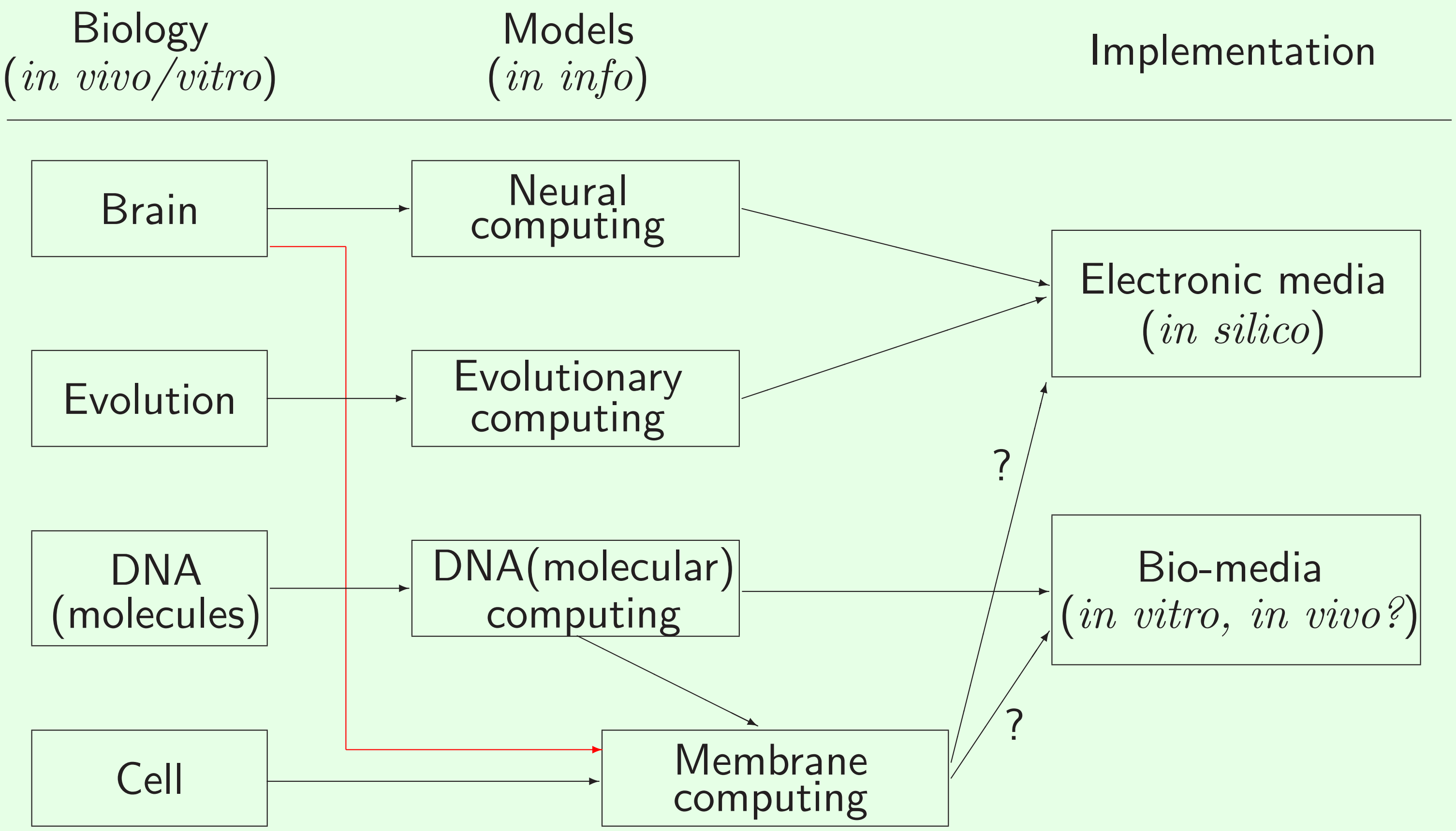
Electronic media
(*in silico*)

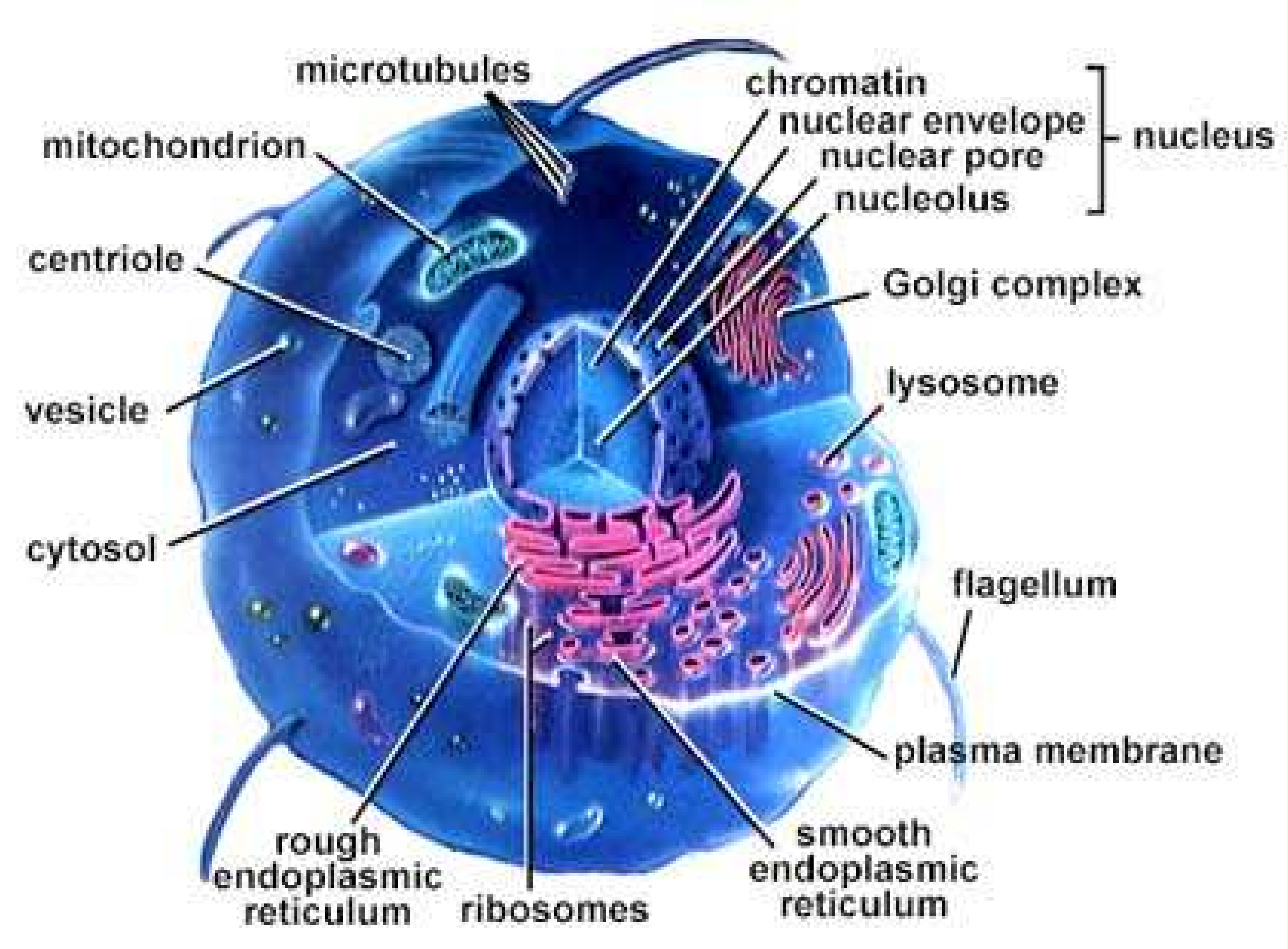
Bio-media
(*in vitro, in vivo?*)

?

?

# FRAMEWORK: Natural computing




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

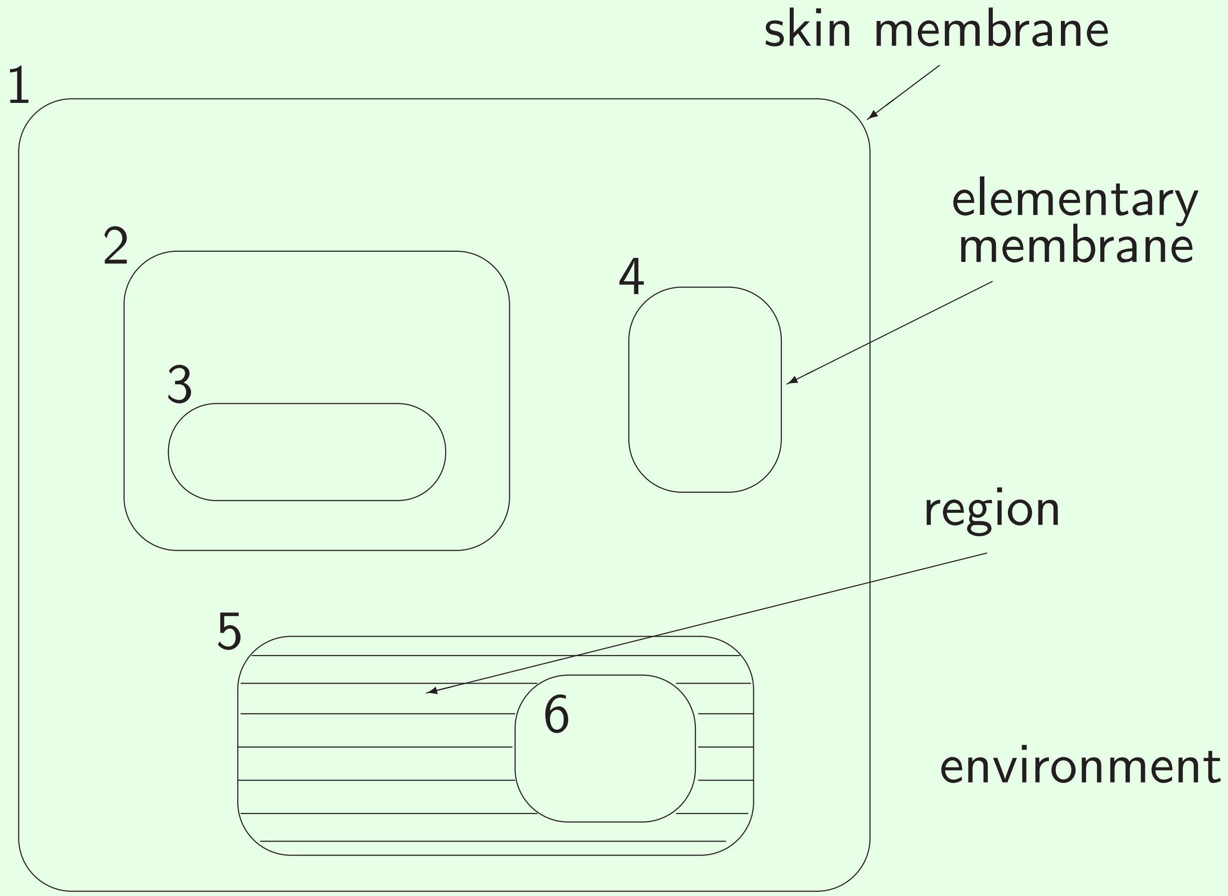
## WHAT IS A CELL? (for a mathematician)

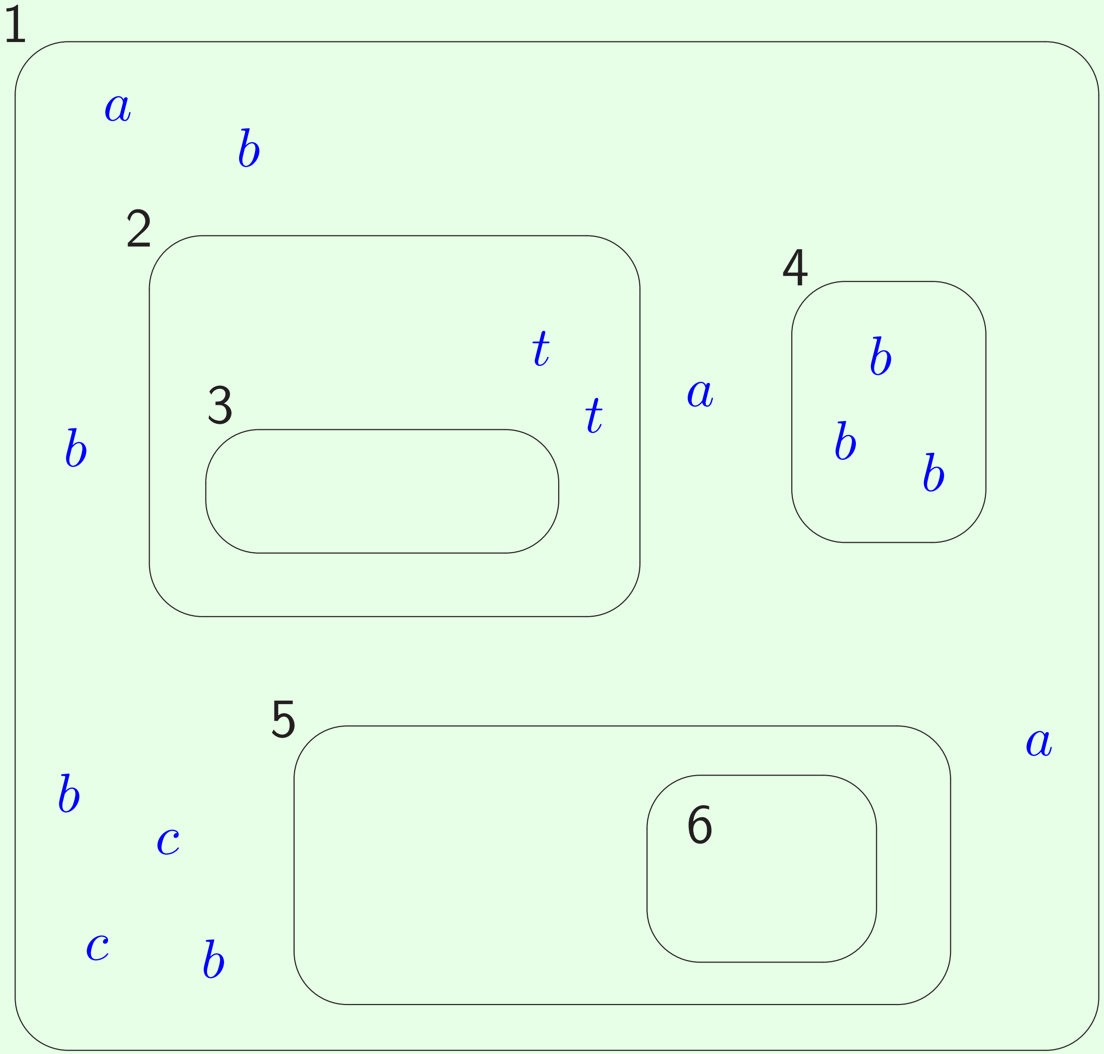
- membranes, separating “inside” from “outside” (hence protected compartments, “reactors”)
- chemicals in solution (hence multisets)
- biochemistry (hence parallelism, nondeterminism, decentralization)
- enzymatic activity/control
- selective passage of chemicals across membranes
- etc.

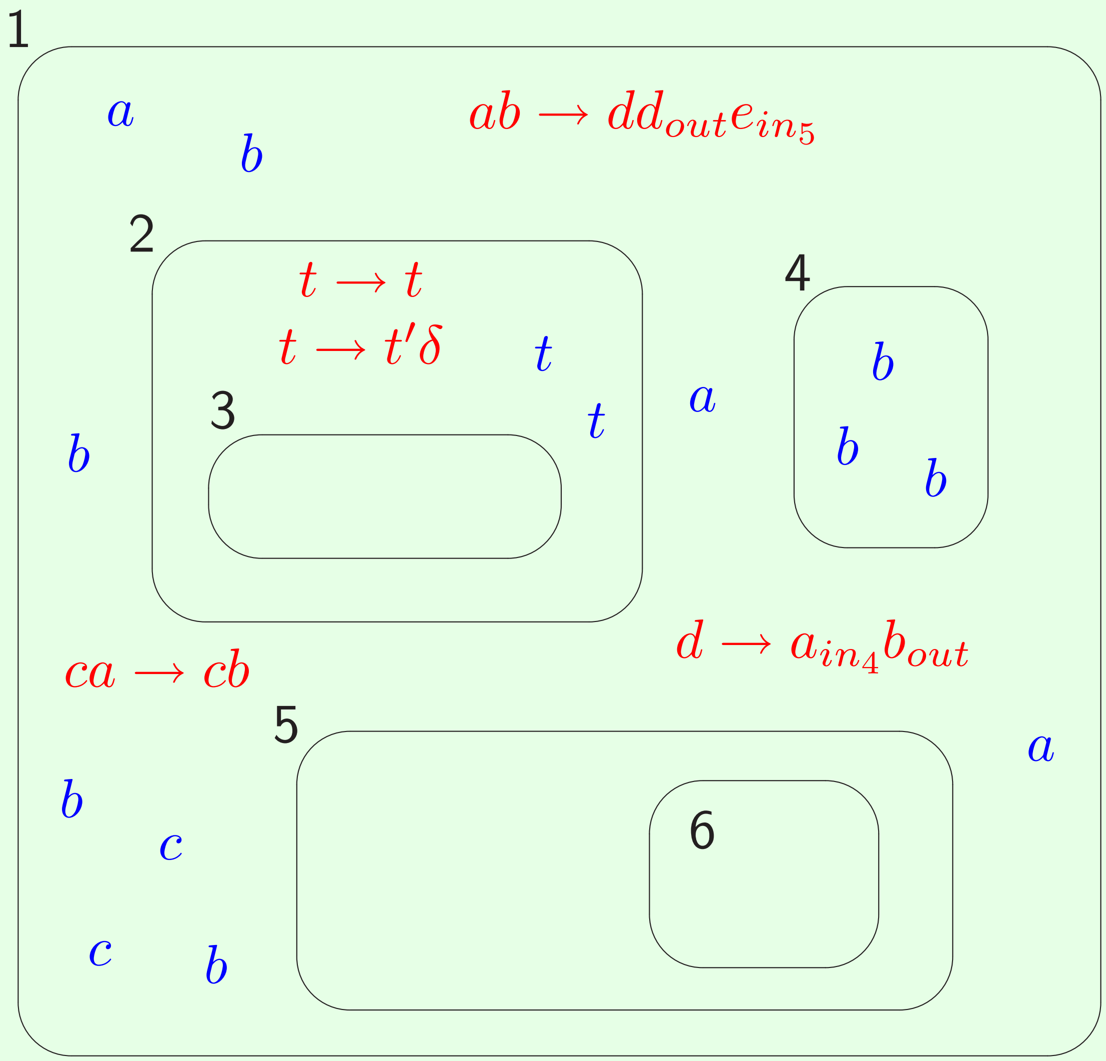
Importance of membranes for biology:. . .

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

## THE BASIC IDEA




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



$ab \rightarrow dd_{out}e_{in_5}$

$t \rightarrow t$

$t \rightarrow t'\delta$

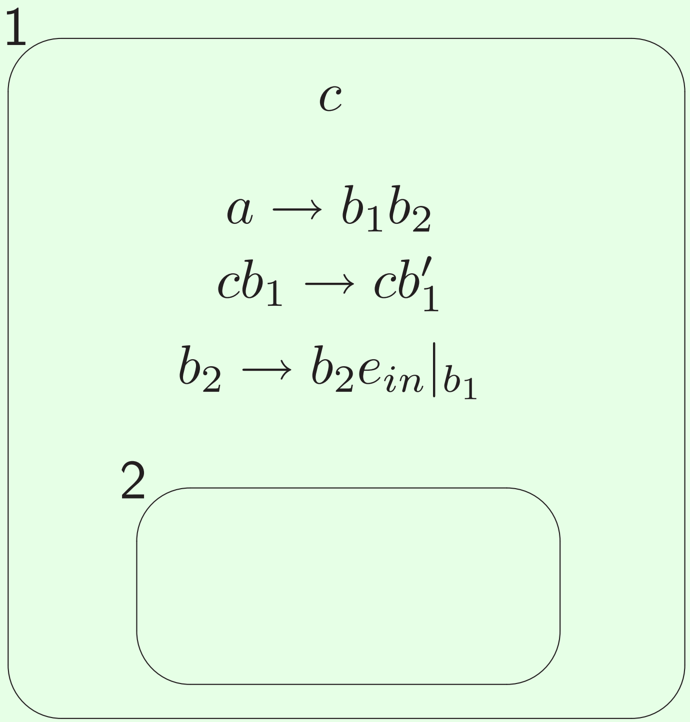
$d \rightarrow a_{in_4}b_{out}$

$ca \rightarrow cb$

## Functioning (basic ingredients):

- nondeterministic choice of rules and objects
- maximal parallelism
- transition, computation, halting
- internal output, external output, traces

## EXAMPLES



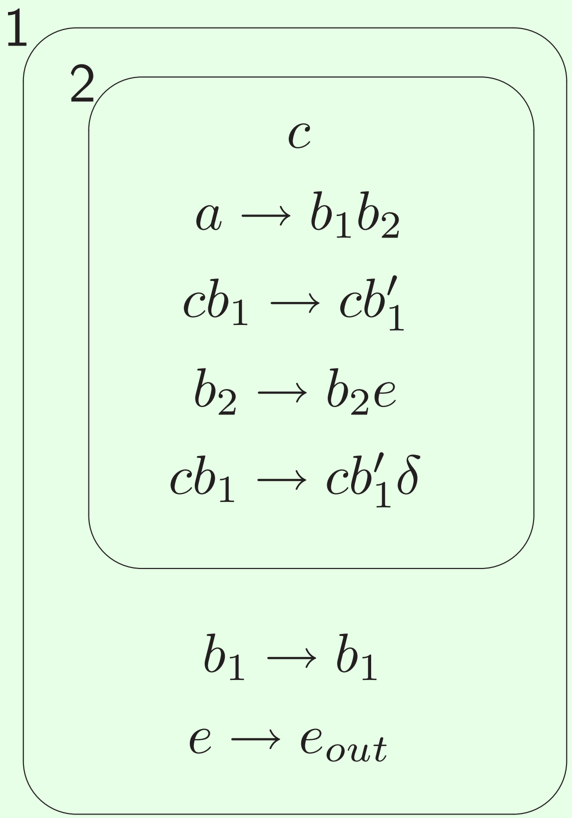
$$c$$

$$a \rightarrow b_1 b_2$$
$$cb_1 \rightarrow cb_1'$$
$$b_2 \rightarrow b_2 e_{in}|_{b_1}$$

Computing system: $n \longrightarrow n^2$ (catalyst, promoter, determinism, internal output)

Input (in membrane 1): $a^n$

Output (in membrane 2): $e^{n^2}$



1

2

$c$

$a \rightarrow b_1b_2$

$cb_1 \rightarrow cb_1'$

$b_2 \rightarrow b_2e$

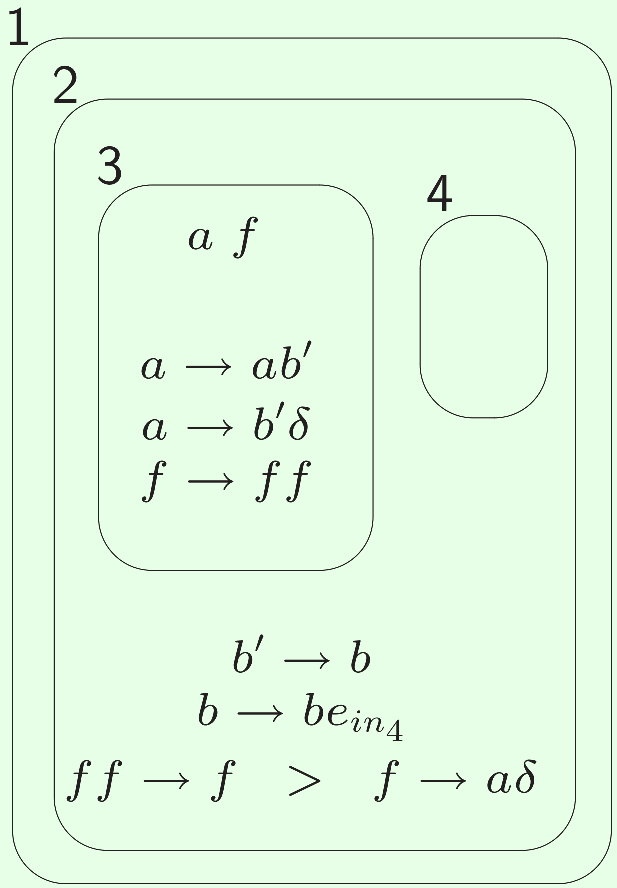
$cb_1 \rightarrow cb_1'\delta$

$b_1 \rightarrow b_1$

$e \rightarrow e_{out}$

The same function ($n \longrightarrow n^2$), with catalyst, dissolution, nondeterminism, external output

# Generative mode : $\{n^2 \mid n \geq 1\}$



$a \; f$

$a \to ab'$
$a \to b'\delta$
$f \to ff$

$b' \to b$
$b \to be_{in_4}$
$ff \to f \quad > \quad f \to a\delta$

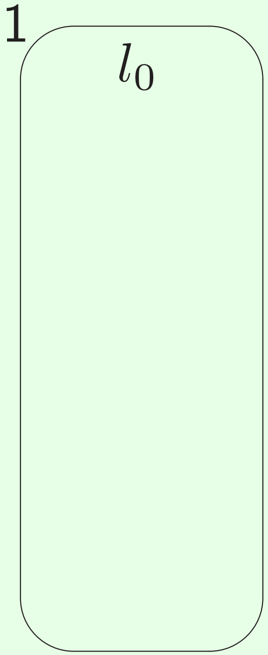
| | | |
|---|---|---|
| $0$ | $af$ | |
| $1$ | $ab'ff$ | |
| $\ldots$ | $\ldots$ | |
| $m \geq 0$ | $ab'^m f^{2^m}$ | |
| $m+1$ | $b'^{m+1} f^{2^{m+1}}$ | $\delta$ |
| $m+2$ | $b^{m+1} f^{2^m}$ | |
| $m+3$ | $b^{m+1} f^{2^{m-1}}$ | $e_{in_4}^{m+1}$ |
| $\ldots$ | $\ldots$ | $\ldots$ |
| $2m+1$ | $b^{m+1} f^2$ | $e_{in_4}^{m+1}$ |
| $2m+2$ | $b^{m+1} f$ | $e_{in_4}^{m+1}$ |
| $2m+3$ | $b^{m+1} a\delta$ | $e_{in_4}^{m+1}$ |
| $m+1$ times | HALT! | |

$$(m+1) \times (m+1)$$

$$N(\Pi) = \{n^2 \mid n \geq 1\}$$

# SIMULATING A REGISTER MACHINE $M = (m, B, l_0, l_h, R)$

$$E = \{a_r \mid 1 \leq r \leq m\} \cup \{l, l', l'', l''', l^{iv} \mid l \in B\}$$



$$\left.\begin{array}{l} (l_1, out; a_r l_2, in) \\ (l_1, out; a_r l_3, in) \end{array}\right\} \text{ for } l_1 : (\mathtt{add}(r), l_2, l_3)$$

$$\left.\begin{array}{l} (l_1, out; l_1' l_1'', in) \\ (l_1' a_r, out; l_1''', in) \\ (l_1'', out; l_1^{iv}, in) \\ (l^{iv} l_1''', out; l_2, in) \\ (l^{iv} l_1', out; l_3, in) \end{array}\right\} \text{ for } l_1 : (\mathtt{sub}(r), l_2, l_3)$$

$$(l_h, out)$$

Symport/antiport rules (of weight 2)

## Types of rules:

$u \to v$ with targets in $v$
(possibly conditional: promoters or inhibitors)
particular cases: $ca \to cu$ (catalytic)
$a \to u$ (non-cooperative)

$(ab, in), (ab, out)$ – symport (in general, $(x, in), (x, out)$)
$(a, in; b, out)$ – antiport (in general, $(u, in; v, out)$)

$u]_i v \to u']_i v'$ – boundary (Manca, Bernardini)

$ab \to a_{tar_1} b_{tar_2}$ – communication (Sosik)
$ab \to a_{tar_1} b_{tar_2} c_{come}$
$a \to a_{tar}$

| | |
|---|---|
| $a[\ ]_i \rightarrow [b]_i$ | go in |
| $[a]_i \rightarrow b[\ ]_i$ | go out |
| $[a]_i \rightarrow b$ | membrane dissolution |
| $a \rightarrow [b]_i$ | membrane creation |
| $[a]_i \rightarrow [b]_j[c]_k$ | membrane division |
| $[a]_i[b]_j \rightarrow [c]_k$ | membrane merging |
| $[a]_i[\ ]_j \rightarrow [[b]_i]_j$ | endocytosis |
| $[[a]_i]_j \rightarrow [b]_i[\ ]_j$ | exocytosis |
| $[u]_i \rightarrow [\ ]_i[u]_{@j}$ | gemmation |
| $[Q]_i \rightarrow [O-Q]_j[Q]_k$ | separation |

and others

Basic classes of cell-like P systems: multiset rewriting P systems:

$$\Pi = (O, \mu, w_1, \ldots, w_m, R_1, \ldots, R_m, i_o),$$

- $O$ = alphabet of objects
- $\mu$ = (labeled) membrane structure of degree $m$ (represented by a string of matching parentheses)
- $w_i$ = strings/multisets over $O$
- $R_i$ = sets of evolution rules
  typical form $ab \to (a, here)(c, in_2)(c, out)$
- $i_o$ = the output membrane

Symport/antiport P systems:

$$\Pi = (O, \mu, w_1, \ldots, w_m, E, R_1, \ldots, R_m, i_o),$$

as above, with $E \subseteq O$ the set of objects which appear in the environment in arbitrarily many copies

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, $\delta, \tau$, activators, etc.
- maximal, bounded, minimal parallelism, sequential/asynchronous, time-, clock-free
- generating, accepting, computing/translating, dynamical system
- computing power, computing efficiency, others
- 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

SAQ:

- 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:

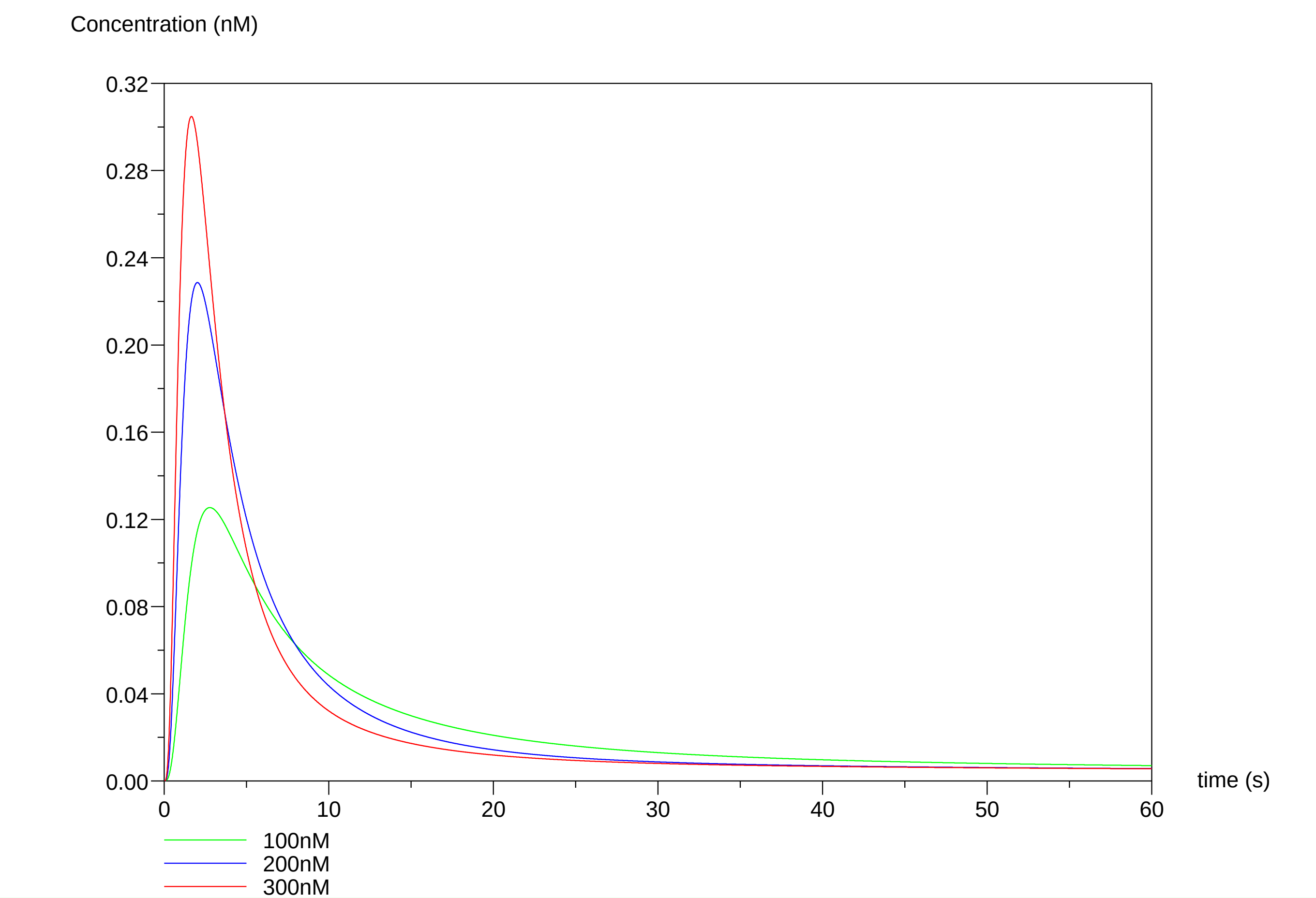
- biology, medicine, ecosystems (continuous versus discrete mathematics) [Sevilla, Verona, Milano, Sheffield, etc.]
- computer science (computer graphics, sorting/ranking, 2D languages, cryptography, general model of distributed-parallel computing) [many]
- linguistics (modeling framework, parsing) [Tarragona]
- optimization (membrane algorithms [Nishida, 2004], [many])
- economics ([Warsaw group], [R. Păun], [Vienna group])

## A typical application in biology/medicine:

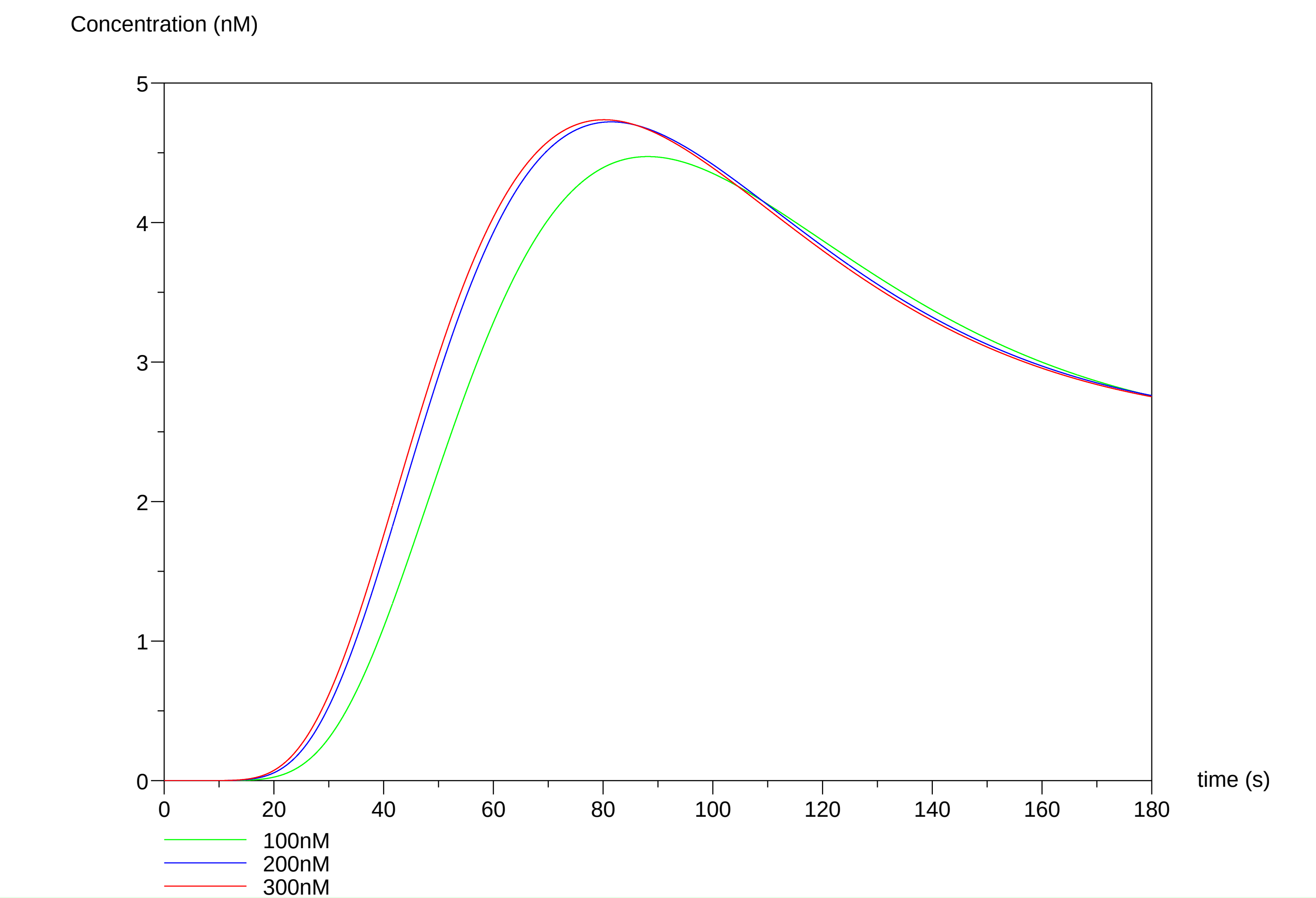
M.J. Pérez–Jiménez, F.J. Romero–Campero:
A Study of the Robustness of the EGFR Signalling Cascade Using Continuous Membrane Systems.
In *Mechanisms, Symbols, and Models Underlying Cognition. First International Work-Conference on the Interplay between Natural and Artificial Computation*, IWINAC 2005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

"Our model consists in the following probabilistic P system of degree 2 with two electrical charges:

$$\Pi = (\Gamma, \mu, \mathcal{M}_0, \mathcal{M}_1, R)$$

where:

- In the alphabet $\Gamma$ we represent the six species of the ecosystem (index $i$ is associated with the species and index $j$ is associated with their age, and the symbols $X$, $Y$ and $Z$ represent the same animal but in different state); it also contains the auxiliary symbols $B$ and $C$.

$$\Gamma = \{X_{ij},\ Y_{ij},\ Z_{ij} :\ 1 \leq i \leq 7,\ 0 \leq j \leq k_{i,5}\} \cup \{\ B,\ C\}$$

- In the membrane structure we represent two regions, the skin (where animals reproduce) and an inner membrane (where animals feed and die): $\mu = [\ [\ ]_1\ ]_0$ (neutral polarization will be omitted)
- In $\mathcal{M}_0$ and $\mathcal{M}_1$ we specify the initial number of objects present in each regions (encoding the initial population and the initial food).

  - $\mathcal{M}_0 = \{X_{ij}^{q_{ij}} : 1 \leq i \leq 7,\ 0 \leq j \leq k_{i,}\}$, where the multiplicity $q_{ij}$ indicates the number of animals, of species $i$ whose age is $j$ that are initially present in the ecosystem.
  - $\mathcal{M}_1 = \{C\ B^{18000}\}$, where the object $B$ represent $0.5$ kg of bones, and 9000 kg is the external contribution of bones to the P system corresponding to the $33\%$ of feeding that come from animals do not modeled in the P system.

- The set $R$ of evolution rules consists of:
  - Reproduction-rules

    Adult males:

    $$r_0 \equiv [X_{ij} \xrightarrow{1-k_{i,14}} Y_{ij}]_0, 1 \leq i \leq 7, 0 \leq j \leq k_{i,4}$$

    Adult females that reproduce:

    $$r_1 \equiv [X_{ij} \xrightarrow{k_{i,5}k_{i,14}} Y_{ij}Y_{i0}]_0,\ 1 \leq i \leq 7,\ k_{i,2} \leq j < k_{i,3}$$

    Adult females that do not reproduce:

    $$r_2 \equiv [X_{ij} \xrightarrow{(1-k_{i,5})k_{i,14}} Y_{ij}]_0,\ 1 \leq i \leq 7,\ k_{i,2} \leq j < k_{i,3}$$

Young animals that do not reproduce:

$$r_3 \equiv [X_{ij} \to Y_{ij}]_0,\ 1 \le i \le 7,\ k_{i,3} \le j < k_{i,2}$$

- Young animals mortality rules:

  Those which survive:

  $$r_4 \equiv Y_{ij}[\ ]_1 \xrightarrow{1-k_{i,7}-k_{i,8}} [Z_{ij}]_1 :\ 1 \le i \le 7,\ 0 \le j < k_{i,1}$$

  Those which die and leaving bones:

  $$r_5 \equiv Y_{ij}[\ ]_1 \xrightarrow{k_{i,8}} [B^{k_{i,12}}]_1 :\ 1 \le i \le 7,\ 0 \le j < k_{i,1}$$

Those which die and do not leave bones:

$$r_6 \equiv Y_{ij}[\ ]_1 \xrightarrow{k_{i,7}} [\ ]_1 :\ 1 \le i \le 7,\ 0 \le j < k_{i,1}$$

- Adult animals mortality rules:

Those which survive:

$$r_7 \equiv Y_{ij}[\ ]_1 \xrightarrow{1-k_{i,9}-k_{i,10}} [Z_{ij}]_1 :\ 1 \le i \le 7,\ k_{i,1} \le j < k_{i,4}$$

Those which die leaving bones:

$$r_8 \equiv Y_{ij}[\ ]_1 \xrightarrow{k_{i,10}} [B^{k_{i,13}}]_1 :\ 1 \le i \le 7,\ k_{i,1} \le j < k_{i,4}$$

Those which die and do not leave bones:

$$r_9 \equiv Y_{ij}[\ ]_1 \xrightarrow{k_{i,9}} [\ ]_1 :\ 1 \le i \le 7,\ k_{i,1} \le j < k_{1,4}$$

Animals that die at an average life expectancy:

$$r_{10} \equiv Y_{ij}[\ ]_1 \rightarrow [B^{k_{i,13} \cdot k_{i,11}}]_1 :\ 1 \le i \le 7,\ j = k_{i,4}$$

- Feeding rules:

$$r_{11} \equiv [Z_{ij} B^{k_{i,16}}]_1 \rightarrow X_{ij+1}[\ ]_1^+ :\ 1 \le i \le 7,\ 0 \le j \le k_{i,4}$$

- Rules of mortality due to lack of food, and the elimination from the system of bones that are not eaten by the Bearded Vulture:

  Elimination of remaining bones:

  $r_{12} \equiv [B]_1^+ \rightarrow [\,]_1$

  External contribution that represent the bones:

  $r_{13} \equiv [C]_1^+ \rightarrow [CB^{18000}]_1$

  Adult animals that die because they have not enough food:

  $r_{14} \equiv [Z_{ij}]_1^+ \rightarrow [B^{k_{i,13} \cdot k_{i,11}}]_1 : \; 1 \leq i \leq 7, \; k_{i,1} \leq j \leq k_{i,4}$

  Young animals that die because they have not enough food:

  $r_{15} \equiv [Z_{ij}]_1^+ \rightarrow [B^{k_{i,12} \cdot k_{i,11}}]_1 : \; 1 \leq i \leq 7, \; j < k_{i,1}$

Figure 1 gives a schematic view of how the P system works.

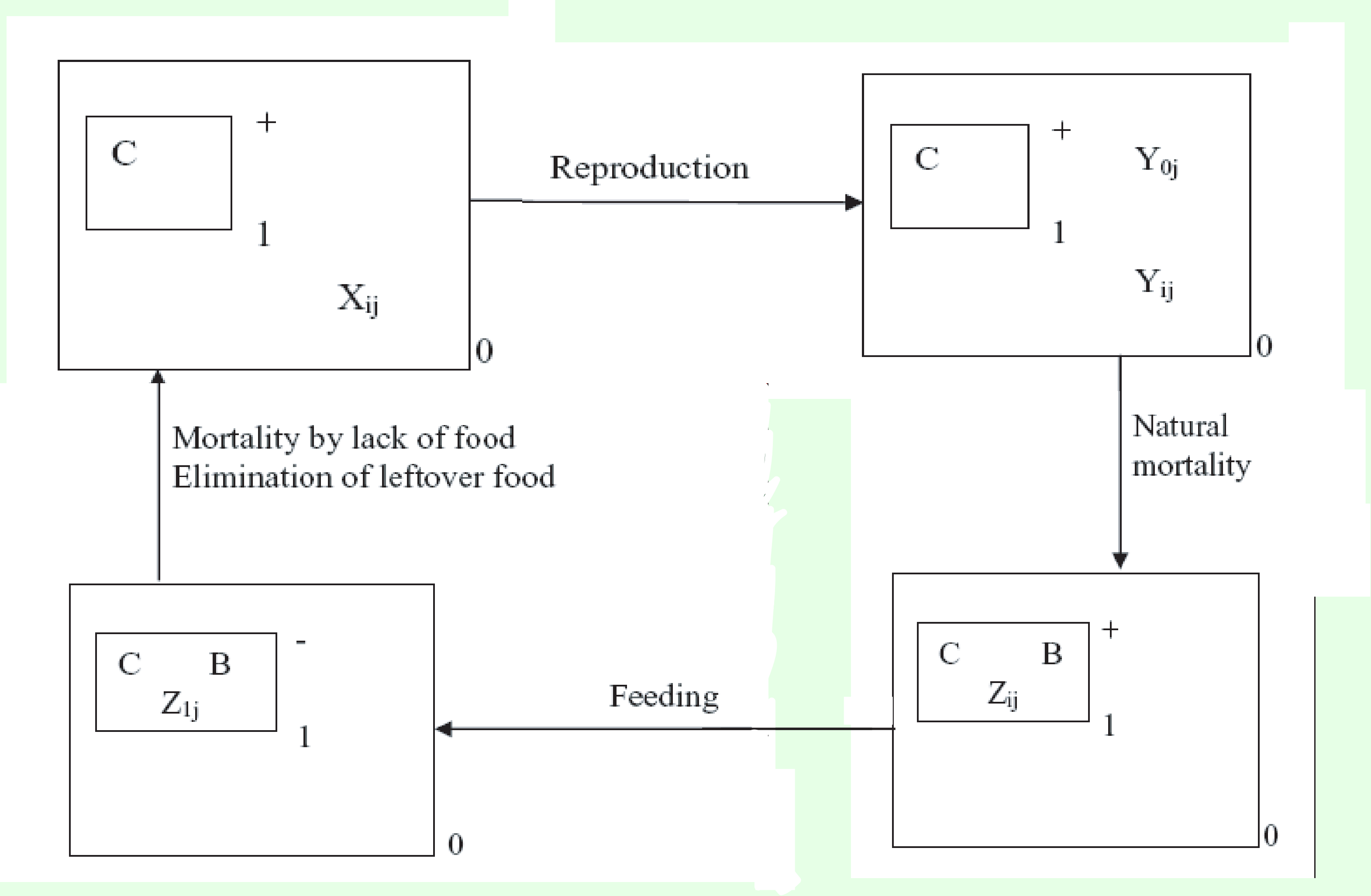


Figure 1: Schema

Table 1: Number of animals, at the moment, in the Pyrenean Catalan

| Species | Number |
|---|---|
| Bearded Vulture | 74 |
| Chamois | 12000 |
| Red deer female | 4400 |
| Red deer male | 1100 |
| Fallow deer | 900 |
| Roe deer | 10000 |
| Sheep | 200000 |

## (Some) Results:

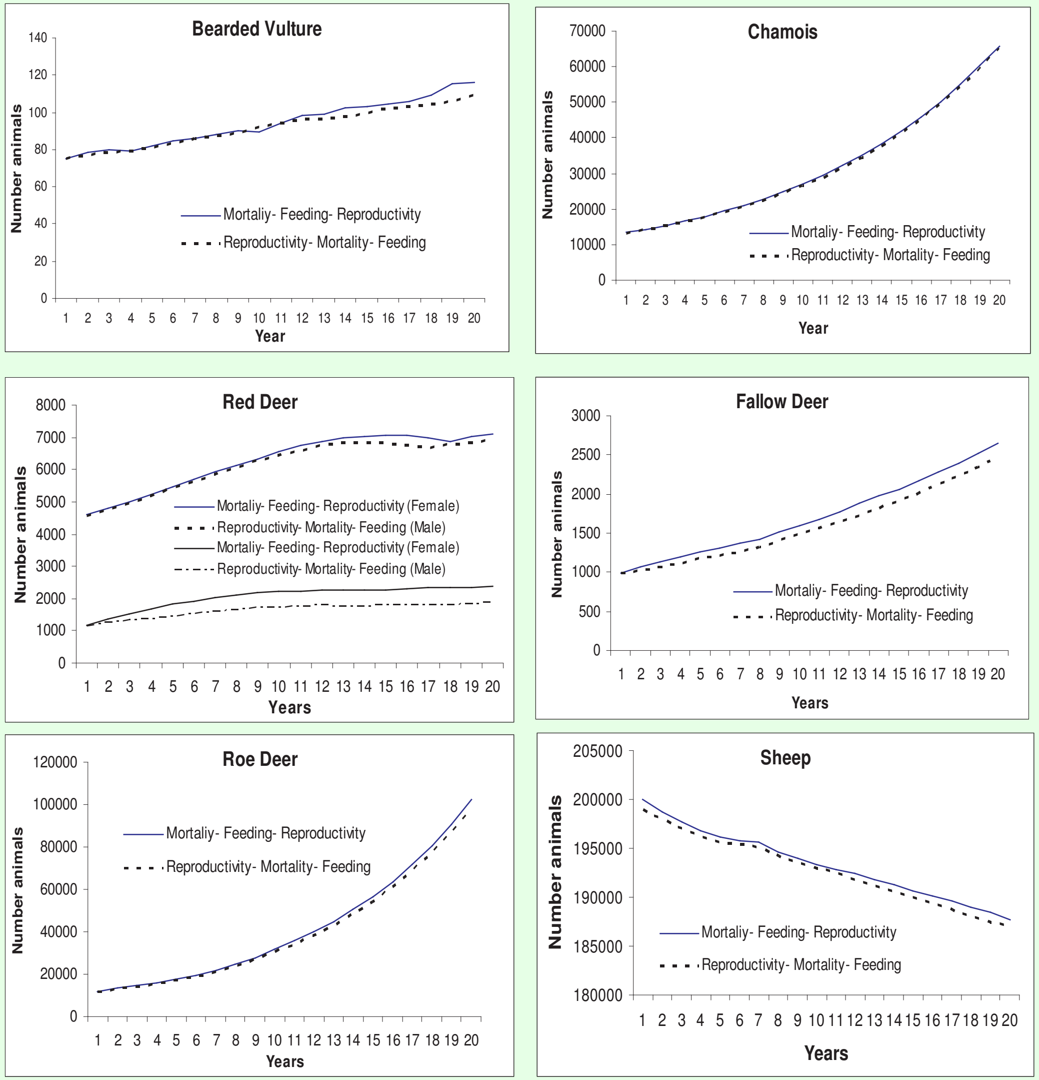


Figure 2: Robustness of the ecosystem

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)

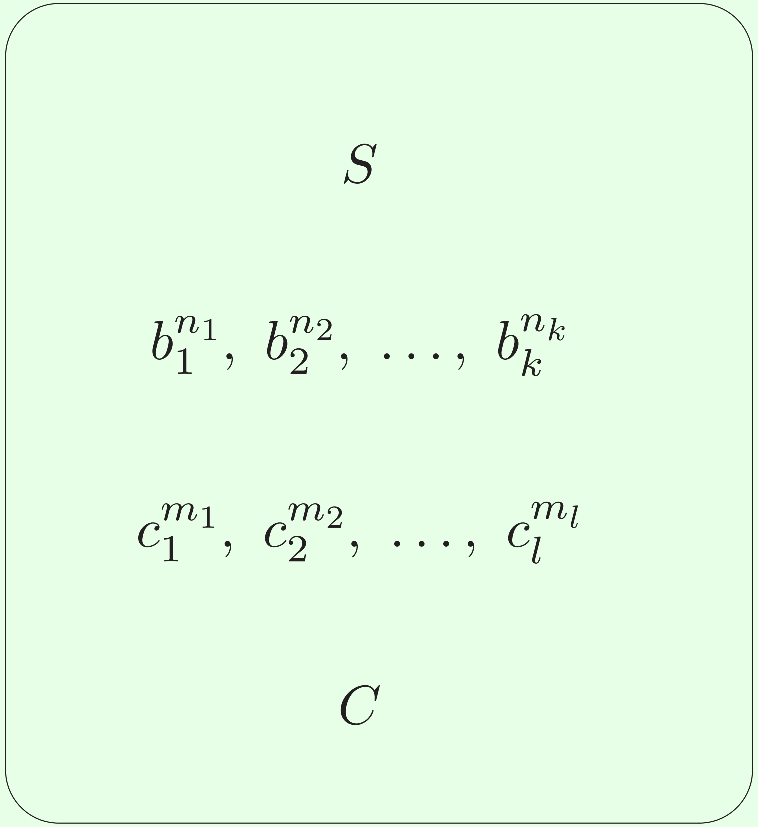
- rapid convergence
- good average and worst solutions (hence reliable method)
- in most cases, better solutions than simulated annealing

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)

## Applications in economics:

- J. Bartosik, W. Korczynski, etc (accounting, human resource management, etc)
- Gh. Păun, R. Păun (general interpretation, paired rules: $([u \rightarrow v]_i; [u' \rightarrow v']_j)$
- Gh. Păun, R. Păun: Numerical P systems
- R. Păun: Modelling producer-retailer transactions



Source of raw materials ($a$)

$S$

Producers ($d$)

$b_1^{n_1},\ b_2^{n_2},\ \ldots,\ b_k^{n_k}$

Retailers (match $d$ with $\bar{d}$)

$c_1^{m_1},\ c_2^{m_2},\ \ldots,\ c_l^{m_l}$

Consumption ($\bar{d}$)

$C$

Examples of rules:

1. $S \to Sa^{N \pm rg}[prob(rg)]$ (no money)

2. $b_i u_i^{p_S(t)} a \to d_i u_S^{p_S(t)}$ (producers pay for $a$)

3. $C \to C\bar{d}^{M \pm rg'} u_C^{p_C(t)(M \pm rg')}[prob(rg')]$ (the general consumer introduces both needs and money)

4. $c_j \bar{d} u_C^{ps_j(t)} \to \bar{d}_j v_j^{ps_j(t)}$ (orders and money pass to retailers)

5. $d_i \bar{d}_j v_j^{pp_i(t)} \to b_i c_j u_i^{pp_i(t)}[Rscore_{i,j}(t)]$ (one copy of $d$ is purchased by $R_j$ from $P_i$, paying for it the price $p_i(t)$ set by $P_i$)

Still more interesting: investments

$u_i^x \rightarrow b_i$ – by producer $i$

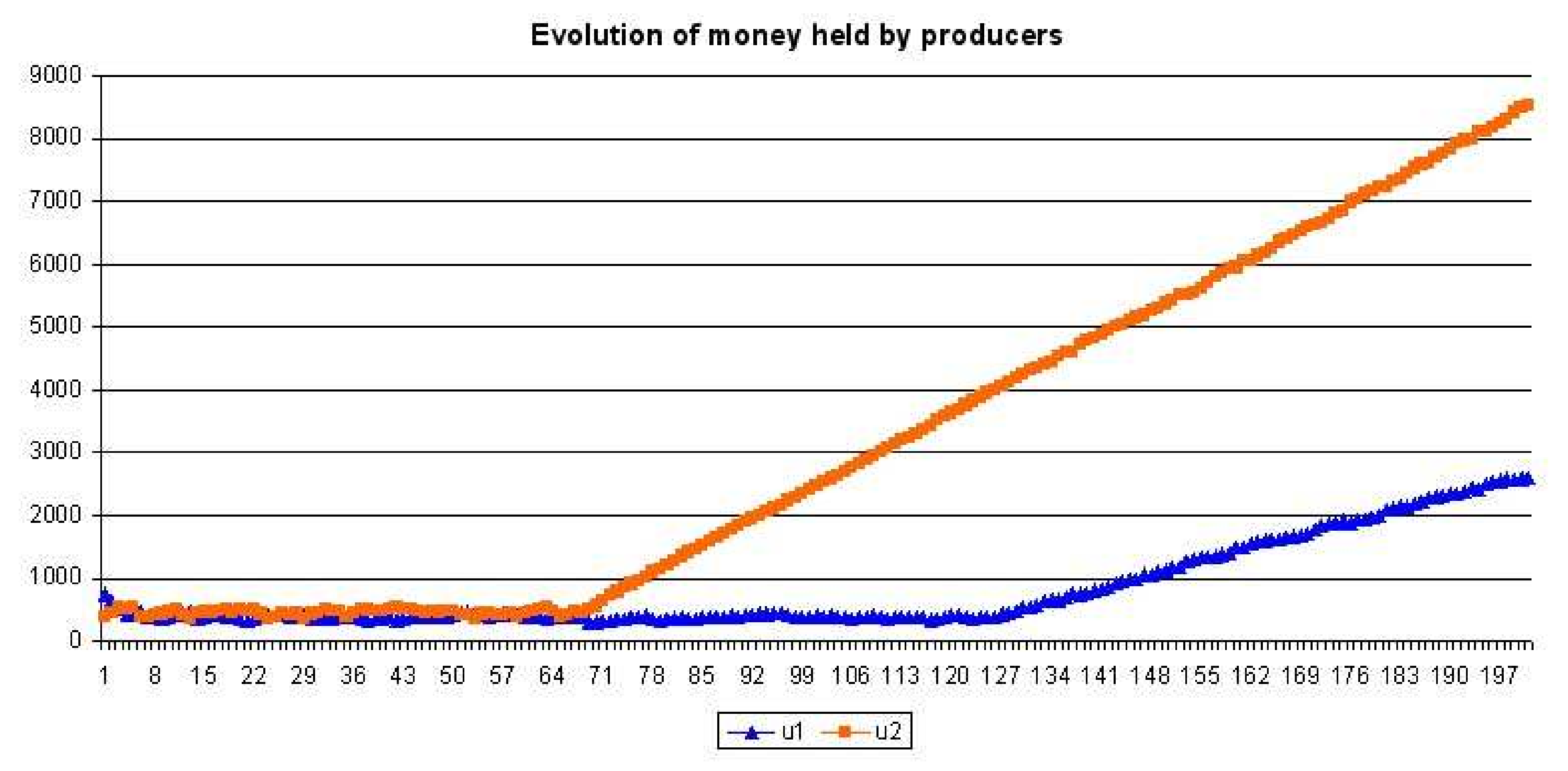
$v_j^y \rightarrow c_j$ – by retailer $j$

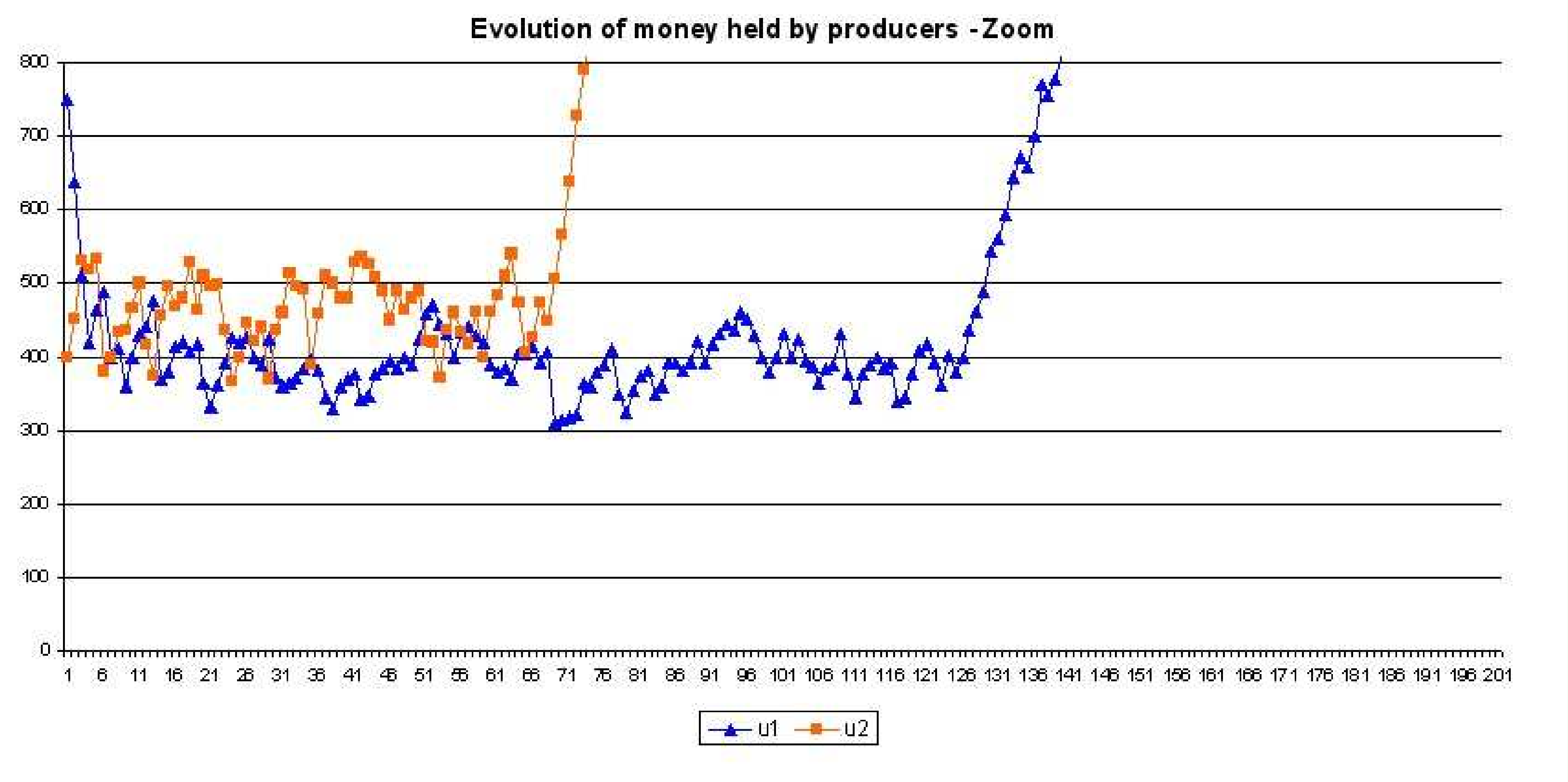
maybe in a bounded amount

$f_i u_i^x \rightarrow b_i$

$g_j v_j^y \rightarrow c_j$

with $z$ copies of each $f_i$ and $g_j$ introduced in the initial configuration


Evolution of money held by producers
9000
8000
7000
6000
5000
4000
3000
2000
1000
0
1
8
15
22
29
36
43
50
57
64
71
78
85
92
99
106
113
120
127
134
141
148
155
162
169
176
183
190
197
u1
u2


Evolution of money held by producers - Zoom
800
700
600
500
400
300
200
100
0
1
6
11
16
21
26
31
36
41
46
51
56
61
66
71
76
81
86
91
96
101
106
111
116
121
126
131
136
141
146
151
156
161
166
171
176
181
186
191
196
201
u1
u2

# 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

*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 Franois Xavier Comte; January 17, 1798 - September 5, 1857): Pilosophie Positive, 1830

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.)

# 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`)