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
  - Nishida's membrane algorithms

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)

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

- Website: http://psystems.disco.unimib.it

(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)

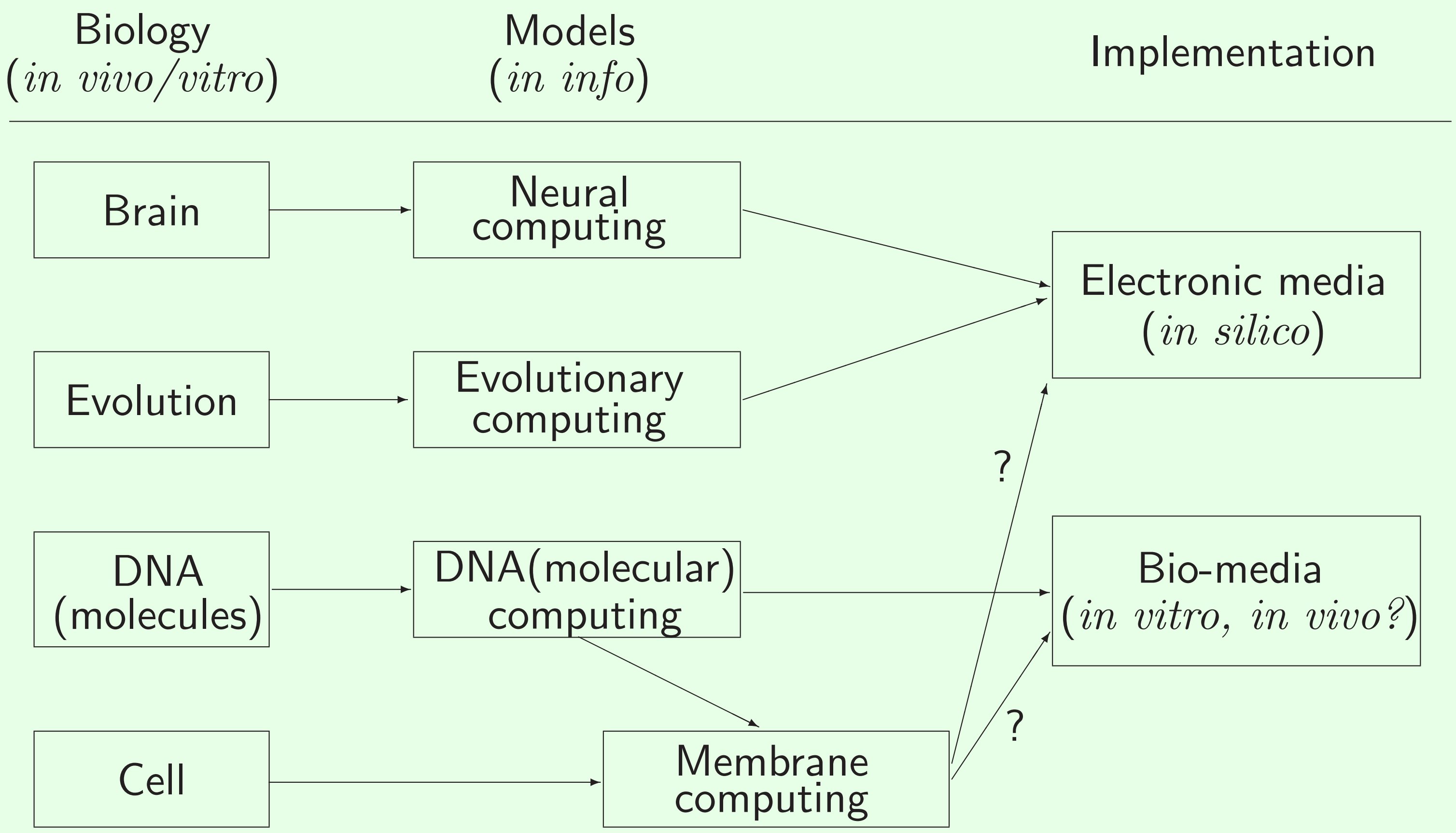
Sheffield (Marian Gheorghe: M.Gheorghe@dcs.shef.ac.uk)

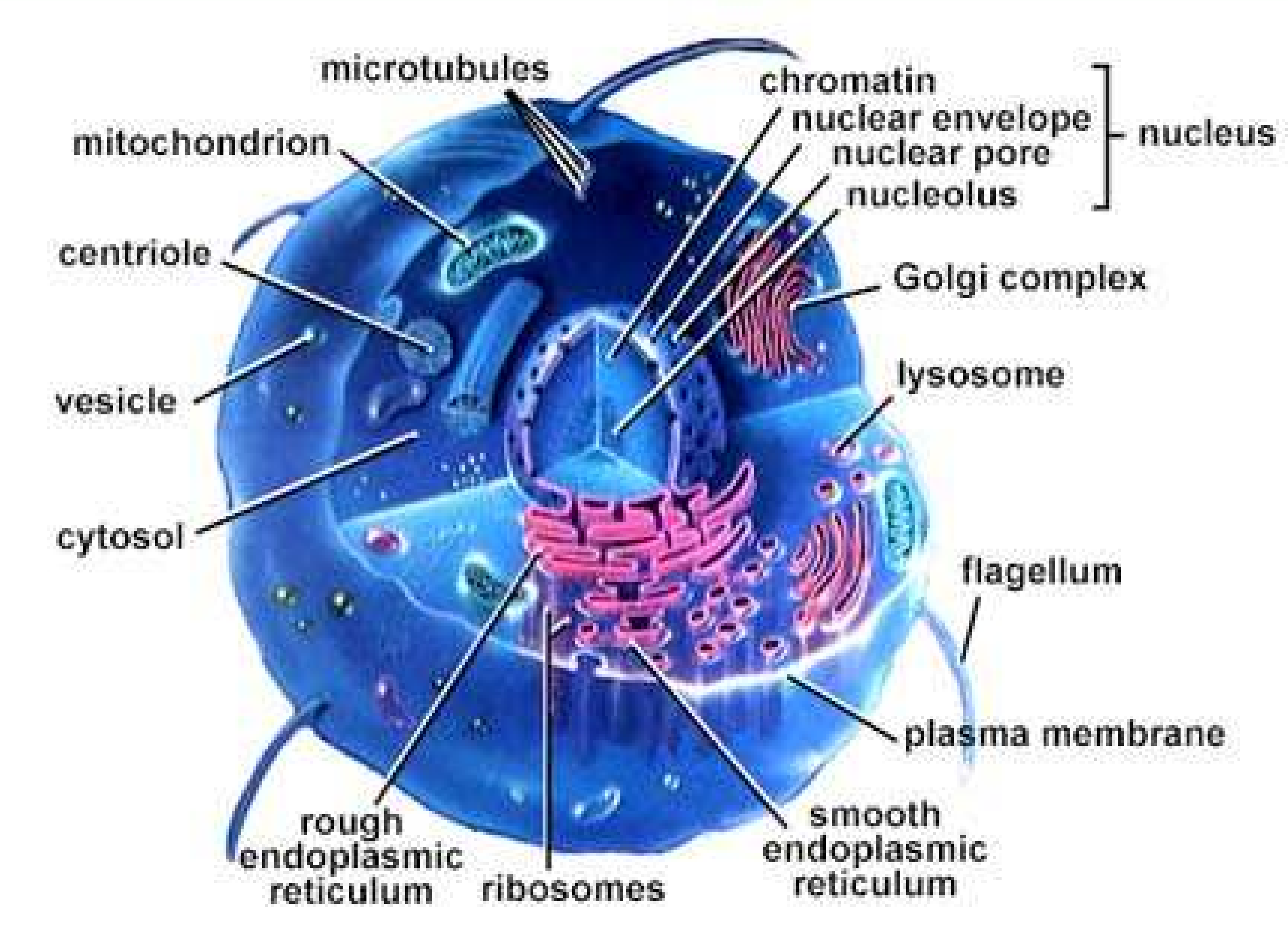
Sevilla (Mario Pérez-Jiménez: marper@us.es)

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

Nottingham, Leiden, Vienna, Evry, Iași

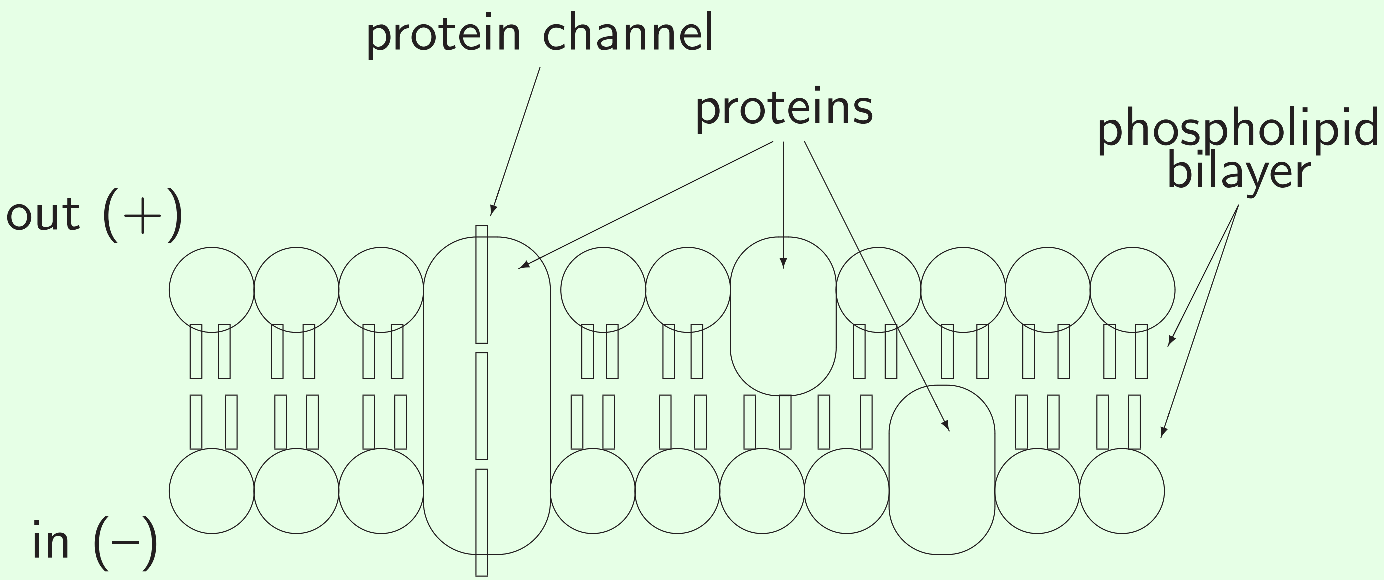
Natural computing

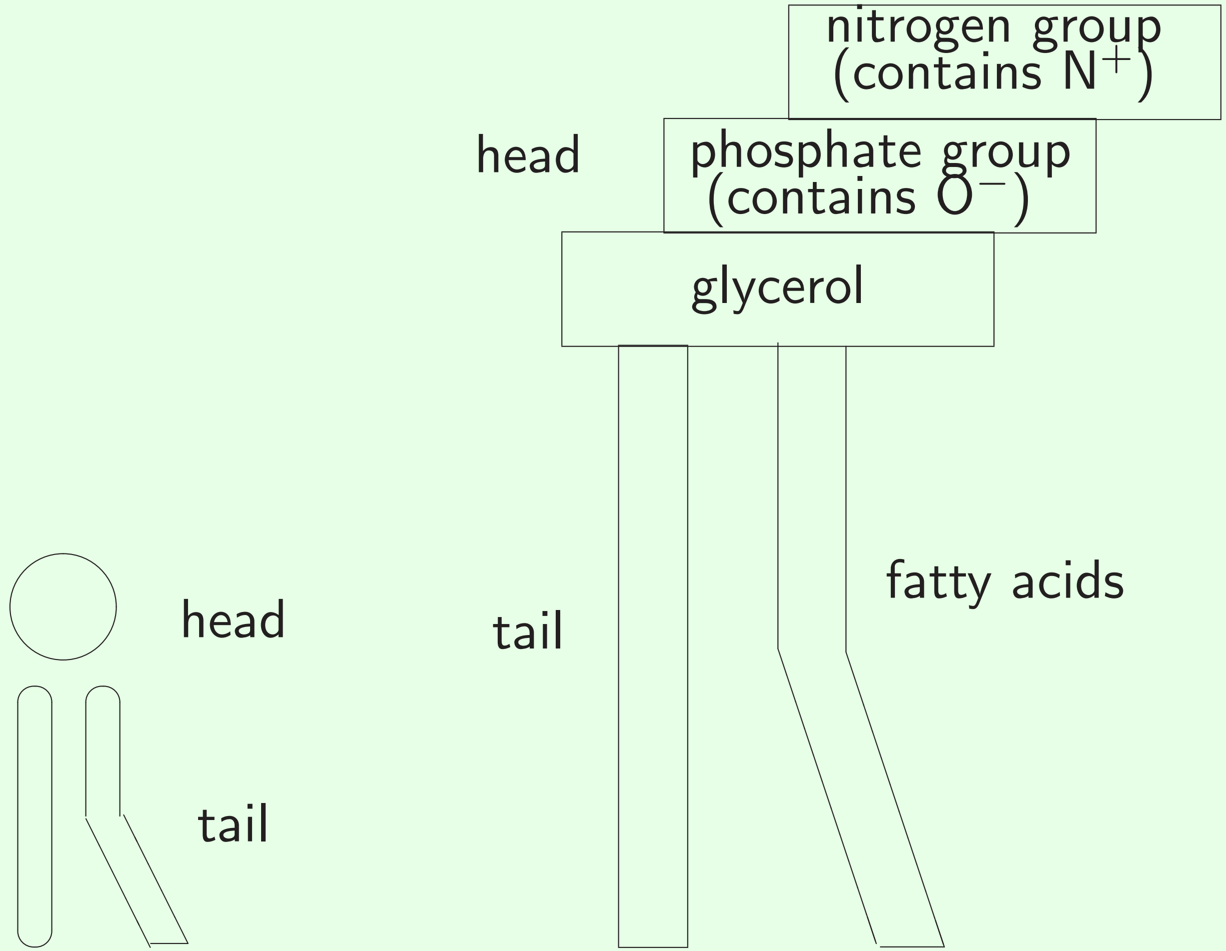



microtubules
mitochondrion
chromatin
nuclear envelope
nuclear pore
nucleolus
nucleus
centriole
Golgi complex
vesicle
lysosome
cytosol
flagellum
plasma membrane
rough endoplasmic reticulum
ribosomes
smooth endoplasmic reticulum

THE STRUCTURE OF PLASMA MEMBRANE

The *fluid–mosaic model*, S. Singer and G. Nicolson, 1972



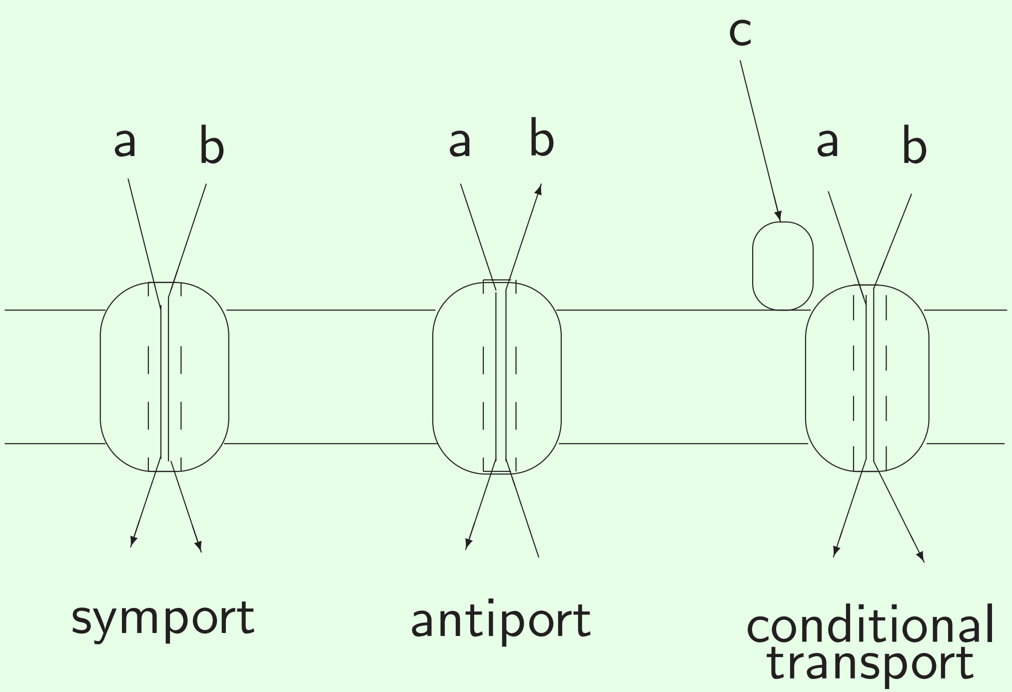

nitrogen group
(contains N+)
head
phosphate group
(contains O−)
glycerol
head
tail
tail
fatty acids

Trans–Membrane Transport

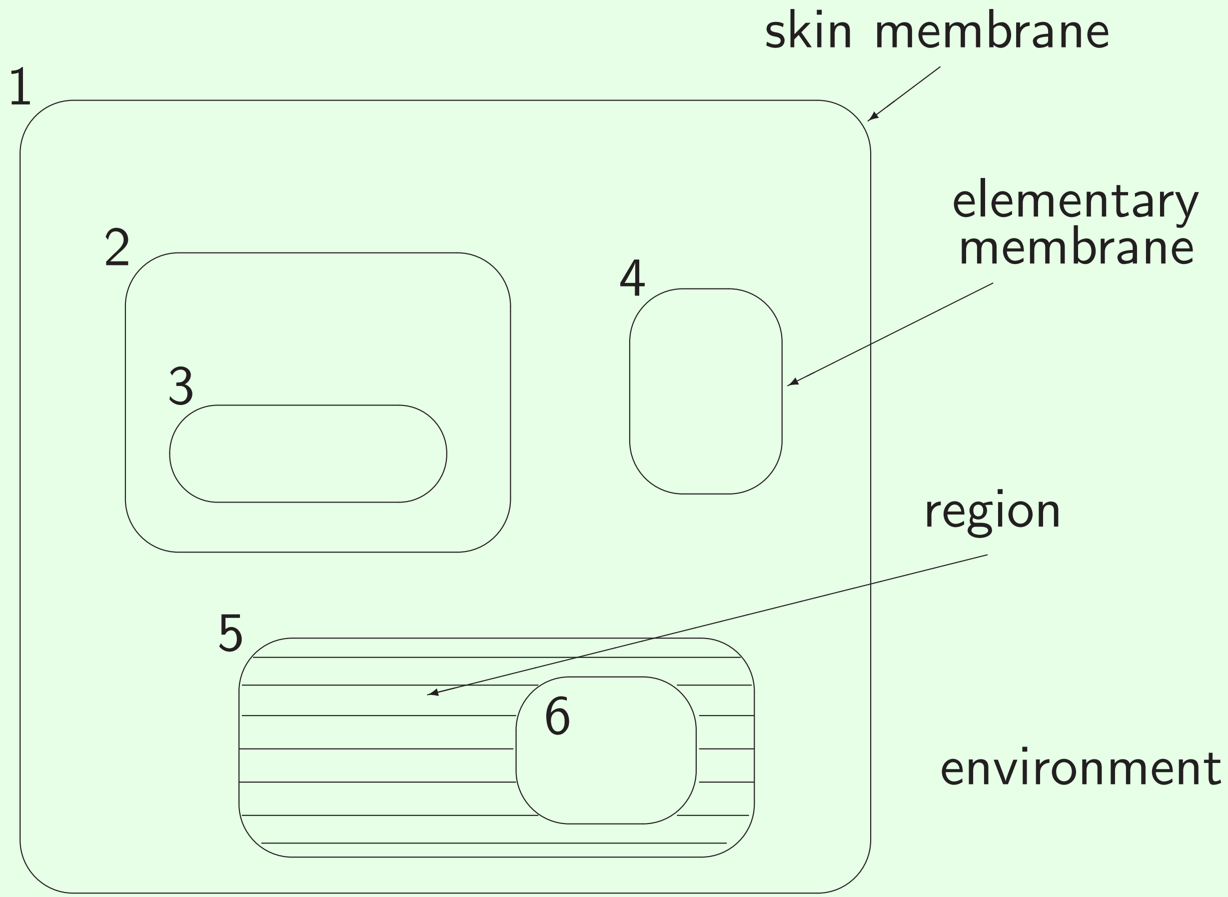
passive (diffusion – concentration based)
active (protein channels)
vesicle–mediated

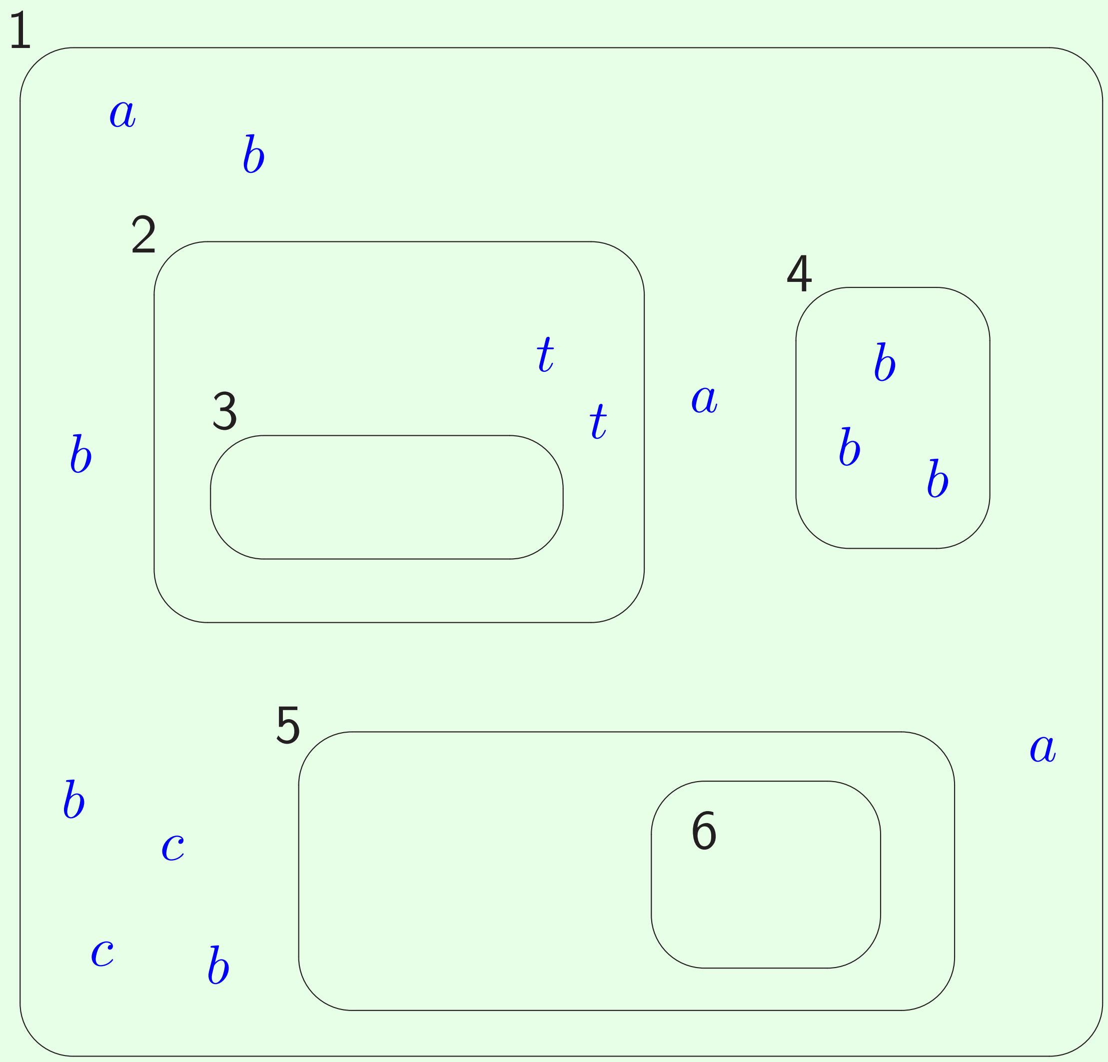
Important case of active transport: coupled transport

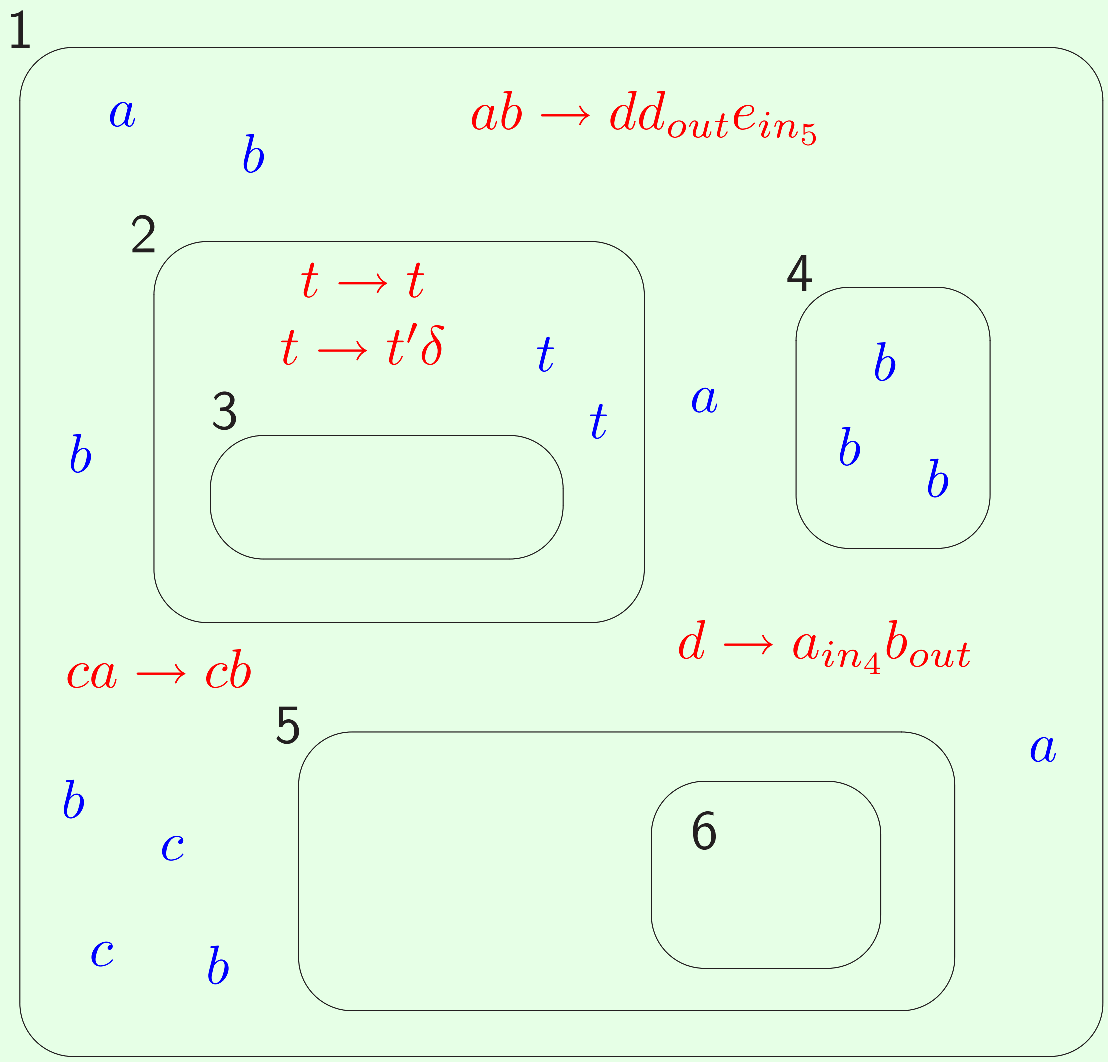
symport
antiport



THE BASIC IDEA



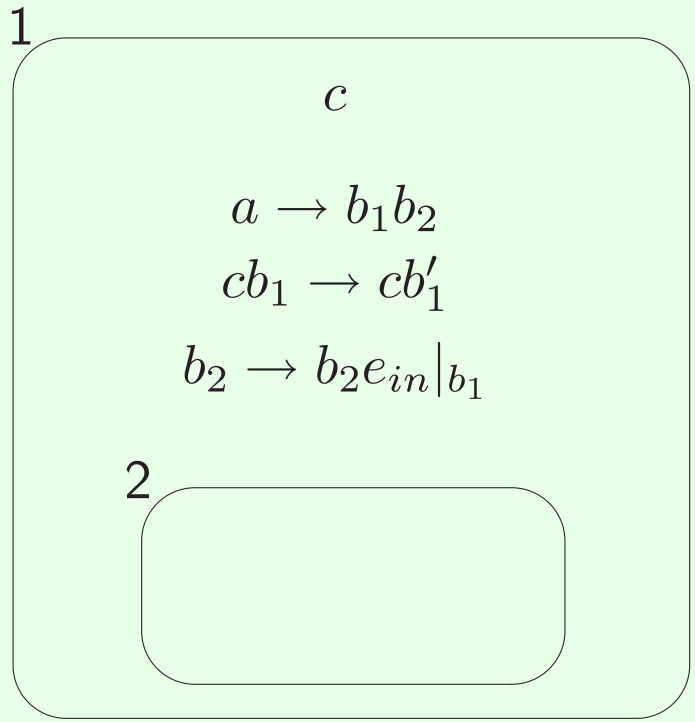

1
a
b
2
3
t
t
a
4
b
b
b
b
5
6
a
b
c
c
b


1
a
b
$ab \rightarrow dd_{out}e_{in_5}$
2
$t \rightarrow t$
$t \rightarrow t'\delta$
t
t
3
4
b
b
b
a
b
$d \rightarrow a_{in_4}b_{out}$
$ca \rightarrow cb$
5
6
a
b
c
c
b

Functioning (basic ingredients):

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

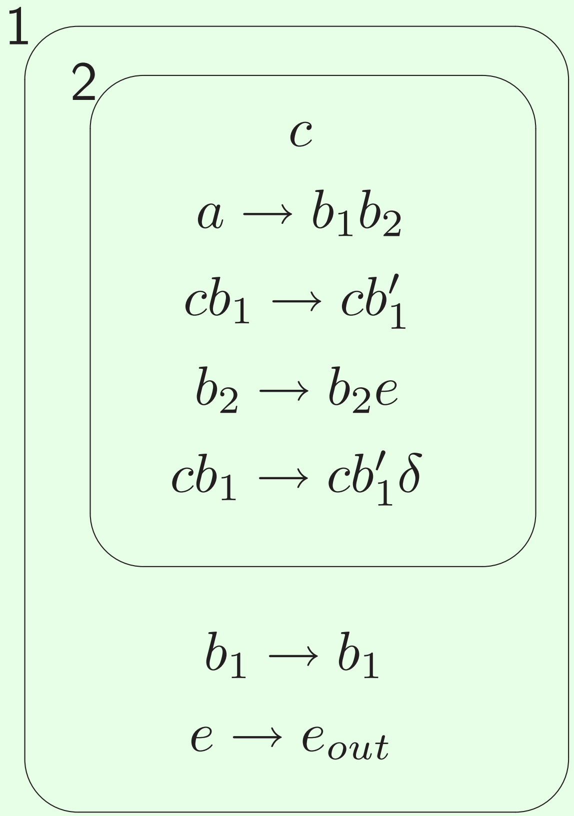
EXAMPLES



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

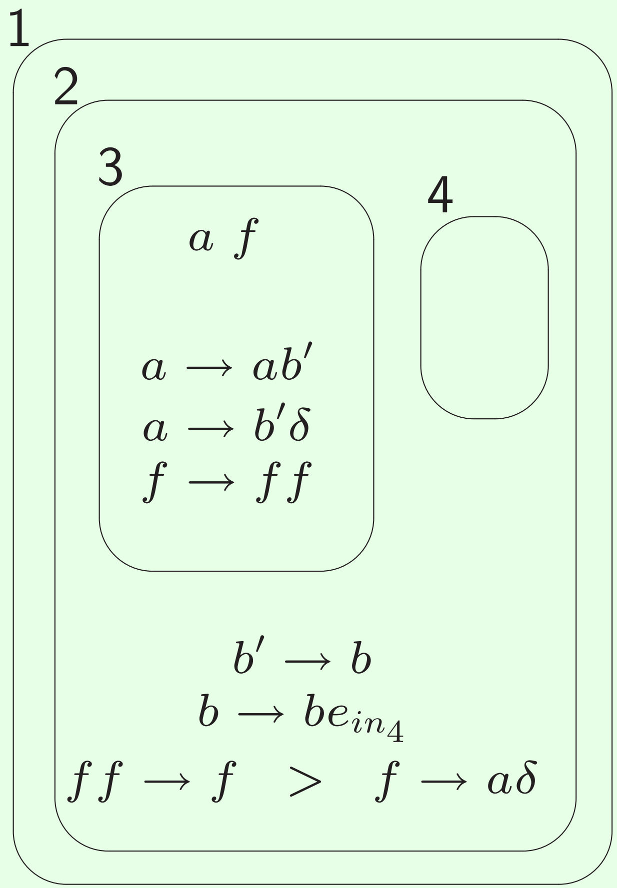
Input (in membrane 1): a^n

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



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

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

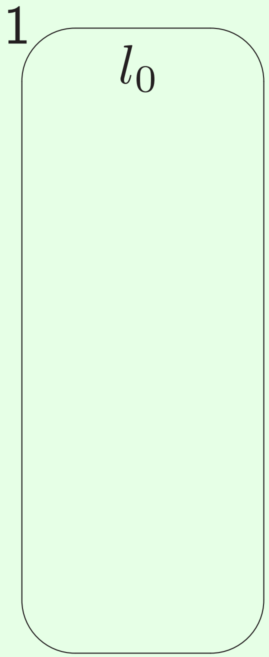


0	af	
1	$ab'ff$	
$\dots$	$\dots$	
$m \geq 0$	$ab'^m f^{2^m}$	
$m+1$	$b'^{m+1} f^{2^{m+1}}$	δ
$m+2$	$b^{m+1} f^{2^m}$	
$m+3$	$b^{m+1} f^{2^{m-1}}$	$e_{in_4}^{m+1}$
$\dots$	$\dots$	$\dots$
$2m+1$	$b^{m+1} f^2$	$e_{in_4}^{m+1}$
$2m+2$	$b^{m+1} f$	$e_{in_4}^{m+1}$
$2m+3$	$b^{m+1} a\delta$	$e_{in_4}^{m+1}$
$m+1$ 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)

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
- synchronized, 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
- quantum
- etc (e.g., brane-membrane bridge)

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
- synchronized, 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.
- quantum-like
- etc. (e.g., brane-membrane bridge)

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)
$ab \to a_{tar_1} b_{tar_2} c_{come}$
$a \to a_{tar}$

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

and others

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]
- 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
- 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 (the power of 1 catalyst, the role of polarizations, dissolution, uniform versus semi-uniform, etc.)
- semantics (events, causality, etc.)
- neural-like systems (more biology, complexity, applications, etc.)
- user friendly, flexible software for bio-applications
- MC and economics

Applications:

- biology, medicine (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])

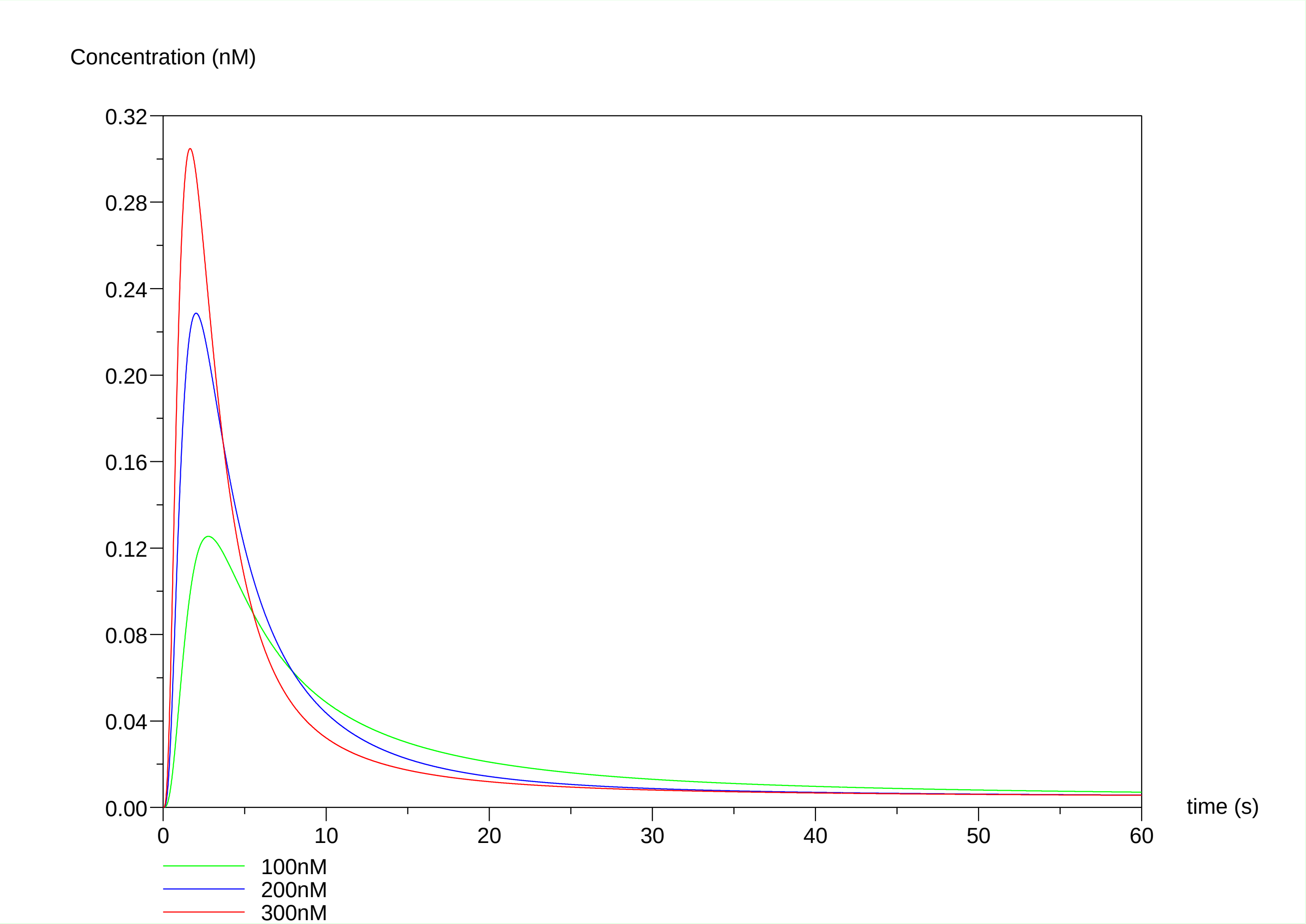
- economics ([Warsaw group], [R. Păun], [Vienna group])

A typical application in biology/medicine:

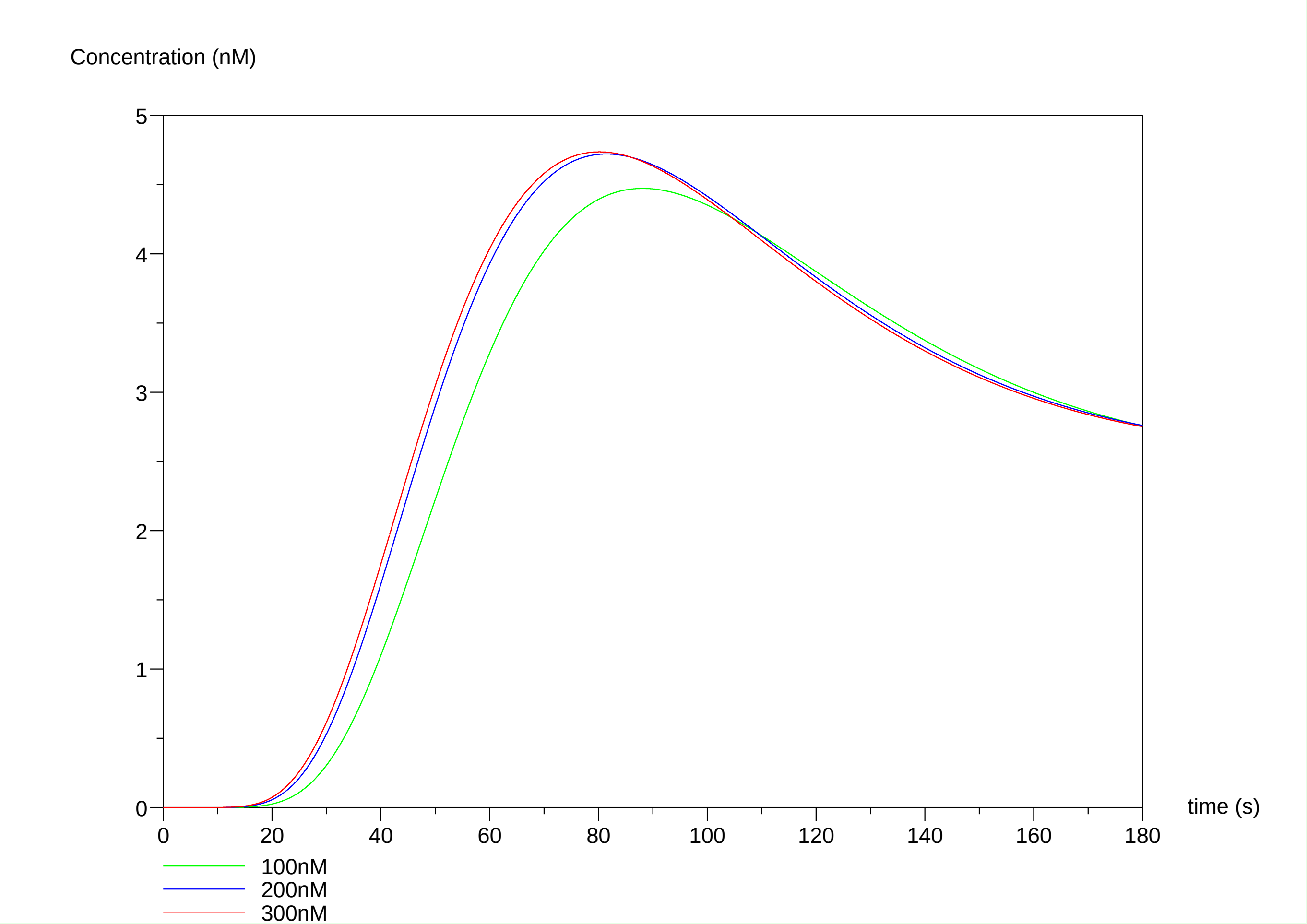
M.J. Pérez–Jiménez, F.J. Romero–Campero:
A Study of the Robustness of the EGFR Signalling Cascade Using Continuous Membrane Systems.
In *Mechanisms, Symbols, and Models Underlying Cognition. First International Work-Conference on the Interplay between Natural and Artificial Computation*, IWINAC 2005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

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

Thank you!

...and please do not forget: `http://psystems.disco.unimib.it`

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