

# P Systems, a New Computational Modelling Tool for Systems Biology

M.J. Pérez-Jiménez; F.J. Romero-Campero

Research Group on Natural Computing  
Department of Computer Science and Artificial Intelligence  
University of Sevilla  
Avda. Reina Mercedes s/n, 41012, Sevilla, Spain  
{marper, fran}@us.es

**Abstract.** In this paper we present P systems as a reliable computational modelling tool for Systems Biology that takes into account the discrete character of the quantity of components of biological systems, the inherently randomness in biological phenomena and the key role played by membranes in the function of living cells. We will introduce two different strategies for the evolution of P systems, namely, *Multi-compartmental Gillespie's Algorithm* based on the well known Gillespie's Algorithm but running on more than one compartment; and *Deterministic Waiting Times Algorithm*, an *exact* deterministic method. In order to illustrate these two strategies we have modelled two biological systems: the EGFR Signalling Cascade and the Quorum Sensing System in the bacterium *Vibrio Fischeri*. Our simulations results show that for the former system a deterministic approach is valid whereas for the latter a stochastic approach like *Multi-compartmental Gillespie's Algorithm* is necessary.

## 1 Introduction

Membrane Computing is an emergent branch of Natural Computing introduced by G. Păun in [15]. Since then it has received important attention from the scientific community. In fact, Membrane Computing has been selected by the Institute for Scientific Information, USA, as a fast *Emerging Research Front* in Computer Science, and [14] was mentioned in [22] as a highly cited paper in October 2003.

This new model of computation starts from the assumption that the processes taking place in the compartmental structure of a living cell can be interpreted as computations. The devices of this model are called P systems. Roughly speaking, a P system consists of a cell-like membrane structure, in the compartments of which one places multisets of objects which evolve according to given rules.

Most variants of membrane systems have been proved to be computationally complete, that is equivalent in power to Turing machines, and computationally efficient, that is able to solve computationally hard problems in polynomial time. Although most research in P systems concentrates on computational powers, lately they have been used to model biological phenomena [2, 3, 5].

As P systems are inspired from the functioning of the living cell, it is natural to consider them as modelling tools for biological systems, within the framework of systems biology, being an alternative to more classical approaches like ODEs. Differential equations have been used successfully to model kinetics of conventional macroscopic coupled chemical reactions. Nevertheless there is an implicit assumption of continuously varying chemical concentration and deterministic dynamics. Two critical characteristics of this approach are that the number of molecules of each type in the reaction mix is large compared to thermal fluctuations in concentration and that for each type of reaction in the system, the number of reactions is large within each observation interval, that is reaction rates are fast.

When concentrations of the reacting species are low and reaction rates are slow, which is frequently the case in genetic circuits, both of the previous pre-suppositions are invalid and the deterministic continuous approach to chemical kinetics breaks down. Instead one has to recognise that the individual chemical reaction steps occur discretely and are separated by time intervals of random length.

In contrast to differential equations, P systems are an unconventional model of computation which takes into consideration the discrete character of the quantity of components and the inherently randomness in biological phenomena. Besides, the key feature of P systems is the so called membrane structure which represents the heterogeneity of the structural organisation of the cells, and where one can take into account the role played by membranes in the functioning of the system, for example the fluctuation effects caused by diffusion.

In this paper we will present P systems as a reliable tool for Systems Biology. We will discuss two different strategies for their evolution. The first strategy is based on the well known Gillespie's Algorithm but running on more than one compartment and so it will be called *Multi-compartmental Gillespie Algorithm*. The second one will be a deterministic strategy called *Deterministic Waiting Times* which consists in an *exact* deterministic method in the sense that we do not approximate infinitesimal intervals of time by  $\Delta t$  as it is the case in ODEs, but we will associate a waiting time, computed in a deterministic way, to each reaction and will use them to determine the order in which the reactions take place.

In order to illustrate these strategies two different biological systems will be modelled; the Epidermal Growth Factor Receptor Signalling Cascade and the Quorum Sensing System in the marine bacterium *Vibrio Fischeri*. The former system is known to play a key role in tumour cell proliferation, angiogenesis and metastasis, being a key biological target for the development of novel anticancer therapies. Whereas the latter phenomenon is one of the best known quorum sensing systems; it consists in a gene regulation system that allows an entire population of bacterial cells to communicate in order to regulate the expression of specific genes involved in the production of light in a coordinated way depending on the size of the population.

The paper is organised as follows. In the next section we introduce P systems. In section 3 two strategies for the evolution of P systems are discussed. A study of EGFR Signalling Cascade and of the Quorum Sensing System in *Vibrio Fischeri* are given in sections 4 and 5. Finally, conclusions are presented in the last section.

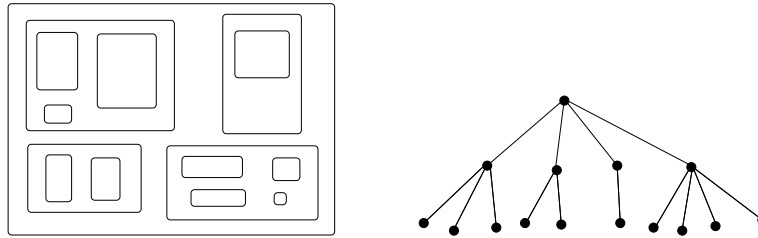
## 2 P systems as a Computational Modelling Tool for Systems Biology

A P system is usually defined as a hierarchical arrangement of a number of membranes identifying a corresponding number of regions inside the system, and with these regions having associated a finite multiset of objects and a finite set of rules. In what follows we give a precise definition of the main ingredients of a P system.

### Definition 1 (membrane structure).

*A membrane structure is a hierarchical arrangement of membranes where all the membranes but one must be included in a unique main membrane, which defines the boundary of the system with respect to the external environment. This particular membrane is called skin membrane. The membrane structure can be represented formally, as a rooted tree, where the nodes are called membranes, the root is called skin, and the relationship of membrane being inside another one is represented by the relationship of the node being the descendent of another one.*

Informally we can represent a membrane structure using Venn diagrams.



**Fig. 1.** A membrane structure represented using a Venn Diagram and a rooted tree

Rules of many different forms have been considered for P systems in order to encode the operation of modifying the objects inside the membrane, moving objects from one region to the other, dissolving, creating, dividing membranes etc. Here, in order to capture the features of most of these rules, we consider rules of the form:

$$u[v]_l \rightarrow u'[v']_l \quad (1)$$

with  $u, v, u', v'$  some finite multisets of objects and  $l$  the label of a membrane. These rules are multiset rewriting rules that operate on both sides of the membranes, that is, a multiset  $u$  placed outside a membrane labelled by  $l$  and a multiset  $v$  placed inside the same membrane can be simultaneously replaced by

a multiset  $u'$  and a multiset  $v'$  respectively. In this way, we are able to capture in a concise way the features of both the communication rules and the transformation rules considered in [4]. Moreover, rules like (1) allow us to express any sort of interactions occurring at the membrane level, and, in particular, they are useful to model the binding of a signal molecule to its corresponding receptor that occurs at the cell-surface level.

We generalise this concept by associating to each rule a boolean predicate  $\pi$  expressing a generic property over the objects contained inside a membrane and the objects contained in the surrounding region or in one of the regions that exists inside the current one. Such a predicate  $\pi$  is meant to specify a condition that need to be satisfied to make the rule applicable inside a given membrane. Finally, we associate to each rule a finite set of attributes which are meant to capture the quantitative aspects that are often necessary to characterise the *reality* of the phenomenon to be modelled. The necessity of taking into account these quantitative aspects has been made clear in a few recent application of P systems to the modelling of biological systems [3, 5–7].

Therefore, we introduce the following notion of program as the basic feature describing a generic process occurring inside a membrane.

**Definition 2 (program).** *Let  $O$  be an alphabet for the objects and let  $L$  be an alphabet of labels. A program is a construct*

$$\langle \pi \gg u[v]_l \rightarrow u'[v']_l, A \rangle$$

*with  $u, v, u', v' \in O^*$  some finite multisets of objects,  $\pi$  a generic boolean predicate,  $l$  is a label from  $L$  and  $A$  a finite set of attributes associated with the rule.*

The predicate  $\pi$  is used to express a condition that needs to be satisfied in order to make the rule applicable inside a membrane. The set of attributes in  $A$  can instead be used to associate to each rule a kinetic constant [3, 6], a probability [2, 5], or a more general function returning the number of occurrences of the multisets  $u, v$  to be consumed and the number of occurrences of the multisets  $u', v'$  to be produced. As well as this, the attributes might be used to associate to a rule some *side-effect* in order to alter other properties of the membrane where the rule is applied.

Our notion of program bears some similarities with, and is somehow inspired by, the notion of attribute grammars used for syntax-directed language translation and automatic code generation [1], as well as the notion of parametric L systems augmented with C code used for modelling plant growth and development [10]. In a sense, our programs also resemble the concept of guarded commands for non-deterministic programming introduced by Dijkstra in 1975 which has then led to CCS, CSP and the modern theory of concurrent systems based on  $\pi$ -calculus [12].

Now, we can define a P system by simply associating a finite multiset of objects to each membrane in a given membrane structure and by considering a finite set of programs to make these objects evolve from one configuration to the other.

**Definition 3 (P system).** A P system is a construct

$$\Pi = (O, L, \mu, M_1, M_2, \dots, M_n, R_1, \dots, R_n)$$

where:

- $O$  is a finite alphabet of symbols representing objects;
- $L$  is a finite alphabet of symbols representing labels for the compartments;
- $\mu$  is a membrane structure containing  $n \geq 1$  membranes labelled with elements from  $L$ ;
- $M_i = (w_i, l_i)$ , for each  $1 \leq i \leq n$ , is the initial configuration of membrane  $i$  with  $l_i \in L$  and  $w_i \in O^*$  a finite multiset of objects;
- $R_i$ , for each  $1 \leq i \leq n$ , is a finite set of programs in membrane  $i$  of the form specified in Definition 2 with objects in  $O$  and labels in  $L$ .

Thus, the initial configuration of each membrane  $i$ , with  $1 \leq i \leq n$ , is given by a finite multiset of objects from  $O$  and by a label from  $L$ . A program consists of a rule of the form  $u[v]_{l_i} \rightarrow u'[v']_{l_i}$ , with  $l_i$  the label of membrane  $i$ ; this program may be either associated with the membrane labelled with  $l_i$  or in the membrane surrounding it. Moreover, we can now precisely say that the evaluation of the predicate  $\pi$  in such a program must be done by considering both the content of membrane  $i$  and the content of the membrane  $upper(\mu, i)$  (i.e., the membrane that directly contains membrane  $i$ ). The evaluation of the predicate  $\pi$ , which has to be done before the application of the rule inside membrane  $i$ , is then denoted by  $\pi(i, upper(\mu, i))$ .

P systems are usually considered as being distributed maximal parallel multiset rewriting systems [16]. That is, in each step, for each membrane, a maximal set of programs to be applied is non-deterministically selected by making sure that no further programs can be applied to the objects left inside the membranes. On the other hand, a few recent works, [3, 5–7], have addressed the issue of introducing new strategies for the application of the programs where the set of programs to be applied in any step is not maximal, but it is somehow bounded. The reasons for the introduction of new derivation/evolution strategies may be different, but, in the context of modelling biological systems, one can say that restrictions to maximal parallelism are often required in order to close the gap between the abstractness of the model and the *reality* of the phenomenon to be modelled. In this work we present two novel strategies for the evolution of P systems substantially different from the ones previously introduced.

### 3 Two Strategies for the Evolution of P systems

At the microscopic level of functioning of cellular processes the interactions between molecules follow the laws of physics. A fundamental result of theoretical statistical physics is the famous  $\sqrt{n}$  law [11], which says that randomness or fluctuations in a system is inversely proportional to the square root of the number of particles. As a result systems with low number of molecules show high fluctuations; whereas the evolution of systems with high concentrations of reactants

are more deterministic. Having this in mind we introduce two strategies for the evolution of P systems, a stochastic strategy based on Gillespie's Algorithm and a deterministic strategy which will be valid for systems with high number of particles.

Gillespie's algorithm [9] (see also [8] for some recent improvements) provides an exact method for the stochastic simulation of systems of bio-chemical reactions; the validity of the method is rigorously proved and it has been already successfully used to simulate various biochemical processes [11]. As well as this, the Gillespie's algorithm is used in the implementation of stochastic  $\pi$ -calculus [18, 25], and in its application to the modelling of biological systems [19]. Here we present an extension of the classical Gillespie's algorithm called *Multi-compartmental Gillespie Algorithm*. This method is developed by taking into account the fact that, with respect to the original algorithm where only one volume is studied, in P systems we have a membrane structure delimiting different regions or compartments, each one can be seen as a volume with its own set of rules, besides the application of a rule inside a compartment can also affect the content of another one; for example the application of a communication rule. Specifically, let  $\Pi = (O, Lab, \mu, M_1, M_2, \dots, M_n, R_1, \dots, R_n)$  be a P system as specified in Definition 3 with the membranes  $M_i = (w_i, L_i)$  and the programs  $R_i$ ,  $1 \leq i \leq n$ . The set  $R_i$  of programs that are active inside membrane contains elements of the form  $(j, \pi_j, r_j, p_j, k_j)$  where:

- $j$  is the index of a program from  $R_i$ ;
- $\pi_j$  is the predicate; in this section this will be always true and will be omitted;
- $r_j$  is the boundary rule contained in the program  $j$ ;
- $p_j$  is the probability of the rule contained in the program  $j$  to be applied in the next step of evolution; this probability is computed by multiplying a stochastic constant  $k_j$ , specifically associated with program  $j$ , by the number of possible combinations of the objects present on the left-side of the rules with respect to the multiset  $w_i$  (or the multiset  $w_{i'}$ , with  $i' = upper(\mu, i)$ ) - the current content of membrane  $i$  ( $i'$ ).

First, each membrane  $i$  will be considered to be a compartment enclosing a volume, therefore the index of the next program to be used inside membrane  $i$  and its waiting time will be computed using the classical Gillespie's algorithm which we recall below:

1. calculate  $a_0 = \sum p_j$ , for all  $(j, r_j, p_j, k_j) \in R_i$ ;
2. generate two random numbers  $r_1$  and  $r_2$  uniformly distributed over the unit interval  $(0, 1)$ ;
3. calculate the waiting time for the next reaction as  $\tau_i = \frac{1}{a_0} \ln(\frac{1}{r_1})$ ;
4. take the index  $j$ , of the program such that  $\sum_{k=1}^{j-1} p_k < r_2 a_0 \leq \sum_{k=1}^j p_k$ ;
5. return the triple  $(\tau_i, j, i)$ .

Notice that the larger the stochastic constant of a rule and the number of occurrences of the objects placed on the left-side of the rule inside a membrane are, the greater the chance that a given rule will be applied in the next step of the simulation. There is no constant time-step in the simulation. The time-step is determined in every iteration and it takes different values depending on the configuration of the system.

Next, the *Multi-compartmental Gillespie's Algorithm* is described in detail:

- **Initialisation**

- set time of the simulation  $t = 0$ ;
- for each membrane  $i$  in  $\mu$  compute a triple  $(\tau_i, j, i)$  by using the procedure described above; construct a list containing all such triples;
- sort the list of triple  $(\tau_i, j, i)$  according to  $\tau_i$ ;

- **Iteration**

- extract the first triple,  $(\tau_m, j, m)$  from the list;
- set time of the simulation  $t = t + \tau_m$ ;
- update the waiting time for the rest of the triples in the list by subtracting  $\tau_m$ ;
- apply the rule contained in the program  $j$  only once changing the number of objects in the membranes affected by the application of the rule;
- for each membrane  $m'$  affected by the application of the rule remove the corresponding triple  $(\tau'_{m'}, j', m')$  from the list;
- for each membrane  $m'$  affected by the application of the rule  $j$  re-run the Gillespie algorithm for the new context in  $m'$  to obtain  $(\tau''_{m'}, j'', m')$ , the next program  $j''$ , to be used inside membrane  $m'$  and its waiting time  $\tau''_{m'}$ ;
- add the new triples  $(\tau''_{m'}, j'', m')$  in the list and sort this list according to each waiting time and iterate the process.

- **Termination**

- Terminate simulation when time of the simulation  $t$  reaches or exceeds a preset maximal time of simulation.

Therefore, in this approach, the waiting time computed by the Gillespie's algorithm is used to select the membranes which are allowed to evolve in the next step of computation. Specifically, in each step, the membranes associated to programs with the same minimal waiting time are selected to evolve by means of the corresponding rules. Moreover, since the application of a rule can affect more than one membrane at the same time (e.g., some objects may be moved from one place to another), we need to reconsider a new program for each one of these membranes by taking into account the new distribution of objects inside them.

As mentioned before in systems with high concentration of chemical substances deterministic approaches are valid; in what follows we present a deterministic *exact* strategy for the execution of the programs, that we will refer to as *Deterministic Waiting Times Algorithm*.

Given a P system, in this strategy, using *mass action law*, we associate a velocity,  $v_i$ , to every program  $i$  in each membrane  $j$  by multiplying the kinetic constant  $k_i$  by the multiplicities of the reactants. Then we compute the waiting

time for the first execution of the program as  $\tau_i = \frac{1}{v_i}$  and return the triple  $(\tau_i, i, j)$ .

Below we give a detailed description of the *Deterministic Waiting Times Algorithm*:

- **Initialisation**

- set time of the simulation  $t = 0$ ;
- for every program  $i$  associated with a membrane  $j$  in  $\mu$  compute the triple  $(\tau_i, i, j)$  by using the procedure described before; construct a list containing all such triples;
- sort the list of triple  $(\tau_i, i, j)$  according to  $\tau_i$ ;

- **Iteration**

- extract the first triple,  $(\tau_j, j, m)$  from the list;
- set time of the simulation  $t = t + \tau_j$ ;
- update the waiting time for the rest of the triples in the list by subtracting  $\tau_j$ ;
- apply the rule contained in the program  $j$  only once changing the number of objects in the membranes affected by the application of the rule;
- for each membrane  $m'$  affected by the application of the rule remove the corresponding all the triples  $(\tau_{j'}, j', m')$  from the list;
- for each membrane  $m'$  affected by the application of the rule  $j$  re-run the Gillespie algorithm for the new context in  $m'$  to obtain a triple  $(\tau_{j'}, j', m')$  for all the programs  $j'$  associated with the membrane  $m'$ ;
- add the new triples  $(\tau_{j'}, j', m')$  in the list and sort this list according to each waiting time and iterate the process.

- **Termination**

- Terminate simulation when time of the simulation  $t$  reaches or exceeds a preset maximal time of simulation.

Note that in this algorithm instead of associating a waiting time to a single program in each membrane (as it is the case in the *Multi-compartmental Gillespie's Algorithm*) every program in each membrane has a waiting time computed in a deterministic way that is used to determine the order in which the programs are executed. Also highlight that this is an *exact* method in the sense that we do not approximate infinitesimal intervals of time by  $\Delta t$  as it is the case in ODEs, but the time step varies across the evolution of the system and it is computed in each step depending on the current state of the system.

These two strategies have been implemented using Scilab, a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications [26].

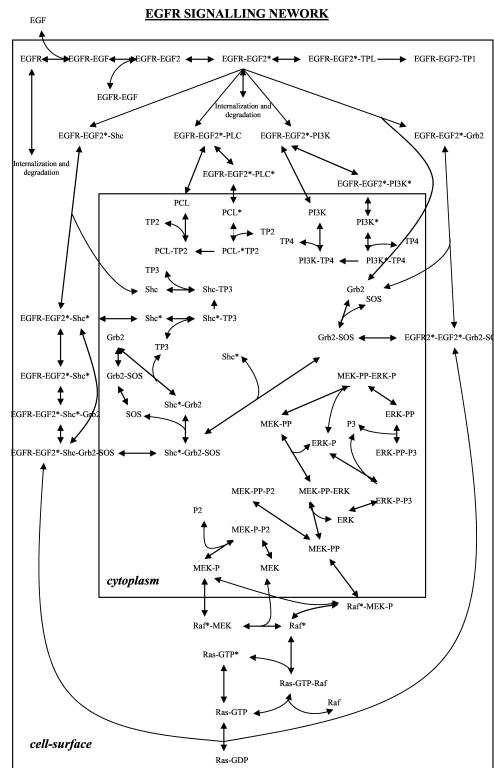
In the following two sections we provide two biological systems to illustrate these strategies. In section 4 we study the EGFR Signalling Cascade where the *Deterministic Waiting Times Algorithm* is suitable for describing its evolution; and in section 5 the Quorum Sensing System in the bacterium *Vibrio Fischeri* is used as an example where stochastic approaches like the *Multi-compartmental Gillespie's Algorithm* are necessary.



## 4 EGFR Signalling Cascade

The epidermal growth factor receptor (EGFR) belongs to the tyrosine kinase family of receptors. Binding of the epidermal growth factor (EGF) to the extracellular domain of EGFR induces receptor dimerisation and autophosphorylation of intracellular domains. Then a multitude of proteins are recruited starting a complex signalling cascade and the receptor follows a process of internalisation and degradation in endosomes. Two principal pathways lead to activation of Ras-GTP by hydrolysis of Ras-GDP. One of these pathways depends on the Src homology and collagen domain protein (Shc) and the other one is Shc-independent. Ras-GTP acts like a *switch* that stimulates the Mitogen Activated Protein (MAP) kinase cascade by phosphorylating the proteins Raf, MEK and ERK. Subsequently phosphorylated MEK and ERK regulates several cellular proteins and nuclear transcription factors. Disregulated EGFR expression, ligand production and signalling have been proved to have a strong association with tumourgenesis. As a result of this, EGFR has been identified as a key biological target for the development of novel anticancer therapies.

Below it is shown a detailed graphical representation of the signalling cascade.



We have developed a model of the signalling cascade described on the previous page using the following P system:

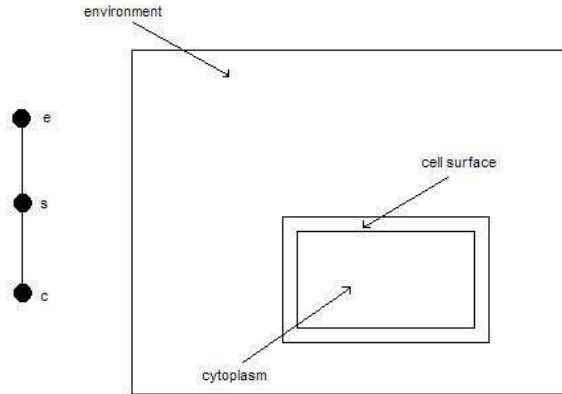
$$\Pi_{EGF} = (O, \{e, s, c\}, \mu, (w_1, e), (w_2, s), (w_3, c), \mathcal{R}_e, \mathcal{R}_s, \mathcal{R}_c)$$

Our model consists of more than 60 proteins and complexes of proteins and 160 chemical reactions. Due to the limitation of space we will not give all the details of the model. A complete description of  $\Pi_{EGF}$  with some supplementary information is available from the web page [www.gcn.us.es/egfr.pdf](http://www.gcn.us.es/egfr.pdf). In what follows we give an outline of our model.

• **Alphabet:** In the alphabet  $O$  we represent all the proteins and complexes of proteins that take part in the signalling cascade. Some of the objects from the alphabet and the chemical compounds that they represent are listed below.

Object	Protein or Complex
EGF	Epidermal Growth Factor
EGFR	EGF Receptor
EGFR-EGF	Receptor and EGF complex
EGFR-EGF <sub>2</sub>	Dimerized Receptor
EGFR-EGF <sub>2</sub> <sup>*</sup>	Phosphorylated Receptor
Shc	Src homology protein
EGFR-EGF <sub>2</sub> <sup>*</sup> -Shc	EGFR-EGF <sub>2</sub> <sup>*</sup> and Shc complex
⋮	⋮
MEK	Mitogenic external regulated kinase
ERK	External regulated Kinase

• **Membrane Structure:** In the EGFR Signalling Cascade there are three relevant regions, namely the *environment*, the *cell surface* and the *cytoplasm*. We represent them in the membrane structure as the membranes labelled with:  $e$  for the environment,  $s$  for the cell surface and  $c$  for the cytoplasm. A Venn diagram representation of the membrane structure can be seen below:



• **Initial Multisets:** In the initial multisets we represent the initial number of molecules (nM) of the chemical substances in the environment, the cell surface and the cytoplasm. These estimations has been obtained from [13, 20].

$$\begin{aligned} w_1 &= \{EGF^{200}\} \\ w_2 &= \{EGFR^{250}, Ras-GDP^{200}\} \\ w_3 &= \{Shc^{250}, PLC_\gamma^{150}, PI3K^{50}, SOS^{40}, Grb2^{80}, TP_1^{100}, TP_2^{450}, TP_3^{450}, TP_4^{125}, \\ &\quad Raf^{80}, MEK^{400}, ERK^{400}, P_1^{80}, P_2^{80}, P_3^{300}\} \end{aligned}$$

• **Programs:** In the programs we model the 160 chemical reactions which form the signalling cascade. Due to the limitation of space only a few programs representing some of the reactions will be presented in this paper. For a detailed enumeration of all the programs, their attributes and the references from where these parameters were taken see the supplementary information available from [www.gcn.us.es/egfr.pdf](http://www.gcn.us.es/egfr.pdf).

As it can be seen in the initial multisets specified before, in the system of the EGFR Signalling Cascade the number of molecules is quite large, as a consequence of the  $\sqrt{n}$  law important fluctuations and stochastic behaviour are not expected in the evolution of the system. Because of this we have chosen the *Deterministic Waiting Times Algorithm* as the strategy for the evolution of the P system  $\Pi_{EGF}$ .

In what follows we show two examples of programs from the set of programs associated with each membrane.

The set of programs associated with the environment,  $\mathcal{R}_e$ , consists only of one program which models the binding of the signal,  $EGF$ , to the receptor  $EGFR$ .

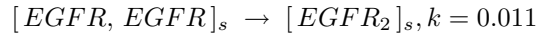


We assume that the previous program is applicable when there are signals and receptors available and so, the predicate  $\pi$  is omitted. The meaning of the previous rule is the following: the object  $EGF$  in the membrane containing the membrane with label  $s$  (the environment), and the object  $EGFR$  inside the membrane with label  $s$  (the cell surface) are replaced with the object  $EGFR-EGF$  in the membrane with label  $s$ ; this object represents the complex receptor-signal on the cell surface. As attributes we associate the kinetic constant  $k$ , which measures the affinity between the signal and the receptor.

The *Deterministic Waiting Times Algorithm* is used in the evolution of the system and the waiting time associated to this program will be computed using the next formula:

$$\tau = \frac{1}{0.003|EGF||EGFR|}$$

One example from the set of programs  $\mathcal{R}_s$  associated to the cell surface is the dimerisation of the receptor, that is the formation of a complex consisting of two receptors:

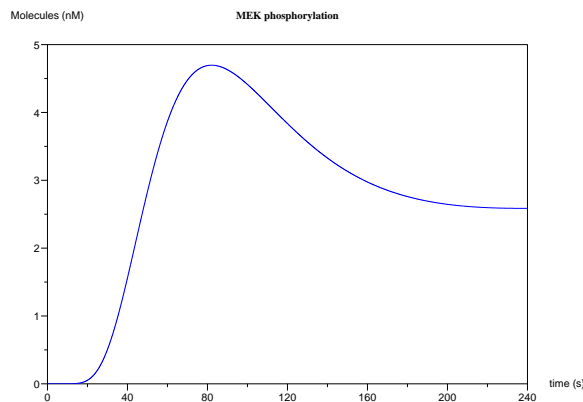
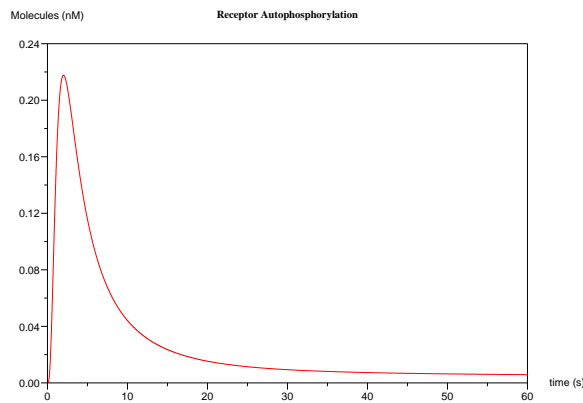


When this program is executed two objects  $EGFR$  representing receptors are replaced with one object  $EGFR_2$ , representing a complex formed with two receptors, in the membrane with label  $s$ , the cell surface. The kinetic constant  $k$  is used to compute the waiting time:

$$\tau = \frac{1}{0.011|EGFR|^2}$$

Using the software mentioned in the previous section and developed in Scilab we run some experiments; in what follows we present some of the results obtained.

Below it is depicted the evolution of the number of autophosphorylated receptors and the number of doubled phosphorylated MEK (Mitogen External Kinase), one of the target proteins of the signalling cascade that regulates some nuclear transcription factors involved in the cell division.

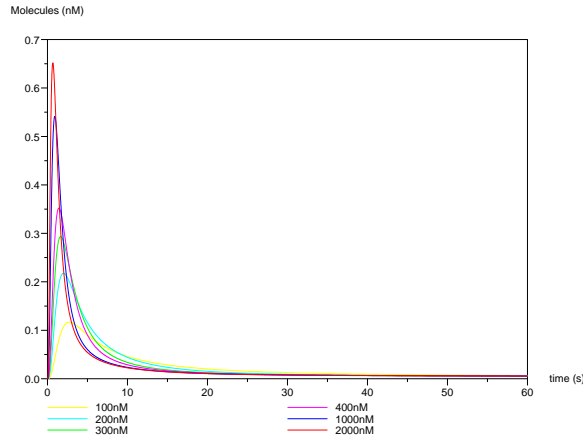


Note that the activation of the receptor is very fast reaching its maximum within the first 5 seconds and then it decays fast to very low levels; on the other hand

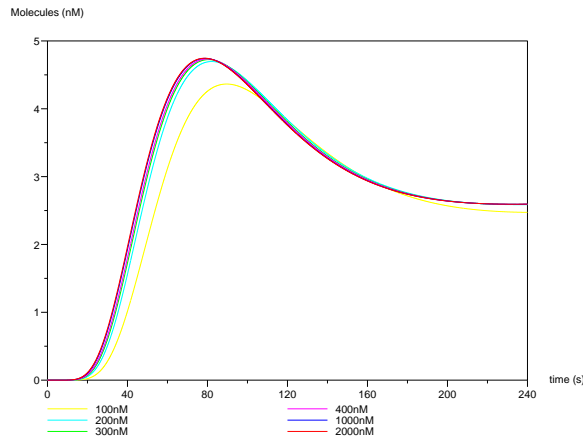
the number of doubled phosphorylated MEK is more sustained around 3 nM. These results agree well with empirical observations, see [13, 20].

In tumours it has been reported an overexpression of signals EGF in the environment and of receptors, EGFR, on the cell surface of cancerous cells. Here we investigate the effect of different EGF concentrations and number of receptors on the signalling cascade.

First, we study the effect on the evolution of the number of autophosphorylated receptors and double phosphorylated MEK of a range of signals, EGF, from 100 nM to 2000 nM.

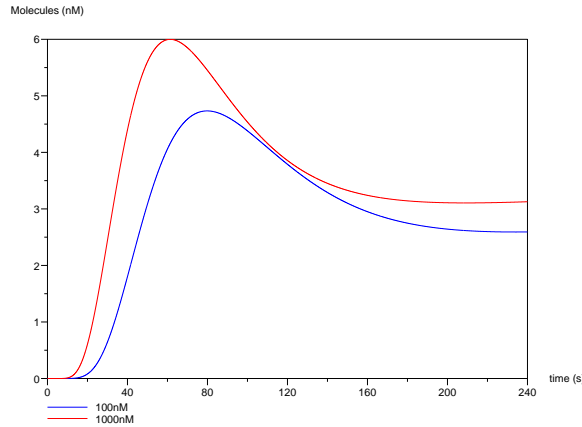


It can be seen that the receptor autophosphorylation is clearly concentration dependent showing different peaks for different number of signals in the environment. According to the variance in the receptor activation it is intuitive to expect different cell responses to different EGF concentrations. Here we will show that this is not the case.



Observe, in the graph depicted on the previous page, that the number of doubled phosphorylated MEK does not depend on the number of signals in the environment. This shows the surprising robustness of the signalling cascade with regard to the number of signals from outside due to EGF concentration. The signal is either attenuated or amplified to get the same concentration of one of the most relevant kinases in the signalling cascade, MEK. Note that after 100 seconds, when the response gets sustained, the lines representing the response to different external EGF concentrations are identical.

Now we analyse the effect on the dynamics of the signalling cascade of different numbers of receptors on the cell surface.



Above it is shown the evolution of the number of doubled phosphorylated MEK when there is 100 nM and 1000 nM of receptors on the cell surface. Note that now the response is considerably different, being the number of activated MEK greater when there is an overexpression of receptors on the cell surface. As a consequence of this high number of activated MEK the cells will undergo an uncontrolled process of proliferation.

The key role played by the overexpression of EGFR on the uncontrolled growth of tumours has been reported before, as a consequence of this, EGFR is one of the main biological targets for the development of novel anticancer therapies.

Finally, stress that for this system we have used a deterministic approach obtaining results that map well experimental data. This is not always the case, in the next section we analyse a system where a stochastic approach is necessary to describe properly its behaviour.

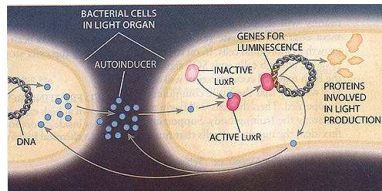
## 5 Quorum Sensing System in the Bacterium *Vibrio Fischeri*

Quorum sensing is a cell density dependent gene regulation system that allows an entire population of bacterial cells to communicate in order to regulate the

expression of certain or specific genes in a coordinated way depending on the size of the population. In this section we present a model of the Quorum Sensing System in the marine bacterium *Vibrio fischeri* using P systems. In this framework each bacterium and the environment are represented by membranes. This allows us to examine the individual behaviour of each bacterium as an agent as well as the behaviour of the colony as a whole and the processes of swarming and recruitment.

The marine bacterium *Vibrio fischeri* exists naturally either in a free-living planktonic state or as a symbiont of certain luminescent squid. The bacteria colonise specialised light organs in the squid, which cause it to luminesce. Luminescence in the squid is thought to be involved in the attraction of prey, camouflage and communication between different individuals. The source of the luminescence is the bacteria themselves. The bacteria only luminesce when colonising the light organs and do not emit light when in the free-living state.

The Quorum Sensing System in *Vibrio Fischeri* relies on the synthesis, accumulation and subsequent sensing of a signal molecule, 3-oxo-C6-HSL, an N-acyl homoserine lactone or AHL (we will call it OHHL). When only a small number of bacteria are present the signal is produced by the bacteria at a low level. OHHL diffuses out of the bacterial cells and into the surrounding environment. At high cell density the signal accumulates in the area surrounding the bacteria and can also diffuse to the inside of the bacterial cells. The signal is able to interact with the LuxR protein to form the complex LuxR-OHHL. This complex binds to a region of DNA called the Lux Box causing the transcription of the luminescence genes, a small cluster of 5 genes, luxCDABE. Adjacent to this cluster are two regulatory genes for the transcription of LuxR and OHHL. In this sense OHHL and LuxR are said to be autoinducer because they activate their own synthesis.



The bacteria are effectively communicating, as a single bacterium is able to detect and respond to signals produced by the surrounding bacteria. Bacteria sense their cell density by measuring the amount of signal present; quorum sensing can therefore explain why the bacteria are dark when in the free living planktonic state at low cell density and light when colonising the light organ of squid at high cell density. A large number of Gram negative bacteria have been found to have AHL-based quorum sensing systems similar to *Vibrio fischeri*.

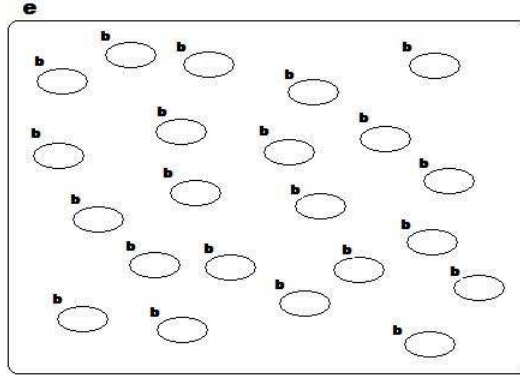
In what follows we present a model of the Quorum Sensing System in *Vibrio fischeri* using a parametric P system,  $\Pi_{V_f}(N)$ ; where  $N$  represents the number of bacteria in the colony. We will study the behaviour of such colony placed inside a *single* environment by examining the evolution of the following P system:

$$\Pi_{V_f}(N) = (O, \{e, b\}, \mu, (w_1, e), (w_2, b) \dots, (w_{N+1}, b), \mathcal{R}_b, \mathcal{R}_e) \quad \text{where:}$$

- (1) **Alphabet:** In the alphabet we represent the signal OHHL, the protein LuxR, the complex protein-signal, the regulatory region LuxBox and the regulatory region occupied by the complex.

$$O = \{\text{OHHL}, \text{LuxR}, \text{LuxR-OHHL}, \text{LuxBox}, \text{LuxBox-LuxR-OHHL}\}$$

- (2) **Membrane Structure and Labels:** In the Quorum Sensing System of *Vibrio fischeri* there are two relevant regions, namely the environment and the bacteria. The environment will be represented by the membrane labelled with  $e$  and each bacterium is represented by a membrane with label  $b$ . We can represent the membrane structure  $\mu$  as a Venn diagram as follows:



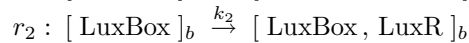
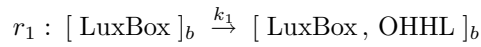
- (3) **Initial Multisets:** In the initial multisets we represent the initial conditions of the system. We are interested in examining how bacteria communicate to coordinate their behaviours and how the system moves from a downregulated state, where the protein and the signal are produced at basal rates, to an upregulated state, where the bacteria produce light. Therefore, in the initial multisets we will suppose that there is nothing in the environment and in the bacteria we will only have the genome (LuxBox) to start the production of the signal and protein at basal rates.

$$w_1 = \emptyset, w_i = \{\text{LuxBox}\} \quad 2 \leq i \leq N + 1$$

- (4) **Programs:** In the programs we model the chemical reactions forming the Quorum Sensing System. Next we specify the set of programs associated with the bacteria,  $\mathcal{R}_b$ , and with the environment,  $\mathcal{R}_e$ ; we also describe briefly the chemical reactions they represent.

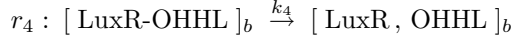
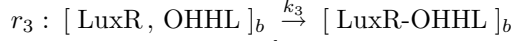
- Set of programs associated with the bacteria,  $\mathcal{R}_b$ :

In an unstressed bacterium the transcription of the signal OHHL and the protein LuxR takes place at basal rates.

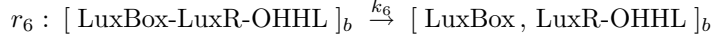
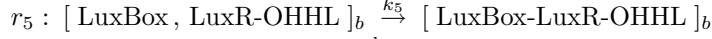




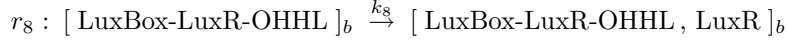
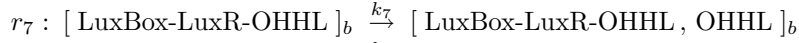
The protein LuxR acts as a receptor and OHHL as its ligand. Both together form the complex LuxR-OHHL which in turn can dissociate into OHHL and LuxR again.



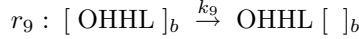
The complex LuxR-OHHL acts as a transcription factor binding to the regulatory region of the bacterium DNA called LuxBox. The complex LuxR-OHHL can also dissociate from the LuxBox.



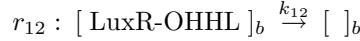
The binding of the complex LuxR-OHHL to the LuxBox produce a massive increase in the transcription of the signal OHHL and of the protein LuxR.



OHHL can diffuse outside the bacterium and accumulate in the environment.

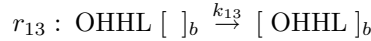


OHHL, LuxR and the complex LuxR-OHHL undergo a process of degradation in the bacterium.



- Set of programs associated with the environment,  $\mathcal{R}_e$ :

The signal OHHL in the environment can diffuse inside the bacteria and undergo a process of degradation.



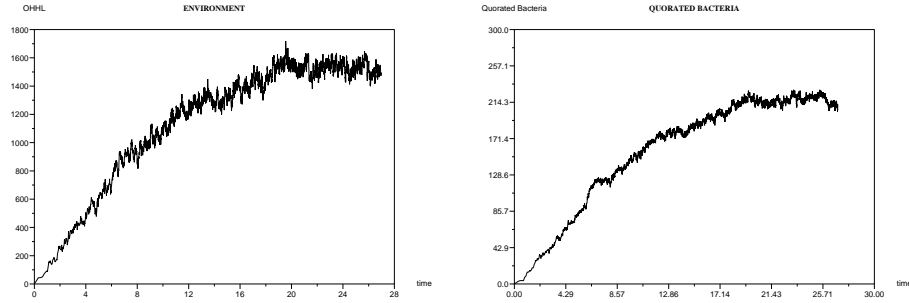
In order to implement our model we have chosen the following set of parameters,  $k_1 = 2, k_2 = 2, k_3 = 9, k_4 = 1, k_5 = 10, k_6 = 2, k_7 = 250, k_8 = 200, k_9 = 1, k_{10} = 50, k_{11} = 30, k_{12} = 15, k_{13} = 20, k_{14} = 20$ , these parameters has been taken or suggested by the literature listed in [21].

As it can be seen in the initial multisets, the Quorum Sensing System has a very small number of molecules. At the beginning of the evolution there is only a single molecule in each bacterium representing its genome. Therefore as a consequence of the  $\sqrt{n}$  law stochastic behaviour is expected. To check if this is true we have used the two strategies introduced in section 3 for the evolution of the P system  $\Pi_{V_f}(N)$ .

First, we use the *Multi-compartmental Gillespie's Algorithm*. We have studied the behaviour of the system for two populations of different size to examine how

bacteria can sense the number of bacteria in the population and produce light only when the number of individuals is big enough.

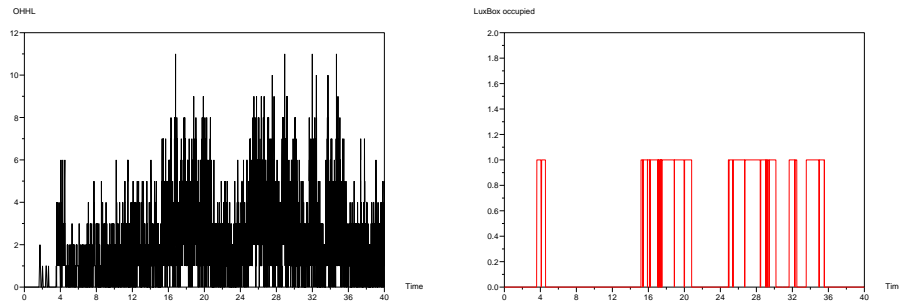
To start with we have considered a population of 300 bacteria. We examine the evolution over time of the number of quorated bacteria <sup>1</sup> (right) and the number of signals (OHHL) in the environment (left).



Observe that the signal, OHHL, accumulates in the environment until saturation and then, when this threshold is reached, bacteria are able to detect that the size of the population is big enough. At the beginning, a few bacteria get quorated and then they accelerate a process of recruitment that makes the whole population behave in a coordinated way.

There exists a correlation between the number of signal in the environment and the number of quorated bacteria such that, when the number of signal in the environment drops, so does the number of quorated bacteria and when the signal goes up it produces a recruitment of more bacteria.

In our approach the behaviour of each individual in the colony can be tracked. Below, for one bacterium it is shown the correlation between the number of signal inside the bacterium (left) and the occupation of the LuxBox by the complex (right), which represents that the bacterium has been quorated.

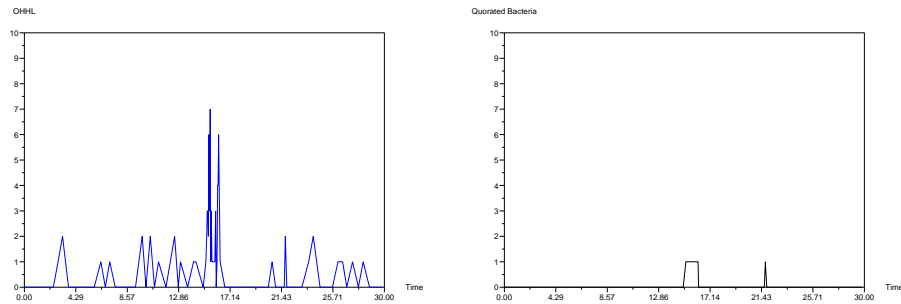


Above it can be seen that the number of signal molecules inside the bacterium has to exceed a threshold of approximately 6 molecules in order to recruit the

<sup>1</sup> We will say that a bacterium is quorated if the LuxBox in this bacterium is occupied by the complex producing the transcription of the enzymes involved in the production of light.

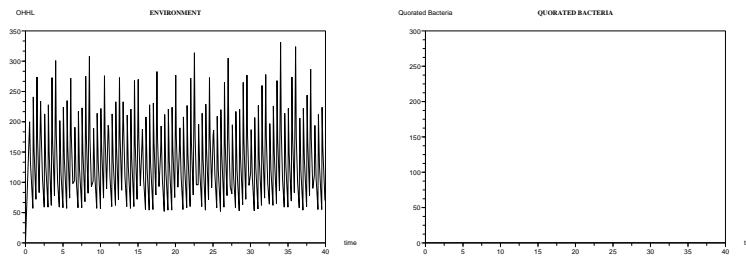
bacterium. Observe that when the number of molecules is greater than 6 the LuxBox is occupied, that is, the bacterium is quorated or upregulated but when there is less than six signals the bacterium switches off the system and stops producing light.

Now, in order to study how bacteria can sense the number of individuals in the colony and get quorated only when the size of the colony is big enough we have examine the behaviour of a population of only 10 bacteria.



In this case the signal does not accumulate in the environment and no recruitment process takes place. Only one of the bacteria guessed wrong the size of the population and got upregulated. But then, after sensing that the signal does not accumulate in the environment, it switched off its systems.

Finally, in order to show the inherently random character of this system, we will show the results obtained using the *Deterministic Waiting Times Algorithm* with the same set of parameters.



Above it is depicted the number of signals (left) and the number of quorated bacteria (right) for a population of 300 bacteria. Note that using the *Deterministic Waiting Times Algorithm* the signal does not accumulate and not even a single bacterium got quorated. These results do not agree with empirical results which show that at high cell densities the signal saturates the environment and the colony of bacteria produces light. Therefore here we have an example of a system with intrinsic random behaviour where stochastic approaches like the Multicompartmental Gillespie's Algorithm are necessary.

## 6 Conclusions

In this paper we have presented P systems as a new computational modelling tool for systems biology; stressing their main characteristics, consideration of the key role played by membranes in the structure and functioning of the cells and the representation of the discrete character of the quantity of components in bio-systems. P systems are also general specification of the biological phenomena that can be evolved using different strategies/algorithms. In this work two novel strategies have been introduced; *Multi-compartmental Gillespie's Algorithm* based on the well known Gillespie's Algorithm but running on more than one compartment; and *Deterministic Waiting Times Algorithm*, an *exact* deterministic method.

Two different biological phenomena have been used to illustrate these strategies. First, in the EGFR Signalling Cascade the *Deterministic Waiting Times Algorithm* was used and second in the Quorum Sensing System in the bacterium *Vibrio Fischeri* the *Multi-compartmental Gillespie's Algorithm* was necessary in order to obtain results in accordance with empirical observations.

The fact that our results agree well with previously formulated hypothesis shows the reliability of P systems as computational modelling tools to produce prediction and perhaps as the field evolves they will be able to produce plausible predictions.

## Acknowledgement

This work is supported by Ministerio de Ciencia y Tecnología of Spain, by *Plan Nacional de I+D+I (2000–2003)* (TIC2002-04220-C03-01), cofinanced by FEDER funds, and by a FPU fellowship from the Ministerio de Ciencia y Tecnología of Spain.

## References

1. Aho, A.V., Sethi, R., Ulmann, J.D. (1986). *Compilers: Principles, Techniques, and Tools*. Addison-Wesley.
2. Ardelean, I., Cavaliere, M. (2003). Playing with a Probabilistic P Simulator: Mathematical and Biological Problems. In: *Brainstorming Week on Membrane Computing, Tarragona, Feb 5-11 2003* (Cavaliere, M., Martin-Vide, C., Păun, Gh., eds.). Tech. Rep. **26/03**, Universitat Rovira i Virgili, Tarragona, Spain, 37–45.
3. Bernardini, F., Gheorghe, M., Muniyandi, R.C., Krasnogor, N., Pérez-Jiménez, M.J., Romero-Campero, F.J. (2005). On P Systems as a Modelling Tool for Biological Systems. In: *Pre-Proceedings of WMC6 - Vienna 2005*, 193–213.
4. Bernardini, F., Manca, V. (2003). P Systems with Boundary Rules. In: [17], 107–118.
5. Besozzi, D. (2004). *Computational and Modelling Power of P systems*, Ph.D. Thesis, Università degli Studi di Milano, Milan, Italy.
6. Bianco, L., Fontana, F., Franco, G., Manca, V. (2005). P Systems for Biological Dynamics. In: *Applications of Membrane Computing* (Ciobanu, G., Păun, Gh., Pérez-Jiménez, M.J., eds.), Springer-Verlag, Berlin, Heidelberg, New York, 81–126.

7. Bianco, L., Fontana, F., Manca, V. (2005). P Systems and the Modelling of Biochemical Oscillation. In: *Pre-Proceedings of WMC6 - Vienna 2005*, 214–225.
8. Gibson, M.A., Bruck, J., (2000). Efficient Exact Stochastic Simulation of Chemical Systems with Many Species and Many Channels. *Journal of Physical Chemistry*, **104**, 25, 1876–1889.
9. Gillespie, D.T. (1977). Exact Stochastic Simulation of Coupled Chemical Reactions. *The Journal of Physical Chemistry*, **81**, 25, 2340–2361.
10. Karwowski, R., Prusinkiewicz, P. (2003). Design and Implementation of the L+C Modelling Language. *Electronics Notes in Theoretical Computer Science*, **82**, 2, 1–19.
11. Meng, T.C., Somani S., Dhar, P. (2004). Modelling and Simulation of Biological Systems with Stochasticity. *In Silico Biology*, **4**, 0024.
12. Milner, R. (1999). *Communicating and Mobile System: The  $\pi$ -Calculus*. Cambridge University Press.
13. Moehren G. et al (2002). Temperature Dependence of the Epidermal Growth Factor Receptor Signaling Network Can Be Accounted for by a Kinetic Model, *Biochemistry* **41** 306–320.
14. Păun, A.; Păun, Gh.: The Power of Communication: P Systems with Symport/Antiport, *New Generation Computing*, **20**, 3 (2002), 295–305.
15. Păun, Gh.: Computing with Membranes, *Journal of Computer and System Sciences*, **61**(1) (2000) 108 – 143.
16. Păun, Gh. (2002). *Membrane Computing. An Introduction*. Springer-Verlag, Berlin, Heidelberg, New York.
17. Păun, Gh., Rozenberg, G., Salomaa, A., Zandron, C., eds. (2003). Membrane Computing. International Workshop, WMC-CdeA 02, Curtea de Arges, Romania, August 19-23, 2002. Revised Papers. *Lecture Notes in Computer Science*, **2597**, Springer-Verlag, Berlin, Heidelberg, New York.
18. Philips, A., Cardelli. L. (2004). A Correct Abstract Machine for the Stochastic Pi-calculus. *Electronical Notes in Theoretical Computer Science*, to appear.
19. Priami, C., Regev, A., Shapiro, E., Silverman, W. (2001). Application of a Stochastic Name-Passing Calculus to Representation and Simulation of Molecular Processes. *Information Processing Letters*, **80**, 25–31.
20. Schoeberl, B. et al. (2002). Computational Modeling of the Dynamics of the MAP Kinase Cascade Activated by Surface and Internalized EGF Receptors, *Nature Biotech.* **20** 370–375.
21. Nottingham Quorum Sensing Web Site <http://www.nottingham.ac.uk/quorum/>
22. ISI Web Site <http://esi-topics.com/erf/october2003.html>
23. Scilab Web Pages: <http://scilabsoft.inria.fr>.
24. The P Systems Web Page: <http://psystems.disco.unimib.it>.
25. The Stochastic Pi-Machine: <http://www.doc.ic.ac.uk/~anp/spim/>.
26. SciLab Web Site <http://scilabsoft.inria.fr/>