A Model of the Quorum Sensing System in Vibrio fischeri using P Systems

Mario J. Pérez-Jiménez, Francisco J. Romero-Campero*

Research Group on Natural Computing
Department of Computer Science and Artificial Intelligence
University of Seville, Avenida Reina Mercedes s/n, 41012 Sevilla, Spain
Phone Numbers: +34 954 557 952, +44 (0) 114 22 21974
email: {marper,fran}@us.es

Abstract. 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 paper we present a model of the Quorum Sensing System in Vibrio fischeri using a variant of membrane systems called P systems. In this framework each bacterium and the environment are represented by membranes and the rules are applied according to an extension of Gillespie's Algorithm called Multicompartmental Gillespie's Algorithm. This algorithm runs on more than one compartment and takes into account the disturbance produced when chemical substances diffuse from one compartment or region to another. Our approach allows us to examine the individual behaviour of each bacterium as an agent as well as the emergent behaviour of the colony as a whole and the processes of swarming and recruitment. Our simulations show that at low cell densities bacteria remain dark while at high cell densities some bacteria start to produce light and a recruitment process takes place that makes the whole colony of bacteria to emit light. Our computational modelling of quorum sensing could provide insights that may enable researchers to develop new applications where multiple agents need to robustly and efficiently coordinate their collective behaviour based only on a very limited information of the local environment.

Keywords: Quorum sensing, Vibrio fischeri, Membrane computing, Gillespie's Algorithm, Emergent Behaviour

* Corresponding Author

1 Introduction

Membrane Computing is an emergent branch of Natural Computing introduced by G. Păun in [18]. 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 [17] was mentioned in [27] as a highly cited paper in October 2003.

This new model of computation has been introduced with the aim of defining a computing device, called P system, which abstracts from the structure and the functioning of living cells. 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. The main results in this area show that P systems are a very powerful and efficient computational model mostly equivalent to Turing machines. Although most research in P systems concentrates on computational powers, it is natural to consider them as modelling tools for biological systems since they are inspired from the functioning of the living cell. Therefore, recently, they have been used to model biological phenomena within the framework of Systems Biology [2–4, 20], to specify artificial cell systems [23], and to describe self-assembly processes [6]. In this case P systems are not used as a computing paradigm, but rather as a formalism for describing the behaviour of the system to be modelled.

In this paper we use the so called Multicompartmental Gillespie's Algorithm introduced in [20]. This algorithm consists of an extension of the well known Gillespie's Algorithm [7] (see [8, 9] for some recent improvements). Gillespie's algorithm 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 [16]. Our algorithm 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 reactions. Moreover, some reactions can disturb the conditions of more than one compartment; for example when molecules move from one compartment to another.

We have used P systems to model the Quorum Sensing System in the marine bacterium Vibrio fischeri. Bacteria are generally considered to be independent unicellular organisms. However it has been observed that certain bacteria have a 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. This cell density dependent gene regulation system is referred to as Quorum Sensing, and it has been described as "the most consequential molecular microbiology story of the last decade" [13].

Up to now, most models of quorum sensing have used differential equations which focus on the description of the change of the average concentration of chemical compounds across the population. Here we use a new formalisation of

this phenomenon in a computational framework which focuses on the description of the behaviour of each individual specifying the compartmental structure of the system (membrane structure) and the chemical reactions (rules) that take place in different regions of the system. This new approach allows us to examine the behaviour of each individual as an agent as well as the emergent behaviour of the whole population as a result of the processes of swarming and recruitment; processes which can also be easily studied in our model. Besides, differential equations treat reactions as continuous fluxes of matter which is correct if there is a very large number of molecules present in the system and the reactions are fast. Several authors argue that this approach is not correct in systems involving very small numbers of molecules and slow reactions. This is the case of quorum sensing systems where a small number of signal molecules inside a bacterium act as transcription regulators binding to a single 'molecule' of the DNA regulatory region. In this case discrete and stochastic approaches are more accurate.

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. 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 diffusion, signalling at the cell surface and selective uptake of substances from the environment. In favour of our approach we also mention the easy understandability and programmability of this model, features which are not easily achieved in approaches which use differential equations.

The paper is organised as follows. P systems are presented as a computational modelling tool for Systems Biology in the next section. In section 3, the Multicompartmental Gillespie's algorithm is described. In section 4 a brief description of the Quorum Sensing System in Vibrio fischeri is given. The model of the Quorum Sensing System is developed in section 5. Results and discussions are exposed in the next section. Finally, conclusions are given in the last section.

2 P Systems as a Computational Modelling Tool for Systems Biology

In the structure and functioning of cells, membranes play an essential role. Cells are separated from the environment by means of a skin membrane, and they are internally compartmentalised by membranes. Inspired by these biological features Gh. Păun introduced P systems as an unconventional model of computation in [18]; for details and updated information on P systems we refer to [19, 28].

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. A P system can be thought as an individual agent and by considering many copies of the same P system placed inside another membrane, that will

be referred as environment, a colony or population of P systems can be studied [1]. Compartments delimited by membranes are considered to be well mixed and homogeneous regions, in this respect if the environment is too large to be considered homogeneous it will be partitioned into different membranes which will be the nodes of a graph whose edges represent how these membranes representing environments are connected.

In what follows we give a precise definition of the main components 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 inclusion of a membrane inside another one is represented by its node being the descendent of another one.*

Informally we can represent a membrane structure using Venn diagrams.

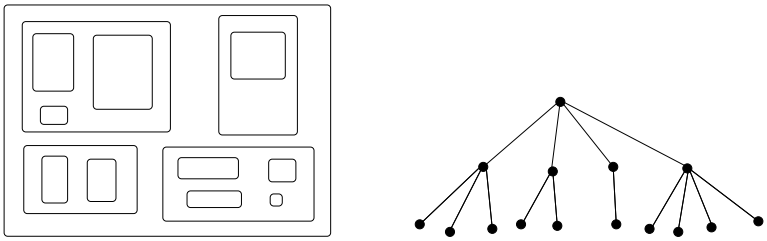

Fig. 1. *A membrane structure represented using a Venn Diagram and a rooted tree*

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

$$u\,[\,v\,]_l \rightarrow u'\,[\,v'\,]_l \tag{1}$$

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 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 with a multiset u' and a multiset v' respectively. 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 [20].

Definition 2 (multienvironment). *A multienvironment is a collection of membranes called environments and communication links between them. Formally, a multienvironment is a graph, $G = (V, S)$, whose nodes V are membranes representing environments. These environments are connected according to the edges S that define the links between them.*

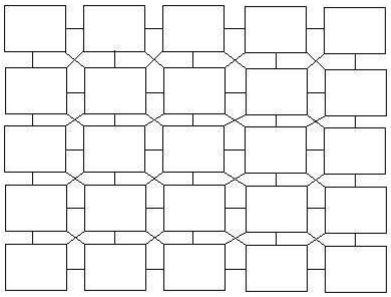

Fig. 2. A multienvironment with 25 different environments.

Each environment will be associated a set of rules of the form (1) and in the case of movement of different substances from one environment to one of the environments connected to it, the following type of rules will be used:

$$[\,u\,]_l - [\ ]_{l'} \rightarrow [\ ]_l - [\,u\,]_{l'} \tag{2}$$

These rules are multiset rewriting rules that operate on two environments, that is, a multiset u placed inside an environment labelled by l that is linked to another environment labelled by l' is removed from the first environment and placed in the second one. In this way, we are able to capture in a concise way the features of diffusion of signals in a large environment from a region to another.

In the environments besides multisets of objects representing chemical substances, we will place a certain number of copies of P systems. These P systems can move from one environment to another by applying rules of the form:

$$[\,[\,]_{l''}\,]_l - [\ ]_{l'} \rightarrow [\ ]_l - [\,[\,]_{l''}\,]_{l'} \tag{3}$$

When one rule of this type is applied a membrane with labelled l'' and all its content, objects and other membranes, is moved from an environment labelled by l to another one connected to it that must be labelled by l'.

We generalise the type of rules in (1), (2) and (3) by associating to each rule a boolean predicate π expressing a generic property over the objects and membranes involved in the rule. Such a predicate π 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.

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

Definition 3 (program). *Let O be an alphabet for the objects and let L be an alphabet of labels. A program is a construct*

$$\langle \pi >> r, A \rangle$$

with π a generic boolean predicate, r is a rule of one of the form (1), (2) or (3) with objects from O and labels from L; and A a finite set of attributes associated with the rule.

The predicate π 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 [2], a probability [20], 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.

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 4 (P system). *A P system is a construct*

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

where:

- *O is a finite alphabet of symbols representing objects;*
- *L is a finite alphabet of symbols representing labels for the compartments;*
- *μ 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 3 with objects in O and labels in L.*

Finally, to study the behaviour of a colony or population of P systems, a number of copies of the same P system is randomly distributed in the different environments of a *multienvironment*.

Definition 5 (Multienvironment P systems). *A multienvironment P system is a construct:*

$$ME = (\Sigma, H, G, E_1, \ldots, E_m, R_1, \ldots, R_m, \Pi, k)$$

where:

- Σ *is a finite alphabet of objects that can be present in the environments;*
- H *is a finite set of labels for the environments;*
- $G = (V, S)$ *is a graph whose nodes* $V = \{1, \ldots, m\}$ *represent environments labelled with elements from* H *and whose edges,* S*, define how the environments are linked;*
- $E_j = (w_j, l_j)$ *for each* $1 \leq j \leq m$*, is the initial configuration of the environment* j *with* $l_j \in H$ *and* $w_j \in \Sigma^*$ *a finite multiset of objects;*
- R_j *for each* $1 \leq j \leq m$*, is a finite set of programs in the environment* j *with objects in* Σ *and labels in* H*.*
- $\Pi = (O, L, \mu, M_1, M_2, \ldots, M_n, R_1, \ldots, R_n)$ *is a P system as in Definition 4.*
- $k \in \mathbb{N}$ *is the number of copies of the P system* Π *that will be distributed randomly in the different environments in the initial configuration of the system.*

3 Multicompartmental Gillespie's Algorithm

A fundamental result of theoretical statistical physics is the famous $\sqrt{n}$ law, which states that noise or fluctuation level in a system are inversely proportional to the square root of the number of particles. Note that the crucial factor is the number of particles, not the concentration. A system with few particles in a very small volume will result in a high concentration but also a large relative noise. Therefore, cellular systems with low number of molecules show high fluctuations; and the deterministic and continuous approach of differential equations is questionable. Instead stochastic and discrete approaches, like Gillespie's Algorithm, are more accurate.

Gillespie's algorithm [7] (see also [8, 9] 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 [16]. Here we will use an extension of the classical Gillespie's algorithm called *Multi-compartmental Gillespie Algorithm* that was introduced in [20]. 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; we also have compartments or membranes representing different regions in the environment. Each one of these compartments can be seen as a volume with its own set of rules, and the application of a rule inside a compartment can also affect the content of another one; for example the application of a rule that moves objects from one membrane to another.

Specifically, let $ME = (\Sigma, H, G, (w_1, l_1), \ldots, (w_m, l_m), R_1, \ldots, R_m, \Pi, k)$ be a multienvironment P system as specified in Definition 5 with
$\Pi = (O, L, \mu, (w'_1, l'_1), \ldots, (w'_n, l'_n), R'_1, \ldots, R'_n)$. The sets R_i $(1 \leq i \leq m)$ and R'_j $(1 \leq j \leq n)$ of programs that are active inside each membrane (including environments) contain elements of the form $(\nu, \pi_\nu, r_\nu, p_\nu, c_\nu)$ where:

- ν is the index of a program from R_i or R'_j;
- π_ν is the predicate; in this section this will be always true and will be omitted;

- r_ν is the rule contained in the program ν;
- p_ν is the probability of the rule contained in the program ν to be applied in the next step of evolution; this probability is computed by multiplying a stochastic constant c_ν, specifically associated as an attribute with program ν, by the number of possible combinations of the objects and membranes present on the left-side of the rules with respect to the multiset w_i (or the multiset $w_{i'}$ contained in the membrane outside membrane i) - the current content of membrane i (i').

Each membrane i (including environments) 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 $(j, r_j, p_j, c_j) \in R_i$;
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 (τ_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 compute a triple (τ_i, j, i) by using the procedure described above; construct a list containing all such triples;
  - sort the list of triples (τ_i, j, i) in decreasing order according to τ_i;
- **Iteration**
  - extract the first triple, (τ_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 τ_m;
  - apply the rule contained in the program j only once changing the number of objects and membranes 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. Note that in this, our approach differs from [22] where only one program is applied at each step without taking into account the rest of programs that are waiting to be applied in the other membranes, neither it is considered the disruption that the application of one program can produce in various membranes.

This algorithm has been implemented using Scilab, a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications [30].

In the following two sections we briefly describe the Quorum Sensing system in the marine bacterium Vibrio fischeri and a model using the formalism discussed on the previous section is presented.

4 Quorum Sensing System in Vibrio Fischeri

Bacteria are generally considered to be independent unicellular organisms. However it has been observed that certain bacteria, like the marine bacterium *Vibrio Fischeri*, exhibit coordinated behaviour which allows an entire population of bacteria to regulate the expression of certain or specific genes in a coordinated way depending on the size of the population. This cell density dependent gene regulation system is referred to as *Quorum Sensing*.

This phenomenon was first investigated in the marine bacterium *Vibrio fischeri*. This bacterium 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 refer to it as OHHL. When only a small number of bacteria are present these molecules are 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. A graphical representation of the system can be seen in figure 3.

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.

For a review on Quorum Sensing in general and the system in Vibrio fischeri see [25].

5 Modelling Quorum Sensing System in Vibrio fischeri

In this section we present a model of the Quorum Sensing System in Vibrio fischeri using a Multienvironment P system. We will study the behaviour of a population of N bacteria placed inside a multienvironment with 25 different environments. Note that we are using a parametric Multienvironment P system, $\mathbf{ME}(N)$, that depends on N, the number of bacteria in the population.

$$\mathbf{ME}(N) = (\{\text{OHHL}\}, \{e\}, G, \{\emptyset, e\}, \ldots, \{\emptyset, e\}, \mathcal{R}_e, \ldots, \mathcal{R}_e, \mathbf{\Pi}, N) \quad \text{where,}$$

(1) In the alphabet we only have OHHL representing the signal,
(2) e is the only label that will be used,
(3) G represents the structure of the environment which consists of 25 environments connected in the way depicted in figure 2,
(4) In the initial configuration every environment will be empty and labelled with e,
(5) The P system $\mathbf{\Pi}$ represents a bacterium. A population of N identical copies of $\mathbf{\Pi}$ will be studied.

$$\mathbf{\Pi} = (\Sigma, \{b\}, [\ ], (\text{LuxBox}, b), \mathcal{R}_b) \quad \text{where:}$$

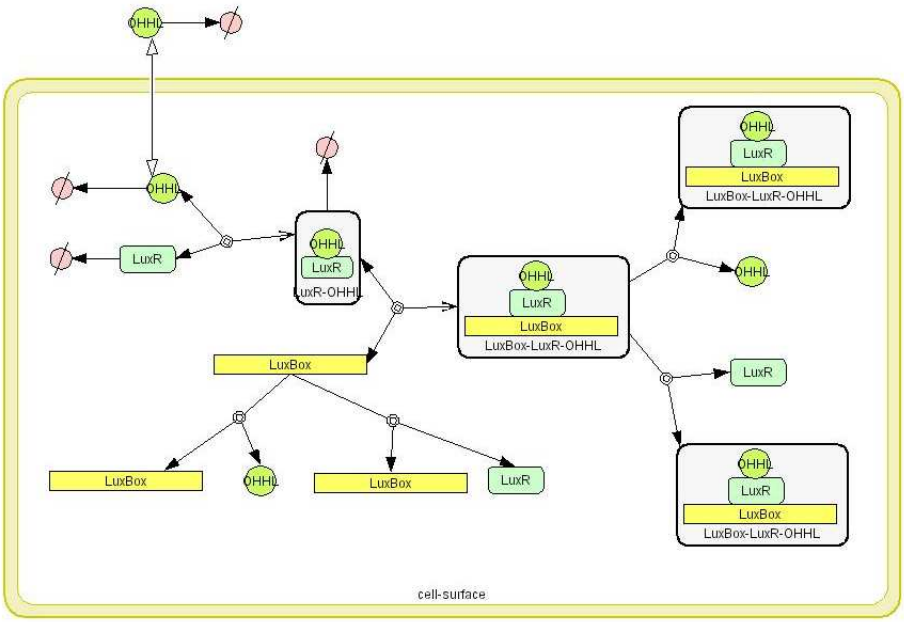

Fig. 3. Quorum Sensing System in Vibrio fischeri

- In the alphabet we represent the signal (3-oxo-C6-HSL), the protein LuxR, the complex protein-signal, the regulatory region LuxBox and the regulatory region occupied by the complex.

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

- The membrane structure consists only of one membrane that represents and separate a bacterium from the environment. This membrane will be labelled with b.
- We are interested in examinig how bacteria communicate to coordinate their behaviours and how the population moves from a downregulated state, where the protein and the signal are produced at basal rates, to an upregulated state. Therefore, in the initial multisets of the bacteria we will have only the genome (LuxBox) to start the production of the signal (OHHL) and protein (LuxR) at basal rates.
- In the rules we model the chemical reactions forming the Quorum Sensing System that take place inside the bacteria. Next we list the rules in $\mathcal{R}_b$ and we briefly describe the chemical reactions they represent:

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

$r_1 :$ [LuxBox]$_b \overset{c_1}{\rightarrow}$ [LuxBox , OHHL]$_b$
$r_2 :$ [LuxBox]$_b \overset{c_2}{\rightarrow}$ [LuxBox , LuxR]$_b$

$(*)$ 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.

$r_3 :$ [LuxR , OHHL]$_b \overset{c_3}{\rightarrow}$ [LuxR-OHHL]$_b$
$r_4 :$ [LuxR-OHHL]$_b \overset{c_4}{\rightarrow}$ [LuxR , OHHL]$_b$

$(*)$ 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.

$r_5 :$ [LuxBox , LuxR-OHHL]$_b \overset{c_5}{\rightarrow}$ [LuxBox-LuxR-OHHL]$_b$
$r_6 :$ [LuxBox-LuxR-OHHL]$_b \overset{c_6}{\rightarrow}$ [LuxBox , LuxR-OHHL]$_b$

$(*)$ 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.

$r_7 :$ [LuxBox-LuxR-OHHL]$_b \overset{c_7}{\rightarrow}$ [LuxBox-LuxR-OHHL , OHHL]$_b$
$r_8 :$ [LuxBox-LuxR-OHHL]$_b \overset{c_8}{\rightarrow}$ [LuxBox-LuxR-OHHL , LuxR]$_b$

$(*)$ OHHL can diffuse outside the bacterium and accumulate in the environment.

$r_9 :$ [OHHL]$_b \overset{c_9}{\rightarrow}$ OHHL [ ]$_b$

$(*)$ OHHL, LuxR and the complex LuxR-OHHL undergo a process of degradation in the bacterium

$r_{10} :$ [OHHL]$_b \overset{c_{10}}{\rightarrow}$ [ ]$_b$
$r_{11} :$ [LuxR]$_b \overset{c_{11}}{\rightarrow}$ [ ]$_b$
$r_{12} :$ [LuxR-OHHL]$_b \overset{c_{12}}{\rightarrow}$ [ ]$_b$

(6) All the environments will have the same set of rules, $\mathcal{R}_e$, described below:

$(*)$ Once the signal OHHL accumulates in the environment it can also diffuse inside the bacteria.

$r_{13} :$ OHHL [ ]$_b \overset{c_{13}}{\rightarrow}$ [OHHL]$_b$

$(*)$ OHHL can also be degraded in the environments.

$r_{14} :$ [OHHL]$_e \overset{c_{14}}{\rightarrow}$ [ ]$_e$

$(*)$ The signal can diffuse from one environment to another.

$r_{15} :$ [OHHL]$_e$ − [ ]$_e \overset{c_{15}}{\rightarrow}$ [ ]$_e$ − [OHHL]$_e$

$(*)$ Bacteria can also move freely from one environment to another.

$r_{16} :$ [[ ]$_b$]$_e$ − [ ]$_e \overset{c_{16}}{\rightarrow}$ [ ]$_e$ − [[ ]$_b$]$_e$

(7) Attributes: As mentioned before the attributes associated to each rule are stochastic constants representing the average number of applications of a rule per time unit. Where possible, we have used estimates derived from the literature. For parameter values not available from these sources we used a "trial-and-error" approach, making an initial "guess" at the values of the missing constants, then compare the resulting behaviour with known properties of the system. If they did not match unknown constants were then adjusted systematically, one parameter at a time.
After this process we have chosen the following set of parameters, $c_1 = 2, c_2 = 2, c_3 = 9, c_4 = 1, c_5 = 10, c_6 = 2, c_7 = 250, c_8 = 200, c_9 = 50, c_{10} = 30, c_{11} = 20, c_{12} = 20, c_{13} = 1, c_{14} = 5, c_{15} = 8, c_{16} = 2$. These values have been set such that the degradation rates compensate the basal production of the signal and the protein and such that the production rates when the regulatory region is occupied produce a massive increase in the transcription of the signal and the protein.
It is also interesting to study how different sets of parameters or attributes produce the same behaviour.

In the initial configuration of our system $\mathbf{ME}(N)$, N copies of $\mathbf{\Pi}$ will be distributed randomly over the 25 different environments. Then, the system will evolve applying the rules from $\mathcal{R}_e$ in the environments and $\mathcal{R}_b$ in the bacteria according to the algorithm presented in section 3. One of the possible initial configurations for a population of 44 bacteria can be seen in figure 4.

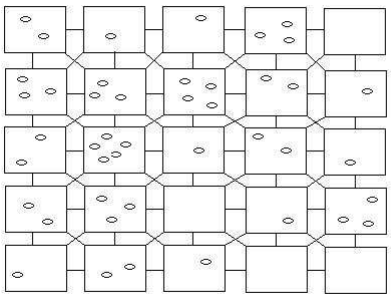

Fig. 4. A possible initial configuration of our Multienvironment P system.

6 Results and Discussions

The model presented in the previous section has been represented in SBML, Systems Biology Markup Language, a computer-readable format for representing models of biochemical reaction networks [29]. The SBML code was generated using CellDesigner, a structured diagram editor for drawing gene-regulatory and biochemical networks [31].

We have run our simulations using a program developed in Scilab, a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications [30]. The Scilab simulator has as input the SBML file specifying our model.

We have studied the emergent 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.

First we have considered a population of 100 bacteria. In figure 5 we show the evolution over time of the number of quorate bacteria [1] and the number of signals (OHHL) in the environment. Observe that the signal, OHHL, accumulates in the

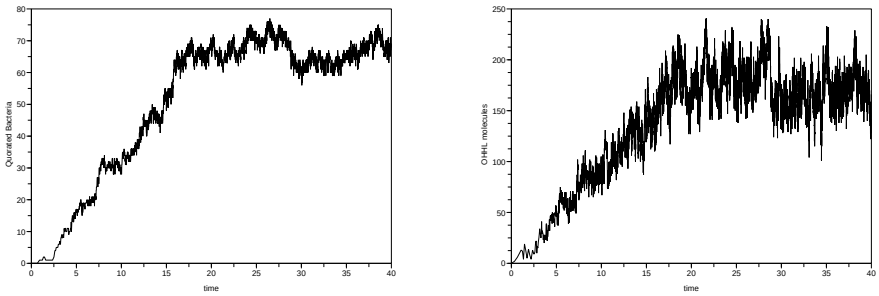

Fig. 5. Number of quorate bacteria (left) and signals in the environment (right)

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 quorate and then they accelerate a process of recruitment that makes the whole population of bacteria behave in a coordinated way.

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

In our approach the behaviour of each individual in the population can be tracked. In figure 6 we have taken a sample of three bacteria and have studied the correlation between the number of signal inside each bacterium (first row) and the occupation of the LuxBox by the complex (second row) which represents that the bacterium is quorate.

Observe in figure 6 that the number of signal molecules inside the bacterium has to exceed a threshold of approximately 7 molecules in order to recruit the

[1] We will say that a bacterium is quorate if the LuxBox in this bacterium is occupied by the complex producing the transcription of the enzymes involved in the production of light.

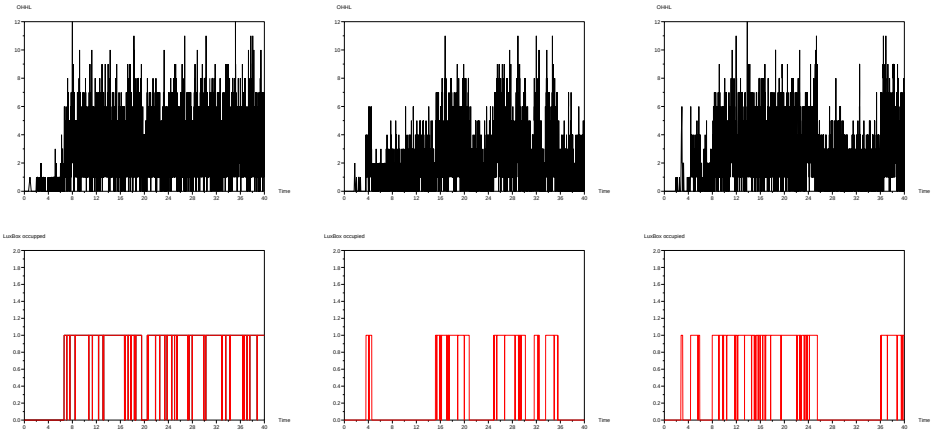

Fig. 6. A sample of three bacteria: Signal and Occupation of the LuxBox

bacterium. When the number of molecules is greater than 7 the LuxBox is occupied, that is, the bacterium is quorate or upregulated but when there is less than seven signals the bacterium switches off the system and it goes downregulated.

We can also study how rules are applied across the evolution of the system. In figure 7, it is shown the evolution of the number of applications of the rules representing the basal production (first two graphs) and the rules representing the production of the signal and protein induced by the binding of the complex to the LuxBox. It is depicted, in figure 7, how at the beginning the basal production rules are the most applied rules while the other two are seldomly applied. But then, as a result of the recruitment process the bacteria sense the size of the population and they behave in a coordinate way applying massively the third and fourth rules. So the system moves from a downregulated state to an upregulated state where the bacteria are luminescence.

Finally, in order to study how bacteria can sense the number of individuals in the colony and get quorate 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 no recruitment process takes place and the signal does not accumulate in the environemnt. Only one of the bacteria guessed wrong the size of the population and got upregulated, see figure 8. But then, after sensing that the signal does not accumulate in the environment, it switched off its system.

In figure 9 it is depicted the behaviour of the bacterium that got quorate. Observe that this bacterium got quorate because the number of signal inside it exceeded the threshold of 7 signals. Then it started to produce signals massively, these signals diffused to the environment where there were not enough bacteria to sense them and they were degraded. When the bacterium sensed that the signal did not accumulated in the environment it switched off its system.

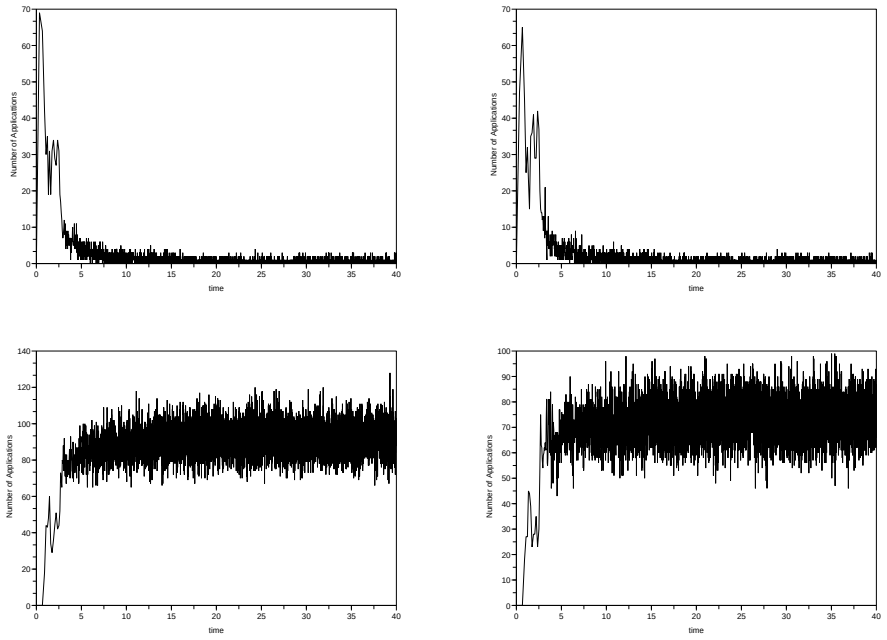

Fig. 7. Number of applications of rules r_1, r_2 (first row) and r_7,r_8 (second row)

Finally, observe in figure 10 that for only 10 bacteria the system remains in an downregulated state only applying the rules representing the basal productions while the rules associated with the production of light are seldomly applied.

Summing up, our simulations show that Vibrio fischeri has a Quorum Sensing System where a single bacterium can guess that the size of the population is big enough and start to produce light. Then this bacterium starts to massively produce signals, if the signal does not accumulate in the environment it means that the guess was wrong then it switches off the system. On the other hand if the signal does accumulate in the environment meaning that the number of bacteria in the colony is big a recruitment process takes place that makes the population of bacteria to luminescence. Observe that this emergent behaviour is a result of local interactions in the environments between different simple agents, bacteria, only able to produce and receive molecular signals. In this respect, our approach and the one in [6] dealing with self-assembly, are very promising initial attempts revealing emergent behaviour, based only in local interactions, in nature and artificial life by using P system models.

7 Conclusions

In this paper we have presented a model of the Quorum Sensing System in Vibrio fischeri within the framework of Multienvironment P systems. Our approach

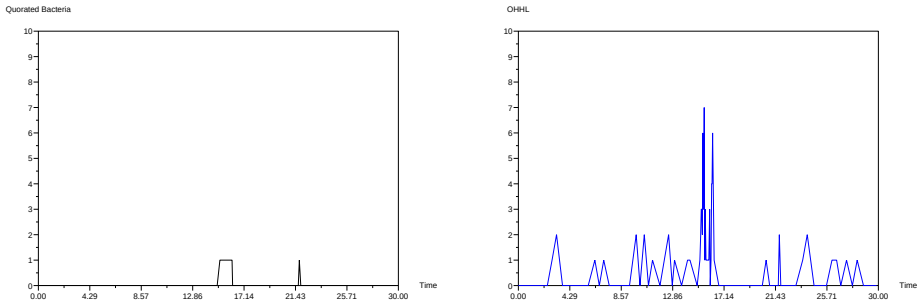

Fig. 8. Quorate bacteria and signals in the environment

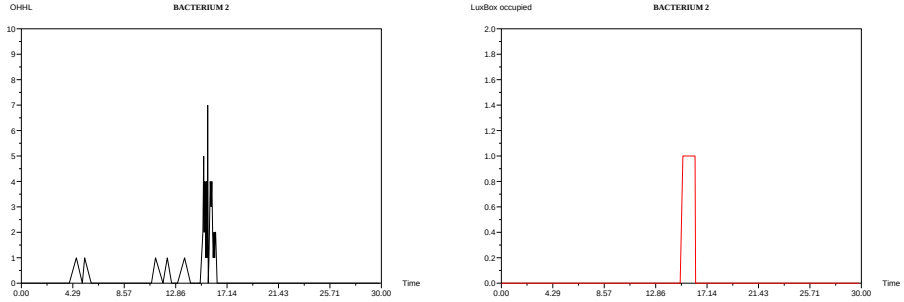

Fig. 9. Behaviour of a bacterium in a population of 10 bacteria

takes into account the discrete character of the components of the system, the level of noise and the role played by membranes. We have been able to study the emergent behaviour of different populations of bacteria in an heterogeneous environment.

The results of our model show that on the one hand bacteria remain dark at low cell densities and on the other hand in big size populations bacteria are able to sense the number of individuals and the population starts to emit light in a coordinated way. This results agree well with in vitro observations.

Quorum sensing is a complex mechanism in its biological details but simple in its fundamental principles. Computational modelling could provide a deeper understanding of its complexity and functioning. These insights could enable researchers to develop new applications where multiple agents need to robustly and efficiently coordinate their collective behaviour based only on a very limited information of the local environment.

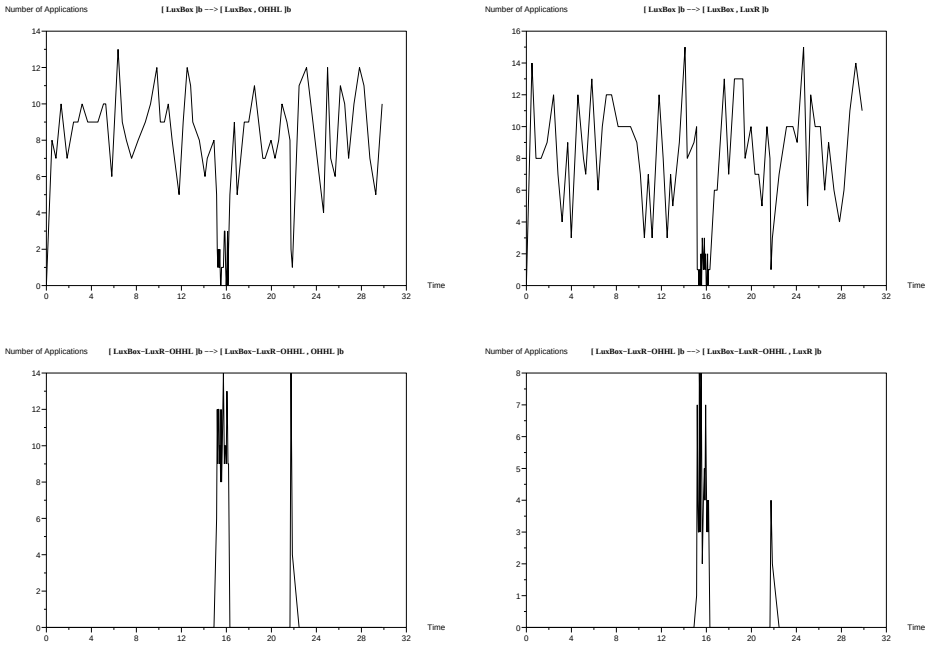

Fig. 10. Number of applications of rules r_1, r_2, r_7 and r_8 in 10 bacteria

References

1. Bernardini, F., Gheorghe, M. (2004) Population P systems. *J. Universal Computer Sci.*, **10**, 509–539.
2. Bernardini, F., Gheorghe, M., Muniyandi, R.C., Krasnogor, N., Pérez-Jiménez, M.J., Romero-Campero, F.J. (2006). On P Systems as a Modelling Tool for Biological Systems. *Lecture Notes in Computer Science*, **3850**, 114–133.
3. Bianco, L., Fontana, F., Manca, V. (2005). P Systems and the Modelling of Biochemical Oscillation. *Lecture Notes in Computer Science*, **3850**, 199–208.
4. Ciobanu, G., Păun, Gh., Pérez-Jiménez, M.J., eds (2005).*Applications of Membrane Computing*, Springer-Verlag, in press.
5. Fargerström, T., James, G., James, S., Kjelleberg, S., Nilsson, P. (2000) Luminescence Control in the Marine Bacterium Vibrio fischeri: An Analysis of the Dynamics of lux Regulation. *J. Mol. Biol.* **296**, 1127–1137
6. Gheorghe, M., Păun, Gh. (2006). Computing by self-Assembly: DNA Molecules, Polyminoes, Cells, submitted
7. Gillespie, D.T. (1977). Exact Stochastic Simulation of Coupled Chemical Reactions. *The Journal of Physical Chemistry*, **81**, 25, 2340–2361.
8. Gillespie, D.T. (2001). Approximate Accelerated Stochastic Simulation of Chemically Reacting Systems. *Journal of Chemical Physics*, **115**, 4, 1716–1733.
9. Gillespie, D.T. (2003). Improved Leap-size Selection for Accelerated Stochastic Simulation. *Journal of Chemical Physics*, **119**, 16, 8229–8234.

10. Greenberg, E.P., Kaplan, H.B. (1985). Diffusion of Autoinducer Is Involved in Regulation of the Vibrio fischeri Luminescence System. *Journal of Bacteriology* **163**(3), 1210–1214
11. Greenberg, E.P., Kaplan, H.B. (1997). Overproduction and Purification of the luxR Gene Product: Transcriptional Activator of the Vibrio fischeri Luminescence System. *Proc. Natl. Acad. Sci.* **84**, 6639–6643
12. Greenberg, E.P., Lupp, C.; Ruby, E.G., Urbanowski, M. (2003). The Vibrio fischeri Quorum-sensing System ain and lux Sequentially Induce Luminescence Gene Expression and Are Important for Persistence in the Squid Host. *Molecular Microbiology*, **50**(1) 319–331
13. Hardie, K.R., Williams, P., Winzer, K. (2002). Bacterial cell-to-cell communication: sorry, can't talk now, gone to lunch. *Current Opinion in Microbiology*, **5**, 216–222
14. Lazdunski, A.M., Sturgis, J.N., Ventre, I. (2004). Regulatory Circuits and Communication in Gram-Negative Bacteria. *Nature Reviews, Microbiology*, **2**, 581–591
15. McCann, J., Millikan, D.S., Ruby, E.G., Stabb, E.V. (2003). Population Dynamics of Vibrio fischeri during Infection of Euprymna scolopes. *Applied and Environmental Microbiology* **69**(10), 5928–5934
16. Meng, T.C., Somani S., Dhar, P. (2004). Modelling and Simulation of Biological Systems with Stochasticity. *In Silico Biology*, **4**, 0024.
17. Păun, A.; Păun, Gh. (2002). The Power of Communication: P Systems with Symport/Antiport, *New Generation Computing*, **20**, 3, 295–305.
18. Păun, Gh. (2000). Computing with Membranes, *Journal of Computer and System Sciences*, **61**(1) 108 – 143.
19. Păun, Gh. (2002). *Membrane Computing. An Introduction*, Springer-Verlag Berlin.
20. Pérez-Jiménez, M.J., Romero-Campero, F.J. (2006) P Systems, a New Computationl Modelling Tool for Systems Biology, *Transactions on Computational Systems Biology*, to appear
21. Stevens, A.M., Greenberg, E.P. (1997). Quorum Sensing in Vibrio fischeri: Essential Elements for Activation of the Luminescence Genes. *Journal of Bacteriology* **179**(2) 557–562
22. Stundzia, A.B., Lumsden, C.J. (1996). Stochastic Simulation of Coupled Reaction-Diffusion Processes. *Journal of Computational Physics*, **127**, 196–207.
23. Suzuki,Y., Tanaka, H. (2000). A New Molecular Computing Model, Artificial Cell Ssytem. *GECO*, 833-840
24. Terrazas, G., Krasnogor, N., Gheorghe, M., Bernardini, F., Diggle, S., Camara, M. (2005) An Environment Aware P-System Model of Quorum Sensing. *Lecture Notes in Computer Science*, to appear.
25. Waters, C.M., Bassler, B.L. (2005) Quorum Sensing: Cell-to-Cell Communication in Bacteria. *Annu. Rev. Cell. Dev. Biol.* **21** 319–346
26. Nottingham Quorum Sensing Web Site `http://www.nottingham.ac.uk/quorum/`
27. ISI Web Site `http://esi-topics.com/erf/october2003.html`
28. P Systems Web Site `http://psystems.disco.unimib.it/`
29. SBML Web Site `http://sbml.org/index.psp`
30. SciLab Web Site `http://scilabsoft.inria.fr/`
31. Cell Designer Web Site `http://www.celldesigner.org/`