

# Towards a P Systems Pseudomonas Quorum Sensing Model

L. Bianco<sup>a</sup>, D. Pescini<sup>b</sup>, P. Siepmann<sup>c</sup>, N. Krasnogor<sup>c</sup>, F. J. Romero Campero<sup>d</sup>  
and M. Gheorghe<sup>e</sup>

<sup>a</sup>Department of Computer Science, University of Verona  
Strada Le Grazie 15, 37134 Verona, Italy  
Email: [bianco@sci.univr.it](mailto:bianco@sci.univr.it)

<sup>b</sup>Dipartimento di Informatica, Sistemistica e Comunicazione  
Università degli Studi di Milano-Bicocca  
Via Bicocca degli Arcimboldi 8, 20126 Milano, Italy  
Email: [pescini@disco.unimib.it](mailto:pescini@disco.unimib.it)

<sup>c</sup>School of Computer Science and Information Technology  
University of Nottingham  
Jubilee Campus, Nottingham, NG81BB, UK  
Email: {[Peter.Siepmann](mailto:Peter.Siepmann@nottingham.ac.uk), [Natalio.Krasnogor](mailto:Natalio.Krasnogor@nottingham.ac.uk)}@nottingham.ac.uk

<sup>d</sup>Research Group on Natural Computing  
Department of Computer Science and Artificial Intelligence  
University of Seville, Avda. Reina Mercedes, 41012 Sevilla, Spain  
Email: [fran@cs.us.es](mailto:fran@cs.us.es)

<sup>e</sup>Department of Computer Science, The University of Sheffield  
Regent Court, Portobello Street, Sheffield S1 4DP, UK  
Email: [M.Gheorghe@dcs.shef.ac.uk](mailto:M.Gheorghe@dcs.shef.ac.uk)

**Abstract.** *Pseudomonas aeruginosa* is an opportunistic bacterium that exploits quorum sensing communication to synchronise individuals in a colony and this leads to an increase in the effectiveness of its virulence. In this paper we derived a mechanistic P systems model to describe the behaviour of a single bacterium and we discuss a possible approach, based on an evolutionary algorithm, to tune its parameters that will allow quantitative simulations of the system.

## 1 Introduction

Quorum sensing is a particular form of cell-to-cell communication in bacteria which exploits the concentration of a particular molecule, called signal, to “sense” the population density of the colony. The quorum sensing regulatory network is used by the individuals of the colony for collective synchronisation and therefore for a coherent control over the gene expression. In *Pseudomonas aeruginosa* this mechanism is responsible for the effectiveness of the virulence of this bacterium [14, 23, 10, 9]. In fact, a single bacterium starts to express its virulence factors only when it senses that the bacteria population has reached a certain threshold level such that the host response will be inadequate.

The activation of a complex cellular response is what distinguishes the quorum sensing as a communication regulatory circuit from other density dependent responses such as the metabolism or detoxification of small molecules.

The simplest quorum sensing network known in Gram-Negative bacteria is also the first one ever discovered [25, 17]. It has been found in the *Vibrio fischeri* bacterium, also known as *Photobacterium fischeri* and is nowadays considered as the paradigm of this cell communication process. In this network two proteins and one signalling molecule are involved. The *R protein* is a transcriptional regulator, while the *I protein* is the synthase for the signalling molecule, also referred to as the *autoinducer*. An important role is also played by the confinement of the bacterial colony. The fact that the *autoinducer* molecule is not dispersed in the environment allows its diffusion inside the individuals and therefore its concentration sensing.

At low cell densities the *I protein* synthesises the *autoinducer* at a basal rate and the signal freely diffuses outside the bacterium. The concentration of the signal inside each bacterium is increased by the combined effect of the confinement and the increase of the population. At this point, the binding of the *R protein* with the *autoinducer* becomes more likely. The binding of the signal molecules activates the *R protein* transcriptional regulator. Since the *I gene* is the target of the *R protein*, the bacterium starts to produce more and more signals. The regulation network autoinduces its transcription. In this way the high concentration of the *autoinducer* coordinates the transcription of all the genes that are target of the *R protein*.

The quorum sensing in *Pseudomonas aeruginosa* is more complex, nevertheless intriguing, since this bacterium uses two different quorum sensing systems which interact each other.

The aim of this work is to provide a P system model [15, 16] of the bacterium *Pseudomonas aeruginosa* quorum sensing, focusing on the communication mechanisms. The parameters of the model will be tuned using an evolutionary algorithm. Our long term aim is to reproduce the characteristic behaviour of the quorum sensing in *Pseudomonas aeruginosa*, namely, the switch between two distinct stable steady solutions: the first describing the behaviour of the non-quorated bacterium (i.e., with low levels of autoinducer), the second modelling its quorated behaviour (i.e., the behaviour obtained with high concentration of the autoinducer molecule). Once the model will be entirely defined several simulations with different strategies [6, 20, 18] will be run.

First of all, we address the modelling of the internal dynamics of one single bacterium, tuning its kinetic constants in a way ensuring its non-quorated behaviour. At a later stage, we intend to exploit compartmentalisation of P systems to model a colony of bacteria each of them internally specified according to the same set of kinetic constants. In this respect we will extend the current model to a Population P systems approach [3] that has been already used to express some aspects of quorum sensing in bacterium *Pseudomonas aeruginosa* [22] and for self-assembly problems [4].



The second signalling system involved in the model is named *rhl* system because it controls the expression of rhamnolipid via the production of *rhlAB* operon. The autoinducer molecule in this case is *C4-HSL* and the active complex is *RhlR-C4-HSL*. It has been shown that this cascade is necessary for the production of some virulence factors like *LasB* elastase and *LasA* protease, as well as pyocyanin, cyanide and alkaline protease. For this reason this signalling system is also known as *vsm* (virulence secondary metabolites).

Although the corresponding autoinducing molecules are highly selective (and thus not interchangeable at all), several interconnections between the *las* and the *rhl* pathways of the quorum sensing in *Pseudomonas aeruginosa* are known. One link between them has been already mentioned and it is constituted by the *LasB* elastase, that needs both *LasR-3-oxo-C12-HSL* and *RhlR-C4-HSL* for its production. More interestingly, the *las* system is at a higher level in the hierarchical regulatory cascade, in fact *LasR-3-oxo-C12-HSL* can activate the expression of the *rhlR* gene. In addition, the active complex *LasR-3-oxo-C12-HSL* can bind to *RhlR* preventing it to form the complex *RhlR-C4-HSL*.

## 2.1 The differential equation model

Many models of the quorum sensing in the *Pseudomonas aeruginosa* are presented in the literature and usually they approach the phenomenon from two different angles. The first one describes the colony behaviour by summarizing individual dynamics as a state change avoiding a precisely detailed representation of each of the bacterium quorum sensing networks [24, 1]. The second one describes in a more detailed fashion the quorum sensing pathway for each bacterium with the purpose to model the emergent behaviour of the whole colony [11].

We think that the P system framework is particularly suitable for this second approach. In fact, the modularity, the compartmentalisation, the hierarchical structure and the rewriting rules (all features of P systems [16]) allow a convenient description of this reality.

In [11] a model of the *las* signalling system has been devised, but no description is given of the *rhl* system. The graphical description of the quorum sensing pathway depicted in Figure 1 has been translated into the following set of eight differential equations:

$$\begin{aligned}
\frac{dP}{dt} &= k_{RA}RA - k_P P \\
\frac{dR}{dt} &= -k_{RA}RA + k_P P - k_R R + k_1 r \\
\frac{dA}{dt} &= -k_{RA}RA + k_P P + k_2 L - k_A A \\
\frac{dL}{dt} &= k_3 l - k_L L \\
\frac{dS}{dt} &= k_4 s - k_S S \\
\frac{ds}{dt} &= V_s \frac{P}{K_s + P} - k_s s \\
\frac{dr}{dt} &= V_r \frac{P}{K_r + P} - k_r r + r_0 \\
\frac{dl}{dt} &= V_l \frac{P}{K_l + P} \frac{1}{K_s + S} - k_l l + l_0
\end{aligned} \tag{2.1}$$

where the correspondence between differential equation symbols and elements in the pathway are summarized in Table 1.

The production of the activated complex  $P$  by means of the autoinducer and the  $LasR$  protein (whose expression is given by the product of the constitutive elements concentrations with a rate  $k_{RA}$ :  $k_{RA}RA$ ) is an example of how cooperative contributions are obtained in the differential equations approach by means of the mass action law.

Basal rates productions and degradations are also taken into account, an example of the former is the  $k_1r$  element giving the basal production of  $LasR$  protein ( $R$ ), while an example of the latter is the degradation of the active complex ( $P$ ) represented by the element  $k_P P$ . The production of messenger RNAs from the corresponding genes is modelled with a Michaelis-Menten-like dynamics depending on the concentration of the transcription factor, as it happens in the case of the production of  $lasR$  and  $rsaL$  mRNAs (respectively  $r$  and  $s$ ), the first modelled by  $V_r \frac{P}{K_r + P}$  and the second by  $V_s \frac{P}{K_s + P}$ . The production of  $lasI$  mRNA ( $l$ ) is also down-regulated by the presence of  $RsaL$  protein ( $S$ ) and this is modelled by  $V_l \frac{P}{K_l + P} \frac{1}{K_S + S}$ , in which the Michaelis-Menten-like dynamics is attenuated by an inversely proportional function of the  $RsaL$  concentration.

Unfortunately, no value is known for the 21 kinetic constants present in the set of differential equations (2.1). To overcome this problem, in [11] several simplifying assumptions are considered, that lead to fewer equations and fewer parameters as well.

Variable	Biological entity
R	<i>LasR</i>
A	<i>3-oxo-C12-HSL</i>
P	<i>LasR-3-oxo-C12-HSL</i>
L	<i>LasI</i>
S	<i>RsaL</i>
r	<i>lasR mRNA</i>
l	<i>lasI mRNA</i>
s	<i>rsaL mRNA</i>

**Table 1.** Variable-concentration correspondence between the differential formulation and the graphical description of the quorum sensing model of *Pseudomonas aeruginosa* (from [11]).

In the following we will describe a possible parameter estimation strategy to tackle this problem (see Section 4). The idea is to rely on this differential equations system as a “synthetic bio-experiment” used to confront our model to.

## 2.2 A first P systems model

Several attempts to simulate quorum sensing in bacteria are present in P systems literature [5, 19], but, as far as we know, none of them deals with the *Pseudomonas aeruginosa* bacterium.

Here we describe a direct P systems translation of the differential equation model previously discussed [11]. Formally, the *Pseudomonas* P System is:

$$\Pi = (V, \mu, w, \mathcal{R})$$

where:

- $V = \{geneR, geneL, R, A, P, L, S, r, l, s\}$  is the alphabet;
- $\mu = [ ]_0$  is the membrane structure: since we address the single bacterium case, it consists of the cellular membrane only;
- $w = geneR geneL$  is the initial configuration that comprises only *LasR* and *LasI* genes, thus is represented as the string;
- $\mathcal{R} = \{r_1, \dots, r_{18}\}$  is the set of the rules:

$$\begin{aligned} r_1 : geneR &\longrightarrow geneR + r \\ r_2 : r &\longrightarrow \lambda \\ r_3 : r &\longrightarrow r + R \\ r_4 : P &\longrightarrow P + r \\ r_5 : R + A &\longrightarrow P \\ r_6 : P &\longrightarrow R + A \\ r_7 : P &\longrightarrow P + s \\ r_8 : s &\longrightarrow \lambda \\ r_9 : S &\longrightarrow \lambda \\ r_{10} : s &\longrightarrow s + S \\ r_{11} : P &\longrightarrow P + l \\ r_{12} : l &\longrightarrow l + L \\ r_{13} : l &\longrightarrow \lambda \\ r_{14} : geneL &\longrightarrow geneL + l \\ r_{15} : L &\longrightarrow \lambda \\ r_{16} : L &\longrightarrow L + A \\ r_{17} : A &\longrightarrow \lambda \\ r_{18} : R &\longrightarrow \lambda \end{aligned}$$

Note that, symbols in  $V$  correspond to the variables of the differential equation and their correspondence to the biological reality is given in Table 1. Two new elements (i.e., *geneR* and *geneL*) are introduced, which account for the genes involved in the basal production of the *LasR* and *LasI* mRNAs.

Each one of the rules in  $\mathcal{R}$  is directly obtained from the differential description of the considered quorum sensing model. For example, we can see that rule  $r_1$  models the basal production of the *LasR* mRNA, while rule  $r_2$  expresses its degradation, moreover rules  $r_5$  and  $r_6$  describe the reversible reaction of the complex  $P$  formation by starting from its fundamental constituents  $R$  and  $A$ .

Due to the different level of abstraction in the representation of different parts of the model (as in the case of the Michaelis-Menten-like kinetics that are modelled with a higher level of abstraction than other components of the system), we can not directly apply mechanistic algorithms to this model. For this reason, we will apply to this set of rules only the strategy known as *Metabolic Algorithm* (for details refer to [6]), whose simulation results, together with some numerical solutions of the set of differential equations (2.1), are shown in Section 2.3 for different choices of parameters.

The metabolic algorithm simulation needs to specify a set of reaction maps, each one associated in a one-to-one manner to the rules of  $\mathcal{R}$ . Reaction maps [6] are functions defined over the state of the system (i.e., multiplicity or concentration of all elements of the system depending on the case), that are used by the Metabolic algorithm to allocate objects to rules. For example, as we will see shortly,  $F_{r_1}$ , that is the reaction map of rule  $r_1$ , is simply the constant rate of production of LasR mRNA. We can have more complicated reaction maps, as in the case of rule  $r_4$  that takes into account the Michaelis-Menten-like production of the LasR mRNA elicited by the *LasR-3oxo-C12-HSL* complex. As in the case of the rules, that specify the physical interactions and connections between the elements of the modelled reality, we can obtain this information from the differential equation formulation. The set of reaction maps employed in our simulations are the following:

$$\begin{array}{ll}
 F_{r_1} = r_0 & F_{r_2} = k_r \\
 F_{r_3} = k_1 & F_{r_4} = \frac{V_r}{K_r + P} \\
 F_{r_5} = k_{RA} & F_{r_6} = k_P \\
 F_{r_7} = \frac{V_s}{K_s + P} & F_{r_8} = k_s \\
 F_{r_9} = k_4 & F_{r_{10}} = k_S \\
 F_{r_{11}} = \frac{V_l}{(K_l + P) \cdot (K_S + S)} & F_{r_{12}} = k_3 \\
 F_{r_{13}} = k_l & F_{r_{14}} = l_0 \\
 F_{r_{15}} = k_L & F_{r_{16}} = k_2 \\
 F_{r_{17}} = k_A & F_{r_{18}} = k_R
 \end{array} \tag{2.2}$$

Note that all reaction maps are constant apart from three of them. We have already discussed the meaning of the reaction map associated to rule  $r_4$ , analogous considerations hold for  $F_{r_7}$  as well. More interesting is the reaction map associated to rule  $r_{11}$  that takes into account the inhibitory effect of *RsaL* protein on the production of the *lasI* mRNA.

Remarkably, the method allows the current description of different parts of the system at different abstraction levels; moreover it is still applicable if all reaction maps are constant, a condition required by mechanistic algorithms.

In the following some simulation results are shown, as well as the numerical solution of the differential equation system, for some chosen parameters.

### 2.3 Simulation results

Here we show how the same reality can be described with two different approaches. As mentioned before, we do not have precise values for the model pa-

rameters, for this reason, as a first comparison attempt, we make a completely fictitious choice for them. As a further work, we plan to adopt some automatic way for the parameter estimation (see Section 4 for more details). The initial choice of parameters is here shown, and all the subsequent changes to this initial parameter set will be explicitly mentioned:

$$\begin{aligned}
 k_{RA} &= 10 & k_P &= 2 \\
 k_R &= 5 & k_1 &= 1 \\
 k_2 &= 1 & k_A &= 1 \\
 k_3 &= 1 & k_L &= 1 \\
 k_4 &= 1 & k_S &= 1 \\
 V_s &= 1 & K_s &= 1 \\
 k_s &= 0.5 & V_r &= 1 \\
 K_r &= 1 & k_r &= 1 \\
 r_0 &= 1 & V_l &= 1 \\
 K_l &= 1 & k_l &= 1 \\
 l_0 &= 1 & K_S &= 1
 \end{aligned} \tag{2.3}$$

The lack of biological information makes this choice completely arbitrary. Moreover, the high level description of some parts of the system prevents us to compute the dynamics of the system by means of stochastic algorithms such as the Gillespie one [12, 13], Dynamical Probabilistic P Systems [20] or the Multi-compartmental Gillespie [18].

In this section we compare the dynamics generated by the metabolic algorithm with the solutions obtained for the corresponding differential equation system. Figure 2 depicts the case in which parameters are chosen according to (2.3), the dynamics of each species reaches a steady state in both approaches, but the relative position of the species is different and this leads to two distinct system dynamics. Moreover, the time of the two systems differs; in the solution of the differential equation system this is measured in arbitrary units (due to the arbitrary choice of parameters), while in the model based on P systems the time is measured in steps of system evolution. In Figure 3 the choice of  $V_l = 0$  switches off rule  $r_{11}$  of the P system model and in this case the results of the two different approaches qualitatively match each other. Finally, the last choice of parameters is aimed at obtaining a quorum sensing consistent behaviour, that is, in the case of a single bacterium in the environment it should not quorate and thus the concentration of the complex  $P$  should reach the basal rate. Accordingly, we set  $K_{RA}$  to the value 0.1. In this case, depicted in Figure 4, the dynamics produced by the two approaches is qualitatively similar again.

### 3 Towards a detailed P systems model

Although the preliminary P system model described in Subsection 2.2 showed that we can obtain comparable results with the current models presented so far, our intention is to refine the model defined above in order to allow the simulation of its dynamics by means of mechanistic methods like Gillespie's



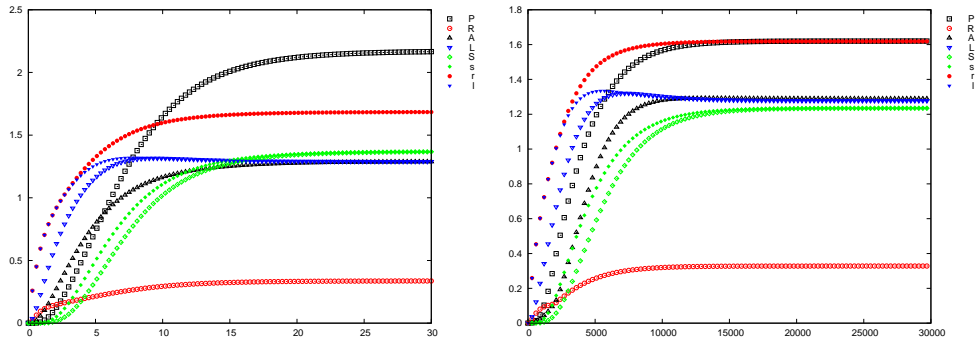


Fig. 2. Results for the Q. S. model with parameters showed in (2.3) using ODE approach (left) and metabolic algorithm (right).

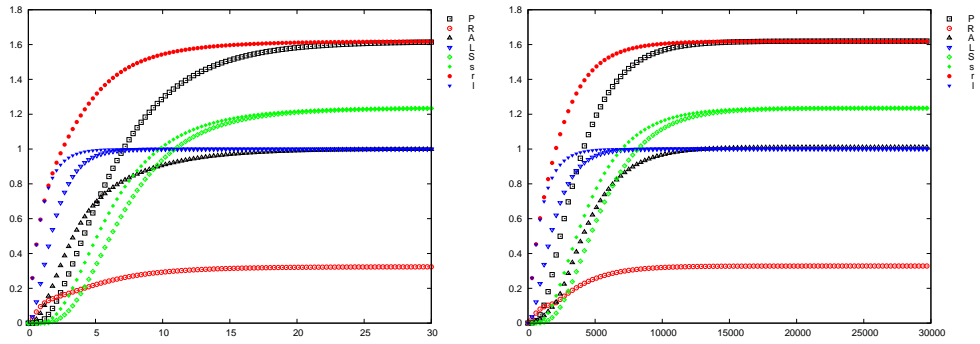


Fig. 3. Results for the Q. S. model with  $V_l = 0$  using ODE approach (left) and metabolic algorithm (right).

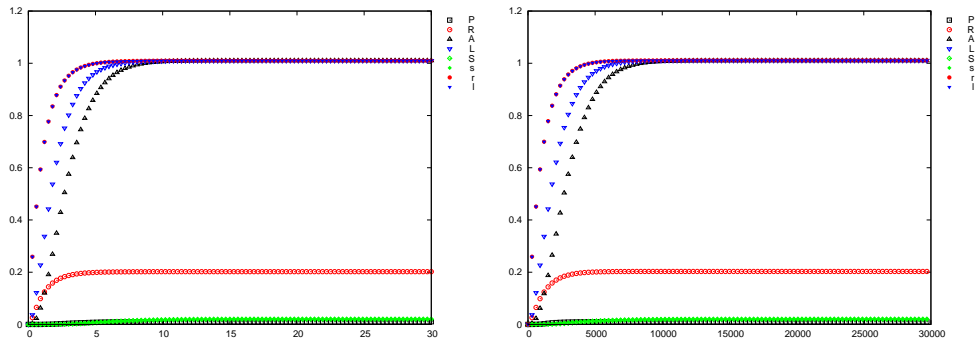


Fig. 4. Results for the Q. S. model with parameters  $k_{RA} = 0.1$  using ODE approach (left) and metabolic algorithm (right).

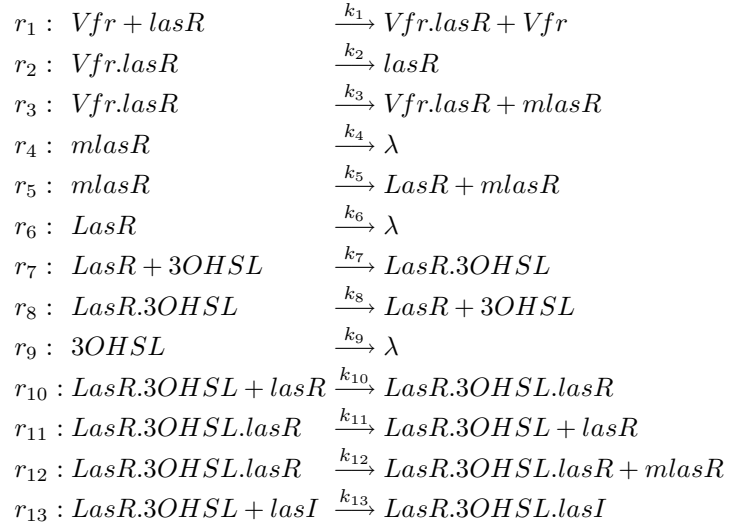
approach [12, 13], Dynamical Probabilistic P Systems algorithm [20] or the Multi-compartmental Gillespie [18]. In addition, this model is completely driven by the set of differential equations and in some cases it is not very biologically accurate. For example, in the case of *rsaL* mRNA production, when different from other mRNAs productions, it does not show any basal rate production. Moreover, it does not consider the binding of the transcription factor to the appropriate gene site necessary to start the transcription process of the DNA into the mRNA.

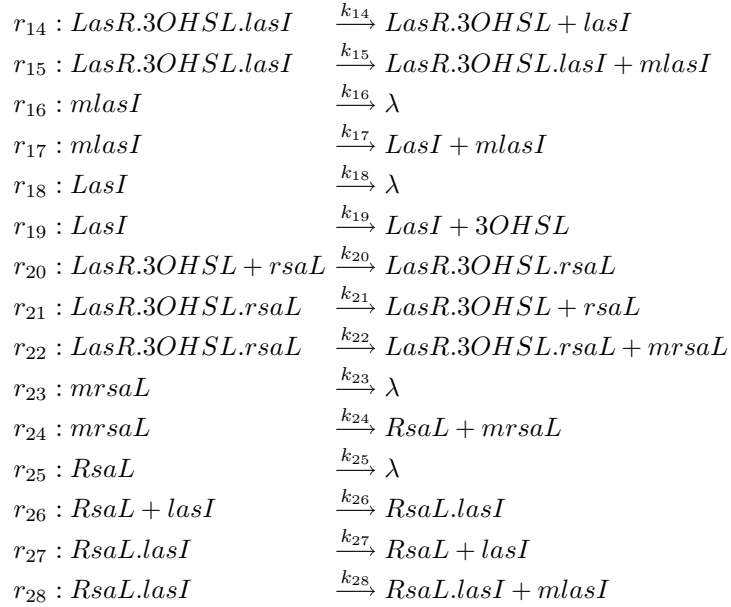
The formal description of the detailed P system model of *Pseudomonas* quorum sensing is the following:

$$\Pi = (V, \mu, w, \mathcal{R})$$

where:

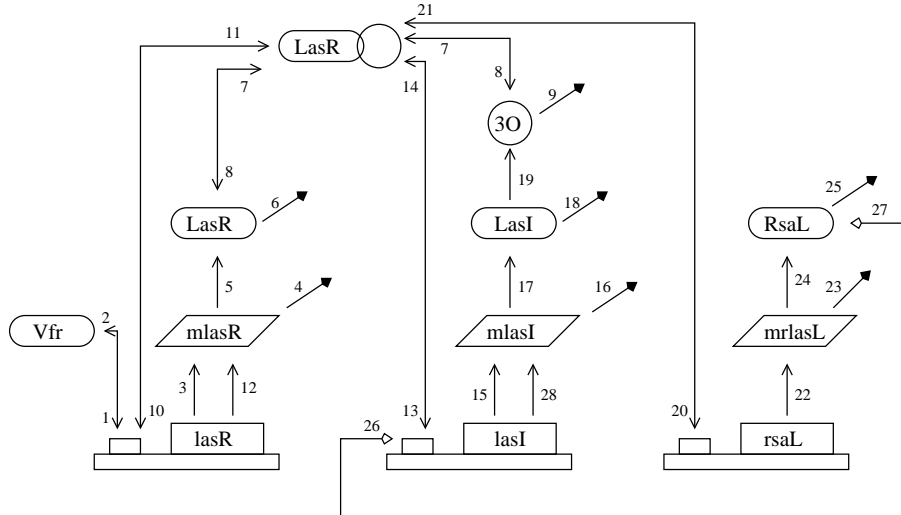
- $V = \{Vfr, lasR, Vfr.lasR, mlasR, LasR, 3OHSL, LasR.3OHSL, LasR.3OHSL.lasR, lasI, LasR.3OHSL.lasI, mlasI, LasI, rsaL, LasR.3OHSL.rsaL, mrsaL, RsaL, RsaL.lasI\}$  is the alphabet;
- $\mu = [ ]_0$  is the membrane structure: since we address the single bacterium case, it contains the cellular membrane only;
- $w = Vfr^n \quad lasR \quad lasI \quad rsaL$  is the initial configuration that comprises only the three genes and the protein *Vfr* that is needed to initiate the transcription and should be initialised with a high amount  $n \in \mathbf{N}$ ;
- $\mathcal{R} = \{r_1, \dots, r_{28}\}$  is the set of the rules:





where  $k_i$ , for  $i = 1, \dots, 28$ , is the rate constant associated with the  $i$ th rule.

This system is depicted in Figure 5 where numbers next to arrows refer to the corresponding rules. Note that arrows with two numbers denote reversible reactions modelled in the P system description with two distinct rules.



**Fig. 5.** The Pseudomonas quorum sensing detailed model analysed here. The number next to each arrow refers to the corresponding P system rule.

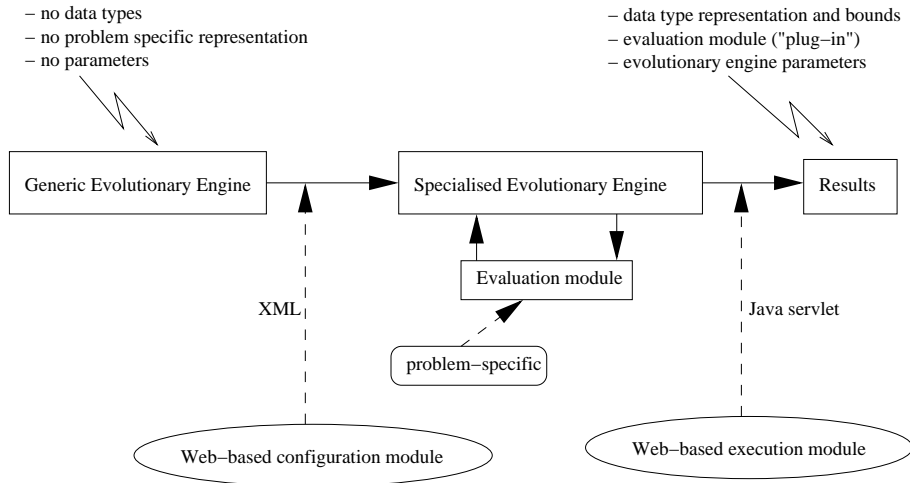
To give some ideas on how the model has been built we explain in detail the process that, starting from the *lasI* gene, leads to the formation of the complex *LasR.3OHSL*, the remaining part of the model follows a similar derivation. The production of *LasI* mRNA (*mlasI*) can be done in two ways depending on the transcription factor bound to the *lasI* gene. In fact, when *LasR.3OHSL* binds to the gene *lasI* (rule  $r_{13}$ ), it activates the transcription of *lasI* gene into *mlasI* mRNA (rule  $r_{15}$ ) with a rate  $k_{15}$ . The *RsaL* protein can bind to the *lasI* gene as well (rule  $r_{26}$ ), but in this case, the same transcription (modelled by rule  $r_{28}$ ) has different rate  $k_{28}$ . Since the biology of the process tells us that *RsaL* protein inhibits the mRNA production, we add the constraint that  $k_{28} \ll k_{15}$ . The *mlasI* mRNA can either be degraded (rule  $r_{16}$ ) or be translated into the *LasI* protein (rule  $r_{17}$ ). The latter can in turn be degraded (rule  $r_{18}$ ) or it can produce the autoinducer molecule *3OHSL* (rule  $r_{19}$ ), that can bind to the *LasR* protein and form the complex *LasR.3OHSL* (rule  $r_7$ ) or be degraded (rule  $r_9$ ).

As far as we know, no value for the kinetic constant necessary for the simulation of this dynamics is known in literature. For this reason we plan to adopt some automatic tools for exploring the huge parameter space. In the following section we describe a Genetic Algorithm fitting approach.

## 4 Parameter Estimation

In previous sections we showed two alternative models of Quorum Sensing and *qualitatively* compared them against a differential equations based model. In this Section we show, as a proof of concept, how P system models can be *quantitatively* fitted to observed data. In this proof of concept section we consider the ODE model as the golden standard against which the P system model must be fitted. That is, the ODE is a proxy for a biological experiment from which we could measure a variety of molecular concentrations. In order to fit the P system models to the ODE's observed data we perform parameter optimisation using an evolutionary algorithm (EA). Our EA has been specially developed for optimising a range of design and manufacturing processes. It has been successfully tested on a variety of complex systems and nano-particles self-organisation system [21]. Our evolutionary system is web-server based and can be tailored to solve a broad range of problems. The number and data types of genes in the chromosome, along with the parameters for the GA, including the users choice of selection, replacement, mutation and crossover mechanisms can be specified in the web-based configuration module. The later builds an XML script as output. This script, along with a plug-in style problem specification class, which most importantly includes the fitness function, configures the evolutionary algorithm to the specific problem at hand. The execution of the evolutionary algorithm can then be started and observed over the internet through a Java servlet. This evolutionary engine also caters for CPU-intensive optimisation problems, like the one we investigate here, by distributing the execution of the algorithm on a large computer cluster. Moreover, the web-server also allows simultaneous executions of the evolutionary engines on different problems. The web-server can be

accessed (under request) from [www.chellnet.org](http://www.chellnet.org). For a schematic representation of the evolutionary engine please see Figure 6.



**Fig. 6.** The ChellNet Evolvable Chellware Engine

In what follows we describe the fitness function used to fit our P systems to the observed time series.

#### 4.1 The Fitness Function

The evolutionary engine is used to adjust the parameters of the P system as to fit the  $w, w \in \mathbf{N}$  observed target time series  $\mathcal{S}_{tgt} = \{s_{tgt}^i\}_{i=1, \dots, w}$  simultaneously, where each of the  $w$  time series corresponds to one of the species concentrations. In turn, the P system model generates  $w$  time series  $\mathcal{S} = \{s^i\}_{i=1, \dots, w}$ . The evolutionary algorithm goal is to minimise the error between  $\mathcal{S}_{tgt}$  and  $\mathcal{S}$ . Although simply put, this error must be done carefully as the sampling of the P system's  $\mathcal{S}$  and that of  $\mathcal{S}_{tgt}$  are different. If  $s_{tgt}^i \in \mathcal{S}_{tgt}$ , with (dropping the super-index for simplicity)  $s_{tgt} = \{y(0), y(\varepsilon), y(2\varepsilon), \dots, y(n\varepsilon)\}$  and  $\varepsilon$  the time step precision for  $\mathcal{S}_{tgt}$ , and  $s = \{y'(0), \dots, y'(t'_j), \dots, y'(t'_m)\}$  there is no direct mapping from  $t'_j$  (in  $s$ ) to  $k\varepsilon$  (in  $s_{tgt}$ ) for some  $k \geq 1$  as the time interval simulated is not uniformly sampled under a Gillespie dynamics. In order to compute the error between a given  $y'(t'_j)$  and a candidate  $\hat{y}$  interpolated from  $s_{tgt}$  we need to interpolate the value  $\hat{y}(t'_j)$  that  $s_{tgt}$  would take at  $t'_j$ . Note that the only point in time that is guaranteed to match in both time series is  $t_0$ , so we can obtain the index

$$k = \lfloor \frac{t'_j - t_0}{\varepsilon} \rfloor.$$

With the index  $k$  we can interpolate  $s_{tgt}$  between the time steps  $t_k$  and  $t_{k+1}$ :

$$q = \frac{y(t_{k+1}) - y(t_k)}{\varepsilon}$$

that is the slope of the segment of line that runs between points  $(t_k, y(t_k))$  and  $(t_{k+1}, y(t_{k+1}))$ . With  $q$  we can interpolate the value of  $s_{tgt}$  at time  $t'_j$  with

$$\hat{y}(t'_j) = y(t_k) + q (t'_j - t_k).$$

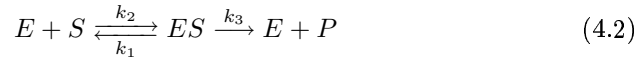
With this provision in mind the parameter learning problem becomes:

$$\min \sum_{s \in S_{tgt}} \frac{\sum_{\forall t'_j \in s^i, s^i \in S} \frac{|\hat{y}(t'_j) - y'(t'_j)|}{\max\{\hat{y}(t'_j), y'(t'_j)\}}}{||s^i||} \quad (4.1)$$

Eq. 4.1 is used by the EA to fit the P system to the data. This fitness measure takes into account all the time series to be approximated and the quality of the sample of each time series.

#### 4.2 A case study: the Michaelis-Menten dynamics

In order to demonstrate the feasibility of automatically tuning a P system model with an evolutionary algorithm we choose a simple case study: we apply the EA to the problem of matching the kinetic constants of a Michaelis-Menten dynamics (MM). The MM dynamics is numerically obtained through a set of differential equations that simulate the following enzymatic reactions:



where  $E$  represent the enzyme catalyzing the reaction transforming the substrate  $S$  into the product  $P$ . The reaction takes place in two different stages, the former being the reversible formation of the active complex  $ES$ , the latter being the production of  $P$ . All the details regarding the MM dynamics can be found in [7, 8].

As mentioned above these reactions are modelled by means of the following set of differential equations:

$$\begin{aligned} \frac{d[S]}{dt} &= -k_1 E_0 [S] + (k_1 [S] + k_2) [ES] \\ \frac{d[ES]}{dt} &= k_1 E_0 [S] - (k_1 [S] + k_2) [ES] \\ \frac{d[P]}{dt} &= k_2 [ES] \end{aligned} \quad (4.3)$$

where  $E_0$  represents the concentration of the total amount of enzyme (i.e., the free enzyme plus that bounded to the substrate to form the complex  $ES$ ), while, as usual in biochemistry,  $[X]$  represent the concentration of the species  $X$ . The reactions (4.2) can be straightforwardly translated into a P system having only one compartment and three rules (each one referring to exactly one of the biochemical reactions mentioned), whose dynamics can be calculated by means of the Gillespie algorithm.

Without loss of generality, we arbitrarily fix the three kinetic constants to  $k_1 = 1000$ ,  $k_2 = 1$  and  $k_3 = 0.05$  and we numerically solve the differential equations. The initial conditions used are 0.001 M for the initial substrate  $S$  and  $0.5 \cdot 10^{-3}$  M for the initial concentration of the enzyme  $E$  (no product  $P$  neither active complex  $ES$  is present at the beginning). We thus obtain three time series that represent the *target behaviour* the P system must imitate. The evolutionary algorithm thus must coerce the P system to mimic as close as possible the MM dynamics (with an imaginary volume fixed to  $1.67 \cdot 10^{-15}$  liters<sup>1</sup>, needed to translate concentrations into objects and deterministic rate constant into stochastic ones).

Figure 7 shows the progress of the evolutionary engine while trying to match with a P system the time series generated by the Michaelis-Menten process. Figure 8 shows the actual display of the evolved P system's concentrations and the target concentrations.

## 5 Conclusions and further work

We have briefly described a part of the quorum sensing network in the *Pseudomonas aeruginosa*. Starting from a differential equations based model we have provided a P systems version of it and we compared the dynamics of the two approaches. In order to apply different simulation strategies on this intriguing phenomenon we provided a more detailed, mechanistic model which, we believe, is closer to the biological reality. The lack of biological information regarding the dynamics of the system led us to use an automatic way for estimating the parameters involved by using an evolutionary algorithm approach that offers a reliable and effective method in this respect.

An immediate step further, after obtaining all the parameters regulating a single bacterium dynamics, is to extend the proposed model at a colony level, exploiting compartments offered by P systems and already established population P systems models.

Other important developments are related to the use of experimental data to tune the dynamics of our specifications such as to simulate real biological processes. In this respect the use of model checking methodologies, already under consideration in a paper under preparation, will contribute towards validating certain properties of the systems modelled.

On the long term we believe that these steps can represent the first stage toward a quantitative analysis that will hopefully lead to a better understanding of the systems under consideration.

## 6 Acknowledgements

N. Krasnogor and P. Siepmann acknowledge the EPSRC for funding project EP/D021847/1.

<sup>1</sup> With this choice a concentration of  $1nM$  corresponds to exactly one molecule. Moreover it is consistent with the order of magnitude of the volume of a bacterium.

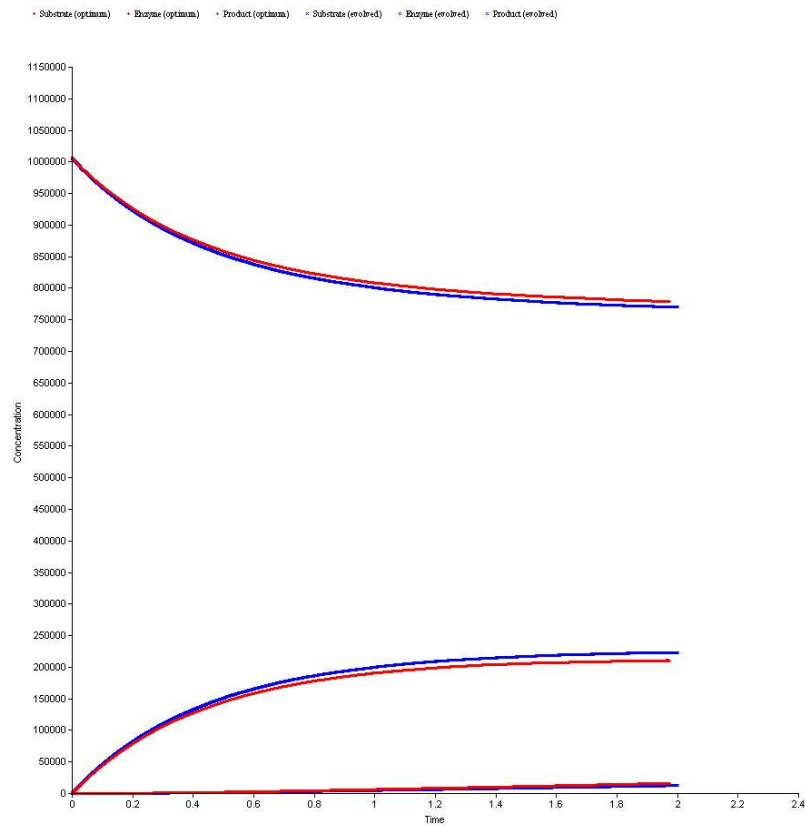


**Fig. 7.** Fitness progress of the parameter learning process. Best individual and average error in the population is shown.

## References

1. K. Anguige, J.R. King, J.P. Ward, and P. Williams. Mathematical modelling of therapies targeted at bacterial quorum sensing. *Mathematical Biosciences*, 192:39–83, 2004.
2. A. P. Arkin. Synthetic cell biology. *Current Opinion in Biotechnology*, 12:638–644, 2001.
3. F. Bernardini and M. Gheorghe. Population P systems. *Journal of Universal Computer Science*, 10:509–539, 2004.
4. F. Bernardini, M. Gheorghe, N. Krasnogor, and J.-L. Giavitto. On self-assembly in population P systems. In C. S. Calude, M. J. Dinneen, G. Păun, and M. J. Pérez-Jiménez, editors, *Unconventional Computation. 4th International Conference, UC 2005*, pages 46–57, 2005.





**Fig. 8.** The target Michaelis-Menten concentrations and the evolved P systems ones.

5. F. Bernardini, M. Gheorghe, N. Krasnogor, R.C. Muniyandi, M.J. Pérez-Jiménez, and F.J. Romero-Campero. On P Systems as a modelling tool for biological systems. In R. Freund, G. Lojka, M. Oswald, and Gh. Pañ, editors, *Pre-Proceedings of the 6Th International Workshop on Membrane Computing (WMC6)*, pages 193–213, 2005.
6. L. Bianco, F. Fontana, and V. Manca. P Systems with Reaction Maps. *International Journal of Foundations of Computer Science*, 17(1):27–48, 2006.
7. G. E. Briggs and J. B. S. Haldane. A note on the kinetics of enzyme action. *Biochem. J.*, 19:338–339, 1925.
8. K. A. Connors. *Chemical kinetics: The study of Reaction Rates in Solution*. VCH, 1990.
9. C. V. Delen and B. H. Iglewski. Cell-to-cell Signalling and *Pseudomonas Aeruginosa* Infections. *Emerging Infectious Diseases*, 4(4):551–560, October-December 1998.

10. S. P. Diggle, K. Winzer, A. Lazdunski, P. Williams, and M. Cámara. Advancing the Quorum in *Pseudomonas aeruginosa*: MvaT and the Regulation of N-Acylhomoserine Lactone Production and Virulence Gene Expression. *Journal of Bacteriology*, 184:2576–2586, 2002.
11. J. D. Dockery and J. P. Keener. A Mathematical Model for Quorum Sensing in *Pseudomonas aeruginosa*. *Bulletin of Mathematical Biology*, 63:95–116, 2001.
12. D. T. Gillespie. A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions. *Journal of Computational Physics*, 22:403–434, 1976.
13. D. T. Gillespie. Exact Stochastic Simulation of Coupled Chemical Reactions. *Journal of Computational Physics*, 81(25):2340–2361, 1977.
14. A. M. Lazdunski, I. Ventre, and J. N. Sturgis. Regulatory Circuits and Communication in Gram-Negative Bacteria. *Nature Reviews, Microbiology*, 2:581–592, 2004.
15. G. Păun. Computing with membranes. *J. Comput. System Sci.*, 61(1):108–143, 2000.
16. G. Păun. *Membrane Computing. An Introduction*. Springer, Berlin, Germany, 2002.
17. J. P. Pearson. Early Activation of Quorum Sensing. *Journal of Bacteriology*, 184:2569–2571, 2002.
18. M. J. Peñez-Jimeñez and F. J. Romero-Campero. P Systems a New Computational Modelling Tool for Systems Biology. *Transactions in Computational Systems Biology*. (In press), 2006.
19. M.J. Pérez-Jiménez and F.J. Romero-Campero. Modelling Vibrio Fischeri's behaviour using P Systems. In *Systems Biology Workshop, ECAL*, 2005.
20. D. Pescini, D. Besozzi, G. Mauri, and C. Zandron. Dynamical probabilistic P systems. *International Journal of Foundations of Computer Science*, 17(1):183, 2006.
21. P. A. Siepmann, G. Terrazas, and N. Krasnogor. Evolutionary Design for the Behaviour of Cellular Automaton-Based Complex Systems. In *Proceedings of the Seventh International Conference on Adapting Computing in Design and Manufacture*.
22. G. Terrazas, N. Krasnogor, M. Gheorghe, F. Bernardini, S. Diggle, and M. Camara. An Environment Aware P System Model of Quorum Sensing. In S. Barry Cooper, B. Löwe, and L. Torenvliet, editors, *New Computational Paradigms. First Conf. on Computability in Europe, CiE2005*, pages 479–485, 2005.
23. A. U. Viretta and M. Fussenegger. Modelling the Quorum Sensing Regulatory Network of Human-Pathogenic *Pseudomonas aeruginosa*. *Biotechnol. Prog.*, 20:670–678, 2004.
24. J. P. Ward, J.R. King, A.J. Koerber, P. Williams, J. M. Croft, and R. E. Sockett. Mathematical modelling of quorum sensing in bacteria. *Journal of Mathematics Applied in Medicine and Biology*, 18:263–292, 2001.
25. K. Winzer, K. R. Hardie, and P. Williams. Bacterial cell-to-cell communication: sorry, can't talk now – gone to lunch! *Current Opinon in Microbiology*, 5:216–222, 2002.