

# Modelling gene expression control using P systems: The Lac Operon, a case study

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

*Research Group on Natural Computing, Department of Computer Science and Artificial Intelligence,  
University of Sevilla, Avda. Reina Mercedes s/n, 41012 Sevilla, Spain*

## Abstract

In this paper P systems are used as a formal framework for the specification and simulation of biological systems. In particular, we will deal with gene regulation systems consisting of protein–protein and protein–DNA interactions that take place in different compartments of the hierarchical structure of the living cell or in different individual cells from a colony. We will explicitly model transcription and translation as concurrent and discrete processes using rewriting rules on multisets of objects and strings. Our approach takes into account the discrete character of the components of the system, its random behaviour and the key role played by membranes in processes involving signalling at the cell surface and selective uptake of substances from the environment. Our systems will evolve according to an extension of Gillespie’s algorithm, called *Multicompartmental Gillespie’s Algorithm*. The well known gene regulation system in the Lac Operon in *Escherichia coli* will be modelled as a case study to benchmark our approach.

*Keywords:* P systems; Systems biology; Gillespie’s algorithm; Gene expression control; Lac Operon

## 1. Introduction

Membrane Computing is an emergent branch of Natural Computing introduced by Păun (2000). 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 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 (Bianco et al., 2005; Ciobanu et al., 2006; Pescini et al., 2006; Pérez-Jiménez and Romero-Campero, 2006).

\* Corresponding author.

*E-mail addresses:* [fran@us.es](mailto:fran@us.es) (F.J. Romero-Campero), [marper@us.es](mailto:marper@us.es) (M.J. Pérez-Jiménez).

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 such as ordinary differential equations (ODEs). Differential equations have been used successfully to model kinetics of conventional macroscopic 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 and that for each type of reaction in the system, the number of reactions is large within each observation interval, that is reactions are fast.

When the number of particles of the reacting species is low and reactions are slow, which is frequently the case in gene expression control in bacteria and viruses, both of the previous presumptions are invalid and the deterministic continuous approach to chemical kinetics is questionable. Instead one has to recognise that the individual chemical reaction steps occur discretely and are separated by time intervals of random length.

Besides it is well known that transcription and translation in bacteria are concurrent and discrete processes that produce a delay in the expression of genes. The order in which different genes are found in operons on the genome of *Escherichia coli* is also a relevant feature in the functioning of the gene expression control. All these characteristics are not easily specified and simulated in classical approaches like ODEs.

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 of the system by using rewriting rules on multisets of objects, that represent chemical substances, and strings, that represent the organisation of genes on the genome. The inherent randomness in biological phenomena is captured by using stochastic strategies. Moreover, the key feature of P systems is the so called membrane structure which represents the compartmentalisation 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 instance, diffusion, selective uptake of molecules from the environment and signalling at the cell surface.

In this paper we will present P systems as a reliable tool for the specification and simulation of cellular processes within the framework of Systems Biology. We will use a strategy, based on the well known Gillespie's algorithm but running on more than one compartment, called *Multi-compartmental Gillespie Algorithm* for the evolution of our systems.

The paper is organised as follows. In the next section we present P systems as a specification language for cellular processes. In Section 3 the *Multicompartmental Gillespie's Algorithm* is described. A brief description of the gene expression control in the Lac Operon is given in Section 4. The next section consists of our model of the Lac Operon. In Section 6 some results are discussed. Finally, conclusions are presented in the last section.

## 2. P Systems as a Specification Language for Cellular Processes

In the structure and the 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.

The processes taking place inside compartments or on the membranes delimiting them are inherently discrete. Chemical reactions can be seen as rewriting rules on multisets of objects whereby some objects (reactants) are replaced with others (products); and genetic processes like transcription and translation can be seen as rewriting rules on strings whereby a substring (site) is replaced with another one.

Inspired by these biological features Păun (2000) introduced P systems as an unconventional model of computation; for details and updated information on P systems we refer to Păun (2002).

A P system is usually defined as a hierarchical arrangement of a number of membranes identifying a corresponding number of regions inside the system. These regions are associated to a finite multiset of objects and a finite set of rules. We will also associate a finite multiset of strings to the membranes representing the genetic information encoded in DNA and RNA.

In what follows we give a precise definition of the main components of a P system.

A P system is a construct

$$\Pi = (\Sigma_{\text{prot}}, \Sigma_{\text{dna}}, \Sigma_{\text{rna}}, L, \mu, M_1, M_2, \dots, M_n, R_1, \dots, R_n)$$

where:

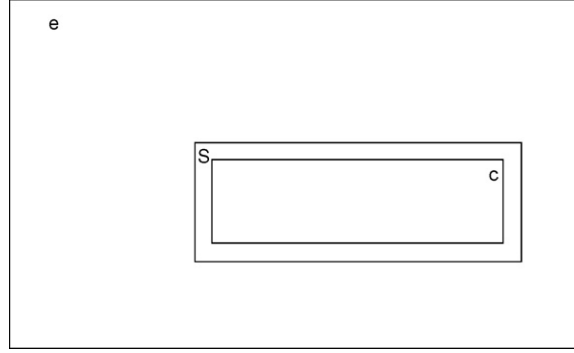


Fig. 1. Membrane structure.

- $\Sigma_{\text{prot}}$  is a finite alphabet of symbols representing proteins;
- $\Sigma_{\text{dna}}$  is a finite alphabet of symbols representing DNA sites;
- $\Sigma_{\text{rna}}$  is a finite alphabet of symbols representing RNA sites;
- $L$  is a finite alphabet of symbols representing labels for the compartments;
- $\mu$  is a membrane structure containing  $n \geq 1$  membranes identified with numbers from  $1, \dots, n$  labelled with elements from  $L$ . Two membranes with the same label will represent two compartments of the same type.

Formally, a membrane structure is defined as 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. The membrane structure can be represented as a rooted tree, where the nodes are called membranes, the root is called skin, and the relationship of a membrane being inside another one is represented by the relationship of the node being the descendent of another one.

We can represent a membrane structure using Venn diagrams.

In this paper, and in some recent attempts to model cell processes (Cheruku et al., 2007; Pérez-Jiménez and Romero-Campero, 2006), a membrane defines an homogeneous and well stirred region; and so it will not necessarily correspond with a biological membrane. Specifically, in the systems we are going to study there are biological membranes that can be considered regions, where proteins, receptors, channels, etc... are located. In this case we will use a region delimited by two membranes, instead of a single membrane, to specify and simulate the processes, taking place on the membrane surface.

For instance, if in the system we are studying there are some receptors placed on the cell surface involved in signalling transduction we will use membranes to represent this three different regions, namely, the environment ( $e$ ), the cell surface ( $s$ ) and the cytoplasm ( $c$ ). This membrane structure can be represented using a Venn diagram as in Fig. 1.

- $M_i = (l_i, w_i, s_i)$ , for each  $1 \leq i \leq n$ , is the initial configuration of membrane  $i$  with  $l_i \in L$  its type,  $w_i \in \Sigma_{\text{prot}}^*$  a finite multiset of object-proteins and  $s_i$  is a finite multiset of strings over  $\Sigma_{\text{dna}}$ ;
- $R_i$ , for each  $1 \leq i \leq n$ , is a finite set of rules associated to membrane  $i$ .

Rules of many different forms have been considered for P systems in order to encode the operation of modifying the objects inside the membrane, of moving objects from one region to the other, dissolving, creating, dividing membranes, etc. Here we will also consider rules working on strings representing DNA or RNA.

The general form of the rules we are going to use is one of the following:

- Protein-protein interaction rules:

$$\mathbf{u}[v]_l \rightarrow \mathbf{u}'[v']_l$$

where  $u, v, u', v'$  are the multisets of objects from  $\Sigma_{\text{prot}}$  and  $l$  is a label from  $L$ .

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 communication rules and transformation rules. These rules are referred to as boundary rules in [Bernardini and Manca \(2002\)](#).

- Genetic rules:

$$[u, s]_l \rightarrow [u', s']_l$$

where  $u, u'$  are the multisets of objects from  $\Sigma_{\text{prot}}$ ,  $s, s'$  the strings over  $\Sigma_{\text{dna}} \cup \Sigma_{\text{rna}}$  and  $l$  is a label from  $L$ .

These rules operate on both objects and strings, that is, a multiset  $u$  placed inside a membrane labelled with  $l$  will be replaced with a multiset  $u'$ ; simultaneously  $s$  will be rewritten by  $s'$  on a string placed inside the same membrane and having  $s$  as substring.

We will also 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 like kinetic or stochastic constants.

In what follows we discuss in more detail the specific rules we will use to model cellular processes.

### 2.1. Protein–Protein Interaction Rules

- Transformation, complex formation and dissociation rules:

$$[a]_l \rightarrow [b]_l$$

$$[a, b]_l \rightarrow [c]_l$$

where  $a, b, c \in \Sigma_{\text{prot}}$  and  $l \in L$

$$[a]_l \rightarrow [b, c]_l$$

These rules are used to specify chemical reactions taking place inside a compartment of type  $l \in L$ , more specifically they represent the transformation of  $a$  into  $b$ , the formation of a complex  $c$  from the interaction of  $a$  and  $b$  and the dissociation of a complex  $a$  into  $b$  and  $c$ .

- Diffusing in and out:

$$[a]_l \rightarrow a[]_l$$

where  $a \in \Sigma_{\text{prot}}$  and  $l \in L$

$$a[]_l \rightarrow [a]_l$$

When chemical substances move or diffuse freely from one compartment to another we use these types of rules, where  $a$  moves from or to a compartment of type  $l$ .

- Binding and debinding rules:

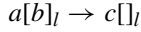
$$a[b]_l \rightarrow [c]_l$$

where  $a, b, c \in \Sigma_{\text{prot}}$  and  $l \in L$

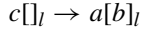
$$[a]_l \rightarrow b[c]_l$$

Using rules of the first type we can specify reactions consisting in the binding of a ligand swimming in one compartment to a receptor placed on the membrane surface of another compartment. The reverse reaction, debinding of substance from a receptor, can be described as well using the second rule.

- Recruitment and releasing rules:



where  $a, b, c \in \Sigma_{\text{prot}}$  and  $l \in L$

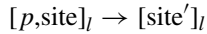


With these rules we represent the interaction between two chemicals in different compartments whereby one of them is recruited from its compartment by a chemical on the other compartment, and then the new complex remains in the latter compartment. In a releasing rule a complex,  $c$ , located in one compartment can dissociate into  $a$  and  $b$ , remaining  $a$  in the same compartment as  $c$ , and  $b$  being released into the other compartment.

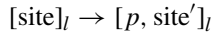
These types of rules have been used in [Cheruku et al. \(2007\)](#); [Pérez-Jiménez and Romero-Campero \(2006\)](#) to model signal transduction pathways consisting of protein–protein interactions. Next, we describe the type of rewriting rules on strings that will be used to model genetic reactions. A previous attempt to model genetic reactions within the framework of P systems, but using rewriting rules only on multisets of objects is presented in [Romero-Campero and Pérez-Jiménez \(in press\)](#). Our attempt is novel in the sense that we use rewriting rules on both objects and strings as it will be described in what follows.

## 2.2. Genetic Rules

- Binding and debinding of a protein to specific site on the DNA:

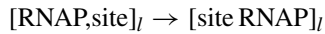


where  $p \in \Sigma_{\text{prot}}$  and  $\text{site}, \text{site}' \in \Sigma_{\text{dna}}$



A protein,  $p$ , can bind a region of the DNA,  $\text{site}$ , yielding a new region  $\text{site}'$ . Reversely  $p$  can dissociate from  $\text{site}$  producing the free protein and a new region  $\text{site}'$ .

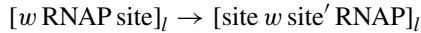
- Polymerase binding to a specific site on the DNA:



where  $\text{RNAP} \in \Sigma_{\text{prot}}$  represents the RNA polymerase and  $\text{site} \in \Sigma_{\text{dna}}$  a promoter in the DNA.

RNA polymerase, RNAP, recognises a specific site on the DNA, binds there and starts transcription. When one rule of this type is applied  $\text{site}$  is replaced with  $\text{site RNAP}$  in the corresponding string.

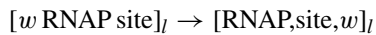
- Transcription:



where  $w \in \Sigma_{\text{rna}}^*$ ,  $\text{site} \in \Sigma_{\text{dna}}$ ,  $\text{site}' \in \Sigma_{\text{rna}}$  and  $\text{RNAP} \in \Sigma_{\text{prot}}$  represents the RNA polymerase.

RNAP transcribes DNA into mRNA producing a strand of complementary RNA that remains attached to it during the process. Therefore, when RNAP transcribes a part of the DNA,  $\text{site}$ , it leaves it behind and attached its complementary part of RNA,  $\text{site}'$ , to the growing mRNA  $w$ .

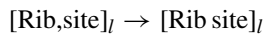
- Polymerase dissociation when reaching a termination site in the DNA:



where  $w \in \Sigma_{\text{rna}}^*$ ,  $\text{site} \in \Sigma_{\text{dna}}$  and  $\text{RNAP} \in \Sigma_{\text{prot}}$  represents the RNA polymerase.

When the RNAP reaches a termination site it ends transcription by dissociating from the DNA and releasing the mRNA sequence that can be, by that time, being translated by some ribosomes.

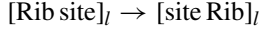
- Binding of a ribosome to mRNA:



where  $\text{Rib} \in \Sigma_{\text{rna}}$  represents a ribosome and  $\text{site} \in \Sigma_{\text{rna}}$  represents a ribosome binding site.

Shortly after RNAP starts transcription and before it is over, ribosomes bind to the growing mRNA to start translation. Using this rule the substring site is replaced with Rib site consuming one object Rib.

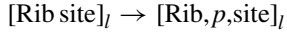
– Translation:



where  $\text{Rib} \in \Sigma_{\text{rna}}$  represents a ribosome and  $\text{site} \in \Sigma_{\text{rna}}$  represents the part of the mRNA that is going to be translated.

Ribosomes translate mRNA into a sequence of amino acids that will yield a protein when the ribosome reaches a termination codon. With these rules we specify how ribosomes move along mRNA, so the substring Rib site is replaced with site Rib representing that site has been translated.<sup>1</sup>

– Dissociation of a ribosome from the RNA:



where  $\text{Rib} \in \Sigma_{\text{rna}}$  represents a ribosome,  $\text{site} \in \Sigma_{\text{rna}}$  represents a termination site on the mRNA and  $p \in \Sigma_{\text{prot}}$  represents the protein encoded on the part mRNA that has been translated.

When a ribosome reaches a termination codon it dissociates from the mRNA releasing the translated protein. The substring Rib site triggers the end of translation, then the objects Rib and  $p$  are produced representing the dissociation of a ribosome and the releasing of the protein whereas the substring Rib site is replaced with Site on the corresponding string representing the mRNA.

### 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 (Gillespie, 1977)(see also (Gillespie, 2001, 2003) 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 (Meng et al., 2004). Here we will use an extension of the classical Gillespie's algorithm called *Multi-compartmental Gillespie Algorithm* that was introduced in Pérez-Jiménez and Romero-Campero (2006). 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 of these compartments 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 rule that moves objects from one membrane to another.

Specifically, let  $\Pi = (\Sigma_{\text{prot}}, \Sigma_{\text{dna}}, \Sigma_{\text{rna}}, L, \mu, M_1, M_2, \dots, M_n, R_1, \dots, R_n)$  be a P system as specified in the previous section with membranes  $M_i = (l_i, w_i, s_i)$  and rules  $R_i$ ,  $1 \leq i \leq n$ . Each rule,  $r_j$ , from the set  $R_i$  of rules in membrane  $i$  will be associated with an attribute  $c_j$  which represents a stochastic constant, that can be interpreted as the average number of applications of the rule per unit time.

This constant is used to compute  $p_j$  (the probability of the rule  $j$  to be applied in the next step of evolution) this probability is computed by multiplying the stochastic constant  $c_j$ , specifically associated as an attribute with rule  $j$ , by the number of possible combinations of the objects and sites present on the left-side of the rules with respect to the multiset  $w_i$  and the string  $s_i$  (or the multiset  $w_i'$  contained in the membrane outside membrane  $i$ ) – the current content of membrane  $i$  ( $i'$ ).

Each membrane  $i$  will be considered as 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

<sup>1</sup> Note that in contrast to transcription it is not necessary to specify the growing sequence of amino acids since we are not interested in studying the processes that take place on it.

recall below:

- (1) calculate  $a_0 = \sum p_j$ , for all  $r_j \in R_i$ ;
- (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  $\tau_i = (1/a_0) \ln(1/r_1)$ ;
- (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 and sites 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  $(\tau_i, j, i)$  by using the procedure described above; construct a list containing all such triples;
  - sort this list of triples  $(\tau_i, j, i)$  in increasing order 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  $j$  in membrane  $i$  only once changing the number of objects and sites 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 rule  $j''$ , 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 rules 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 rule and waiting time for each one of these membranes by taking into account the new distribution of objects and strings inside them. Note that in this point our approach differs from [Stundzia and Lumsden \(1996\)](#) where only one rule is applied at each step without taking into account the rest of rules that are waiting to be applied in other membranes, neither it is considered the disruption that the application of one rule 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.

In the following two sections we briefly describe the gene expression control in the Lac Operon in *E. coli* and we present a model using the formalism discussed in the previous section.

#### 4. Gene Expression Control in the Lac Operon

Many of the genes in *E. coli* are expressed constitutively; that is, they are always turned "on". Others, however, are active only when their products are needed by the cell, so their expression must be regulated. The most direct way

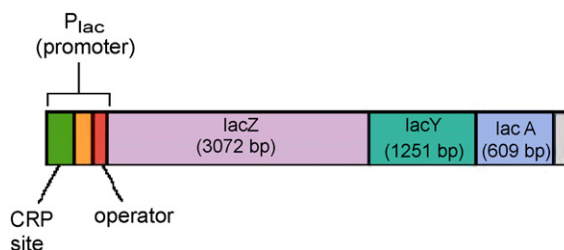


Fig. 2. Lac Operon.

to control the expression of a gene is to regulate its rate of transcription; that is, the rate at which RNA polymerase transcribes the gene into molecules of messenger RNA (mRNA).

Adding a new substrate to the culture medium may induce the formation of new enzymes capable of metabolising that substrate. An example of this phenomenon happens when we take a culture of *E. coli* that is feeding on glucose and transfer some of the cells to a medium containing lactose instead, a revealing sequence of events takes place. At first the cells are quiescent: they do not metabolise lactose, their other metabolic activities decline and cell division ceases. Soon, however, the culture begins growing rapidly again with the lactose being rapidly consumed. During the quiescent interval, the cells began to produce three enzymes that they had not been producing before: a permease, LacY, that transports lactose across the plasma membrane from the culture medium into the interior of the cell;  $\beta$ -galactosidase which hydrolyses lactose into glucose and galactose, and a transacetylase, LacA, whose function is still uncertain.

The genes encoding these proteins and other proteins involved in the regulation of their transcription are located on a region of the *E. coli* genome called *Lac Operon* (Fig. 2).<sup>2</sup>

The gene *lacI* encodes a protein called LacI that acts as a repressor. The lac repressor is made up of four identical polypeptides (the protein product of the gene *lacI*). Part of this molecule has a site that enables it to recognise and bind to 24 base pairs of the lac promoter called the lac operator, *op*, preventing the RNA polymerase from transcribing the structural genes *lacZ*, *lacY* and *lacA* that encode  $\beta$ -galactosidase, the permease and the transacetylase, respectively.

Nonetheless, sometimes the repressor drops from the promoter allowing transcription at basal rate. Besides, the repressor contains another site where allolactose, a product of the reaction of lactose with  $\beta$ -galactoside, can bind producing a conformational change. As a result of this change, it can no longer bind to the operator region and falls off. RNA polymerase can then bind to the promoter and transcribe the lac genes.

Thus, when lactose is added to the culture medium, it causes the repressor to be released from the operator so RNA polymerase can transcribe the three structural genes of the operon into a single molecule of messenger RNA. Hardly does transcription begin before ribosomes attach to the growing mRNA molecule and move down it to translate it into the three proteins. Transcription and translation are concurrent processes in bacteria which play an important role in gene expression control.

Absence of active lac repressor is essential but not sufficient for effective transcription of the Lac Operon. The presence of glucose in the culture medium, even in the presence of lactose, seems to repress or inhibit the synthesis of  $\beta$ -galactosidase. The molecular mechanism of this effect is called *catabolite repression*.

Catabolite repression is mediated through the effects that glucose transport into the cell has on the internal concentration of cyclic AMP (cAMP). If glucose is abundant in the growth medium it will be transported in to the cell by the action of the glucose transport system. As it is being transported, glucose is phosphorylated with the phosphate group being donated by a component of the transport system called EIIA ~ P. The same component also activates the enzyme, adenylate cyclase (AC). As long as the component is participating in glucose transport, it is not able to activate adenylate cyclase. The result is that as glucose is transported into the cell, the concentration of cAMP falls (because adenylate cyclase is not being activated to synthesise any more). If there is little or no glucose in the growth medium, the glucose transport system is not operational. The phosphate donor component is now free to activate adenylate cyclase.<sup>3</sup>

<sup>2</sup> An operon is a group of genes physically linked on the chromosome and under the control of the same promoters. In an operon, the linked genes give rise to a single mRNA that is translated into the different gene products. This type of mRNA is called a polycistronic mRNA.

<sup>3</sup> EIIA, the non-phosphorylated state of EIIA ~ P, inhibits the permease involved in the uptake of lactose inside the bacterial cell preventing lactose from entering the cytoplasm.



The result is that in the absence of glucose, the concentration of cAMP rises. Thus there is an inverse relationship between the external concentration of glucose and the cytoplasmic concentration of cAMP. As one rises, the other falls.

Therefore, when glucose is scarce in the medium, cAMP is abundant in the cytoplasm and it can be bound by the cAMP receptor protein (CRP), which is also known as catabolite activator protein (CAP). As its name suggests, this protein is responsible for mediating the phenomenon of catabolite repression through its ability to activate transcription. The complex CRP–cAMP<sub>2</sub> binds to the Lac Operon just upstream of the promoter. In this position it can assist RNA polymerase to bind by direct protein–protein contacts increasing the rate of transcription hugely.

Summing up, the lactose operon is subject to both negative and positive control. The lac repressor, LacI, negatively regulates expression and, the activator, cAMP–CRP<sub>2</sub>, positively activates expression.

There are, as a result, four basic states of expression of the Lac Operon:

- No glucose and no lactose

Under these conditions, there will be a large number of cAMP molecules in the cell and CRP–cAMP<sub>2</sub> will be bound at its binding site upstream of the lac promoter. It will assist RNA polymerase to bind to the promoter but it will not activate transcription because the lactose repressor will remain bound to the operator site since there is no inducer, allolactose, present.

There will be essentially no transcription of the Lac Operon.

This makes physiological sense. Without sugar substrates the cell cannot carry out much metabolism; however, it remains poised to use whatever it can whenever it can. In this case, if lactose does become available, the cell can and will immediately respond because lactose permease will transport the lactose into the cell and RNA polymerase is positioned to start the expression of β-galactosidase so that the lactose can be utilized immediately.

- Glucose present but no lactose

Under these conditions, there will be a low number of cAMP molecules in the cell so CRP–cAMP<sub>2</sub> will not be bound at the lac promoter. In addition, the activity of lactose permease will be inhibited.

There will be no transcription of the Lac Operon.

This also makes physiological sense. As long as glucose is present in the growth medium there is little need to metabolise lactose and since lactose is not present there is no need to transport lactose into the cell or to express the genes of the Lac Operon.

- Glucose and lactose present

Under these conditions, there will be a low number cAMP molecules in the cell so CRP–cAMP<sub>2</sub> will not be bound at the lac promoter. Lactose permease will be inhibited but some lactose will still enter the cell.

There will be a low level transcription of the Lac Operon.

Again, this makes physiological sense. As long as glucose is present in the growth medium there is little need to metabolise lactose. However, since lactose is now present, the cell would be foolish to ignore a sugar supply completely. The Lac Operon will be induced but, since CRP is not bound, the amount of transcription is relatively low.

- No glucose but abundant lactose

Under these conditions, there will be a large number of cAMP molecules in the cell so CRP–cAMP<sub>2</sub> will be bound at the lac promoter. Lactose permease is not inhibited, so it will transport the lactose into the cell.

There will be maximal transcription of the Lac Operon.

This also makes physiological sense. With lactose as the sole sugar source, the cell must use every available molecule for its own benefit. Thus the lactose permease transport system will transport lactose into the cell and the Lac Operon will be both induced and activated.

The presence of two separate control systems allows the cell to respond more sensitively to the needs imposed by changing growth conditions. Many bacterial operons have dual control systems. The details are different in the different cases, however (Ptshane, 2004; Ptashne and Gann, 2002).

## 5. A Model of the Lac Operon

In this section we present a model of the gene expression control in the Lac Operon using P systems. We will study the behaviour of system for different initial conditions with/without glucose and with/without lactose.

Our model consists in the following P system:

$$\Pi = (\Sigma_{\text{prot}}, \Sigma_{\text{dna}}, \Sigma_{\text{rna}}, \{e, s, c\}, [[[\ ]_3]_2]_1, M_1, M_2, M_3, R_1, R_2, R_3)$$

where:

- In the alphabets  $\Sigma_{\text{prot}}$ ,  $\Sigma_{\text{dna}}$  and  $\Sigma_{\text{rna}}$  we represent the proteins, DNA sites and mRNA sites involved in the Lac Operon regulation.

$$\begin{aligned} \Sigma_{\text{prot}} = & \{ \text{EIIA}, \text{EIIA} \sim \text{P}, \text{EIICB}, \text{EIICB} \sim \text{EIIA} \sim \text{P}, \text{EIICB} \sim \text{P}, \text{Gluc}, \text{Gluc} \sim \text{P}, \text{EIICB} \\ & \sim \text{P} \sim \text{Gluc}, \text{Lact}, \text{LacY}, \text{Lact} \sim \text{LacY}, \text{AC}, \text{AC} \sim \text{EIIA} \sim \text{P}, \text{ATP}, \text{AC} \sim \text{EIIA} \\ & \sim \text{P} \sim \text{ATP}, \text{LacY} \sim \text{EIIA}, \beta\text{-Galac}, \text{Lact}, \text{Allolact}, \text{Galac}, \text{LacI}, \text{LacI} \sim \text{Allolact}, \text{CRP}, \text{CRP} \\ & \sim \text{cAMP}, \text{CRP} \sim \text{cAMP}_2, \text{RNAP} \} \end{aligned}$$

$$\Sigma_{\text{dna}} = \{ \text{cap}, \text{cap}^{\text{CRP} \sim \text{cAMP}_2}, \text{op}, \text{op}^{\text{LacI}}, \text{lacZ}_s, \text{lacZ}_m, \text{lacZ}_e, \text{lacY}_s, \text{lacY}_m, \text{lacY}_e, \text{lacA}_s, \text{lacA}_m, \text{lacA}_e \}$$

$$\Sigma_{\text{rna}} = \{ \text{Rib}, \overline{\text{op}}, \overline{\text{lacZ}_s}, \overline{\text{lacZ}_m}, \overline{\text{lacZ}_e}, \overline{\text{lacY}_s}, \overline{\text{lacY}_m}, \overline{\text{lacY}_e}, \overline{\text{lacA}_s}, \overline{\text{lacA}_m}, \overline{\text{lacA}_e} \}$$

- The labels will be used to identify the compartments involved in the Lac Operon regulation, namely, the environment ( $e$ ), the cell surface ( $s$ ) and the cytoplasm ( $c$ ).
- In the membrane structure we represent the relevant regions of the system, environment, cell surface and cytoplasm. In Fig. 1 Section 2 it is depicted a Venn diagram representing the membrane structure.
- In  $M_1$ ,  $M_2$  and  $M_3$  we specify the initial number of objects and strings present in each compartment and the labels associated with them.
- We will study the behaviour of the system when four different initial conditions  $M_1^1$ ,  $M_1^2$ ,  $M_1^3$  and  $M_1^4$  are associated with the environment, namely:

$$\text{No glucose and no lactose: } M_1^1 = (e, \emptyset, \lambda)^4$$

$$\text{No glucose but abundant lactose: } M_1^3 = (e, \{\text{Lact}^{300000}\}, \lambda)$$

$$\text{Abundant glucose and no lactose: } M_1^2 = (e, \{\text{Gluc}^{300000}\}, \lambda)$$

$$\text{Abundant glucose and lactose: } M_1^4 = (e, \{\text{Gluc}^{300000}, \text{Lact}^{300000}\}, \lambda)$$

- In  $M_2$  we specify the initial configuration of the cell surface:

$$M_2 = (s, \{\text{EIICB}^{2500}, \text{EIICB} \sim \text{P}^{15000}, \text{AC}^{10000}, \text{LacY}^{3000}\}, \lambda)$$

- In  $M_3$  we specify the initial configuration of the cytoplasm:

$$M_3 = (c, \{\text{RNAP}^{300}, \text{Rib}^{3000}, \text{EIIA}^{2000}, \text{EIIA} \sim \text{P}^{13000}, \text{ATP}^{1000000}, \beta\text{-Galac}^{3000}, \text{LacI}^{1500}, \text{CRP}^{10000}\}, s_3)$$

The following string,  $s_3$ , represents the Lac Operon in *E. coli*.

$$\text{cap op} \overbrace{\text{lacZ}_s \text{lacZ}_m \dots \text{lacZ}_m \text{lacZ}_e}^{30} \overbrace{\text{lacY}_s \text{lacY}_m \dots \text{lacY}_m \text{lacY}_e}^{12} \overbrace{\text{lacA}_s \text{lacA}_m \dots \text{lacA}_m \text{lacA}_e}^6$$

Due to their size polymerase occupy around 100 nucleotides, and so each site lacZ, lacY and lacA will represent 100 nucleotides of the corresponding gene, instead of a single one, where RNAP is transcribing. Therefore, we have only 30 sites lacZ, 12 lacY and 6 lacA representing the 3000, 1200 and 600 nucleotides of the corresponding genes.

The CAP binding site and the operator are represented by cap and op.

<sup>4</sup>  $\lambda$  represents the empty string and it is used to specify that no string is placed in the compartment in the initial configuration.

- $R_1$ ,  $R_2$  and  $R_3$  are the sets of rules associated with the environment, cell surface and cytoplasm.

$$R_1 = \{r_4, r_8\}$$

$$R_2 = \{r_2, r_3, r_5, r_6, r_9, r_{11}, r_{18}, r_{20}, r_{22}\}$$

$$R_3 = \{r_1, r_7, r_{10}, r_{12}, r_{13}, r_{14}, r_{15}, r_{16}, r_{17}, r_{19}, r_{21}, r_{23}, \dots, r_{57}\}$$

These rules represent the chemical reactions that take place in each compartment or region of the cell. Next we list the rules involved in the gene expression control in the Lac Operon and the attributes associated with them; which are stochastic constants representing the average number of applications of the rule per unit time. These parameters were taken from the literature listed on the references or were computed, as described in Gillespie (1977), from some deterministic kinetic constants used in ODE models assuming a bacterial volume of  $10^{-15}$  l (Huber et al., 1975; Kennell and Riezman, 1977; Kierzek et al., 2001; Lolkema et al., 1991; Rohwer et al., 2000; Wong et al., 1997).

### 5.1. Protein–Protein Interaction Rules

#### 5.1.1. Rules for the Transport of Glucose Inside the Cell

The uptake of glucose consists of a transfer pathway where a phosphoryl group is transferred sequentially along a series of proteins to glucose.



The cytoplasmic phosphorylated protein  $\text{EIIA} \sim \text{P}$  is recruited by the transmembrane protein EIICB, it can also dissociate from it.



Once  $\text{EIIA} \sim \text{P}$  is bound to EIICB, it donates the phosphoryl group producing the protein  $\text{EIICB} \sim \text{P}$  in the cell surface and releasing EIIA into the cytoplasm.

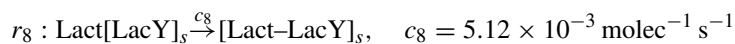


Glucose in the environment can bind to the phosphorylated protein  $\text{EIICB} \sim \text{P}$  on the cell surface and then it is transported and released into the cytoplasm being phosphorylated in the process.

#### 5.1.2. Rules for the Transport of Lactose Inside the Cell



The permease LacY is the product of the second gene on the Lac Operon, lacY. It is produced in the cytoplasm and transported to the cell surface. Note that the cell surface is outside the cytoplasm.



Lactose present in the environment is transported to the cytoplasm by binding first to LacY on the cell surface and then being released into the cytoplasm.

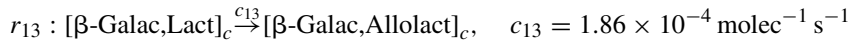


EIIA inhibits the uptake of lactose when it is recruited from the cytoplasm by the permease LacY.

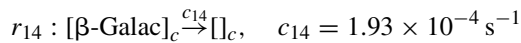
These reactions are important in catabolite repression; a high number of non-phosphorylated EIIA in the cytoplasm is a consequence of the presence of glucose in the environment. EIIA ~ P is depleted from the cytoplasm by EIIBC that transfers the phosphoryl group to the glucose present in the environment, releasing EIIA and Gluc ~ P into the cytoplasm.

Since the bacterium prefers glucose to lactose the uptake of lactose is inhibited in the presence of a high number of EIIA molecules.

### 5.1.3. $\beta$ -Galactosidase Rules



$\beta$ -Galactosidase cleaves lactose into glucose and galactose so the bacterium can consume it. Allolactose is also a product of the interaction between  $\beta$ -galactosidase and lactose and act as a signal of the presence of lactose in the environment.

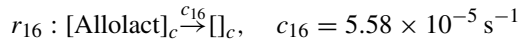


$\beta$ -Galactosidase is also degraded in the cytoplasm.

### 5.1.4. Allolactose Rules



Allolactose is a signal of the presence of lactose in the environment and so it inhibits the repressor, LacI, preventing it from binding to the operator and repressing transcription.



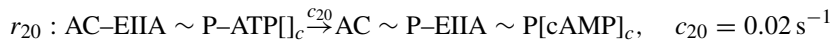
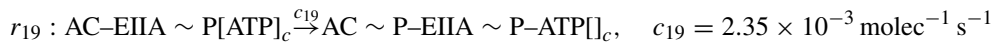
Degradation of allolactose in the cytoplasm.

### 5.1.5. Adenylate Cyclase Rules

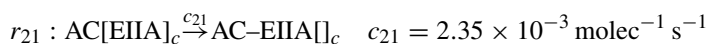
Adenylate cyclase (AC) regulates the production of cAMP, which is inversely proportional to the number of glucose molecules in the environment.



AC recruits the phosphorylated EIIA ~ P from the cytoplasm, which in turn can dissociate from it and be released again.

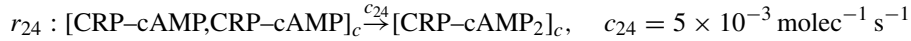


Once bound to the AC, EIIA ~ P recruits ATP from the cytoplasm and transforms it into cAMP which after released into the cytoplasm.



As mentioned before the non-phosphorylated number of molecules of EIIA is a signal of the presence of glucose in the environment. And so it inhibits the activation of the Lac Operon by repressing the activity of AC and therefore the production of cAMP and the activator CRP-cAMP<sub>2</sub>.

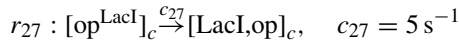
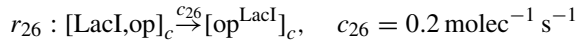
### 5.1.6. Activator Rules



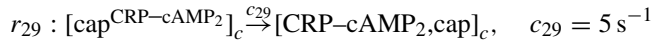
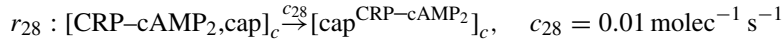
CRP by itself cannot bind to the promoter, it needs to combine with cAMP to produce the complex CRP-cAMP which in turn will form a dimer CRP-cAMP<sub>2</sub> that is able to bind to the promoter and increase the rate of transcription of the genes encoded in the Lac Operon.

### 5.2. DNA-Protein Interaction Rules

The switch of the Lac Operon is made of the CAP site, cap, and the operator, op. It has four different configurations cap op, cap op<sup>LacI</sup>, cap<sup>CRP-cAMP<sub>2</sub></sup> op and cap<sup>CRP-cAMP<sub>2</sub></sup> op<sup>LacI</sup>. These configurations can be reached by applying the following binding and dissociation rules:



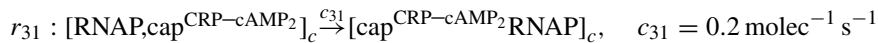
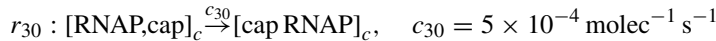
In absence of lactose the repressor LacI will be active and it will bind the operator inhibiting the transcription of the Lac Operon. Nonetheless, occasionally the LacI drops from the operator allowing a basal production of the proteins.



In absence of glucose there is a high number of cAMP molecules that will produce a high number of activators, CRP-cAMP<sub>2</sub>. The activator binds to the CAP site and recruits RNAP increasing the rate of transcription.

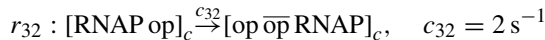
### 5.3. Genetic Rules

#### 5.3.1. Polymerase Binding to the Promoter

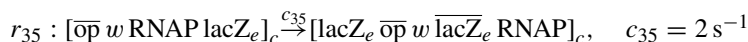
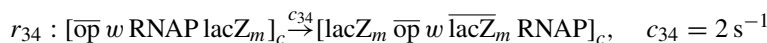
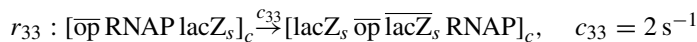


The RNA polymerase binds to the Lac Operon and starts transcription with different affinities depending on the configuration of the switch. On the one hand, when the CAP site is free RNAP seldomly binds producing a basal rate of transcription. On the other hand, if the activator CRP-cAMP<sub>2</sub> occupies the CAP site it produces an increase in the rate of transcription of around 40-fold.

#### 5.3.2. Transcription



RNAP starts transcription by leaving the operator free behind so another polymerase can bind and start transcription before the first one finishes transcribing the operon. Therefore, we are explicitly simulating transcription by different polymerase as concurrent processes.



where  $w \in \Sigma_{\text{rna}}^*$

RNAP moves along the lacZ gene transcribing it into mRNA. First it attaches  $\overline{\text{lacZ}}_s$  to the growing mRNA, which represents the ribosome binding site (RBS) for this gene. Once this site is produced a ribosome binding rule can be applied and then translation will start. Therefore, in our approach transcription and translation take place in parallel. During transcription a complementary strand of mRNA is produced. When the RNAP reaches the end of the gene it attaches  $\overline{\text{lacZ}}_e$  to the growing string. When reaching this site ribosomes will dissociate releasing the protein encoded by the lacZ gene,  $\beta$ -galactosidase.

$$\begin{aligned}
r_{36} : [\overline{\text{op}} w \text{RNAP lacY}_s]_c &\xrightarrow{c_{36}} [\text{lacY}_s \overline{\text{op}} w \overline{\text{lacY}}_s \text{RNAP}]_c, & c_{36} = 2 \text{ s}^{-1} \\
r_{37} : [\overline{\text{op}} w \text{RNAP lacY}_m]_c &\xrightarrow{c_{37}} [\text{lacY}_m \overline{\text{op}} w \overline{\text{lacY}}_m \text{RNAP}]_c, & c_{37} = 2 \text{ s}^{-1} \\
r_{38} : [\overline{\text{op}} w \text{RNAP lacY}_e]_c &\xrightarrow{c_{38}} [\text{lacY}_e \overline{\text{op}} w \overline{\text{lacY}}_e \text{RNAP}]_c, & c_{38} = 2 \text{ s}^{-1} \\
r_{39} : [\overline{\text{op}} w \text{RNAP lacA}_s]_c &\xrightarrow{c_{39}} [\text{lacA}_s \overline{\text{op}} w \overline{\text{lacA}}_s \text{RNAP}]_c, & c_{39} = 2 \text{ s}^{-1} \\
r_{40} : [\overline{\text{op}} w \text{RNAP lacA}_m]_c &\xrightarrow{c_{40}} [\text{lacA}_m \overline{\text{op}} w \overline{\text{lacA}}_m \text{RNAP}]_c, & c_{40} = 2 \text{ s}^{-1}
\end{aligned}$$

where  $w \in \Sigma_{\text{ma}}^*$

These rules specify the transcription of the other genes, lacY and lacA, following the same scheme as that of lacZ.

$$r_{41} : [\overline{\text{op}} w \text{RNAP lacA}_e]_c \xrightarrow{c_{41}} [\text{RNAP, lacA}_e, \overline{\text{op}} w \overline{\text{lacA}}_e]_c, \quad c_{41} = 2 \text{ s}^{-1}$$

where  $w \in \Sigma_{\text{ma}}^*$

When the polymerase reaches the end of the Lac Operon, represented by site lacA<sub>e</sub> it dissociates from it and releases the polycistronic mRNA encoding the proteins  $\beta$ -galactosidase, LacY and LacA.

### 5.3.3. Ribosome Binding Rules

$$\begin{aligned}
r_{42} : [\text{Rib, } \overline{\text{lacZ}}_s]_c &\xrightarrow{c_{42}} [\text{Rib } \overline{\text{lacZ}}_s]_c, & c_{42} = 0.16 \text{ molec}^{-1} \text{ s}^{-1} \\
r_{43} : [\text{Rib, } \overline{\text{lacY}}_s]_c &\xrightarrow{c_{43}} [\text{Rib } \overline{\text{lacY}}_s]_c, & c_{43} = 0.16 \text{ molec}^{-1} \text{ s}^{-1} \\
r_{44} : [\text{Rib, } \overline{\text{lacA}}_s]_c &\xrightarrow{c_{44}} [\text{Rib } \overline{\text{lacA}}_s]_c, & c_{44} = 0.16 \text{ molec}^{-1} \text{ s}^{-1}
\end{aligned}$$

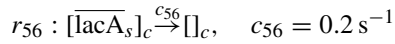
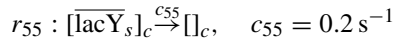
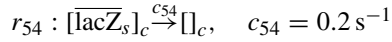
Ribosomes attach to their RBS on the growing or already transcribed mRNA and start translation.

### 5.3.4. Translation and Ribosome Dissociation Rules

$$\begin{aligned}
r_{45} : [\text{Rib } \overline{\text{lacZ}}_s]_c &\xrightarrow{c_{45}} [\overline{\text{lacZ}}_s \text{ Rib}]_c, & c_{45} = 0.3 \text{ s}^{-1} \\
r_{46} : [\text{Rib } \overline{\text{lacZ}}_m]_c &\xrightarrow{c_{46}} [\overline{\text{lacZ}}_m \text{ Rib}]_c, & c_{46} = 0.3 \text{ s}^{-1} \\
r_{47} : [\text{Rib } \overline{\text{lacZ}}_e]_c &\xrightarrow{c_{47}} [\beta\text{-Galac, Rib, } \overline{\text{lacZ}}_e]_c, & c_{47} = 0.3 \text{ s}^{-1} \\
r_{48} : [\text{Rib } \overline{\text{lacY}}_s]_c &\xrightarrow{c_{48}} [\overline{\text{lacY}}_s \text{ Rib}]_c, & c_{48} = 0.3 \text{ s}^{-1} \\
r_{49} : [\text{Rib } \overline{\text{lacY}}_m]_c &\xrightarrow{c_{49}} [\overline{\text{lacY}}_m \text{ Rib}]_c, & c_{49} = 0.3 \text{ s}^{-1} \\
r_{50} : [\text{Rib } \overline{\text{lacY}}_e]_c &\xrightarrow{c_{50}} [\text{LacY, Rib, } \overline{\text{lacY}}_e]_c, & c_{50} = 0.3 \text{ s}^{-1} \\
r_{51} : [\text{Rib } \overline{\text{lacA}}_s]_c &\xrightarrow{c_{51}} [\overline{\text{lacA}}_s \text{ Rib}]_c, & c_{51} = 0.3 \text{ s}^{-1} \\
r_{52} : [\text{Rib } \overline{\text{lacA}}_m]_c &\xrightarrow{c_{52}} [\overline{\text{lacA}}_m \text{ Rib}]_c, & c_{52} = 0.3 \text{ s}^{-1} \\
r_{53} : [\text{Rib } \overline{\text{lacA}}_e]_c &\xrightarrow{c_{53}} [\text{LacA, Rib, } \overline{\text{lacA}}_e]_c, & c_{53} = 0.3 \text{ s}^{-1}
\end{aligned}$$

During translation ribosomes move along the mRNA and when they reach the termination site,  $\overline{\text{lacZ}}_e$ ,  $\overline{\text{lacY}}_e$  or  $\overline{\text{lacA}}_e$ , they release the proteins,  $\beta$ -galactosidase, LacY or LacA.

### 5.3.5. mRNA Degradation Rules



mRNA is degraded and the RBS for the different genes are removed so ribosomes cannot bind and start translation.

## 6. Results and Discussion

Using the Multicompartmental Gillespie's Algorithm described in Section 3 and a simulator developed in Scilab, a scientific software package for numerical computations providing a powerful open computing environment for engineering and scientific applications, we have studied the behaviour of the system for different environmental conditions. Namely, absent of lactose and glucose; no glucose but abundant lactose; abundant glucose but no lactose and abundant glucose and lactose.

For these four different conditions we describe the behaviour of the system, as well as, we show the evolution of the number of molecules of some key proteins over time. The configuration of the switch in the Lac Operon for each different condition will also be discussed.

### 6.1. No Glucose and No Lactose

When glucose is not present in the environment EIIA ~ P is not consumed by the glucose transport system and it activates AC on the cell surface which in turn produces a high number of cAMP molecules, as shown in Fig. 3, and as a consequence a high number of activator molecules, CRP-cAMP<sub>2</sub>.

With a high number of activators, the CAP site will be occupied by a CRP-cAMP<sub>2</sub> molecule, which will assist RNA polymerase in binding to the promoter. Nonetheless, since there is no lactose in the environment no allolactose will be produced in the cytoplasm and therefore the repressor will be active and bound to the promoter. This will yield to the configuration of the switch that can be observed in italics on the next string representing the Lac Operon.

```
cap-CRP-cAMP_2 RNAP op -LacI lacZ_s lacZ_m -op Rib -lacZ_s
-lacZ_m RNAP lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m
-op Rib -lacZ_s -lacZ_m -lacZ_m -lacZ_m -lacZ_m -lacZ_m
-lacZ_m -lacZ_m -lacZ_m RNAP lacZ_m lacZ_m lacZ_m lacZ_m
lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m
lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m lacZ_m
lacZ_m lacZ_e lacY_s lacY_m lacY_m lacY_m lacY_m lacY_m lacY_m
lacY_m lacY_m lacY_m lacY_m lacY_m lacY_e lacA_s lacA_m lacA_m
lacA_m lacA_m lacA_e
```

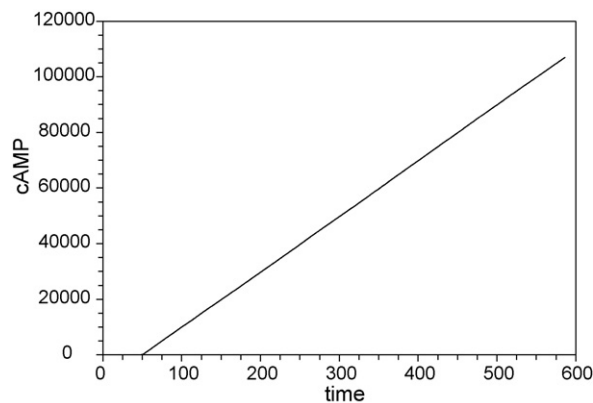


Fig. 3. Number of cAMP molecules over time in absence of glucose and lactose.

Observe that because  $\text{CRP-cAMP}_2$  is bound to the CAP site, RNAP (in bold) will be ready to start transcription whenever the repressor drops from the operator. Note that on the previous string there are two polymerases, highlighted in bold, transcribing the operon. This will produce a slight increase in the expression of the genes encoded in the Lac Operon. This makes physiological sense, with no sugar in the environment the bacterium sets the operon such that it can respond immediately and efficiently to the presence of lactose.

## 6.2. No Glucose but Abundant Lactose

As mentioned before the absence of glucose produces a high number of activators and so a  $\text{CRP-cAMP}_2$  molecule will be bound to the CAP site activating the recruitment of the RNAP. Besides when lactose is abundant in the environment it will be transported by the permease LacY, that is express at a basal level, into the cytoplasm.  $\beta$ -Galactosidase is also present at a basal level in the cytoplasm and as soon as lactose is present it starts to cleave it into galactose and glucose, and also occasionally allolactose appears as a product of the interaction between  $\beta$ -galactosidase and lactose. Allolactose acts as an inducer binding to the repressor LacI and preventing it from binding to the operator. In Fig. 4 it is depicted how the number of active repressors is rapidly inhibited when lactose is abundant.

Under these conditions the Lac Operon will be both induced (no repressor will be bound to the operator) and activated (the activator will be bound to the CAP site). Therefore, the configuration of the switch will be  $\text{cap}^{\text{CRP-cAMP}_2}$  op and the genes encoded in the operon will be transcribed massively, as it can be deduced by the number of RNAP transcribing the operon (Fig. 4). This will result in a drastic increase in the number of  $\beta$ -galactosidase and LacY molecules, see Fig. 5.

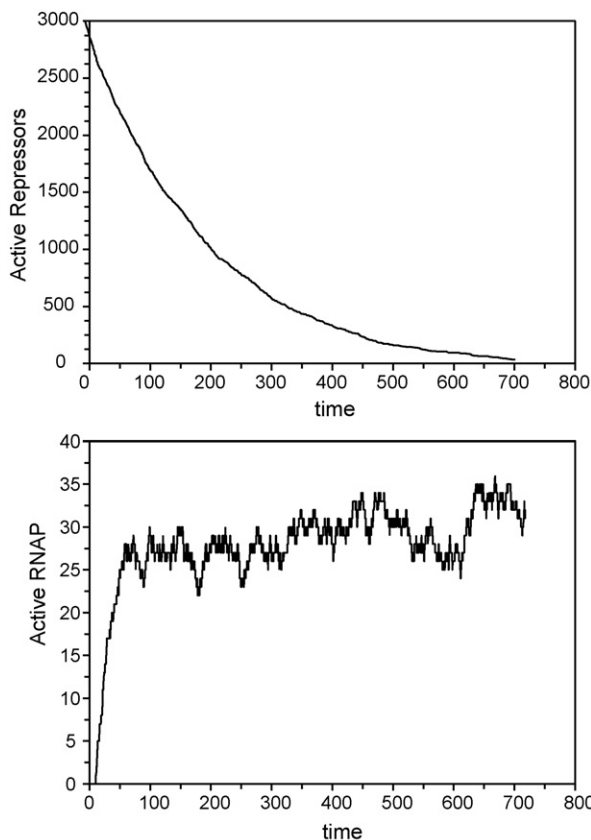


Fig. 4. Number of active repressors and RNAP.



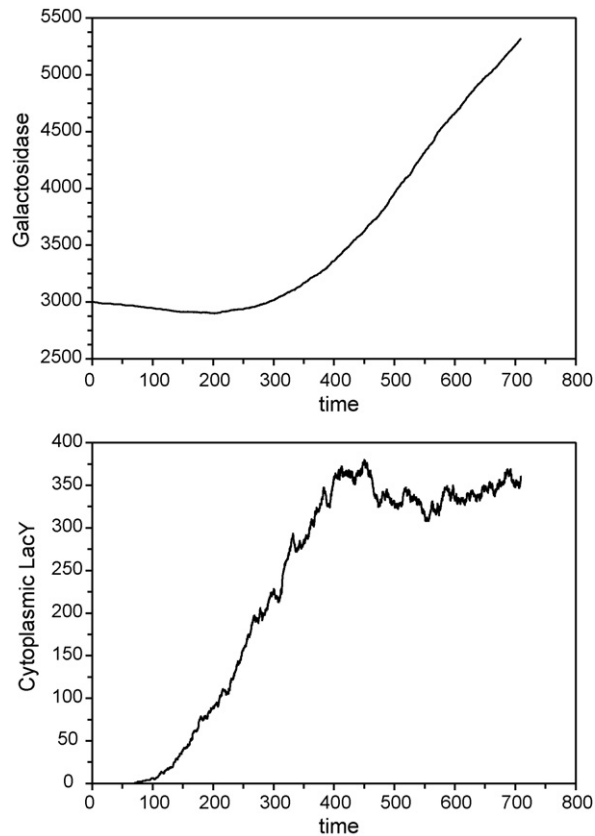


Fig. 5. Number of  $\beta$ -galactosidase and permease LacY over time.

### 6.3. Glucose Present but No Lactose

Under these conditions,  $EIIA \sim P$  is depleted rapidly from the cytoplasm by the glucose transport system and the activity of AC is repressed producing a low number of activators, see Fig. 6. Therefore, no CRP will not be bound at the lac promoter. In addition, the activity of lactose permease will be inhibited.

Since there is not lactose in the environment the repressor will be active and bound to the operator producing only transcription of the Lac Operon at a basal rate, as it can be seen in the low number of active polymerases in Fig. 7.

This makes physiological sense. As long as glucose is present in the growth medium there is little need to metabolise lactose and the lactose operon is switch off.

### 6.4. Glucose and Lactose Present

Again the presence of glucose produces a low number of activators and the Lac Operon will not be activated by the binding of a  $CRP-cAMP_2$  molecule to the CAP site. Therefore, even in the presence of lactose the genes encoded in the Lac Operon will be off,<sup>5</sup> there will be little uptake of lactose from the environment.

Observe in Fig. 8 how the number of glucose molecules decreases in the environment whereas lactose remains almost constant.

<sup>5</sup> This phenomenon is known as catabolite repression.

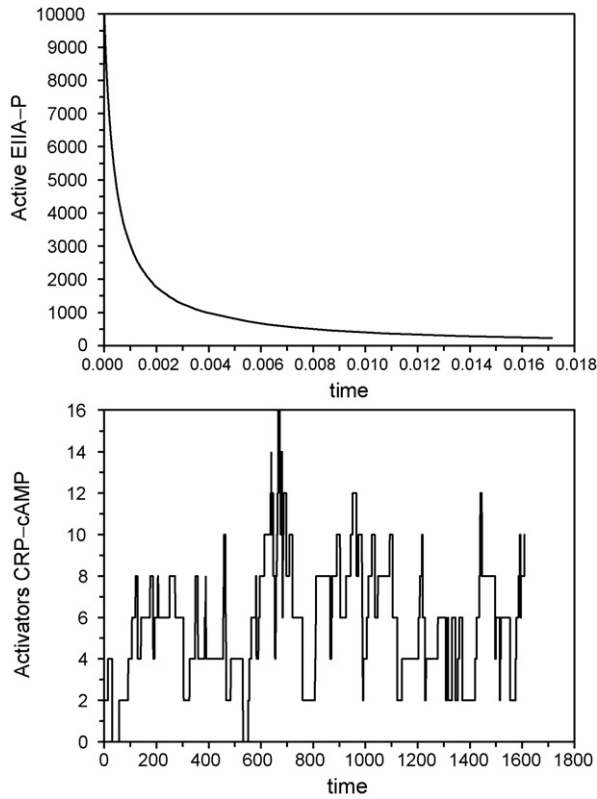


Fig. 6. Number of active EIIA ~ P and activator CRP-cAMP.

Nonetheless, some lactose will be present in the cytoplasm which will produce allolactose able to inhibit the repressor to some extent producing a low transcription of the operon. This can be seen in Fig. 9 where the number of galactosidase and LacY start to increase slowly.

Again, this makes physiological sense. As long as the bacterium can metabolise glucose there is little need to metabolise lactose. However, since lactose is now present, the cell would not ignore a sugar supply completely. The Lac Operon will be induced but, since CRP is not bound, the amount of transcription is relatively low.

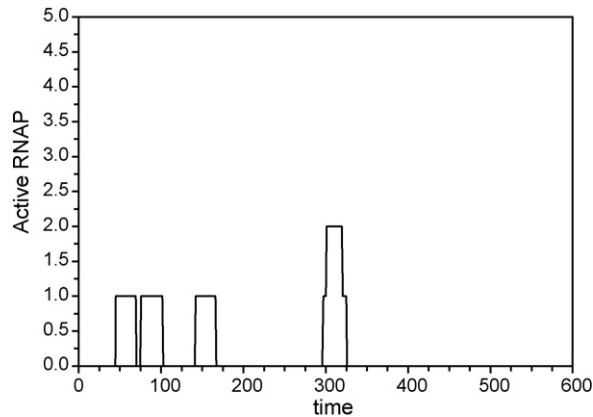


Fig. 7. Number of polymerases transcribing the Lac Operon in presence of glucose.

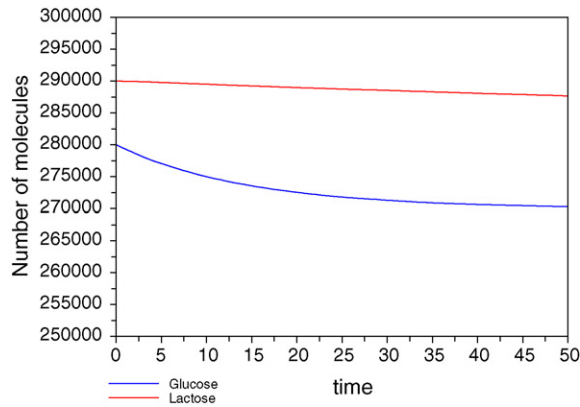


Fig. 8. Glucose and lactose molecules in the environment.

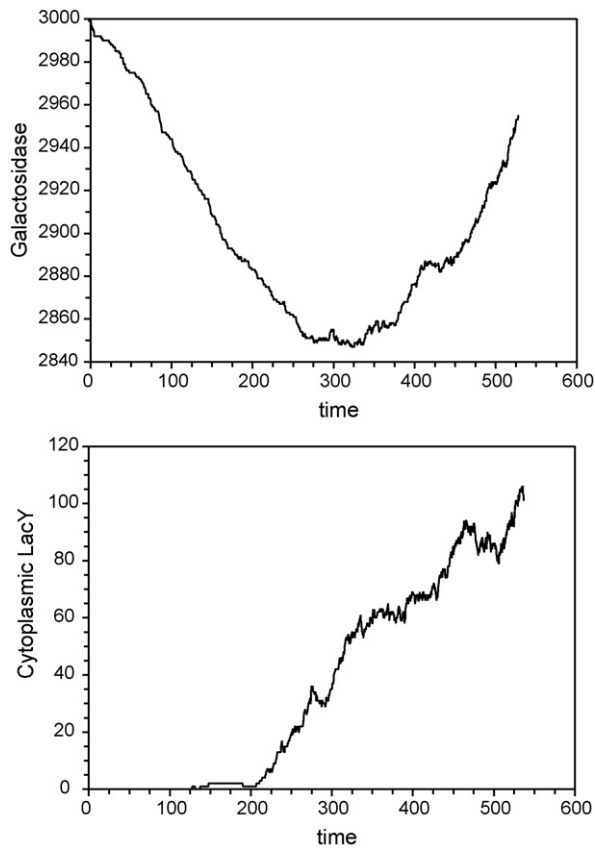


Fig. 9. Number of  $\beta$ -galactosidase and permease LacY over time.

## 7. Conclusions

In this paper we have used P systems as a formal framework for the specification and simulation of biological systems involving protein–protein and protein–DNA interactions. Our approach takes into account the key role played by membranes in the structure and functioning of the cells and the discrete and concurrent character of processes in bio-systems.

The models specified using P systems can evolve using different strategies/algorithms. In this work we have used the *Multi-compartmental Gillespie's Algorithm*. This strategy requires much computational resources and so the authors will study the possibility to adapt improved version of Gillespie's algorithm, like Gillespie (2001, 2003), in order to develop more efficient strategies for the evolution of P systems.

A model of the gene expression control in the Lac Operon in *E. coli* has been also presented as a case study. We have studied the behaviour of the system for different environmental conditions to see how the system is able to sense the presence of different substrates (glucose and lactose) using the cell surface and react according to them by synthesising in the cytoplasm the enzymes necessary to consume them. Note that we have modelled explicitly transcription and translation as rewriting and concurrent processes on strings, see string on page 21. The delay between the sensing of the signal and the expression of different genes is not modelled explicit but emerge as a consequence of the formulation of our approach.

Our results agree well with experimental observations and results obtained using other approaches. This shows the reliability of P systems as computational modelling tools to produce *postdiction* and perhaps as the field evolves they will be able to produce plausible predictions.

## Acknowledgements

This work is supported by Ministerio de Ciencia y Tecnología of Spain, by *Plan Nacional de I + D + I* (2005–2007) (TIN2005-09345-C04-01), cofinanced by FEDER funds, and by a FPU fellowship from the Ministerio de Ciencia y Tecnología of Spain.

## References

- Bernardini, F., Manca, V., 2002. P systems with boundary rules. In: Păun, Gh., Rozenberg, G., Salomaa, A., Zandron, C. (Eds.), *Membrane Computing, International Workshop*, vol. 2597 of *Lecture Notes in Computer Science*. Springer, Curtea de Arges, Romania, pp. 107–118.
- Bianco, L., Fontana, F., Manca, V., 2005. P systems and the modelling of biochemical oscillation. In: Freund, R., Păun, Gh., Rozenberg, G., Salomaa, A. (Eds.), *Membrane Computing, International Workshop*, vol. 3850 of *Lecture Notes in Computer Science*. Springer, Vienna, Austria, pp. 199–208.
- Cheruku, S., Păun, A., Romero Campero, F.J., Pérez Jiménez, M.J., Ibarra, O.H., 2007. Simulating FAS induced apoptosis by using P systems. *Prog. Nat. Sci.* 17 (4), 424–431.
- Ciobanu, G., Păun, Gh., Pérez-Jiménez, M.J. (Eds.), 2006. *Applications of Membrane Computing, Natural Computing Series*. Springer, Berlin.
- Gillespie, D.T., 1977. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.* 81 (25), 2340–2361.
- Gillespie, D.T., 2001. Approximate accelerated stochastic simulation of chemically reacting systems. *J. Chem. Phys.* 115 (4), 1716–1733.
- Gillespie, D.T., 2003. Improved leap-size selection for accelerated stochastic simulation. *J. Chem. Phys.* 119 (16), 8229–8234.
- Huber, R.E., Wallenfels, K., Kurz, G., 1975. Action of  $\beta$ -galactosidase on allolactose. *Can. J. Biochem.* 53, 1035–1038.
- Kennell, D., Riezman, H., 1977. Transcription and translation initiation frequencies of the *Escherichia coli* Lac Operon. *J. Mol. Biol.* 114, 1–21.
- Kierzek, A.M., Zaim, J., Zielenkiewicz, P., 2001. The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression. *J. Biol. Chem.* 276 (11), 8165–8172.
- Lolkema, J.S., Carrasco, N., Kaback, R., 1991. Kinetic analysis of lactose exchange in proteoliposomes reconstituted with purified lac permease. *Biochemistry* 30, 1284–1290.
- Meng, T.C., Somani, S., Dhar, P., 2004. Modelling and simulation of biological systems with stochasticity. *In Silico Biol.* 4, 0024.
- Păun, Gh., 2000. Computing with membranes. *J. Comput. Syst. Sci.* 61 (1), 108–143.
- Păun, Gh., 2002. *Membrane Computing. An Introduction*. Springer, Berlin.
- Pescini, D., Besozzi, D., Mauri, G., Zandron, C., 2006. Dynamical probabilistic P systems. *Int. J. Found. Comput. Sci.* 17 (1), 183–204.
- Pérez-Jiménez, M.J., Romero-Campero, F.J., 2006. P systems, a new computational modelling tool for systems biology. *Trans. Comput. Syst. Biol.* VI 4220, 176–197.
- Ptashne, M., 2004. *A Genetic Switch. Phage Lambda Revisited*, third ed. Cold Spring Harbor Laboratory Press, New York.
- Ptashne, M., Gann, A., 2002. *Genes and Signals*. Cold Spring Harbor Laboratory Press, New York.
- Romero-Campero, F.J., Pérez-Jiménez, M.J. A model of the quorum sensing system in *Vibrio fischeri* using P systems. *Artif. Life J.*, in press.
- Rohwer, J.M., Meadow, N.D., Roseman, S., Westerhoff, H., 2000. Understanding glucose transport by the bacterial phosphoenolpyruvate: glucose phosphotransferase system on the basis of kinetic measurements in vitro. *J. Biol. Chem.* 275 (45), 34909–34921.
- Stundzia, A.B., Lumsden, C.J., 1996. Stochastic simulation of coupled reaction—diffusion processes. *J. Comput. Phys.* 127, 196–207.
- Wong, P., Gladney, S., Keasling, J.D., 1997. Mathematical model of the Lac Operon: inducer exclusion, catabolite repression, and diauxic growth on glucose and lactose. *Biotechnol. Prog.* 13, 132–143.