

Document downloaded from:

<http://hdl.handle.net/10251/124694>

This paper must be cited as:

Baquero, F.; Campos Frances, M.; Llorens, C.; Sempere Luna, JM. (2018). A Model of Antibiotic Resistance Evolution Dynamics Through P Systems with Active Membranes and Communication Rules. *Lecture Notes in Computer Science*. 11270:33-44.
https://doi.org/10.1007/978-3-030-00265-7_3



The final publication is available at

https://doi.org/10.1007/978-3-030-00265-7_3

Copyright Springer-Verlag

Additional Information

A model of antibiotic resistance evolution dynamics through P systems with active membranes and communication rules

Fernando Baquero¹, Marcelino Campos^{1,2}, Carlos Llorens³, and José M. Sempere²

¹ Department of Microbiology
Ramon y Cajal University Hospital, IRYCIS
baquero@bitmailer.net

² Department of Information Systems and Computation
Universitat Politècnica de València
{mcampos, jsempere}@dsic.upv.es

³ Biotechvana
carlos.llorens@biotechvana.com

Abstract. In this work we describe a model of antibiotic resistance evolution dynamics based on a membrane computing approach. The model was implemented in a simulator tool first proposed in [3], with a naive set of rules and characteristics. In this paper, we describe the improvements over the first version of the model, we introduce new P system rules to manage all the elements of the system, and we explain a scenario in order to illustrate the experiments that can be carried out in the proposed framework.

Keywords: Membrane Computing, Systems Biology, Antibiotic Resistance, Bacteria Population Dynamics.

1 Introduction

P systems were proposed by Gh. Păun in 1998 [7] and gave birth to the membrane computing research area [9]. Basically, P systems are computational models inspired by eukaryotic cells and the exchange of biomolecules and information that is carried out through the physical membranes that structure the organelles of the cell. The computational ingredients of P systems are: (1) the arrangement of different regions organized in a tree-like structure, (2) a set of rules that guide the operation of the model during any computation, and (3) a set of objects that evolve and move throughout the structure of the system. The model can be defined non-deterministic and working in a maximally parallel manner. Hence, the interest in using it for the resolution of highly complex problems and with a stochastic component. In fact, membrane computing has proven to be a useful and versatile tool in systems biology, giving rise to different models in different areas such as ecology [5], bioprocesses [2] or the mechanisms for genetic regulation [1], to name a few areas of interest. A compendium of some works on systems

biology addressed by membrane computing is [8], while other bio-applications of membrane computing have been shown in [4]. In this work, we propose a modeling of the population dynamics of bacteria referred to the evolution of antibiotic resistance. Antibiotic resistance is a priority problem in Public Health. As it has been described in [12]: *”Antimicrobial resistance (AMR) with a wide range of infectious agents is a growing public health threat of broad concern to countries and multiple sectors. Increasingly, governments around the world are beginning to pay attention to a problem so serious that it threatens the achievements of modern medicine. A post-antibiotic era, in which common infections and minor injuries can kill, far from being an apocalyptic fantasy, is instead a very real possibility for the 21st century.”* That is why the study of possible scenarios where resistance to antibiotics behaves epidemically is of great interest. Our work is oriented to the efficient design of a computational simulator of such scenarios as it has already been tested in [3].

The structure of this work is as follows: In Section 2, we formally define P systems with active membranes and we explain its semantics. We define a cell-like P system with active membranes and communication rules to model the population dynamics related to the antibiotic resistance evolution in bacteria. In Section 3, we describe a scenario to test the proposed model, and in Section 4 we report some experimental results from the described model. Finally, we state some conclusions about this work.

2 Basic concepts

In this section we introduce basic concepts about P systems. We assume that the reader is familiar with the basic concepts of membrane computing. If this is not the case, we recommend the reading of [10] and [11]. In this work we use cell-like P systems with active membranes and without polarization that are defined as follows.

Definition 1. *A P system with active membranes of degree $m \geq 1$ is defined by the tuple $\Pi = (V, H, \mu, w_1, w_2, \dots, w_m, R, i_0)$ where*

1. V is the alphabet of objects
2. H is the alphabet of labels for membranes
3. μ is the initial membrane structure, of degree m , with all membranes labeled with elements of H and no polarizations. A membrane with label h is represented as $[]_h$
4. w_1, w_2, \dots, w_m are strings over V specifying the multiset of objects present in the compartment of μ
5. R is a finite set of rules of the following types
 - (a) $[v \rightarrow w]_h$ with $v, w \in V^*$ (evolution rules)
 - (b) $v[]_h \rightarrow [w]_h$ with $v, w \in V^*$ (‘in’ communication rules)
 - (c) $[v]_h \rightarrow w[]_h$ with $v, w \in V^*$ (‘out’ communication rules)
 - (d) $[v]_h \rightarrow [[w]_j]_h$ with $v, w \in V^*$ (membrane creation with object evolution)

6. $i_0 \in \{0, \dots, m\}$ indicates the region where the result of a computation is obtained (0 represents the environment).

The rules of the P system are applied in a non-deterministic maximally parallel manner. The computation of the system finishes whenever no rule can be applied. A configuration of the system at time t during a computation is defined by the membrane structure μ_t and the multisets of objects at every region in μ_t .

2.1 Modifications of the system used in the model definition

We are going to specify some of the components of the P system that have been previously defined, since to perform the simulation of the model we have introduced some modifications on the rules and the way in which the system works, that is, how the rules are applied in each step of computing. Fundamentally, we will specify the rules, the object alphabets and the working manner of the system.

Alphabets and membranes

In each region of the system there are two parameters that can regulate the application of the rules within the region: the *capacity* and the *occupation*. We consider that the alphabet V is partitioned into two alphabets V_c and V_{nc} so that $V = V_c \cup V_{nc}$, with $V_c \cap V_{nc} = \emptyset$. V_c denotes the set of objects that consume capacity of the membrane when they enter into the region delimited by it, while V_{nc} is the rest of objects. Each object from V_c consumes one unit of the capacity of the membrane. In addition, we will consider that the internal membranes of a region also consume only one unit, independently of its internal structure and the objects it may contain. In this way, if in the region delimited by the membrane h with a capacity γ there is an occupation index of δ objects and membranes then we will denote it as $[]_h^{\gamma-\delta}$. We will call *effective capacity* the difference between capacity and occupation. Observe that the effective capacity of a membrane is always greater than or equal to zero, and we will not allow negative values. This concept is dynamic since the occupation of a membrane can vary depending on the communication rules that are executed during the computation.

Another aspect in which we will make distinction is with respect to the objects that can be duplicated in the membrane duplication rules that we will explain later. Again, we will define a partition of the alphabet V into two sets V_d and V_{nd} so that $V = V_d \cup V_{nd}$, with $V_d \cap V_{nd} = \emptyset$. Here, V_d denotes the set of objects that are duplicated in a membrane duplication, while objects from V_{nd} are not replicated.

Rules

The rules at every region are ordered by priorities in the usual manner. In addition, we can associate to each rule a numerical parameter that identifies its suitability in terms of its possible application or not in a computation step.

Therefore, we can define the *suitability* function: $\mathcal{P} : R \rightarrow [0, 1]$ that approximates our model to a probabilistic/stochastic model in a way similar to PDP systems [6], although in our case we only use a cell-like P system instead of a multienvironment P system with active membranes. This function allows the application of the rules to only a subset of the objects. For example, if a rule $a \rightarrow b$ has a suitability index of 0.5 and there are two objects a in the region, then only one object a will be transformed into b according to the rule application (always taking this fact on average over the number of times we apply that rule in that configuration).

Next we will describe the semantics of the rules that we have defined in our model. Some of them differ slightly from the semantics that have been habitually assigned to them, while others are new rules that adapt to the needs of the reality that is intended to model:

1. evolution rules: $[u_i]_j^\alpha \rightarrow [u'_i]_j^{\alpha'}$
 In the membrane j the multiset u_i is substituted by the multiset u'_i . The effective capacity of the membrane is adjusted according to the multisets u_i and u'_i , and the elements from V_c and V_{nc} .
2. 'in' object communication rules: $[u_i]_j^\alpha]_k^\beta \rightarrow [[u'_i]_j^{\alpha'}]_k^{\beta'}$
 The effective capacity of the membrane j may be decreased and the effective capacity of membrane k may be increased according to the multisets u_i and u'_i , and the elements from V_c and V_{nc} .
3. 'out' object communication rules: $[[u_i]_j^\alpha]_k^\beta \rightarrow [u'_i]_j^{\alpha'}]_k^{\beta'}$
 The effective capacity of the membrane j may be increased and the effective capacity of membrane k may be decreased according to the multisets u_i and u'_i , and the elements from V_c and V_{nc} .
4. 'between' object communication rules: $[[u_i]_j^\alpha []_k^\beta]_p^\omega \rightarrow [[]_j^{\alpha'} [u'_i]_k^{\beta'}]_p^\omega$
 The effective capacity of the membrane j may be increased and the effective capacity of membrane k may be decreased according to the multisets u_i and u'_i , and the elements from V_c and V_{nc} . Observe that, at the membrane p , the effective capacity ω does not change since the membranes j and k remain inside the membrane p .
5. membrane dissolution rules (removing content): $[[u_i]_k^\alpha]_p^\beta \rightarrow []_p^{\beta'}$
 If the membrane k has all the objects of the multiset u_i , it is dissolved and the objects and membranes contained inside disappear. The effective capacity of p is increased due to the dissolution of k .
6. membrane dissolution rules (leaving the content): $[[u_i]_k^\alpha]_p^\beta \rightarrow [w]_p^{\beta'}$
 If the membrane k contains the objects denoted by u_i , it is dissolved and all the objects inside (denoted by w) remain in the membrane p . The effective capacity of the membrane p changes according to the multiset w , and the elements from V_c and V_{nc} , and the disappearance of the membrane k .

7. 'in' membrane communication rules: $[[]_i^\alpha []_j^\beta]_k^\omega \rightarrow [[[]_i^\alpha]_j^{\beta'}]_k^{\omega'}$
If the membranes i and j are in the region k , the membrane i is introduced into the membrane j . The effective capacities of membranes j and k are changed according to the new membrane structure.
8. 'out' membrane communication rules: $[[[]_i^\alpha]_j^\beta]_k^\omega \rightarrow [[]_i^\alpha []_j^{\beta'}]_k^{\omega'}$
If the membrane j contains the membrane i , the membrane i is pushed out of the membrane j . The effective capacities of membranes j and k are changed according to the new membrane structure.
9. 'between' membrane communication rules: $[[[]_i^\alpha]_j^\beta []_k^\gamma]_l^\omega \rightarrow [[]_j^{\beta'} [[]_i^\alpha]_k^{\omega'}]_l^\gamma$
If the membrane j and k are in the same region and the membrane j contains the membrane i , the membrane i is moved from the membrane j to the membrane k . The effective capacities of membranes j and k are changed according to the new membrane structure. Observe that the capacity of membrane l does not change.
10. membrane duplication rules: $[[]_i^\alpha]_j^\beta \rightarrow [[]_i^{\alpha'} []_i^{\alpha''}]_j^{\beta'}$
The membrane i duplicates itself. The effective capacity of membrane j is changed according to the new membrane structure. For the membrane i the process is the following: We start with one membrane i and we finish with two membranes i . All the membranes and the objects inside i belonging to V_d , are duplicated while the objects belonging to V_{nd} are distributed non-deterministically between the two copies of i .

Computation mode

We define a parameter ψ that denotes the computation mode of the system. We consider three different computation modes that are regulated according to the effective capacity of every membrane. Observe that the effective capacity allows the population regulation of the simulated system. In other works such as [5] the population regulation is carried out by the objects defined in the alphabets of the P system.

We have considered three different computation modes that we describe as follows:

- $\psi = 1$
All the rules are executed according to the priorities and the function \mathcal{P} provided that they can be executed and the effective capacity of the membranes are not exceeded.
- $\psi = 2$
All the rules are executed according to the priorities and the function \mathcal{P} . When a rule is applied if a membrane exceeds its effective capacity receiving objects or membranes, some elements generated by the rules are nondeterministically removed to preserve the membrane effective capacity to zero.

– $\psi = 3$

All the rules are executed according to the priorities and the function \mathcal{P} . At the end of a calculation step, the membranes with their exceeded effective capacity eliminate the objects and the membranes proportionally to their quantity until the effective capacity is set to zero.

3 Description of a scenario

With the objective of testing the proposed system, we are going to model a scenario where the rules, objects and structures are adapted to those described in the previous section. We model two populations: a hospital and a community. Each population contains hosts and each host contains bacteria. Guests can move from one population to another and bacteria can spread from one host to another. Different types of antibiotics are administered to the guests that can kill the bacteria. Bacteria can have some resistance genes that allow them to survive an antibiotic. An antibiotic removes some intestinal bacteria and other bacteria can occupy this place while the intestinal bacteria are recovering.

Before beginning to define the scenario more formally, we must take into account some details. First, the capacity of the membranes, especially in the hosts, is essential for the dynamics of the simulation; a host has a finite capacity to contain bacteria. On the other hand, bacteria have a very high level of growth but bacteria can not replicate if there is no space. Then, when we apply antibiotic to a host and eliminate bacteria, a new space appears and can be affected by other bacteria. And second, to make a meaningful simulation, we need many membranes: in a hospital there are many patients, in a community many people, and a host can have three billion bacteria. For a simulation, these numbers are intractable. To solve this problem we made two decisions: for the number of hosts the important is the proportion of people between the hospital and the community. We can take a single sample but we must maintain the proportion of a host in the hospital for one hundred people in the community. Another decision we have made is that we consider only one small sample of bacteria instead of the 3 billion that a host can have. The main reason for this decision is that each bacterium is represented by a membrane in the P system and the practical computability of the simulation could be compromised if all the bacteria are encoded. Taking a representative sample of the elements of the scenario seems an appropriate approach to obtain significant results.

We will detail the scenario taking into account the following aspects:

1. One step of computation is equivalent to one hour in the real time. One step simulates one hour because *E. coli* usually takes about an hour to divide into optimal conditions, being one of the fastest bacteria to do so. Therefore, one hour is a significant unit of time to work.
2. There are two populations, one hospital with one hundred hosts and one community with ten thousand hosts.
3. Every four steps (four hours) the hospital and the community exchange one host.

4. We work with the bacteria of the intestine. The five types of bacteria studied are EC (*Escherichia coli*), EF1 and EF2 (*Enterococcus faecium*), KP (*Klebsiella pneumoniae*) and PA (*Pseudomonas aeruginosa*). These bacteria represent the 1% of all the bacteria in the intestine.
5. In this scenario there are three types of antibiotics: A1 (*aminopenicillins*), A2 (*cefotaxime*) and A3 (*fluoroquinolones*).
6. An antibiotic treatment consists in one dose every 6 hours for a seven days. Each dose tries to kill the 30% of the bacteria in the first hour and a 15% in the second hour.
7. The 20% of the hosts in the hospital are under treatment, as well as the 1.3% in the community.
8. At the hospital, the 30% of the treatments are for antibiotic A1, 40% for antibiotic A2, and 30% for the A3 antibiotic. In the community, the 75% of the treatments are antibiotic A1, 5% of antibiotic A2 and 20% of antibiotic A3.
9. When a host is treated with antibiotic A1, the 25% of the bacteria in the intestine dies, with the antibiotic A2 dies 20% and for the antibiotic A3 the 10%. These bacteria take two months to recover their normal number. Meanwhile, this space can be occupied by EC, EF1, EF2, KP and PA.
10. A bacterium can have two different types of resistance, a static resistance (resistances in the genome) or a mobile resistance (resistances in plasmids or transposons).
11. A bacterium can only contain two mobile resistances.
12. The static resistance AR1 resists the antibiotic A1, AR2 resists the antibiotic A2 and AR3 resists the antibiotic A3.
13. The mobile resistance PAR1 resists A1 and the resistance PAR2 resists A1 only at the 10% if the resistance is in EC, KP, or PA and A2 if the resistance is in EC, KP or PA. The resistance PAR2 resists only A2 if it is in EF1 or EF2.
14. Each host starts with the bacteria configuration shown in table 1:

Table 1. Table of bacteria configuration

Bacteria	number	static resistance	mobile resistance
EC	5000	no resistance	no resistance
EC	2500	no resistance	PAR1
EC	1000	AR3	no resistance
EC	100	AR3	PAR1
EF1	995	AR2	no resistance
EF2	200	AR2 and AR3	PAR1
KP	200	AR1 and AR3	PAR2
PA	5	AR1	PAR2

15. A mobile resistance can move between EC and KP with a rate of 0.0001, between EF1 and EF2 with a rate of 0.0001, from PA to EC or KP at a rate of 0.000000001 and from EC or KP to PA at a rate of 0.000000001. Mobile resistance movements between bacteria of the same type can also be considered. For example, the movement rate from PA to PA is 0.0001.
16. There is a possibility by mutation that a bacterium without AR3 genetic resistance takes this resistance at a rate of 0.00000001.
17. When one host propagates to another, it passes the 0.1% of EC, EF1, EF2, KP, and PA. In one hour, the 5% of the hospital hosts extended other hosts and the 1% in the community.
18. The growth of each bacterium is different. Growth 1 represents that in one hour, if there is space, the bacteria double their number. EC has growth 1, EF1 and EF2 have 0.85, KP has 0.9 and PA has 0.15. A resistance in a bacterium means an additional cost for growth because every element must be replicated in mitosis. In this scenario we put a penalty of 0.03 (subtracted from the growth rate) for each AR3, PAR1 and PAR2.

4 Experiments and results

For experimentation, we write the scenario described above in a XML language created expressly for the simulator of this calculation model. We perform 50,000 steps because for the study of resistances in bacteria this number of steps is needed to show significant results. The simulator uses the computation mode $\psi = 3$ of the model. The initial membrane structure for the simulation is showed in the Figure 1.

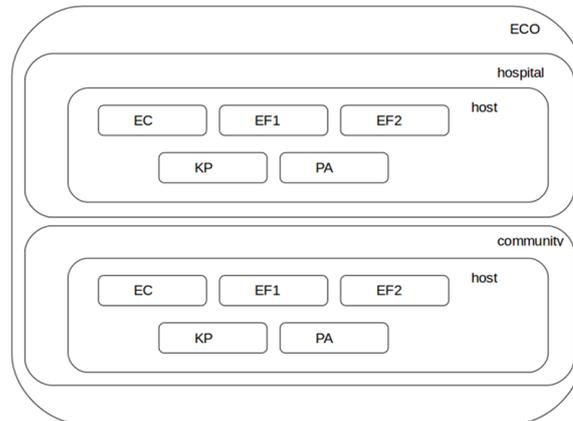


Fig. 1. Initial membrane structure for the described scenario.

The proposed P system described in Section 2 was created to simulate scenarios with bacteria. This means that in a computation there is a large number of membranes, and to be able to execute a simulation we need to work with membranes with multiplicity. When we use membranes with multiplicity and rules with suitability, the power of the multiplicity is lost because we separate the membranes to apply the rules. For example, if we have a membrane with multiplicity two and we apply a rule with 0.5 of suitability, the rule is only executed in one of the two membranes and the result produces two different membranes with multiplicity one. After some calculation steps, some membranes appear that represent exactly the same bacteria but are not represented with only one membrane with multiplicity. To solve this problem, at the end of each step, we check all the membranes that do not have inside other membranes (in this case the bacteria), and we fuse the equivalent membranes in only one membrane with multiplicity that is equal to the sum of the fused membranes.

From the simulation, we obtain two types of results: the first is the counting of each object and membrane in the hospital and the community (the elements in the contained membranes are also counted), and the second is the test, a test is defined by two sets of elements: what is *necessary* and what is *forbidden*. A membrane passes the test if it contains all the necessary elements and none of the elements is forbidden. So, we can count the number of membranes that pass the test (for example, how many EC bacteria have a resistance but do not have another). In this scenario, we focused the study on two points: the evolution of the bacteria in the hospital and the evolution of the resistance of the EC in the hospital.

In the Figure 2, we can see the evolution of the bacteria in the hospital. The x axis in the graph represents the calculation steps (one step simulates one hour), and the y axis represents the number of elements that we analyze. The bacteria in smaller number are PA, these bacteria have resistance to A1 and A2, but they have a very low fitness in front of the fitness of the other bacteria. The PA bacteria are outside the graph due to its low number with respect to the other bacteria. The next two bacteria are EF1 and EF2. Although EF1 and EF2 have the same physical state, EF2 appear in greater numbers because, at the beginning, EF2 bacteria have resistance to all antibiotics and EF1 only to A2. The KP bacteria have resistance to all antibiotics but they appear in more quantity than EF1 because their fitness is a little higher than EF2. Finally, the EC have the highest growth among all the bacteria under study. It is because they have the best physical form and, at the beginning, some EC have resistance to A3 and A1.

We must say that some resistances are transferred throughout the experiment. The most important are EF1 that take the resistance PAR1 from EF2. In addition, EF1 still do not have resistance to A3 (they can take this resistance by mutation, but with very little probability). The other important resistance acquisition is the resistance of EC to A2. EC can take this resistance from KP and, with less probability, from PA. When the EC take resistance to A2 they take resistance to all the antibiotics.

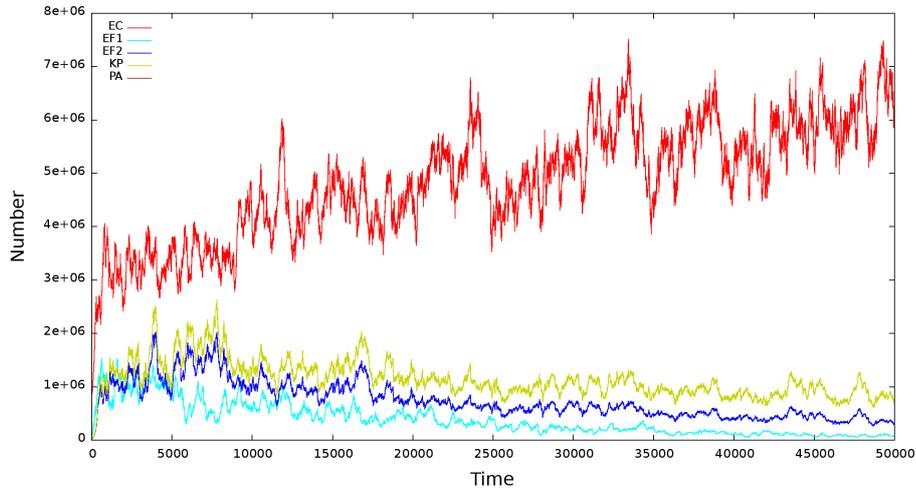


Fig. 2. Evolution of the bacteria in the hospital along about 70 month.

The Figure 3 shows the evolution of the resistance of the EC bacteria in the hospital. At the beginning, the ECs with only resistance to A1 are the more numerous. This is due to the initial configuration of the ECs, and because the resistance to A1 is the best one, given that A1 is the most used antibiotic (75% in the community). The ECs with resistance to A1 and A3 start with a number lower than ECs with resistance to A1 but they grow faster. The ECs with resistance to A1 and A3 exceed the number of the ECs with only resistance to A1, this is due to the resistance to more antibiotics. Finally, throughout the experiment, little by little we can see how ECs with resistance to all antibiotics grow and, eventually, they appear in greater numbers than the others. The ECs with resistance to all antibiotics are ECs with PAR1, PAR2 and AR3.

5 Conclusions

In this work we have presented a P system that allows the modeling of the evolution dynamics of antibiotic resistance. This is a public health problem of the first order and we believe that models such as the one presented in this work contribute to a greater understanding of this type of processes of an infectious nature. The first approach we made with this type of systems was published in [3]. However, subsequent refinements of the model and its adjustments led us to a substantial modification of the model with the aim of giving it a greater capacity to represent the real problem as well as greater computational benefits. The main modifications that we have been incorporating since our initial proposal has been exposed in this work.

The first conclusion we can reach is that P systems are a useful tool when dealing with some problems of Systems Biology, such as the one we have dis-

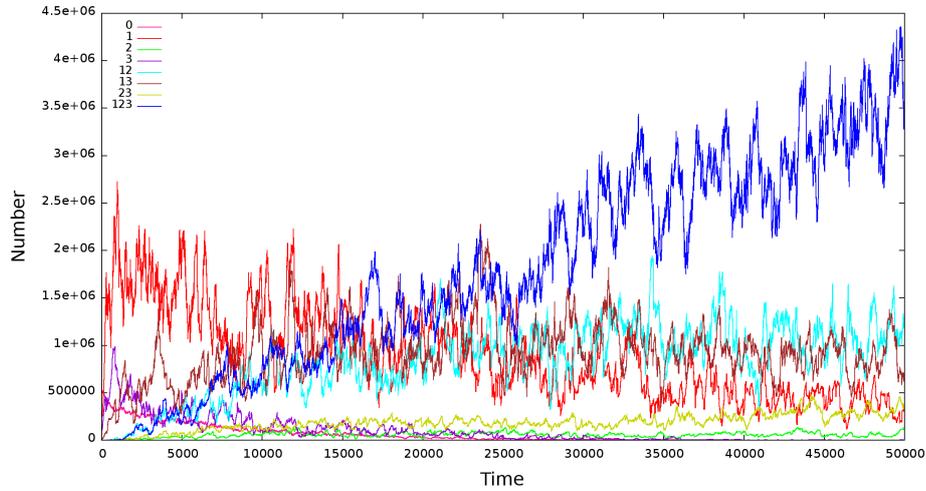


Fig. 3. Evolution of the EC bacteria depending of their resistances in the hospital along about 70 month. For this study we take into account the static resistance AR3 and the mobile resistances PAR1 and PAR2. 0 shows EC without AR3, PAR1 and PAR2, 1 shows EC with PAR1, 2 shows EC with PAR2, 3 shows EC with AR3, 12 shows EC with PAR1 and PAR2, 13 shows EC with PAR1 and AR3, 23 shows EC with PAR2 and AR3 and 123 shows EC with PAR1, PAR2 and AR3.

cussed in this paper. In addition, the formalization of the P systems allows a fairly intuitive understanding of what happens in the system, even for those scientists and professionals who do not have a great knowledge about computer models. In our case, the explanation of the gene transfer could be done in a very graphic way, assimilating some regions of the system to the carriers of the resistance and the populations in interaction as regions at the same level.

This work allows us to continue advancing towards more complex simulators that allow a greater understanding of reality.

Acknowledgements

Our work would not have been possible without the previous contributions of the community in P systems that allowed us to formulate this model as a valid tool and, in some cases, the best tool for modeling biological processes. In particular, all the works on ecological systems of the Natural Computing Group of the University of Seville, led by Prof. Mario de Jesús Pérez-Jiménez, have been our source of inspiration to address the design of our model.

José M. Sempere is indebted to Mario for his generosity and sincere friendship during all these years. This anniversary is a good opportunity to convey to Mario my gratitude for his human qualities, scientific rigor and ethical standards that

I have been fortunate to witness and share. My sincere congratulations and best wishes for all that we still have left to share.

References

1. N. Barbacari, A. Profir, C. Zelinschi. Gene regulatory network modeling by means of membrane computing. In *Proceedings of the 7th International Workshop on Membrane Computing WMC 2006*. LNCS, Vol. 4361, pp162-178. 2006.
2. D. Besozzi, P. Cazzaniga, S. Cocolo, G. Mauri, D. Pescini. Modeling diffusion in a signal transduction pathway: the use of virtual volumes in P systems. *International Journal of Foundations of Computer Science*, Vol. 22, No. 1, pp 89-96. 2011.
3. M. Campos, C. Llorens, J.M. Sempere, R. Futami, I. Rodríguez, P. Carrasco, R. Capilla, A. Latorre, T. Coque, A. Moya, F. Baquero. A membrane computing simulator of trans-hierarchical antibiotic resistance evolution dynamics in nested ecological compartments (ARES). *Biology Direct* 10(41). 2015.
4. G. Ciobanu, Gh. Păun, M.J. Pérez-Jiménez. *Applications of Membrane Computing*. Springer, Germany. 2006
5. M.A. Colomer, A. Margalida, D. Sanuy, M.J. Pérez-Jiménez. A bio-inspired model as a new tool for modeling ecosystems: the avian scavenger as a case study. *Ecological Modelling*, 222(1), pp 33-47. 2011.
6. M.A. Colomer, M.A. Martínez-del-Amor, I. Pérez-Hurtado, M.J. Pérez-Jiménez, A. Riscos-Núñez. A Uniform framework for modeling based on P systems, in *IEEE Fifth International Conference on Bio-inspired Computing: Theories and Applications (BIC-TA 2010)* vol. 1, ed. by K. Li, A.K. Nagar, R. Thamburaj, pp 616-621. 2010.
7. J. Dassow, Gh. Păun. On the Power of Membrane Computing. TUCS Technical Report No. 217. 1998
8. P. Frisco, M. Gheorghe, M. Pérez-Jiménez (eds.) *Applications of Membrane Computing in Systems and Synthetic Biology*. Springer, Switzerland. 2014.
9. Gh. Păun. Computing with Membranes. *Journal of Computer and System Sciences* 61(1): pp. 108-143. 2000.
10. Gh. Păun. *Membrane Computing. An Introduction*, Springer, 2002.
11. Gh. Păun, G. Rozenberg, A. Salomaa (eds.). *The Oxford Handbook of Membrane Computing*. Oxford University Press, Oxford. 2010.
12. Antimicrobial Resistance. Global Report on Surveillance. World Health Organization. 2014.