

RESEARCH

Open Access



A membrane computing simulator of trans-hierarchical antibiotic resistance evolution dynamics in nested ecological compartments (ARES)

Marcelino Campos^{1,3}, Carlos Llorens^{2*}, José M. Sempere³, Ricardo Futami², Irene Rodríguez^{1,4,5}, Purificación Carrasco⁶, Rafael Capilla², Amparo Latorre^{5,6,7}, Teresa M. Coque^{1,4,5}, Andres Moya^{5,6,7} and Fernando Baquero^{1,4,5*}

Abstract

Background: Antibiotic resistance is a major biomedical problem upon which public health systems demand solutions to construe the dynamics and epidemiological risk of resistant bacteria in anthropogenically-altered environments. The implementation of computable models with reciprocity within and between levels of biological organization (i.e. essential nesting) is central for studying antibiotic resistances. Antibiotic resistance is not just the result of antibiotic-driven selection but more properly the consequence of a complex hierarchy of processes shaping the ecology and evolution of the distinct subcellular, cellular and supra-cellular vehicles involved in the dissemination of resistance genes. Such a complex background motivated us to explore the P-system standards of membrane computing an innovative natural computing formalism that abstracts the notion of movement across membranes to simulate antibiotic resistance evolution processes across nested levels of micro- and macro-environmental organization in a given ecosystem.

Results: In this article, we introduce ARES (Antibiotic Resistance Evolution Simulator) a software device that simulates P-system model scenarios with five types of nested computing membranes oriented to emulate a hierarchy of eco-biological compartments, i.e. a) peripheral ecosystem; b) local environment; c) reservoir of supplies; d) animal host; and e) host's associated bacterial organisms (microbiome). Computational objects emulating molecular entities such as plasmids, antibiotic resistance genes, antimicrobials, and/or other substances can be introduced into this framework and may interact and evolve together with the membranes, according to a set of pre-established rules and specifications. ARES has been implemented as an online server and offers additional tools for storage and model editing and downstream analysis.

(Continued on next page)

* Correspondence: carlos.llorens@biotechvana.com; baquero@bitmailer.net

²Biotechvana, Valencia, CEEI Building, Benjamin Franklin Av. 12, Valencia Technological Park, 46980 Paterna, Spain

¹Department of Microbiology, Ramón y Cajal University Hospital, IRYCIS, Carretera de Colmenar Viejo, km. 9,100, 28034 Madrid, Spain

Full list of author information is available at the end of the article



(Continued from previous page)

Conclusions: The stochastic nature of the P-system model implemented in ARES explicitly links within and between host dynamics into a simulation, with feedback reciprocity among the different units of selection influenced by antibiotic exposure at various ecological levels. ARES offers the possibility of modeling predictive multilevel scenarios of antibiotic resistance evolution that can be interrogated, edited and re-simulated if necessary, with different parameters, until a correct model description of the process in the real world is convincingly approached. ARES can be accessed at <http://gydb.org/ares>.

Reviewers: This article was reviewed by Eugene V. Koonin, and Eric Baptiste.

Keywords: Membrane computing, P-system, Antibiotic resistance, Essential nesting

Background

Antibiotic resistance (AR) is a serious biomedical problem upon which public health systems urge for new R&D strategies to prevent the emergence and dissemination of resistant bacteria in human inhabited environments [1–6]. The discovery and clinical use of antimicrobials during the past 20th century has changed the course of medicine and the human lifestyle by reducing mortality from bacterial infections. However, the use or misuse of antibiotics, medicines and drugs may lead bacteria to become tolerant to the action of antimicrobials, eventually making standard treatments ineffective and thus challenging key medical practices as intensive care medicine, transplantation, or the therapy of immuno-compromised patients [2, 5, 7]. AR is also a subject of particular interest to food-chain stakeholders who use antimicrobials as growth-promoting additives until recently or for preventing cross-infection in food production areas [7–12]. Under antibiotic exposure, farms have become significant reservoirs for antibiotic resistant microorganisms, which in turn can be transmitted from food-producing animals and plants to humans [7–13]. The matter is to understand the risks and the unintended consequences of the anthropogenic use and release of antimicrobial agents into the biosphere, which without appropriate measures might drive the world towards a post-antibiotic era where mortality due to untreatable and fatal infections may become customary, particularly in underdeveloped areas of the world [3–5, 12, 14]. Understanding AR evolution is however a complex issue, since it is not just the result of antibiotic-driven selection of mutant resistant bacterial clones as it is frequently considered, but more properly the consequence of a variety of trans-hierarchical interactions between all biological vehicles involved in the dissemination of the genetic information involving AR [15–18]. We are talking about the following biological entities: genetic platforms, transposons and/or plasmids (here called subcellular replicators), bacterial cells, genetic exchange communities (GEC, communities in which the interchange of genetic material occurs frequently), microbiota, host individuals and host communities. In other words, not only AR genes can be carriers of AR genetic information, but also all other biological units

in which AR genes can be successively located at different subcellular, cellular and supra-cellular nested levels of the ecosystem [19–21]. It is important to note the need of understanding the effect of the whole nested frames associated to AR to establish a comprehensive “parameter space” able to describe the multidimensional evolution of antibiotic resistance (Table 1, for more details see [22]). All the aforesaid carriers are units of selection that can be simultaneously and independently chosen at different environmental levels including “invironmental” or microbial ecosystems [23]. Therefore, we should expect a high complexity in between-hosts demographical dynamics, as the host colonized or infected by resistant bacteria usually

Table 1 Key-nested frames associated to AR: a complex parameter space

-
- a) Density of colonized and colonizable hosts with antibiotic resistant bacteria
 - b) Population sizes of bacteria per host during colonization and infection
 - c) Susceptibility to colonization of hosts, including age, gender, ethnicity, nutrition, illness-facilitated colonization
 - d) Frequency of between-hosts interactions i.e. ,human-to-human or animal-human interactions
 - e) Host natural and acquired immune response to colonizing organisms
 - f) Ecological parameters of colonizable areas, including interaction with local microbiota and frequency and type of antibiotic-resistant commensals
 - g) Migration and dispersal
 - h) Antibiotic and biocide exposure and overall density of antibiotic use, type of antibiotics and mode of action, dosage and duration of therapy, adherence to therapy, selective antibiotic concentrations, antibiotic combinations
 - i) Mode of transmission of resistant organisms from the environment to hosts
 - j) Transmission rates between hosts (antibiotic treated and not-treated, infected, and not-infected)
 - k) Time of contact between hosts
 - l) Hygiene, infection control, sanitation
 - m) Food, and drinking water contamination by resistant bacteria and host exposure
 - n) Environmental contamination by resistant organisms, including soil, sewage and water
-

belongs to a population of interacting individuals that in turn belong to a community of interacting bacterial species. Note, for instance, that lice and keds of pets and farm animals (*Melophagus ovinus*, *Linognathus vituli*, *Heterodoxus spiniger*) or pest-insects such as the house fly (*Musca domestica*) and the German cockroach *Blattella germanica*, which are known to play a significant role as reservoirs and vectors of opportunistic bacterial pathogens that are often resistant to antibiotics [24–26] as they move, freely and indiscriminately, from filth and animal waste to food, allowing resistant bacteria to explore new habitats in hospitals, community settings and food facilities [27–29]. This type of complex scenarios, where bacterial resistant populations and their genetic platforms containing AR genes move because they are selected, and evolve because of their promiscuous migration, establish the precondition to visualize AR not only as a matter of biological function, but also as information flow processing. For these reasons, AR evolution is without doubt one of the major biomedical challenges for researchers in epidemiology and systems biology.

Interestingly, from the epistemological interaction of system biology, computer science and mathematics, a variety of models have arisen in the last decades to connect performances at different scales. These models are known as nested or embedded models (for a review see [30]) and have been used to acceptably address specific questions involving within-host dynamics enclosed in a model of between-host epidemiological scenarios. Nested models are classified as “inessential” when the within-host dynamic influences between-host processes but not *vice versa*, or “essential” when there is a reciprocal feedback between levels of organization. In particular, AR modeling requires an essential nested model; any alteration of the carriers in any specific resistance trait, or in their mechanisms of variation and mobilization (mutation, recombination, transposition, horizontal gene transfer, migration) may influence the dynamics of other units of higher and lower hierarchy, having logical consequences on the frequency and dissemination of AR genes and, therefore, evolutionary and/or ecological consequences on bacterial population [31, 32]. Until not too long ago, the difficulty to model this type of scenarios with essential nesting was an important limitation to feasibly study AR evolution processes. However, exciting new opportunities have recently arisen from a natural computing formalism inspired on the structure and functioning of biological cells, called membrane computing [33–35]. Membrane computing conceives any biological system as a hierarchical construct where the flow of materials can be interpreted as computing processes. In particular, membrane computing offers a versatile framework known as P-system that consists of a hierarchical membrane structure of nested compartments where multisets of objects are located and can move

across membranes evolving according to a finite number of given rules. Membrane computing have been proved to be universal models of computation [34] and has been successfully used to model oscillatory systems [36], processes of signal transduction [37, 38], gene regulation control [39], quorum sensing [40], meta-populations [41] and ecosystems [42, 43] thus suggesting that anything that can be computed can be done so as a P-system. For more details on the different approaches reached under membrane computing, see [44], or refer to the official websites of the membrane computing community [45] and P-Lingua [46] the programming language used for the development of P-systems.

In this paper we introduce a new P-system model designed for computing at three levels of organization (subcellular, cellular, supra-cellular) through the software implementation of a simulator we call Antibiotic Resistance Evolution Simulator (ARES). The general aim of ARES is to facilitate predictive computational models on the potential trans-hierarchical response of AR to particular interventions in specific scenarios. The simulator’s project is a work in progress, requiring constant refinements derived from the experiences (“experiments”) of costumers. The first version here introduced, is a prototype that offers a predefined layout composed of five types of nested-membranes that conceptually emulate an ecosystem hierarchy of biological boundaries based on population environmental areas, reservoirs, host populations and bacterial lineages of opportunistic pathogens. Granted to the implementation a friendly-to-use front-end interface, the user is allowed to define a starting configuration of elements (subcellular vehicles, antimicrobials and other substances) inhabiting the aforesaid membranes, specifications and rules according to which both elements and membranes evolve through a number of iterations. ARES is hosted at the GyDB Project [47] a database for research of mobile genetic elements (relevant carriers in the study of AR), and has been launched as an online server accessible at <http://gydb.org/ares>.

Methods

P-system model for simulating ecosystems with nested ecological boundaries

AR is a process of multilevel selection of nested units where the distinct resistance-carriers (gene vehicles) influence each other for selection and introgressive crossing of resistance to antibiotics at different environmental levels (subcellular, cellular and supracellular) [18, 20, 48]. This is an eco-biological model that can be formally generalized according to the following tuple:

$$\Pi = (V, \mu, w_1, w_2, \dots, w_n, (R_1, \rho_1), \dots, (R_n, \rho_n))$$

where V is a working alphabet of objects; μ is a membrane structure consisting of n membranes labeled $1, 2, \dots, n$ represents a rooted tree; and w_1, \dots, w_n are strings over V that represent multisets of objects initially placed in the structure of n membranes, which, from that point on will be referred as ecological boundaries (EBs) through the rest of this article.

In the V alphabet, our model takes into account the following entities to be treated as objects:

- Bacterial resistance genes computed either as an independent unit or constituting a combination together with a particular subcellular replicator (for instance a plamid). Here, we use the set of symbols AR_i to describe genes encoding AR, where i denotes the object's identity. Should an AR gene is designated in its single form (AR-like) it will be considered as a genomic gene (i.e. present as a locus within the bacterial host genome) by the model, but if it is attached to a particular subcellular replicator then it will be considered to be part of the subcellular replicator (i.e. carried by the subcellular replicator).
- Subcellular replicators inside bacterial cells such as plasmids, integrative-conjugative elements (ICE), transposons, or any other genetic element with self-replication ability. For the sake of simplicity, in this first version we only consider plasmid-like objects, which our model computes with the set of symbols PL_i . As previously indicated above, the model permits simulation of plasmids carrying AR genes by introducing a regular expression that define complex objects as follows: let us to consider k different plasmids and j different AR genes, then a complex object for the combination of plasmids with genes belongs to the following expression (with λ being an empty string in the absence of a name): $(PL_i + \dots + PL_k)(AR_1 + \lambda) \dots (AR_j + \lambda)$. For example the strings "PL₁", "PL₁-AR₁", "PL₁-AR₂", and "PL₁-AR₁-AR₂" respectively correspond to objects representing four different forms of the same plasmid – "not carrying AR genes", "carrying gene AR₁", "carrying gene AR₂" and "carrying both AR genes". For computational sake, the current version of our P-system model permits only simulation of plasmids carrying up to two different AR genes.
- External chemicals and/or biomaterials released into the environment, including any kind of molecule inhibiting bacterial growth such as antibiotics and biocides. We use the set of symbols A_i to describe these objects.
- Management Clocks are objects labeled with G_i symbols used to periodically add objects to specific membranes according to ecosystem influences expected to be cyclic.

In the membrane structure μ the model implements five hierarchical EBs labeled as ECO , P_i , RS_i , H_i and B_i

- ECO is the skin EB representing the peripheral ecosystem or ultimate container for all populations and environments.
- P_i is the second EB type in the membrane framework level that designates simulation of particular environmental areas of the ecosystem for the spread of hosts and resistant bacteria and eventually other bacterial pathogens (inhabitable spaces, areas for food acquisition, and other eventual infrastructures).
- H_i is a type of EB on the third level of nesting within the membrane framework hierarchy. We use H -like EBs to define host individuals carrying microbiota (host-specific assemblies of bacterial microorganisms, defined by its microbiota composition).
- RS_i is another type of EB, also on the third framework level, that the model uses in order to abstract (when suitable) the computation of three different types of physical or conceptual reservoirs. Two of these are called Food and Water supplies and are used during the simulation as reservoir-EBs of these resources. It is worth to note that for computational sake food and water are not treated as objects but as quantitative internal resources of the P-system being necessary for hosts' live that must therefore be periodically generated by management clocks and consumed by the simulated hosts according to the rates stated by the users in the configuration of rules. Once the host population growth outpaces the availability of food and water the model activates an internal malthusian rule that randomly kills (eliminates from the simulation) a number of hosts equivalent to the population surplus. The third RS-like EB is called "Sewage" and it refers to any body of water conveying all water-carried waste (either natural or anthropogenic) being removed from a community. Sewage can also be used to simulate the stool remains (or fecal droppings) periodically released by H-like individuals to the environment. The three RS-like types of EBs are represented only once within each (P-like) environment and although they are providers of water and food they may also contain substances and microbial contamination released through the droppings of animal hosts and the conversion of dead animals into food. In other words, the use of reservoirs allows the user to simulate supplies of food and water but also recycling of microbiota released into the ecosystem by animal hosts (H-like membranes) during the final act of digestion or to

turn into food any organism that dies or is predated by other organisms.

- B_{ij}^{\pm} is the last EB level of the hierarchy contemplated in our model and it is used to simulate bacterial cells. Each cell has several attributes here defined as follows: the superscript “plus/minus” is used to indicate if the cell is gram-negative or gram positive; the subscript i is used to represent cell populations as lineages term here used to highlight that the user can design the simulation of a microbiome according to a common historical offspring of cells at any taxonomical level (a lineage can therefore refer to a domain, phylum, class, order, family, genus, species and clones, depending on the experiment); j is used to assign two or more cells to a particular or unique community of cells or GEC. For instance, within a microbiome, the user can define a subset of cells to belong to GEC_1 and another subset to belong to GEC_2 using the subscript differentiation.

The dynamics, specificity and behavior of the distinct membranes and objects during the P-system simulation are administrated by a finite set of rules (R) fixed to each membrane that can be ranked by order of priorities (ρ). Particularly, our P-system model considers rules for the processes of transition, interactions and social behavior, birth, death, inactivation, and evolution of the distinct objects and membranes. The model also considers rules called specifications for ecosystem resource limitations in space and time (as limits in space, or life expectancy). Following are some examples of rules described in the formal definition followed by the membrane computing community, and where for the sake of simplicity, we omit priorities and stochastic parameters.

Example 1: transition rules with movement to hierarchically adjacent regions:

“Substance A_i enters bacterial cells” or “Substance A_i leaves bacterial cells”.

$$[u_i []_j]_k \rightarrow [u'_i []_j]_k \text{ (inside)}$$

$$[[u_i]_j]_k \rightarrow [u'_i []_j]_k \text{ (outside)}$$

Example 2: transition rule with movement to twin adjacent regions:

“A bacterial cell transmits a plasmid to other members of the same GEC”.

$$[u_i]_j []_i \rightarrow []_j [u'_i]_i$$

Example 3: active membrane rules with movement to twin adjacent regions:

“An individual host transmits bacteria to another individual”.

$$[[]_k]_j []_j \rightarrow []_j [[]_k]_i$$

Here, region k is moved from region j to region i , that is at the same level. Then, regions inside region k can be moved according to their corresponding rules.

Example 4: active membrane rules for membrane division:

“Growth of bacterial cells j in region i .”

$$[[]_j]_i \rightarrow [[]_j []_j]_i$$

Here, the content of region j is copied together with its rules and all the membranes it contains in a hierarchical manner.

In summary, our P-system consists of a membrane structure composed of five types of EBs and a working alphabet V of objects whose interactive feedback is determined by the set of rules assigned at every EB. This framework can be graphically represented as a Venn diagram (Fig. 1). As shown in the figure, the container diagram (ECO) represents the skin EB. Within ECO, the two next diagrams designated as $-P_i$ and P_j represent two P-like environmental EBs (the user can however design as many P-like EBs as required). P-like EBs are allowed to contain RS-like, H-like and B-like EBs (represented as diagrams of smaller size). RS-like EBs (designated as i, j, k) represent food, water and sewage reservoirs and are allowed to contain B-like membranes (but not H-like membranes). H-like EBs can be distinguished in subtypes (social classes, species, etc.) using subscript assignments. For example, in the figure we contemplate 3 populations (i, j, k) that may be respectively composed of a number of individuals (for instance 100, 50 and 150, etc.). Each H-like EB is allowed to contain a number of internal B-like EBs (but not RS-like EBs) defining its intrinsic microbiota. B-like EBs can be placed not only within RS-like and H-like EBs but also in P-like EBs and can be differentiated in lineages to which gram and GEC status can be assigned using sub- and superscripts. The status of Gram positive or Gram-negative organisms is assigned using a superscript with two states (minus and plus). In the figure we observe four subtypes (i, j, k, l) according to the left subscript. Those labeled with the left subscript j belong the GEC- j and those labeled with the subscript k belong to GEC- k . Those having the superscript plus are considered to be gram-positive cells, and those assigned the superscript minus are gram-negative. Logically, population size can also be assigned to each lineage (for example 10^9 cells per bacterial lineage). The working alphabet is composed of four types of objects (also differentiated in subtypes) summarized below the Venn diagram. In particular the

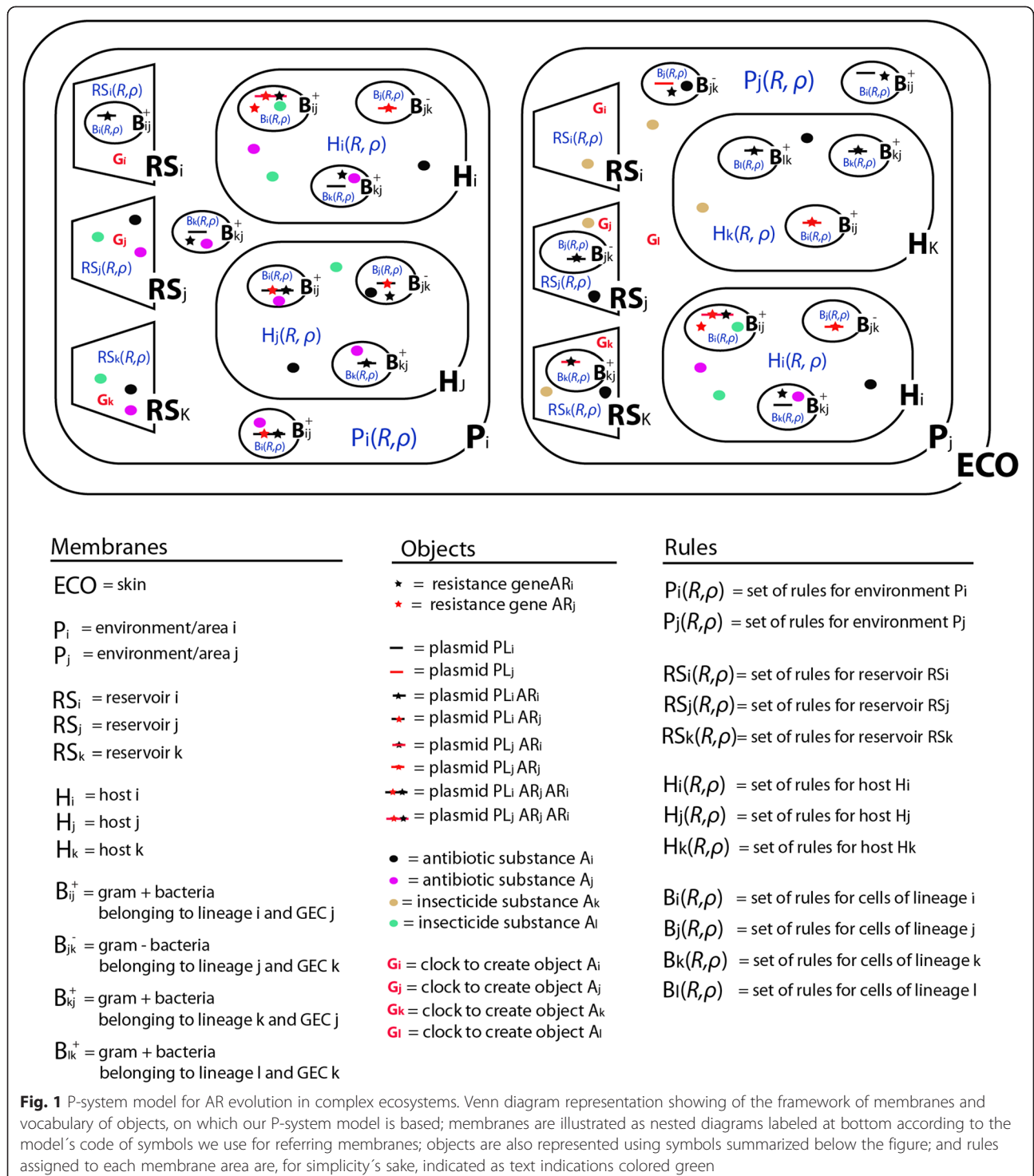


Fig. 1 P-system model for AR evolution in complex ecosystems. Venn diagram representation showing of the framework of membranes and vocabulary of objects, on which our P-system model is based; membranes are illustrated as nested diagrams labeled at bottom according to the model's code of symbols we use for referring membranes; objects are also represented using symbols summarized below the figure; and rules assigned to each membrane area are, for simplicity's sake, indicated as text indications colored green

figure shows two AR-like objects (*i, j*) defining two different AR genes; four A-like objects (*i, j, k, l*) defining four distinct substances (for instance two antibiotics and two insecticides with different properties); eight PL-like objects representing two plasmids (*i, j*) each one with four possible states (without AR genes, carrying an AR_i

gene, an AR_j gene or carrying both AR genes); and four G-like objects (*i, j, k, l*) representing management clocks. AR-like and PL-like objects are restricted to B-like EBs but they can move from a B-like EB into another (to emulate horizontal transfer events). Note, however, that AR objects can be only transferred when carried by a

plasmid object. A-like and G-like objects are allowed in all EBs (excepting ECO which is the ultimate container) as they either define substances expected to spread across all environments or periodical actions (in the case of G-like management clocks) stated by the user. Finally, every EB is assigned a set of specific rules (R) designed and tuned with the aim to govern the dynamic of interactions and evolutionary events within each EB according to given priorities (ρ), parameters and conditions indicated by the user.

Results and discussion

Introducing ARES: simulator device core and server implementation

The P-system model previously defined in Methods was implemented as a simulator software, which was programmed using the Java object-oriented computer programming language [49] and P-lingua foundations [50]. In particular, the simulator reads an xml file that contains the starting configuration of a P-system and run simulation of the P-system case study configured during a user-defined number of interactions contemplating four steps per iteration; 1) *step of evolution* where all rules for evolution and interaction are applied; 2) *step of movement* where all rules for movement apply; 3) *step of growth* where all membranes allowed to divide do so. After each step, an updating pre-step prepares the model for the next one. The rules applying to each region act simultaneously, in a parallel way, to all the objects and regions, and the stochastic behavior of the system is achieved by applying the rules according to their stochastic parameters in a naive probabilistic manner. Rules apply if: a) the objects in need of the rule application are in the region and b) there is no rule with higher priority that uses common objects. The transition from one configuration to the next is carried out by applying all the rules at every region in a non-deterministic maximal parallel mode; the system is always running from one configuration to the next and only halts if no rule can be applied. Hence, the halting configuration contains the output of the system and the final configuration. In addition, stochastic parameters to model the population dynamics of the system are also introduced in the rules.

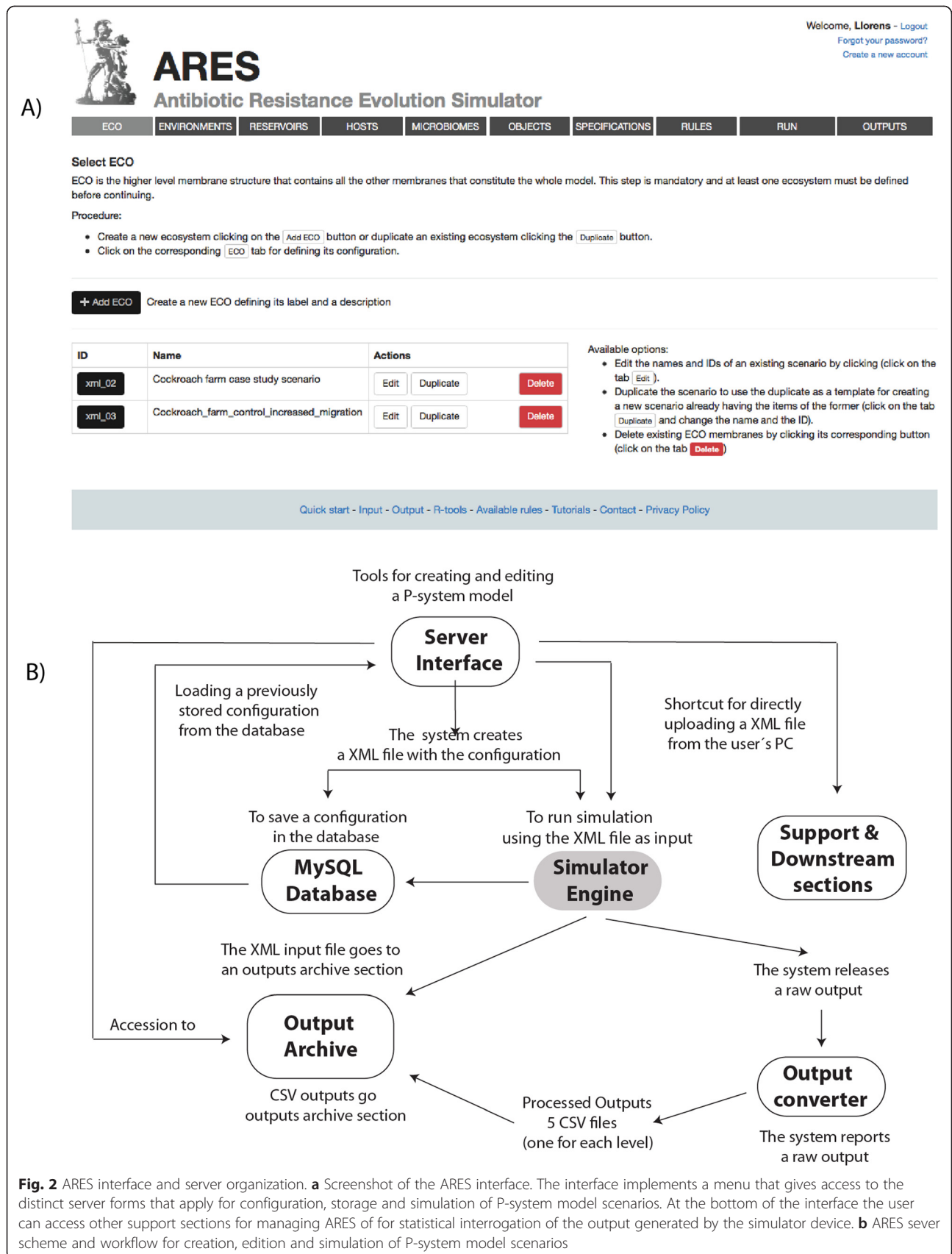
The simulator has been installed within an engine core within a 4× 6 Core Server with Linux OS and 128 GB of RAM, and has been coupled with the following sub-systems; 1) a MySQL Management System for storage of P-system configurations; 2) an engine for output conversion to CSV format; 3) the output's archive, which is a repository of output folders; 4) a server section to upload training tutorials; 5) a collection of scripts for statistical analysis developed using the R programming language [51]; 6) A front-end interface layout to manage all other sub-systems, programmed in a PHP framework on Laravel 4 following the Model-View-Controller pattern

an architectural model where server interfaces are interchangeable [52]. This Infrastructure is what we call ARES.

Managing ARES

Users can simulate dynamics of AR evolution using ARES to design a P-system model scenario adapted to the case study specified by the user and then run simulation of this scenario as many times as necessary correcting parameters until a realistic description of the AR process is approximated or validated if real world observations are available. Use of ARES is free, but its accession is password protected in order to allow the users to open and maintain a user account that will be needed to store and run model projects in private session (a simulation may last hours or even days depending on the complexity of a P-system scenario). ARES is managed via an easy-to-use interface that implements a centralized menu (Fig. 2a) for accessing the system of forms the user needs to sequentially complete in order to introduce the starting configuration of a P-system scenario, run a simulation, and access the results. All menu-forms accessible with this menu can be navigated back and forth for editing the P-system configuration, change or add EBs, objects and rules where or when necessary. A scheme of the whole ARES infrastructure and the workflow for configuration and simulation of P-system scenarios is depicted on Fig. 2b. The usual procedure can be synthesized in the following steps. The form designated as "ECO" must be first accessed (via menu) and completed to create the P basal skin EB; then "ENVIRONMENTS" has to be accessed to configure as many P-like (environmental) EBs as needed within ECO; next, "RESERVOIRS" and "HOSTS" must be filled to configure RS-like (reservoirs) and H-like (hosts) EBs within the previously created P EBs; after this, "MICROBIOMES" must be used to configure a series of B-like (bacterial) EBs that can be either placed within the previously created P-, RS- and H-like EBs; then "OBJECTS" must be used to create as many as PL- (plasmids), AR- (AR genes), A-like (antibiotics and/or other substances), and G-like (clocks) objects as required within the previously created EBs (except for ECO, since it is the skin membrane); finally, the forms "SPECIFICATIONS" and "RULES" must be completed to state the rules assigned to each EB by selecting them from a list of pre-designed rules provided in an understandable and generalized way allowing the user to choose and tune rules with the values and the parameters needed to approximate the frequencies, behaviors, conditions and priorities that govern the dynamic of interactions among the different membranes and objects of the P-system model to be simulated.

Once the P-system starting configuration has been defined, it is automatically written to an xml file (which is the input into the simulator), which is stored in ARES.



The form “*RUN*” is an interface that provides accession to the simulator engine core accompanied by a list of all xml files ready for simulation. The user only needs to select an xml file from such a list, then determine the number of iterations the simulation will last (each iteration is set to be correspond to one day) and run the simulation. It is worth to note that xml files can be exported or imported from the user’s PC as convenience, using the “*RUN*” interface.

Once the simulation starts, ARES automatically creates a folder (labeled with the name given to the P-system during the configuration) in the outputs’ archive assigned to this particular simulation and then generates the xml input file that is placed within this output folder. When the simulation finishes the system first delivers a raw output, which is a plain file containing the sampling counts per iteration of all simulated objects and EBs. The raw output is difficult to manage because of the default format it encloses. To overcome this difficulty, ARES implements an output converter engine that processes and splits the raw output into a set of 5 formatted csv files appointed as *ECO-like.csv*, *P-like.csv*, *RS-like.csv*, *H-like.csv*, *B-like.csv*. These files constitute together the output that ARES delivers after simulation for the user into the output’s archive (the output’s archive can be accessed using the “*OUTPUTS*” tab in the ARES menu). Each csv contains the counts for each EB and object sampled at the P-system EB level referred in the file name. For instance, *ECO-like.csv* has the counts of all EB and objects sampled at the ecosystem level, *H-like.csv* has the counts of all objects and EBs sampled in the H-like EBs but not in those of higher levels (i.e. ECO and P-like); and *B-like.csv* file has the counts of all objects sampled in B-like EBs. In each csv, iterations correspond with the rows and are organized in ascending order (being the starting configurations the first count in the file) and membranes and objects correspond with the columns.

ARES also offers other sections for user support, which can be accessed via the submenu available at the footer of the ARES interface. 3 of these sections called “*R-TOOLS*”, “*TUTORIALS*”, and “*AVAILABLE RULES*” are of particular interest. The first section (*R-TOOLS*) gives access to an interface (also managed via menu) offering the users different scripts mainly but not exclusively developed in “R”, for downstream interrogation of csv outputs. Note however that the use of *R-TOOLS* is not a mandatory task as csvs are open plain files that can be processed using any other tool or statistical packages such as Excel, Gnuplot, Matlab and Mathematica, etc. The second section (“*TUTORIALS*”) is a repository where users can upload and download tutorials for P-system configuration and management of ARES (see also the section below, “Tutorials and training material”). Finally the third section (“*AVAILABLE RULES*”) is a

section where we summarize all pre-designed rules to date available for each type of EB in order to let the user to make preliminary evaluations of the rules to take for a particular simulation before creating the starting configuration of the P-system. This section also includes a form for users to make if necessary any specific suggestion for the implementation of new rules not yet available.

Tutorials and training material

Although management of ARES is quite intuitive, the design and preparation of the starting configuration of a P-system model scenario can be an arduous task for researchers not familiarized with membrane computing (configuration of items, assignation of rules, etc.). Taking this into primary consideration, we have prepared two tutorials aimed to give the reader some training material that can be downloaded from the “*TUTORIALS*” section of ARES under the labels “Nosocomial Scenario” and “Two cockroach farms” respectively. The first tutorial contemplates a simplified nosocomial scenario provided with the sole objective of allowing the user to take the first steps in learning how to configure an exemplary P-system creating two membrane environments (a community and a hospital environment), host EBs (for instance, patients), microbial communities within hosts, composed of distinct bacterial EBs and plasmids, carrying AR genes objects within the bacterial EBs. The tutorial also exemplifies how to create clocks to introduce other objects such as antibiotics in the simulation or how to configure and tune a basic package of rules. The second tutorial contemplates a more elaborated scenario that can be addressed after completing the first tutorial. This tutorial specifically focuses on the simulation of two populations of *B. germanica* (a model insect organism able to implant an intestinal microbiota similar to that of humans [53–56]) respectively emplaced in two separate cages with the possibility of migrating from one to the other. These two boxes conceptually represent environmental EBs for hospital and urban-community individuals, here designated as P_1 and P_2 . Cockroaches of both farms are hosts (H-like EBs) carrying the same intestinal microbiota, which according to Carrasco et al. [56] is predominantly composed of eight bacterial lineages (B-like EBs). Four of these cellular lineages will be simulated as Gram-negative, while the four other will be Gram-positive. All bacterial cells of all lineages are allowed to carry three distinct types of intracellular plasmids (PL_1 , PL_2 and PL_3) capable of horizontal transfer. One of these (PL_1) is carrier of an AR gene (AR_1) conferring resistance to a gram-negative specific antibiotic designated as “ A_1 ” while another plasmid type (PL_2) carries an AR gene (AR_2) offering resistance to a gram-positive specific antibiotic labeled as A_2 . In the starting configuration, the third plasmid type (PL_3) does not carry AR genes but during the course of the

simulation it is allowed to recruit any of two types of AR genes simulated. Each P-environment has food and water supplies (RS-like membranes) to simulate the feeding of H individuals. The overall aim of this tutorial is to compare the response of the microbial communities in two main scenarios (control and case study) simulated under three distinct interventions (increase of the rate of migration, increase of antibiotic dosage, and fumigation). In total, the tutorial considers eight P-system scenarios (designated from xml_1 to xml_8) and provides indications about how to prepare, configure and simulate the aforesaid scenarios during 600 iterations (equivalent to 600 days). Although the second tutorial is provided solely for demonstration purposes, it is worth noting that the ecological scenario of this tutorial is in advanced process of being implemented in the real world, such a way providing a powerful tool for “experimental epidemiology” of antibiotic resistance. Such an epidemiological model will be useful for the validation of ARES, confronting predictions with real outcomes when the changes are introduced in the computing model and in the experimental “two cockroach farms” setting.

Conclusions

ARES is a new membrane computing simulator we have launched with the aim to help researchers develop computational models oriented to help elucidation of hidden aspects of the epidemiological and ecological complex patterns of AR that cannot be easily traced in the real world, due to both practical and complexity reasons. The underlying computational model of ARES is a P-system that differs from other models (including other previously published P-systems) in that both the framework and set of rules permit the user to simulate stochastic dynamics at different environmental (subcellular, cellular and supracellular) levels of the simulated ecosystem. This ability is what allows the user to assess the reciprocal feedback between the different carriers involved in the dissemination of AR genes, edit the configuration of the model scenario and then re-run the simulation with changing the parameters until a correct description of the AR process is approximated according to real world observations. ARES is a project in continuous progress and, therefore, a prototype in which we are working for future implementations and improvement, including an experimental epidemiology model (the “two cockroach farms model”). The project is open to all other experts interested in contributing expertise and criticisms.

Reviewers' comments and response

Reviewer's report 1: Eugene Koonin

Reviewer's comment:

I must indicate that I am not an expert in membrane computing. That said, my impression is that ARES is a

highly promising, flexible platform for modeling the complex dynamics of antibiotic resistance evolution. The description of the model is quite logical and meticulous. Given the obvious, overarching importance of the study of antibiotic resistance, I expect that this tool will be in high demand in the research community.

Authors' response: *Thank you very much for your positive comments and feedback. We hope the device and the formalism to be of interest for researchers working in predictive models for AR evolution or in System Biology. It is important to also recognize all the previously work done by the membrane computing community whose know-how (cited in the “Background” of this article), has been an important starting reference for us in order to conceptualize and design the P-system model we introduce in this article.*

Reviewer's report 2: Erik Bapteste

Reviewer's comment:

This work is stimulating to read and remarkable by its ambition: simulating complex systems and following the evolutionary dynamics of a diversity of objects within these systems. This manuscript introduces a rather intuitive formalization for these two tasks, taking advantage of P-systems. However intuitive these approaches are, I felt that an additional illustration, typically a Venn diagram for a simulation considered worthy of interest by the authors, would greatly help most readers to go beyond some of the rather abstract (and potentially discouraging) formalism used in the main text, e.g. when P-systems are described by pseudo-equations. Such a Venn diagram would also probably help to immediately appreciate what hierarchically adjacent regions? or twin adjacent regions? are on a concrete example.

Authors' response: *Thank you very much for your positive evaluation and criticisms. This second version of the manuscript includes a Venn diagram (designated as Fig. 1) where the framework of membranes and objects contemplated by the model are represented (and discussed in text).*

Reviewer's comment:

When I tested ARES on line, the website was operational, and user friendly. The proposed implemented specifications seem sound and numerous enough to be of use to interested scientists, or to make them feel scared by the many decisions one must make to fully benefit from this approach. Typically, not being an expert on the evolution of such systems, it was difficult for me to appreciate what sets of parameters and what rules were realistic ones. While I realize that in principle comparing simulations results with the biological reality may help one to a posteriori decipher what parameters were indeed realistic, I am a bit skeptical that in practice

this strategy will necessarily work so well for several reasons:

- How can one measure the similarity between simulated results and the biological reality? It might be useful to implement such comparative measures in ARES.
- How can one know when the simulation results are significantly close to the reality to be approximated? Some statistics would be needed here.
- Consistently, how can one discriminate between multiple scenarios producing comparable results what scenario is more realistic, if any, especially with so many parameters? Is there a way to compare, say, the complexity of two models with equally likely outputs? I guess these might become tasks for the future, should ARES evolve in a way that helps its users to explore parameters ranges in a statistically meaningful framework.

Authors' response: *Conceptually speaking, membrane computing is easy and intuitive but it is true that designing a P-system and preparing the xml input file for running the P system with an appropriate starting configuration is a daunting task because of the reasons you indicate. This motivated us to design ARES as friendly as possible in order to let the users to deal with membrane computing without being an expert in membrane computing. However, it is also true that to appropriately manage ARES, the user must do an first effort in getting familiar with at least the basic principles of membrane computing and also make another effort in getting some training. These issues, motivated us to create several support sections in ARES (see also our response to the minor comments). One of these sections (that called R-TOOLS) consists of a collection of R-scripts managed via interface for statistical interrogation of the ARES outputs. Evidently, ARES, and of course the aforementioned R-TOOLS section, are both work in progress with obvious limitations but the project is scalable. In this first release we have designed the basal P-system model and have programmed the software implementation of this P-system. At present, we are preparing new implementations. The 3 questions you address are excellent examples of new improvements we take note in order to implement them as soon as possible. Meanwhile, it is worth noting that the output released by ARES is a plain file the user is free to process with any third-party statistical package and statistical model or test (ANOVA, MANOVA, etc.) for comparing/discriminating the outputs of one or more scenarios among them or with empirical observations, which in turn can help the user to tune the starting configuration and re-run a simulation as many time as needed until a particular configuration approximates the real world observation.*

Reviewer's minor comment 1:

- Some generic terms related to P-systems, such as membrane structure? or regions?, could be misleading in this particular context of medicine and microbial evolution, and may deserve to be adjusted.

Authors' response: *In this version we have used the term "Ecological Boundary" (EB) when referring to particular membranes of the P-system framework. Hope you will find this term more appropriate.*

Reviewer's minor comment 2:

- It is easy to get lost in the numerous options (specifications and such): maybe having examples of what are considered as realistic parameters in some known environment (i.e. having default values associated with particular environments), or the possibility to run a simple pre-implemented case study might help the user to perform meaningful analyses? Maybe such a pre-implemented P-system is just what the import environments? option already offers, but this option did not seem to work for me on line.

Authors' response: *We have created a section called "TUTORIALS" within ARES that permits users to upload and share new tutorials with other users and where we also provide 2 tutorials (one very simple and the other a bit more complex) with material and indications allowing the user to get some training before starting with his/ own P-systems. In addition we have also created a contact section for users support as well as another section for frequently asked questions (FAQs). Also, and considering your feedback, we have created another section called "AVAILABLE RULES" where all pre-designed rules to date implemented in ARES are listed with the aim to let the users pre-study the whole set of rules and then evaluate which rules are appropriate or not for their interests. This new section also includes a form for users to make us recommendations in regards of new rules not yet contemplated that we will also try to program as soon as possible. Finally, let us to make one clarification; at present, there are not pre-implemented P-systems in ARES but the possibility to re-use the complete (or partial) configuration of a P-system previously introduced and stored in ARES by the user. The option did not work for you because you do not have any P-system configuration in ARES previously stored. We have clarified this in FAQs and where correspond in the system of forms of ARES but we also take note in any case of this interesting suggestion - have a collection of P-system configuration modules (i.e. pieces) pre-implemented in ARES - for further improvements.*

Reviewer's minor comment 3:

- For GEC descriptions, please explain what distinct numerical values will mean (i.e. will the same number mean that bacteria belong to the same gene exchange community, or will the value? 0? mean that bacteria do not belong to any GEC? Or does it indicate that a particular set of bacteria can be split into 3 GEC, when 3 is the value chosen?)

Authors' response: *Done in text.*

Reviewer's minor comment 4:

- The password on ARES must contain numbers, but it does not say this right away.

Authors' response: *Amended.*

Reviewer's minor comment 5:

- It might be nice to also have a Venn diagram as the output to compare the overall picture before?, and after simulation?

Authors' response: *Originally we aimed to implement Venn diagrams in the way you suggest (at the beginning or at the end of a simulation) but we dismissed the idea because it is only viable when plotting small P-systems. Bear in mind that one expect to find distinct types and subtypes of EBs and objects in the starting (or final) configuration of a more or less regular P-system for AR evolution, as well as a variety of rules assigned to each EB subtype. Note that although the population size of some membranes and objects to plot can be defined in single units (no more than a ten), the population size of just one bacterial lineage could reach thousands or even millions of EBs. We are however, working in order to find a satisfactory graphical solution when representing P-system complex scenarios via ARES.*

Abbreviations

AR: Antibiotic resistance; ARES: Antibiotic Resistance Evolution Simulator; EB: Ecological boundary; GEC: Genetic community; ICE: Integrative-conjugative elements; MVC: Model-view-controller.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MC, CL, JMS, AL, TMC, AM and FB designed the study; MC, CL, JMS, TMC and FB designed the P-system model; MC and JMS designed and programmed the simulator core engine; RF, RC and CL designed the server infrastructure; MC, RF, PC, IR, CL and FB performed most of the experimental work; and CL, JMS and FB wrote the manuscript. All authors read and approved the manuscript.

Acknowledgements

This work was mainly funded in the HRYC laboratory by research grants of the European Commission (EvoTAR-282004), and the support of the members of the Consortium, in particular Rob Willems (Utrecht), Dan Andersson (Uppsala), and Marc Woolhouse (Edinburgh) is particularly acknowledged. HRYC was also funded by the Ministry of Economy and Competitiveness of Spain (FIS-PI12-01581, PI10-02588, CB06/02/0053) and the regional government of Madrid (PROMPT-S2010/BMD2414). This work has also been supported by grants BFU2012-39816-C02-01 (co-financed by FEDER funds and the Ministry of Economy and Competitiveness, Spain) to AL and Prometeo/2009/092 (Ministry of Education, Government of Valencia, Spain) and Explora Ciencia y Explora Tecnología/SAF2013-49788-EXP (Spanish Ministry of Economy and Competitiveness) to AM. IRF is recipient of a "Sara Borrell" postdoctoral fellowship (Ref. CD12/00492) from the Ministry of Economy and Competitiveness (Spain). We are also grateful to the Spanish Network for the Study of Plasmids and Extrachromosomal Elements (REDEEX) for encouraging and funding cooperation among Spanish microbiologists working on the biology of mobile genetic elements (Spanish Ministry of Science and Innovation, reference number BFU2011-14145-E).

Author details

¹Department of Microbiology, Ramón y Cajal University Hospital, IRYCIS, Carretera de Colmenar Viejo, km. 9,100, 28034 Madrid, Spain. ²Biotechvana, Valencia, CEEI Building, Benjamin Franklin Av. 12, Valencia Technological Park, 46980 Paterna, Spain. ³Department of Information Systems and Computation (DSIC), Polytechnic University of Valencia, Camino de Vera, 46022 Valencia, Spain. ⁴Antibiotic Resistance and Bacterial Virulence Unit (HRYC-CSIC), associated to the Superior Council of Scientific Investigations (CSIC), Madrid, Spain. ⁵Network Research Center for Epidemiology and Public Health (CIBER-ESP), Madrid, Spain. ⁶Cavanilles Institute of Biodiversity and Evolutionary Biology, University of Valencia, C/ Catedrático José Beltrán 2, 46980 Paterna, Valencia, Spain. ⁷Foundation for the Promotion of Health and Biomedical Research in the Valencian Community (FISABIO) - Public Health, Avenida de Cataluña 21, 46020 Valencia, Spain.

Received: 8 April 2015 Accepted: 31 July 2015

Published online: 05 August 2015

References

1. Baquero F, Coque TM, Canton R. Counteracting antibiotic resistance: breaking barriers among antibacterial strategies. *Expert Opin Ther Targets*. 2014;18:851–61.
2. Baquero F, Lanza VF, Canton R, Coque TM. Public health evolutionary biology of antimicrobial resistance: priorities for intervention. *Evol Appl*. 2014;8:223–39.
3. Baquero F, Coque TM, de la Cruz F. Ecology and evolution as targets: the need for novel eco-evo drugs and strategies to fight antibiotic resistance. *Antimicrob Agents Chemother*. 2011;55:3649–60.
4. Carlet J, Jarlier V, Harbarth S, Voss A, Goossens H, Pittet D, et al. Ready for a world without antibiotics? The perspectives antibiotic resistance call to action. *Antimicrob Resist Infect Control*. 2012;1:11.
5. Laxminarayan R, Duse A, Wattal C, Zaidi AK, Wertheim HF, Sumpradit N, et al. Antibiotic resistance—the need for global solutions. *Lancet Infect Dis*. 2013;13:1057–98.
6. G8-Science-Ministers-Statement. 2013. <https://www.gov.uk/government/news/g8-science-ministers-statement>.
7. Levy SB, Marshall B. Antibacterial resistance worldwide: causes, challenges and responses. *Nat Med*. 2004;10:S122–9.
8. Wellington EM, Boxall AB, Cross P, Feil EJ, Gaze WH, Hawkey PM, et al. The role of the natural environment in the emergence of antibiotic resistance in gram-negative bacteria. *Lancet Infect Dis*. 2013;13:155–65.
9. Marshall BM, Levy SB. Food animals and antimicrobials: impacts on human health. *Clin Microbiol Rev*. 2011;24:718–33.
10. Marshall BM, Ochieng DJ, Levy SB. Commensals: underappreciated reservoir of antibiotic resistance. *Microbe*. 2009;4:231–8.
11. Forsberg KJ, Reyes A, Wang B, Selleck EM, Sommer MO, Dantas G. The shared antibiotic resistome of soil bacteria and human pathogens. *Science*. 2012;337:1107–11.
12. Heuer H, Schmitt H, Smalla K. Antibiotic resistance gene spread due to manure application on agricultural fields. *Curr Opin Microbiol*. 2011;14:236–43.

13. Teillant A, Laxminarayan R. Economics of Antibiotic Use in U.S. Swine and Poultry Production. *Choices*. 2015;30:1. 1st Quarter 2015.
14. ANTIBIOTIC RESISTANCE THREATS in the United States. <http://www.cdc.gov/drugresistance/threat-report-2013/pdf/ar-threats-2013-508.pdf>.
15. Gillings MR. Evolutionary consequences of antibiotic use for the resistome, mobilome and microbial pangenome. *Front Microbiol*. 2013;4:4.
16. Davies J, Davies D. Origins and evolution of antibiotic resistance. *Microbiol Mol Biol Rev*. 2010;74:417–33.
17. Palmer AC, Kishony R. Understanding, predicting and manipulating the genotypic evolution of antibiotic resistance. *Nat Rev Genet*. 2013;14:243–8.
18. Baquero F, Tedim AP, Coque TM. Antibiotic resistance shaping multi-level population biology of bacteria. *Front Microbiol*. 2013;4:15.
19. Partridge SR. Analysis of antibiotic resistance regions in Gram-negative bacteria. *FEMS Microbiol Rev*. 2011;35:820–55.
20. Baquero F, Coque TM. Multilevel population genetics in antibiotic resistance. *FEMS Microbiol Rev*. 2011;35:705–6.
21. Martinez JL, Baquero F, Andersson DI. Predicting antibiotic resistance. *Nat Rev Microbiol*. 2007;5:958–65.
22. Martinez JL, Baquero F. Emergence and spread of antibiotic resistance: setting a parameter space. *Upsala Journal of Medical Sciences*. Upsala J Med Sci. 2014, Early Online: 1–10, doi:10.3109/03009734.2014.901444.
23. Baquero F, Nombela C. The microbiome as a human organ. *Clin Microbiol Infect*. 2012;18 Suppl 4:2–4.
24. Kumsa B, Socolovschi C, Parola P, Rolain JM, Raoult D. Molecular detection of *Acinetobacter* species in lice and keds of domestic animals in Oromia Regional State. *Ethiopia PLoS One*. 2012;7:e52377.
25. Ahmad A, Ghosh A, Schal C, Zurek L. Insects in confined swine operations carry a large antibiotic resistant and potentially virulent enterococcal community. *BMC Microbiol*. 2011;11:23.
26. Graczyk TK, Knight R, Gilman RH, Cranfield MR. The role of non-biting flies in the epidemiology of human infectious diseases. *Microbes Infect*. 2001;3:231–5.
27. Limoe M, Enayati AA, Khassi K, Salimi M, Ladonni H. Insecticide resistance and synergism of three field-collected strains of the German cockroach *Blattella germanica* (L.) (Dictyoptera: Blattellidae) from hospitals in Kermanshah, Iran. *Trop Biomed*. 2011;28:111–8.
28. Salehzadeha A, Tavacolib P, Mahjubc H. Bacterial, fungal and parasitic contamination of cockroaches in public hospitals of Hamadan, Iran. *J Vect Borne Dis*. 2007;44:105–10.
29. Akinjogunla OJ, Odeyemi AT, Udoinyang EP. Cockroaches (*periplaneta americana* and *blattella germanica*): reservoirs of multi drug resistant (MDR) bacteria in Uyo, Akwa Ibom State. *Scientific J Biol Sci*. 2012;1:19–30.
30. Mideo N, Alizon S, Day T. Linking within- and between-host dynamics in the evolutionary epidemiology of infectious diseases. *Trends Ecol Evol*. 2008;23:511–7.
31. Gillings MR, Stokes HW. Are humans increasing bacterial evolvability? *Trends EcolEvol*. 2012;27:346–52.
32. Baquero F. Environmental stress and evolvability in microbial systems. *Clin Microbiol Infect*. 2009;15 Suppl 1:5–10.
33. Paun G, Rozemberg G, Salomaa A. *The Oxford Handbook of Membrane Computing*. Oxford, London. Oxford University Press. 2010.
34. Paun G. *Membrane Computing. An Introduction*. Berlin, Heidelberg. Springer-Verlag GmbH. 2002.
35. Paun G. Computing with membranes. *J Comput Syst Sci*. 2000;61:108–43.
36. Fontana F, Biancom L, Manca V. P systems and the modeling of biochemical oscillations. *Lect Notes Comput Sci*. 2006;3850:199–208.
37. Cheruku S, Paun A, Romero-Campero FJ, Perez-Jimenez MJ, Ibarra OH. Simulating FAS-induced apoptosis by using P systems. *Prog Nat Sci*. 2007;4:424–31.
38. Perez-Jimenez MJ, Romero-Campero FJ. P systems, a new computational modelling tool for systems biology. *Transactions on computational systems*. *Lect N Bioinform*. 2006;Biology VI:176–97.
39. Romero-Campero FJ, Perez-Jimenez MJ. Modelling gene expression control using P systems: The Lac Operon, a case study. *Biosystems*. 2008;91:438–57.
40. Romero-Campero FJ, Perez-Jimenez MJ. A model of the quorum sensing system in *Vibrio fischeri* using P systems. *Artif Life*. 2008;14:95–109.
41. Besozzi D, Cazzaniga P, Pescini D, Mauri G. Modelling metapopulations with stochastic membrane systems. *Biosystems*. 2008;91:499–514.
42. Cardona M, Colomer MA, Perez-Jimenez MJ, Sanuy D, Margalida A. Modelling ecosystems using P Systems: The Bearded Vulture, a case of study. *Lect Notes Comput Sci*. 2009;5391:137–56.
43. Cardona M, Colomer MA, Margalida A, Perez-Hurtado I, Perez-Jimenez MJ, Sanuy D. A P system based model of an ecosystem of some scavenger birds. *Lect Notes Comput Sci*. 2010;5957:182–95.
44. Frisco P, Gheorghe M, Perez-Jimenez M. *Applications of Membrane Computing in Systems and Synthetic biology*. Cham. Springer International Publishing. 2014.
45. Membrane Computing Community. <http://ppage.psysteams.eu>.
46. P-Lingua. http://www.p-lingua.org/wiki/index.php/Main_Page.
47. Llorens C, Futami R, Covelli L, Dominguez-Escriba L, Viu JM, Tamarit D, et al. The Gypsy Database (GyDB) of mobile genetic elements: release 2.0. *Nucleic Acids Res*. 2011;39:D70–4.
48. Baquero F. From pieces to patterns: evolutionary engineering in bacterial pathogens. *Nat Rev Microbiol*. 2004;2:510–8.
49. Java. <http://www.java.com>.
50. Garcia-Quismondo M, Gutierrez-Escudero R, Martinez-del-Amor MA, Orejuela-Pinedo E, Pérez-Hurtado I. P-Lingua 2.0: a software framework for cell-like P systems. *Int J Comput Commun*. 2009;1V:234.
51. R programming language. <http://www.r-project.org>.
52. Maciel A, Sankaranarayanan G, Halic T, Arikatla VS, Lu Z, De S. Surgical model-view-controller simulation software framework for local and collaborative applications. *Int J Comput Assist Radiol Surg*. 2011;6:457–71.
53. Dethlefsen L, McFall-Ngai M, Relman DA. An ecological and evolutionary perspective on human-microbe mutualism and disease. *Nature*. 2007;449:811–8.
54. Ley RE, Lozupone CA, Hamady M, Knight R, Gordon JL. Worlds within worlds: evolution of the vertebrate gut microbiota. *Nat Rev Microbiol*. 2008;6:776–88.
55. Pallen MJ, Wren BW. Bacterial pathogenomics. *Nature*. 2007;449:835–42.
56. Carrasco P, Perez-Cobas AE, Van de Pol C, Baixeras J, Moya A, Latorre A. Succession of the gut microbiota in the cockroach *Blattella germanica*. *Int Microbiol*. 2014;17:99–109.

Submit your next manuscript to BioMed Central and take full advantage of:

- Convenient online submission
- Thorough peer review
- No space constraints or color figure charges
- Immediate publication on acceptance
- Inclusion in PubMed, CAS, Scopus and Google Scholar
- Research which is freely available for redistribution

Submit your manuscript at
www.biomedcentral.com/submit

