Biochemical and Biophysical Research Communications 498 (2018) 342–351

Contents lists available at ScienceDirect

Biochemical and Biophysical Research Communications

journal homepage: www.elsevier.com/locate/ybbrc

Stochastic simulation of multiscale complex systems with PISKaS: A rule-based approach

Tomas Perez-Acle [a, b, *, 2], Ignacio Fuenzalida [a, 2], Alberto J.M. Martin [a, b, 1], Rodrigo Santibañez [a, c], Rodrigo Avaria [b], Alejandro Bernardin [a, b], Alvaro M. Bustos [a], Daniel Garrido [c], Jonathan Dushoff [d, e], James H. Liu [f]

[a] Computational Biology Lab, Fundación Ciencia & Vida, Santiago, Chile
[b] Centro Interdisciplinario de Neurociencia de Valparaiso, Universidad de Valparaíso, Valparaíso, Chile
[c] Department of Chemical and Bioprocess Engineering, School of Engineering, Pontificia Universidad Católica de Chile, Santiago, Chile
[d] Department of Biology, McMaster University, Hamilton, ON, Canada
[e] Institute of Infectious Disease Research, McMaster University, Hamilton, ON, Canada
[f] Massey University, Palmerston North, New Zealand

ARTICLE INFO

Article history:
Received 21 July 2017
Received in revised form
2 November 2017
Accepted 20 November 2017
Available online 21 November 2017

Keywords:
Agents
Rules
Gene regulation
Infectious disease
Prisoner's dilemma
Trust
Game theory

ABSTRACT

Computational simulation is a widely employed methodology to study the dynamic behavior of complex systems. Although common approaches are based either on ordinary differential equations or stochastic differential equations, these techniques make several assumptions which, when it comes to biological processes, could often lead to unrealistic models. Among others, model approaches based on differential equations entangle kinetics and causality, failing when complexity increases, separating knowledge from models, and assuming that the average behavior of the population encompasses any individual deviation. To overcome these limitations, simulations based on the Stochastic Simulation Algorithm (SSA) appear as a suitable approach to model complex biological systems. In this work, we review three different models executed in PISKaS: a rule-based framework to produce multiscale stochastic simulations of complex systems. These models span multiple time and spatial scales ranging from gene regulation up to Game Theory. In the first example, we describe a model of the core regulatory network of gene expression in *Escherichia coli* highlighting the continuous model improvement capacities of PISKaS. The second example describes a hypothetical outbreak of the Ebola virus occurring in a compartmentalized environment resembling cities and highways. Finally, in the last example, we illustrate a stochastic model for the prisoner's dilemma; a common approach from social sciences describing complex interactions involving trust within human populations. As whole, these models demonstrate the capabilities of PISKaS providing fertile scenarios where to explore the dynamics of complex systems.

1. Introduction

Complex Systems (CSs) encompass a variety of phenomena where the interaction between constituent elements produces emergent properties. Among other characteristics, CSs exhibit degeneracy being highly robust to random failures [1]. Such systems are ubiquitous in nature and society and, when it comes to their analysis, they are usually represented as networks. In these networks, edges represent the interactions occurring between the entities composing the system, which are typically depicted as nodes. Nevertheless, networks are static pictures of CSs: they disregard its dynamic behavior precluding the study of some of its fundamental properties such as evolvability [2]. Among other methods, two main approaches for the dynamic modeling of CSs prevail: deterministic methods based on ordinary differential equations (ODEs) or agent-based modeling, and stochastic approaches based either on stochastic differential equations (SDEs), or on the Stochastic Simulation Algorithm (SSA).

Approaches based on ODEs and SDEs require the definition of an

* Corresponding author. Computational Biology Lab, Fundación Ciencia & Vida, Santiago, Chile.
E-mail address: tomas@dlab.cl (T. Perez-Acle).
[1] Current Address: Centro de Genómica y Bioinformática, Facultad de Ciencias, Universidad Mayor, Santiago, Chile.
[2] Equally contributors.

https://doi.org/10.1016/j.bbrc.2017.11.138

equation for each different reaction or interaction in the model. Despite their broad applicability [3], ODEs models rely on several unrealistic assumptions. For instance, ODEs require a homogeneous distribution of components, and furthermore, a continuous distribution of interactions and quantities that, in real systems, are of discrete nature. On the other hand, SDEs assume that the stochasticity of the systems can be modeled as a source of noise acting as a modulator of the average dynamic of the population. In contrast, ODE models assume a deterministic behavior, focusing on the mean distribution of the system under study. On top of this, ODE- and SDE-based models usually entangle kinetics and causality. This characteristic excels when unrelated simultaneous processes produce an apparent causality. Last but not least, approaches based on differential equations require as many equations as variables, hindering continuous model improvement, and therefore separating domain knowledge from the models.

Although both ODEs and SDEs are well-established methods, SSA-based methods are gaining momentum among other approaches as both reliable and versatile strategies to model a variety of CSs [4]. An SSA model occurs in a reactor where a set of species or agents interact by a set of reactions or rules. While the reactor is a well-mixed and homogeneous environment containing the initial quantities of every agent, the application of the set of rules will produce the dynamic of the system. Within the reactor, stochastically selected rules are applied as discrete events generating trajectories over time to produce solutions representing feasible temporal paths of the system. Importantly, as time depends on the execution of rules, time intervals are asymmetric over the trajectory of the system and also between parallel reactors. Current variants of SSA use their own language to describe the systems under simulation. Some examples are the BioNetGen language [5] and the Kappa language [6]. Importantly, for each one of these languages, a proper simulation engine should be used: NFSim [7] executes models written in BioNetGen and KaSim [8], executes models written in Kappa. Several simulation engines based on the SSA have been developed. While some implementations account for large number of species [5,9–11], some others deal with systems formed by numerous particles [12,13], or with slow-scale systems [14,15], and some others were developed to add computational power to the simulation [16,17].

Despite the increasing relevance of SSA and the expanded features included in several of its implementations, these fail to deal properly with the combinatorial diversity and spatial heterogeneity of biological systems. To overcome these issues, we developed PISKaS, a parallel implementation of a spatial Kappa simulator [18–21]. PISKaS is a multiscale simulation tool suitable to perform stochastic simulations on distributed memory computing architectures. PISKaS expands the Kappa language by allowing the explicit declaration of compartments interconnected by links to simulate heterogeneous environments. In PISKaS, these links account for different types of transport between compartments. Each PISKaS compartment is executed in parallel, running its own SSA. The execution of a communication rule between compartments is treated as a perturbation modifying the number of agents in both the source and the destination compartment. Due to these new features, Kappa models are easier and more compact to write and to understand than those employed by other implementations. While the original motivation for both the SSA and the Kappa language comes from the modelling of chemical reactions, PISKaS can be used to model systems unrelated to chemistry, or where the species or agents involved do not represent chemical units but instead more complex entities (v.g. genes, individuals or communities).

In the following pages, three models of CSs will be used to demonstrate the versatility and capabilities of PISKaS to deal with multiscale models. In doing so, we present a partial model for the transcriptional regulation of *Escherichia coli*; a highly compartmentalized model to study the spread of Ebola virus disease; and a well-known model from Game Theory (GT): the Prisoner's dilemma (PD) which is aimed the at study of cooperation and competition between individuals in a society. While in the *E. coli* model we exploit PISKaS capabilities to deal with continuous model improvement, on the Ebola model we rely on a highly compartmentalized environment, and in the PD, we follow an approach where rules operate considering the value of certain agent properties. As a whole, we propose PISKaS as a suitable tool to produce stochastic simulations of multiscale CSs using a rule-based approach.

2. Methods

PISKaS was developed from a forked branch of the Kappa language simulator, KaSim v3.5 [8], adding new features to improve performance by implementing parallelism, and to allow spatial declaration of the model environment by using compartments and links. KaSim is written in Ocaml (http://www.ocaml.org), a programming language following both the functional and object-oriented paradigms. PISKaS employs OcamlMPI 1.01 (https://forge.ocamlcore.org/projects/ocamlmpi/), taking advantage of the MPI library to implement the parallel execution of compartments. To execute a simulation, PISKaS starts an iterative process determining the probability that the next event on the simulation occurs at time τ, selecting rule R_j, given the state of the system x in time t (Equation (1)):

$$p(T,\ R_j|x,t) = a_j(x)e^{-a_0(x)T} \tag{1}$$

where $\tau=t+\Delta t$, a_0 is the total reactivity of the system and a_j is the reactivity of rule j. Importantly, τ is calculated from a random number r_1 distributing uniformly between 0 and 1 according to Equation (2):

$$T = \frac{1}{a_0(x)} \ln\left(\frac{1}{r_1}\right) \tag{2}$$

By doing so, each temporal step over the simulation randomly selects a single rule that *creates*, *destroys*, *binds*, *unbinds* agents or *modifies* some property of an agent: the five primitives of the Kappa language [6]. Importantly, PISKaS proposes a new algorithm to deal with the heterogeneous spatiality of CSs. This modified version of the traditional SSA partitions the initial state of the system into several compartments. Each compartment is considered as a homogeneous volume following the SSA fundamental assumption; reactions operating over agents are considered as collisions between particles and, no collisions occur between agents belonging to different compartments. As a consequence, every compartment in the model executes its own SSA. Transport between compartments is produced by rules executing perturbations to modify the number of agents in both linked compartments: whereas a transport rule removes an agent from a compartment, it will then create that agent in the linked compartment. To produce the system dynamics, PISKaS implements a parallel algorithm which supports, at the cost of some accuracy that is tunable by the modeler, the scalability of the model by adding compartments, links, and transport rules. These features provide a more accurate physical description of the environment, allowing the execution of millions of agents in thousands of compartments, depending on hardware capabilities.

The performance of PISKaS implementation depends on several parameters of the model. The main parameter regulating performance is the synchronization step h, i.e. the elapsed time between

synchronization processes occurring in different compartments. A small value for *h* produces a slower but more accurate simulation compare to that of a larger *h*. Unfortunately, finding the optimum *h* value requires a trial-and-error approach because it is tightly linked to both the spatial complexity of the model and its rules. Nevertheless, our divide-and-conquer approach allows to create orthogonal reactors in the simulation where each SSA is run in parallel, using its own agents and rules.

To deal with the problem of finding a proper *h* parameter, at least 1000 simulations were run for every model varying the value of *h*, considering 10 time-intervals equally distributed, starting from the smallest time ticking between all linked compartments, up to the largest one. To do so, an initial simulation was run for every model, monitoring the behavior of time ticking in all different compartments. To compute statistics, trajectory ensembles were obtained by averaging over time and calculating the standard deviation. Importantly, as the SSA can be defined as a state algorithm dedicated to solving numerically a continuous first order Markov chain, there is no need to use a more complex process such as the block average, to compute the error over the average trajectory of the system.

All simulations where run in a distributed memory cluster composed by two computing nodes of 64 cores with 128 GB RAM interconnected by a Mellanox Infiniband switch. Each node has 4 AMD Opteron 6376 processors running at 2.3 Ghz.

The development of PISKaS is under the GNU license and can be downloaded from GitHub at https://github.com/DLab/PISKaS/wiki. All models used in this work can be downloaded from GitHub at https://github.com/DLab/PISKaS/wiki/Models.

3. Results: application cases

3.1. Simulation of transcriptional networks in E. coli

The regulation of gene expression occurring in living organisms is an epitome of biological CSs. Even though it can be argued that gene regulation is an interconnected process highly dependent on gene expression, whole-genome control is usually modeled by dividing it into more or less independent processes [22,23]. Partitioning has two main purposes; to facilitate knowledge generation and to simplify parameter optimization. To demonstrate PISKaS capabilities to support continuous improvement of models, two systems were studied: the core gene regulatory network of *E. coli* and the replication of the ColEI plasmid. Both models were subsequently combined into a more detailed system, taking advantage of the natural compartmentalization of biological systems (Fig. 1). By using the gene regulatory network (GRN) of *E. coli* MG1655 and a custom-made script written with the Python PySB library [24], all *Kappa* models were written automatically following the *Kappa* Biobrick Framework [25]. The GRN was created combining literature [26] and the EcoCyc database [27] in a format that permitted the creation of models where agents, rules, parameters and the initial condition of the system were predefined. Specific rules created and employed in all three models are described using graphical notation in Fig. S1.

3.1.1. Core regulatory network: Sigma factors

The core regulatory network in *E. coli* is controlled by seven proteins called Sigma Factors (SFs) [26]. A SF binds to the RNA Polymerase (RNAP) core enzyme giving specificity and causing competition for different transcription factor binding sites [28]. This model is composed by 10 agents representing genes, three of them encoding for each RNAP subunits and the remaining seven encoding for each SFs. The assembly of every possible complex RNAP-SF was modeled as a reversible process as reported previously [28,29]. Gene regulation conducted by the binding of each RNAP-SF complex to each of the promoters, was also modeled as a reversible process, whereas transcription and termination were irreversible events. Our model also accounted for the transcription units present in this system. Specifically, rpoC is the only unit transcribed right after rpoB, and both genes coding for RNAP subunits are regulated by the same RNAP-SF.

Three different settings were explored with 1000 simulations each: in the first setting, only one RNAP could be assembled, while seven were possible in the second and 28 in the third setting: one RNAP for each non-redundant regulation according to Fig. 1. Moreover, in both the first and second settings, only one copy of each SF was available, while in the third setting, there was a copy of every regulation present in the GRN. The setting with higher number of SFs resulted in a better stability of the number of RNAP-SF complexes, showing an average slightly lower than 20% (Fig. 1B). Notably, having a higher number of copies of the SFs, this setting resulted the closest one to previously reported experimental data [30] where the fraction of free RNAP-SF complexes was 30%, rendering a more realistic version of the model. The fraction of free RNAP-SF is regulated by additional layers of regulatory elements and the rate of interaction between components. Several antagonists regulate the *E. coli* SF activity. Of note is the rsd protein that binds rpoD and controls the transition to a stationary state [31]. Other Sigma antagonists are the FlgM that binds FliA and controls flagellar synthesis, and the rseAB protein that regulates rpoE activity, necessary for stress response. Eleven components have sigma factor antagonist activity and anti-sigma factor antagonist activity in *E. coli*. Therefore, the incorporation of the aforementioned elements in the core regulatory model could reduce the notorious distance between simulated and experimental values, prior to parameter adjustment.

3.1.2. ColEI replication model

The regulatory network controlling the replication of the ColEI plasmid is formed by two non-coding RNAs [32]. The first is a RNA which binds tightly to the DNA, making a hybrid DNA-RNA that after processed by RNase E serves as a primer for the DNA Polymerase [33]. To control its replication, this plasmid encodes an antisense RNA, called RNA I, which binds the 5′ end of the RNA primer, destabilizing the hybrid and effectively inhibiting DNA replication. To further explore PISKaS capabilities, our ColEI replication model simulates the formation of an RNA-RNA pairing to inhibit DNA replication (Fig. 1C). Even though arbitrary rates were employed, a pseudo-stationary state was reached with near 300 plasmid copies. On the other hand, without the RNA-RNA interaction, the plasmid reached approximately 48,000 copies over the simulation, which is far beyond the numbers obtained from the model with both non-coding RNAs.

3.1.3. Combined model

Given that both previous models share common components (SFs, RNAP, RNase E), a more realistic setting was developed by joining them, demonstrating the continuous improvement strategy supported by PISKaS. During the combined simulation, despite the explicit communication between compartments, each type of agent behaves in a very similar way to that observed in the separated simulations. However, the dynamic of the plasmid replication was heavily impacted, reaching only 20 copies of the plasmid instead of the 200 obtained in the separated model, with some simulations stalled after only one copy was produced.

3.2. A hypothetical outbreak of the Ebola virus

Other interesting models of CSs are those employed to the study

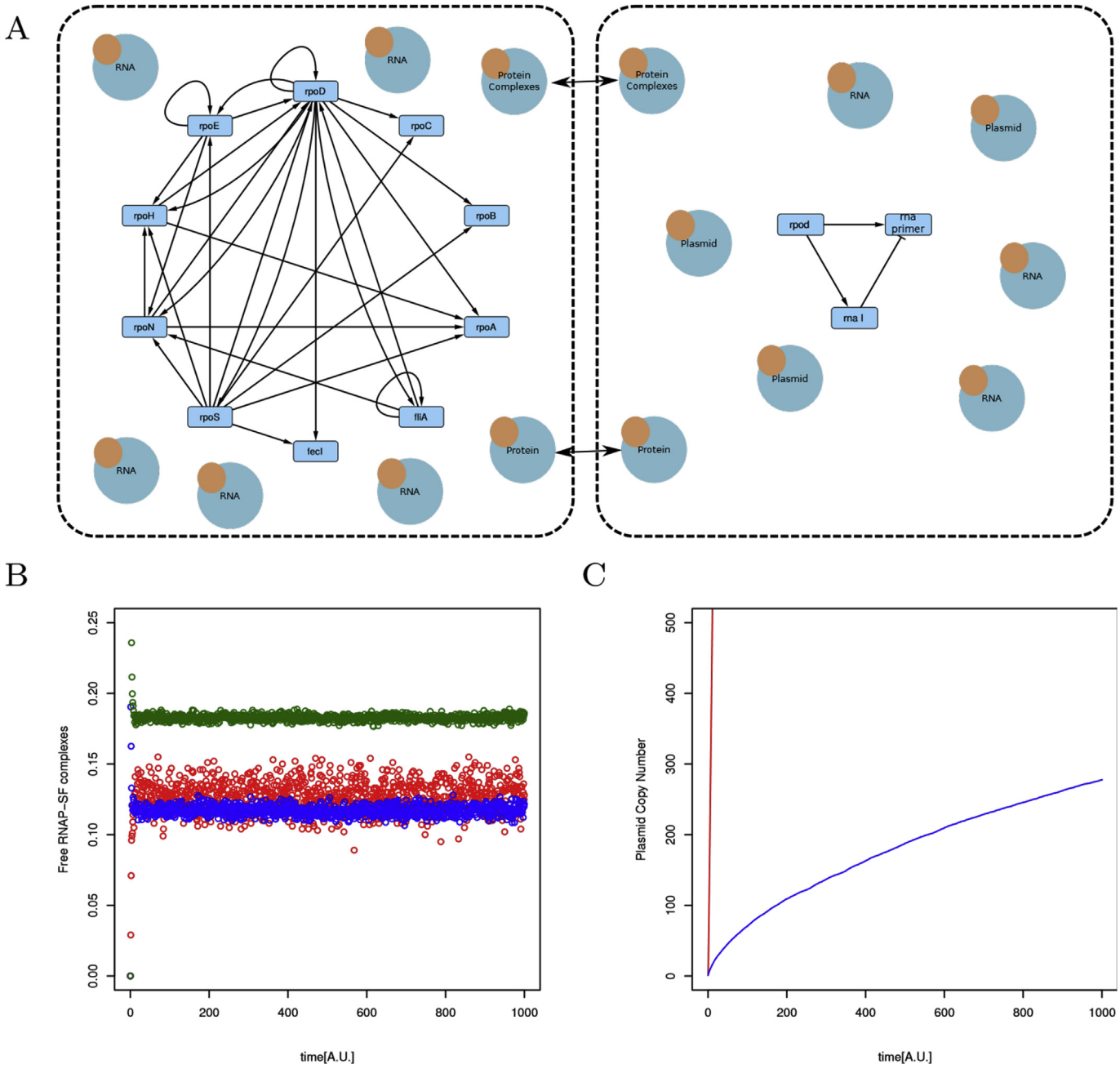

Fig. 1. The gene regulatory network simulated in both the core regulatory network and in the ColEI replication models. The regulation of transcription (upper-left) and plasmid replication (upper-right) were compartmentalized and allowed to share free proteins and protein complexes (arrows). Outcome of the simulation of transcriptional networks in *E. coli*. The bottom left panel, Presence of RNAP-SF complexes in the three settings of the core regulatory network model: Red dots represent the percentage of free RNAP-SF complexes, while blue and green represent one and seven RNAP-SF complexes, respectively. The bottom right panel, two simulations of ColEI replication model: red line represents the unconstrained situation where RNA I and RNA primer show no interaction, and the blue line indicates RNA interaction inhibit replication. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

of the spread of infectious diseases. Of note, the accurate modeling of infectious diseases leads to several significant consequences such as predictions on the impact of an outbreak over the population; determining the number of infected, exposed and total cases. Moreover, a proper model could lead to identifying the rate of disease growth, and to the definition of the most effective policies aimed at avoiding the spread of the disease, such as vaccination strategies. One of the earliest mathematical models ever developed to study the propagation of an infectious disease was the SIR model developed by Kermack and McKendrick in 1927 [34]. In this seminal work, the authors defined a compartmentalized ODE model where people could be categorized in three groups. As a consequence, the SIR model defines susceptibles (S), i.e. those who can be infected; infected (I), i.e. those who have been infected; and removed (R), those who cannot further participate in the dynamics either by acquiring immunity or by being dead. This simple but effective model predicts the number of infected people over time by finding both the basic reproductive number R_0, i.e. the number of individuals that each infected person could infect over time, and the exponential growth rate of the disease r_0, i.e. a number defining

how quickly the growth of the disease follows the exponential trend.

Despite its broad applicability [35–37], this model lacks important features of some contagious diseases such as the transmission of viruses from dead people, as was recently observed for the Ebola virus [38]. A model explicitly dealing with this type of transmission was recently developed [39]. In this SEIRD model, individuals are classified into susceptible (S), exposed (E), infected (I), removed (R) and dead (D) (Fig. 2A). To assess the capabilities of PISKaS to deal with such compartmentalized models, we developed a SEIRD model to simulate an outbreak of the Ebola disease in a hypothetical country composed by 4 cities (Fig. 2B), named C1 to C4, and having a bidirectional highway system interconnecting them. In doing so, we produced a country-level model by including compartments resembling four large cities with population densities of 9,200, 3,912, 3867 and 3958 habs./km^2 for C1, C2, C3 and C4, respectively. Moreover, our model allows the free travel of agents between cities from all S, E and I states. This is particularly important when considering that both E and I agents could spread the disease by travelling along the hypothetical country. Rates of exposition from infected people (β_i), exposition from dead people (β_D), the time needed to transit from the exposed to the infected state (T_E), and the time (T_I) in which infected people remain until the application of the rate of either death (f) or removal ($1 - f$), were all obtained from the ODEs adjustment of the SEIRD model to actual data coming from the last Ebola outbreak in Africa, as proposed in Ref. [39]. *Kappa* rules employed to generate the set of stochastic SEIRD simulations are shown in Fig. S2.

As an initial condition, ten infected persons appeared in city C1 on day 0. Of note, this relatively large number of initially infected people ensured the expansion of the disease since the disease did not spread in the majority of the simulations initialized with fewer infected individuals (data not shown). These simulations were employed to characterize the dynamic properties of the disease spreading in each city, including the exponential growth rate r_0 averaged after 1000 simulations. City-level r_0 (Fig. 2D) were calculated using Equation (3), where $I(t)$ is the number of infected people at time t.

$$I(t) = \exp(r_0 t) \tag{3}$$

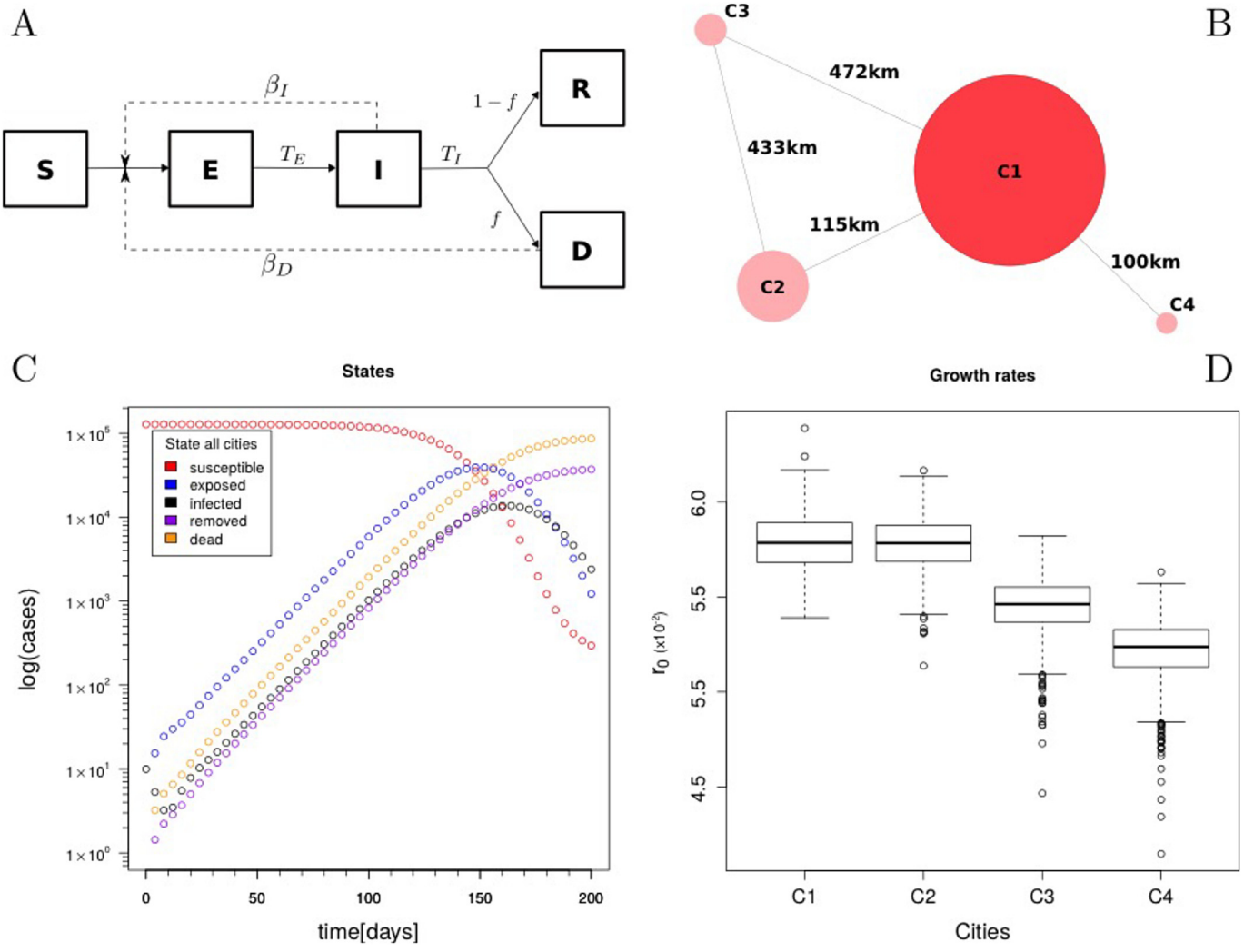

Fig. 2. A hypothetical outbreak of the Ebola virus. Panel A), graphical representations of the SEIRD model. Black boxes describe each state that can be achieved by any agent of the simulation: Susceptible, Exposed, Infected, Removed and Dead. Solid arrows represent how these states evolve over time, while dashed arrows indicate the transmission of the virus. Panel B), topology of connectivity between cities of our model. The size of each city is proportional to the population density normalized by the population density of C1. Panel C) and D), results from the simulations of the spread of Ebola. Panel C), averaged occurrence of the five different states for agents participating in the SEIRD model using a log scale. Panel D), exponential growth rates calculated for each city in the model according to the 1000 independent simulations. Data is shown using a box-plot where central black line is the average, limits of the boxes represent the standard deviation and the whiskers represent the variance.

As expected, the average trajectory over an ensemble of 1000 stochastic simulations shows the typical exponential spread of the Ebola disease (Fig. 2C). It is worth noting that in the first day of the disease spread, the number of infected people decreases due to two factors: first the transit of infected people to either the dead or the removed states, and also because of the time that people spend in the susceptible state, around 6 days after the exposition to the virus. This can be noted in Fig. 2C, by considering that people of types R, D and E, appear around day 6 from the simulation start. Previously to this day, only infected and susceptible people are present in the simulation. When considering the growth rates for different cities (Fig. 2D), our results indicate that in each city the disease spreads at different rates. This behavior could be the consequence of both differences between the population density of each city, and the highway connectivity between cities. Of note, while it is expected that city C1 being the largest city will exhibit the higher r_0 (Fig. 2D), it is interesting that the second highest r_0 belongs to city C2 instead of city C4, which is the closest city to the origin of the disease (city C1). Importantly, population densities of cities C2, C3 and C4 are similar therefore the important parameter influencing r_0 in city C2 seems to be the topology of connectivity. Despite the degree of connectivity between cities C2, and C3 is the same ($D=2$), the larger value of r_0 for C2 must be influenced by the close proximity of C2 to the origin of the infection. Notably, city C4 is the closest city to the origin of the infection and also is the smallest city in the model (Fig. 2D). On the contrary to what we expected, r_0 for city C4 is the smallest one, denoting the importance of the degree of connectivity for the spread of the disease. As a whole, our data indicates that both proximity and degree of connectivity should be key factors to consider when developing contingency policies to avoid the spread of contagious diseases such as Ebola.

3.3. Exploring human collaboration using the Prisoner's dilemma

Social Dilemmas (SDs) encompass different situations in which people may cooperate or defect, thus obtaining a personal benefit and producing a detriment to their counterpart, represented by other individuals, and by extension to society at large [40]. In general, SDs include a variety of problems spanning economics, political sciences, anthropology and cultural evolution. To formally treat these SDs, mathematical approaches from Game Theory (GT) provide a theoretical framework suitable to produce computational models such as the Prisoner's Dilemma (PD) or the investors game, among others [41]. These games have guided quantitative research on social sciences for decades, predicting under which circumstances cooperation or defection may surge, providing a rich scaffold to understand the evolution of collaboration in human societies [40–45]. In each matrix-based game, such as the PD, there are two players: each player, or agent, has two possible moves or actions interpreted as Cooperation (C) or Defection (D), resulting in four possible outcomes of the game (Table 1). The possible outcomes from this game are: reward from mutual cooperation (R); punishment arising from mutual defection (P); suckers outcome, obtained by the player who cooperates against a defecting partner (S); and temptation outcome, achieved by defecting against a cooperating partner (T).

Table 1
The pay-off matrix used for our simulation of the PD. Temptation (T) produces +2 units of goods, reward (R) produces +1 units, punishment (P) produces −1 and sucker's pay-off produces −2 units.

		Player 1	
		C	D
Player 2	C	+1 (R), +1 (R)	−2 (S), +2 (T)
	D	+2 (T), −2 (S)	−1 (P), −1 (P)

In its simplest form, a PD is a matrix-based game described by the following relations (Equations (4) and (5)):

$$T > R > P > S \tag{4}$$

$$2R > T + S \tag{5}$$

where Equation (4) ensures that exploiting a cooperating partner is preferred over mutual cooperation, which is preferred to mutual defection, and the latter is preferred over being exploited. Importantly, Equation (5) ensures that mutual cooperation is preferred to both unilateral cooperation and defection [40,41,45].

Despite GT providing a versatile theoretical framework to tackle several SDs, usual approaches do not consider the relationship between trust among agents participating in the game and the probability of cooperation or defection. Of note, the creation and production of social capital in human societies depends on the level of trust between people belonging to the same social class (i.e. intra-class trust), and also between people belonging to different social classes (i.e. inter-class trust) [46]. Moreover, societies with higher trust levels usually exhibit higher social capital and, as a consequence, higher tendency to collaborate [47]. Therefore, we decided to integrate trust as part of the social variables belonging to the agents. In doing so, the economic prosperity of a society, expressed in the form of gross domestic product (GDP), will depend on the creation and development of social capital, which is our simulation is in direct relationship with both intra- and inter-class trust.

To explore PISKaS capabilities to produce a model where the operation of rules depends on the value of certain properties belonging to the agents, we produced a PD where the probability of defection or cooperation will depend on the level of trust held among the agents. Consequently, reward or punishment will be received by the agents of the simulation according to the pay-off matrix of the PD (Table 1). This behavior will increase or decrease individual wealth over the simulation and summing over the agents of the simulation, will produce the GDP of the society.

To define the players participating in the PD, our agents has the following interface:

Person(X, T~1–2~3–4, W)

Where *Person* is the name of the agent, *X* corresponds to the interaction site between agents, *T* defines the level of trust, and *W* represents a wallet where every agent accumulates the reward obtained from the last interaction, according to the pay-off matrix. To explore the hypothesis whether the size of the population is directly related to the creation of GDP, we decided to model two isolated cities, City 1 and City 2, with 800 and 1000 agents, representing the urban and rural population of New Zealand, respectively. The distribution of trust for both cities was obtained from the trust profiles of the urban population of New Zealand, as revealed by the "The Global Trust Inventory" (Liu, Milojev & Gil de Zuniga, to be published). In brief, the trust profile of the population is composed by a 28,3% of High Trust (HT), 46,4% Moderate Trust (MT), 19,0% Low Institutional Trust (LIT), and 6,2% Low Trust (LT). Therefore, the number of agents belonging to each class of trust was obtained by multiplying the number of total agents by the trust distribution. *Kappa* rules to run the simulation were written for each trust profile (Fig. S3) describing how players engage in interactions with other players during the PD games. Importantly, independent of the level of trust, all agents have the same probability of interaction over the simulation. However, as the number of

agents corresponding to each trust profile is different, generally speaking, the probability of interactions will be higher for those larger groups compared to that of the smaller groups.

To produce an ensemble trajectory of our PD model, each simulation was run for 200 arbitrary units (a.u.) of time, and their results were averaged between 1000 independent simulations. Results shown in Fig. 3 correspond to a model composed by two isolated compartments, named City 1 (Fig. 3A, B, E, and F) and City 2 (Fig. 3C, D, G and H). The normalized social wealth (GDP) was calculated by summing up over the *W* site of the individual agents and dividing by the number of agents (Fig. 3A and C). We also computed the normalized contribution to the GDP of each trust profile, as shown in Fig. 3B and D. The percentage of different outcomes coming from the interactions between trust profiles is shown in Fig. 3E and H. In general, the average normalized GDP generated over the simulation by City 2 is higher compared to that of City 1. Moreover, the slope of the curve showing the creation of the GDP over time is higher in City 2 compared to that of City 1.

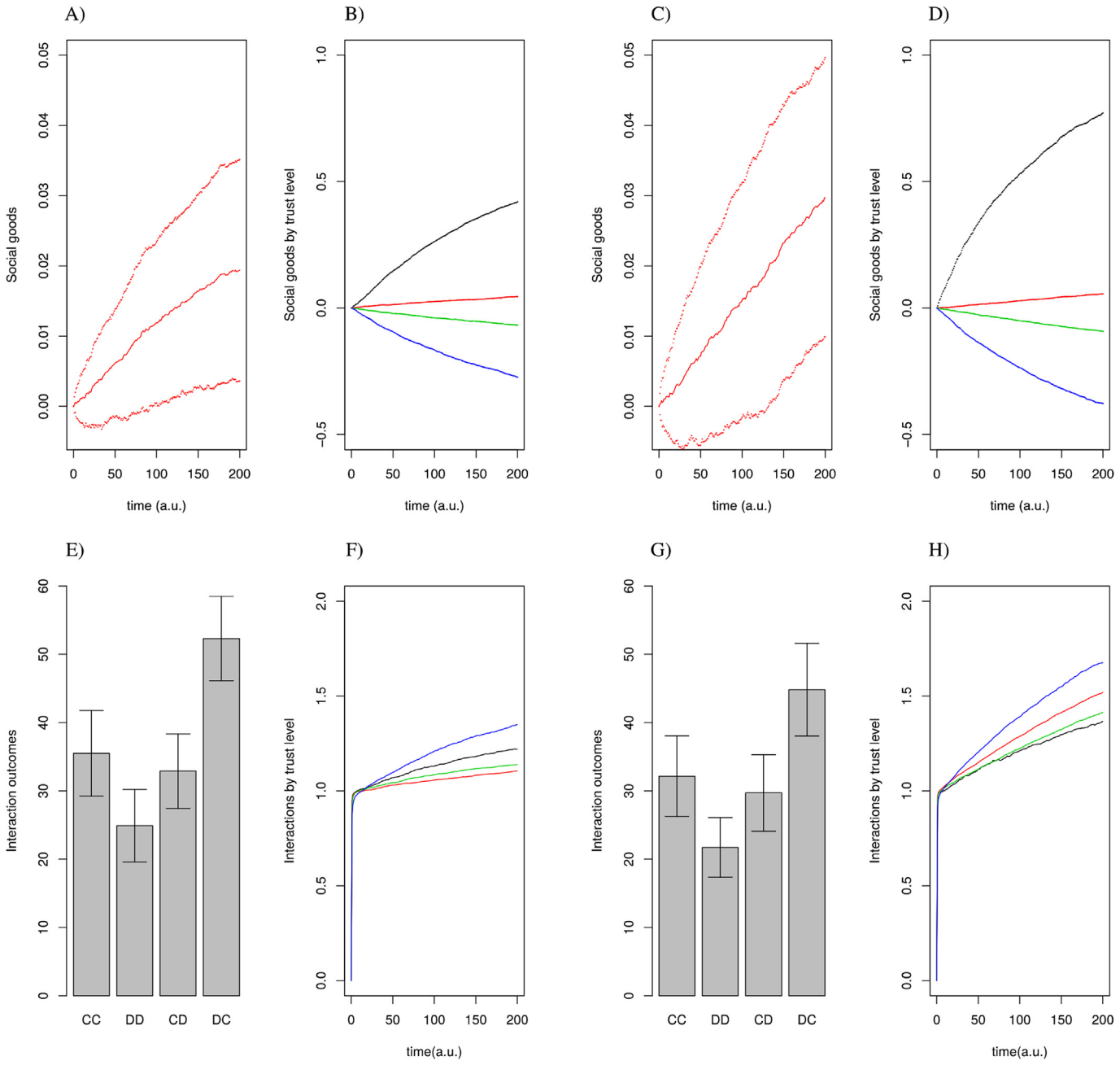

Fig. 3. Results of the PD model. The outcome of 1000 independent simulations for City 1 and for City 2 are presented in Panels A), B), E), and F); and Panels C), D), G) and H), respectively. The accumulation of social goods as a measurement of the average normalized GDP (solid red line) for City 1 and City 2 is shown in Panels A) and C), respectively. Dotted lines represent the standard deviation of the GDP over 1000 simulations. The normalized individual accumulation of goods for agents belonging to different trust profiles of City 1 and City 2, is presented in Panels B) and D), respectively. The total outcome of the PD interactions between agents from City 1 and City 2 over the simulation is shown using percentages in Panels E) and G), respectively. The four possible outcomes of the PD matrix-game are denoted as CC (cooperate-cooperate), DD (defect-defect), CD (cooperate-defect) and DC (defect-cooperate). The number of interactions from the PD occurring between agents of the same trust profile in City 1 and City 2 are shown in Panels F) and H), respectively. For Panels B, D), F) and H), black line represents HT, red line represents MT, green line represents LIT and blue line represents LT. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

When we consider the standard deviation (Fig. 3A and C) generated by 1000 independent simulations (see methods), both GDP curves seem to be generated from the same distribution. Therefore, no significant differences can be observed between both curves. However, a slightly significant difference between the standard deviation of the GDP curve of City 2 compared to that of City 1, at both the lower and higher extremes, can be seen in Fig. 3A and C. This behavior implies that at the beginning of some simulations —between 0 and 80 a. u. of time— City 2 will generate a lower GDP. On the other hand, for some other simulations, City 2 will generate a higher GDP at the end of the simulation, between 110 and 220 a. u. of time. Consequently, these differences could be explained by the accumulation of individual wealth of the population.

To further explore the role of wealth accumulation by different trust profiles over the creation of GDP, we calculated the distribution of wealth by trust level, normalized by the number of agents of each profile, as seen in Fig. 3B and D. As whole, the average individual accumulation of wealth by both MT and LIT is similar when comparing both cities. When considering the average individual accumulation of wealth for both HT and LT, at the end of the simulation City 2 exhibits both the largest and the lowest values compared to that of City 1. This behavior could explain the differences shown in Fig. 3A and C denoting the importance of individual wealth accumulation. Despite that a higher GDP for City 2 appears as the obvious result —considering that City 1 has a 20% lower number of agents—, unexpectedly, the accumulation of individual wealth per trust profile in City 2 do not follow a proportional increase. Notably, only the accumulation of individual wealth of both HT and LT are the ones that seem to be affected by the number of agents. To determine whether this effect is generated by the interactions between agents, we calculated the percentage of interactions coming from the PD, as seen in Fig. 3E and G. As a whole, the percentage of interactions of types cooperate-cooperate (CC), defect-defect (DD) and cooperate-defect (CD) from both cities is similar, with higher differences shown for interactions of type defect-cooperate (DC), which is significantly higher in City 1 compared to that of City 2. When considering the number of interactions normalized per trust profile (Fig. 3F and H), is interesting to note that the total number of interactions for every trust profile in City 2 is higher than that of City 1. Moreover, the ranking of interactions between trust profiles is different when comparing both cities: while in both cities the profile showing the largest interaction is LT, in City 2 the profile showing the lowest interaction is HT, whereas in City 1 the lowest one is MT. As the probability of interactions between all profiles is the same (see above), the most important factor influencing the increase of the GDP of City 2 compared to that of City 1, is the number of interactions occurring in the larger profiles named MT and LIT which in City 2 appear at the middle of the interaction ranking. In contrast, these two larger groups appear at the bottom of the interactions ranking in City 1. Therefore, the higher GDP of City 2 compared to that of City 1 could be the result of an elevated number of interactions of the two larger trust profiles of the society.

4. Discussion

Through the description of three multiscale examples representing CSs from different domains of knowledge, we have explored the versatility of PISKaS. Despite we have explored different features from our models, an exhaustive parameter exploration, including topology of connectivity between compartments, sensibility of the models to changes on rates of rules, adjustment of rates to real data, among other, are all efforts we are currently executing and whose results will be published elsewhere.

Of note, PISKaS can scaled up compared to the original implementation by adding distributed memory computational resources and, in doing so, it allows for the definition of more realistic models with increased complexity. Due to the explicit declaration and usage of compartments, models in PISKaS overcome the limitation of running the SSA only on homogeneously mixed volumes, using a divide-and-conquer approach to speed-up the calculations.

As denoted by the simulation of the transcriptional networks of *E. coli*, non-compartmentalized approaches to conduct computational simulations may render unrealistic or over-simplified results: a limitation that is overcome by the declaration of compartments in PISKaS. It is also well known that transcription and other biological processes at the molecular level are stochastic; therefore, simulations based on differential equations oversimplify the inherent dynamics of these biological processes disregarding the importance of individual components. Our models, first simulated independently and then combined, reproduced properties of their natural counterpart. Of note, this simple model could be further expanded following the same modular procedure depicted above: separated modules, representing different parts of the whole system, can be modeled independently to then be added to produce a global model. In addition, once the parameters of the smaller models are determined, their re-optimization in the final model should be less expensive, requiring only smaller fine-tuning. This is particularly relevant when considering that by mixing the tow compartments Given the simple scheme that enables PISKaS to simulate compartments and transportation of agents, we are expanding this model to include the regulation of gene expression for the entire genome of *E. coli* (to be published elsewhere).

By studying a hypothetical outbreak of the Ebola virus in Chile, we demonstrated how explicit compartmentalization permits the identification of important properties of a system that may be overlooked otherwise. Traditional modeling of disease spread calculates a single exponential growth rate r_0 for each disease. In contrast, our simulations identified different r_0 for each city, demonstrating the importance of both the population density and the connectivity pattern between cities. Despite the obvious idea that an infectious agent is transmitted at higher rates in densely populated cities than in the countryside, a compartmentalized model combining both population density and highway topology is very hard to produce using differential equations. In the case of our PISKaS model, to produce such model is worth only a dozen of lines of *Kappa* rules.

Assessing human collaboration scenarios by producing a computational model of the PD, highlights a completely different level of abstraction when modeling CSs. Interactions between humans follow complex rules, difficult to understand and quantify, where the relevance of an individual could be of radical importance for the whole [48]. To simplify the complexity of these interactions, we have produced a PISKaS model of the PD where, apart from the classical GT approach, we introduced trust as a proxy to the creation of social wealth. As a consequence, we quantified the GDP or our artificial societies that is generated by the dynamics of the PD. Notably, our simulations suggest that to increase the wealth of a society, the most relevant actor is the accumulation of goods by the middle class, usually the largest segment in western societies. Moreover, to support the advancement of societies, a relevant factor is the creation and development of trust between individuals belonging to the same social class but more importantly, is the creation and development of trust between different social classes.

5. Conclusion

In this work, we illustrate an alternative to the deterministic simulation of CSs based on the stochastic simulation paradigm. Our three examples introduce the most relevant characteristics of

PISKaS, our rule-based stochastic simulation engine. As a whole, PISKaS versatility provides a suitable framework to the study of complex multiscale systems with explicit spatial definitions. As discussed above, these features are particularly important to model biological systems where the spatial heterogeneity and the stochasticity generated by individual components are both key elements to understand the dynamics of the system.

Conflicts of interest

All authors declare no conflict of interest.

Funding

This work has been partially supported by the Proyecto Financiamiento Basal CONICYT-PIA [PFB16], ICM-Economia project to Instituto Milenio CINV [P09-022-F], and USA Air Force Office of Scientific Research Awards [FA9550-16-1-0111] and [FA9550-16-1-0384]. AJMM and RS received economic support from FONDECYT Iniciacion [11140342]. RS received support from CONICYT-PCHA Doctorado Nacional [2014-21140377]. AB received support from a FIB-UV fellowship. Powered@NLHPC: This research was also supported by the Chilean National Laboratory for High-Performance Computing (NLHPC) [ECM-02].

Acknowledgements

We acknowledge other members of the Computational Biology Lab for useful insights and discussions about this work.

Appendix A. Supplementary data

Supplementary data related to this article can be found at https://doi.org/10.1016/j.bbrc.2017.11.138.

References

[1] H. Kitano, Systems biology: a brief overview, Science (80-) 295 (2002) 1662–1664, https://doi.org/10.1126/science.1069492.
[2] C. Mayer, T.F. Hansen, Evolvability and robustness: a paradox restored, J. Theor. Biol. 430 (2017) 78–85, https://doi.org/10.1016/j.jtbi.2017.07.004.
[3] J. Fisher, T.A. Henzinger, Executable cell biology, Nat. Biotechnol. 25 (2007) 1239–1249, https://doi.org/10.1038/nbt1356.
[4] M. Barrio, K. Burrage, A. Leier, T. Tian, Oscillatory regulation of Hes1: discrete stochastic delay modelling and simulation, PLoS Comput. Biol. 2 (2006) 1017–1030, https://doi.org/10.1371/journal.pcbi.0020117.
[5] J.R. Faeder, M.L. Blinov, W.S. Hlavacek, Rule-based modeling of biochemical systems with BioNetGen, Methods Mol. Biol. 500 (2009) 113–167, https://doi.org/10.1007/978-1-59745-525-1_5.
[6] E. Murphy, D. Vincent, J. Feret, J. Krivine, R. Harmer, Rule based modeling and model refinement, in: Elem Comput Syst Biol, John Wiley {&} Sons, Inc., Hoboken, 2010, pp. 83–114.
[7] L.A. Chylek, L.A. Harris, J.R. Faeder, W.S. Hlavacek, Modeling for (physical) biologists: an introduction to the rule-based approach, Phys. Biol. 12 (2015) 45007, https://doi.org/10.1088/1478-3975/12/4/045007.
[8] J. Krivine, et al., KaSim User Manual, 2010.
[9] M.A. Gibson, J. Bruck, Efficient exact stochastic simulation of chemical systems with many species and many channels, J. Phys. Chem. A 104 (2000) 1876–1889, https://doi.org/10.1021/jp993732q.
[10] L. Lok, R. Brent, Automatic generation of cellular reaction networks with Moleculizer 1.0, Nat. Biotechnol. 23 (2005) 131–136, https://doi.org/10.1038/nbt1054.
[11] V. Danos, W. Fontana, J. Krivine, Scalable simulation of cellular signaling networks, in: Program Lang Syst, Springer, Berlin Heidelberg, 2007, pp. 139–157, https://doi.org/10.1007/978-3-540-76637-7.
[12] Y. Cao, D.T. Gillespie, L.R. Petzold, Efficient step size selection for the tau-leaping simulation method, J. Chem. Phys. 124 (2006) 44109, https://doi.org/10.1063/1.2159468.
[13] D.T. Gillespie, Approximate accelerated stochastic simulation of chemically reacting systems, J. Chem. Phys. 115 (2001) 1716, https://doi.org/10.1063/1.1378322.
[14] Y. Cao, D.T. Gillespie, L.R. Petzold, The slow-scale stochastic simulation algorithm, J. Chem. Phys. 122 (2005) 14116, https://doi.org/10.1063/1.1824902.
[15] M. Rathinam, L.R. Petzold, Y. Cao, D.T. Gillespie, Stiffness in stochastic chemically reacting systems: the implicit tau-leaping method, J. Chem. Phys. 119 (2003) 12784, https://doi.org/10.1063/1.1627296.
[16] L. Dematté, D. Prandi, GPU computing for systems biology, Brief. Bioinform 11 (2010) 323–333, https://doi.org/10.1093/bib/bbq006.
[17] C. Dittamo, D. Cangelosi, Optimized parallel implementation of Gillespie's first reaction method on graphics processing units, in: 2009 Int Conf Comput Model Simul, IEEE, 2009, pp. 156–161, https://doi.org/10.1109/ICCMS.2009.42.
[18] F. Nunez, C. Ravello, H. Urbina, T. Perez-Acle, in: A Rule-based Model of a Hypothetical Zombie Outbreak: Insights on the Role of Emotional Factors during Behavioral Adaptation of an Artificial Population, 2012 arXiv Prepr arXiv12104469.
[19] I. Fuenzalida, A. Bustos, D. Inostroza, A. Bernardin, A. Martin, T. Perez-Acle, PISKaS: a HPC tool for stochastic agent/rule-based modeling of spatially explicit biological systems [slides, version 1; not peer reviewed], F1000Research (ISCB Comm. J.) 6 (2017).
[20] I. Fuenzalida, A. Martin, T. Perez-Acle, PISKa: a parallel implementation of spatial Kappa [poster, version 1; not peer reviewed], F1000Research (ISCB Comm. J.) 4 (2015).
[21] I. Fuenzalida, A. Martin, B. Alejandro, T. Perez-Acle, Modeling multiscale complex biological systems using PISKa [poster, version 1; not peer reviewed], F1000Research (ISCB Comm. J.) 6 (2017).
[22] K. Nishimura, S. Tsuru, H. Suzuki, T. Yomo, Stochasticity in gene expression in a cell-sized compartment, ACSs Synth. Biol. 4 (2015) 566–576, https://doi.org/10.1021/sb500249g.
[23] W.A. Lim, C.M. Lee, C. Tang, Design principles of regulatory networks: searching for the molecular algorithms of the cell, Mol. Cell. 49 (2013) 202–212, https://doi.org/10.1016/j.molcel.2012.12.020.
[24] C.F. Lopez, J.L. Muhlich, J.A. Bachman, P.K. Sorger, Programming biological models in Python using PySB, Mol. Syst. Biol. 9 (2013) 646, https://doi.org/10.1038/msb.2013.1.
[25] D. Stewart, J.R. Wilson-Kanamori, Modular modelling in synthetic biology: light-based communication in *E. coli*, Electron Notes Theor. Comput. Sci. 277 (2011) 77–87, https://doi.org/10.1016/j.entcs.2011.09.037.
[26] B.-K. Cho, D. Kim, E.M. Knight, K. Zengler, B.O. Palsson, Genome-scale reconstruction of the sigma factor network in Escherichia coli: topology and functional states, BMC Biol. 12 (2014) 4, https://doi.org/10.1186/1741-7007-12-4.
[27] I.M. Keseler, A. Mackie, A. Santos-Zavaleta, R. Billington, C. Bonavides-Martínez, R. Caspi, C. Fulcher, S. Gama-Castro, A. Kothari, M. Krummenacker, M. Latendresse, L. Muñiz-Rascado, Q. Ong, S. Paley, M. Peralta-Gil, P. Subhraveti, D.A. Velázquez-Ramírez, D. Weaver, J. Collado-Vides, I. Paulsen, P.D. Karp, The EcoCyc database: reflecting new knowledge about Escherichia coli K-12, Nucleic Acids Res. 45 (2017) D543–D550.
[28] M. Mauri, S. Klumpp, A model for sigma factor competition in bacterial cells, PLoS Comput. Biol. 10 (2014) 29–34, https://doi.org/10.1371/journal.pcbi.1003845.
[29] A. Ishihama, Subunit of assembly of Escherichia coli RNA polymerase, Adv. Biophys. 14 (1981) 1–35.
[30] M. Patrick, P.P. Dennis, M. Ehrenberg, H. Bremer, Free RNA polymerase in Escherichia coli, Biochimie 119 (2015) 80–91, https://doi.org/10.1016/j.biochi.2015.10.015.
[31] J.E. Mitchell, T. Oshima, S.E. Piper, C.L. Webster, L.F. Westblade, G. Karimova, D. Ladant, A. Kolb, J.L. Hobman, S.J.W. Busby, D.J. Lee, The Escherichia coli regulator of sigma 70 protein, Rsd, can up-regulate some stress-dependent promoters by sequestering sigma 70, J. Bacteriol. 189 (2007) 3489–3495, https://doi.org/10.1128/JB.00019-07.
[32] B. Polisky, ColEI replication control Circuitry: sense from antisense minireview, Cell 55 (1988) 929–932, https://doi.org/10.1016/0092-8674(88)90235-8.
[33] A. Robinson, A.M. van Oijen, Bacterial replication, transcription and translation: mechanistic insights from single-molecule biochemical studies, Nat. Rev. Microbiol. 11 (2013) 303–315, https://doi.org/10.1038/nrmicro2994.
[34] W.O. Kermack, A.G. McKendrick, A contribution to the mathematical theory of epidemics, in: Proc R Soc London a Math Phys Eng Sci, 1927, pp. 700–721.
[35] B. Shulgin, L. Stone, Z. Agur, Pulse vaccination strategy in the SIR epidemic model, Bull. Math. Biol. 60 (1998) 1123–1148.
[36] M.E.J. Newman, Spread of epidemic disease on networks, Phys. Rev. E 66 (2002) 16128.
[37] H.W. Hethcote, The mathematics of infectious diseases, SIAM Rev. 42 (2000) 599–653, https://doi.org/10.1137/S0036144500371907.
[38] W.H.O. Team, Ebola Response, Ebola virus disease in West Africa—the first 9 months of the epidemic and forward projections, N. Engl. J. Med. 2014 (2014) 1481–1495.
[39] J.S. Weitz, J. Dushoff, Modeling post-death transmission of Ebola: challenges for inference and opportunities for control, Sci. Rep. 5 (2015) 8751, https://doi.org/10.1038/srep08751.
[40] V. Capraro, A model of human cooperation in social dilemmas, PLoS One 8 (2013), e72427, https://doi.org/10.1371/journal.pone.0072427.
[41] T. Veloz, P. Razeto-Barry, P. Dittrich, A. Fajardo, T. Veloz, P. Razeto-Barry, P. Dittrich, A. Fajardo, Mathematical Biology Reaction networks and evolutionary game theory, J. Math. Biol. 68 (2014) 181–206, https://doi.org/10.1007/s00285-012-0626-6.
[42] M. Nowak, K. Sigmund, The evolution of stochastic strategies in the Prisoner's

Dilemma, Acta Appl. Math. 20 (1990) 247–265, https://doi.org/10.1007/BF00049570.
[43] D. Ashlock, N. Rogers, The Impact of Long - term Memory in the Iterated Prisoner ' s Dilemma, (n.d.).
[44] E. El-Seidy, E.M. Elshobaky, K.M. Soliman, Two population three-player prisoner's dilemma game, Appl. Math. Comput. 277 (2016) 44–53, https://doi.org/10.1016/j.amc.2015.12.047.
[45] Steven Kuhn, 'Prisoner's Dilemma', the Stanford Encyclopedia of Philosophy, Spring 201, Metaphysics Research Lab, Stanford University, 2017.
[46] R.D. Putnam, Social capital: measurement and consequences, Can. J. Policy Res. (2001) 41–51.
[47] F. Fukuyama, Trust: the social virtues and the creation of prosperity, Free Press Pap. (1995) 23–32. Part I, Chapter 3.
[48] Isaac Asimov, Foundation, Pengiun Random House LLC, 1991, 0-553-29335-4.