Biochemical and Biophysical Research Communications xxx (2017) 1–10

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 xxx

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

Please cite this article in press as: T. Perez-Acle, et al., Stochastic simulation of multiscale complex systems with PISKaS: A rule-based approach, Biochemical and Biophysical Research Communications (2017), https://doi.org/10.1016/j.bbrc.2017.11.138