

Experimenting the Simulation Strategy of Membrane Computing with Gillespie Algorithm by Using Two Biological Case Studies

¹Ravie Chandren Muniyandi and ²Abdullah Mohd Zin

¹School of Computer Science,

²School of Information Technology,

Faculty of Information Science and Technology,

University Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Abstract: Problem statement: The evolution rules of membrane computing have been applied in a nondeterministic and maximally parallel way. In order to capture these characteristics, Gillespie's algorithm has been used as simulation strategy of membrane computing in simulating biological systems. **Approach:** This study was carried to discuss the simulation strategy of membrane computing with Gillespie algorithm in comparison to the simulation approach of ordinary differential equation by analyzing two biological case studies: prey-predator population and signal processing in the Ligand-Receptor Networks of protein TGF- β . **Results:** Gillespie simulation strategy able to confine the membrane computing formalism that used to represent the dynamics of prey-predator population by taking into consideration the discrete character of the quantity of species in the system. With Gillespie simulation of membrane computing model of TGF- β , the movement of objects from one compartment to another and the changes of concentration of objects in the specific compartments at each time step can be measured. **Conclusion:** The simulation strategy of membrane computing with Gillespie algorithm able to preserve the stochastic behavior of biological systems that absent in the deterministic approach of ordinary differential equation. However the performance of the Gillespie simulator should be improved to capture complex biological characteristics as well as to enhance the simulation processes represented by membrane computing model.

Key words: Membrane computing, Gillespie algorithm, biological system

INTRODUCTION

Membrane computing enriches the model of molecular computing by providing a spatial structure for molecular computation, inspired by the hierarchical structure of living cells. In the structure and the functioning of cell, membranes play an essential role in which objects pass in a regulated fashion within and across the membranes. The cell is separated from its environment by means of the plasma membrane and it is internally compartmentalized by means of internal membranes. The membrane computing model formalizes this fundamental feature of the living cell, namely, membrane structure (Paun, 2000). Membrane computing is introduced as a class of parallel, distributed and nondeterministic computing devices (Paun, 1998). The fundamental features that are used in this computing model are a membrane structure where objects evolve according to specified evolution rules, which also determine the communication of objects

between membranes. The evolution rules are applied in a nondeterministic and maximally parallel way, which means all the objects that can evolve, must evolve (Paun, 1998).

As a result of a membrane computing model, a computing device is obtained, starting from an initial configuration and letting the system evolve. A universal clock is assumed to exist which means at each step, all rules from all regions are simultaneously applied to all objects that can be the subject of an evolution rule. When no further rule can be applied, the computation halts and we get the result in a prescribed way. On the contrary, if there is at least one rule that can be applied forever, then the computation is unsuccessful and no output is obtained.

Since its establishment, membrane computing has been a branch of theoretical computer science, an area of mathematical investigations. However, the interest has increased in investigating membrane computing into practical computing applications. Since membrane

Corresponding Author: Ravie Chandren Muniyandi, School of Computer Science, Faculty of Information Science and Technology, University Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

computing originated from biology, question of why we do not use membrane computing back into where it has originated has arisen (Muniyandi and Abdullah, 2009). Most of mathematical models of biological processes have been done by using continuous mathematics, especially system of Ordinary Differential Equations (ODE) in which the variation of concentration of each chemical substance or object is modeled as a global entity. In this respect, it is assumed that the number of objects of each type in the reaction mix is large so that the reactions are fast. Therefore, when the number of objects in reactions is small and the reactions are slow, which is common in most biological systems, the ODE assumption is questionable. The ODE approach also makes it difficult to develop a modular and scalable design for hierarchical system. The stochastic, discrete and hierarchical characteristics of membrane computing is being considered as an alternative to solve these limitations by taking into considerations its essential features that are of interest for biological applications.

Several programming paradigms and programming languages were selected for implementing membrane systems simulators: Lisp, Haskell, MzScheme as functional programming languages (Suzuki and Tanaka, 2000; Arroyo *et al.*, 2003; Noval *et al.*, 2003; Baranda *et al.*, 2002) prolog, CLIPS as declarative languages (Cordon-Franco *et al.*, 2004; Perez-Jimenez and Romero-Campero, 2004), C, Visual C++, Java as imperative and object-oriented languages (Ciobanu and Paraschiv, 2002). Membrane computing was also described as executable specifications in Maude (Andrei *et al.*, 2005). Therefore, there are many attempts to simulate membrane computing on the existing computers. They have both didactic and scientific values. However the membrane computing is inherently parallel and, in many variants, they also exhibit an intrinsic non-determinism, hard to be captured by sequential computers. By simulating parallelism and non-determinism on a sequential machine, one can lose the real power of parallelism and attractiveness of membrane computing.

On one hand, maximum parallelism could be obtained with an unbound number of processors, assuming no scheduling overheads. Therefore there is special interest to implement membrane computing by simulations on multiple processors. Parallel and cluster implementation for membrane computing is presented using C++ and Message Passing Interface (MPI) (Ciobanu and Guo, 2004). In the research by Ciobanu and Guo (2004) a simplest variant of membrane computing, the transition membrane computing is implemented. The program is implemented and tested on a Linux cluster consists of 64 dual processor nodes

in which each node consists of two 1.4 GHz Intel PIII CPUs with 1 GB of memory and connected by Myrinet as well as gigabit Ethernet. This research implies that this implementation achieves better performance when such hardware is available (Ciobanu and Guo, 2004). Although this could be possible direction for membrane computing in future, the cost of establishing the hardware at this stage would be paramount if we considering testing huge amount of membrane systems by setting up a processor for each membrane.

On the other hand, it is important to underline the fact that implementing a membrane system on an existing electronic computer cannot be a real implementation, it is merely a simulation. However, the simulations already circulated in this area have at least three important merits: they have structure resemblance to membrane structure, they have a didactic value and they can be used for simulating membrane systems which model certain real-life phenomena, with relevance for other domains than computer science itself, such as artificial life, ecology and biochemistry (Paun, 2000).

In the P Systems webpage, there are some applications have been described. Some of them are: PSim is a Java Simulator for membrane computing based on the metabolic algorithm; SimCM is a Java simulator for transition membrane computing; SubLP-Studio is a software simulator for the Sub LP-Systems model, a variant of L-Systems and membrane computing.

Furthermore, different simulation strategies have also been employed to investigate membrane computing such as dynamic probabilistic (Pescini *et al.*, 2006), metabolic algorithm (Manca, 2008) and Gillespie algorithm. Dynamic probabilistic and metabolic algorithm are using deterministic approach. Gillespie algorithm is using stochastic approach.

This study investigates and evaluates membrane computing simulation strategy based on Gillespie algorithm compared to deterministic approach of ODE with experiments with two biological case studies. The deterministic approach refers to no randomness that involved in the development of future states of the system in which the computation evolve under time when every action produces a reaction and every reaction, in turn, becomes the action of subsequent reactions. The stochastic approach refers to non-deterministic behavior of reaction in which a system's future evolution is described by probability distributions to determine its subsequent state.

This investigation has two purposes. The first is to examine the consequence of number of objects in the reaction mix in the stochastic approach compared to

deterministic approach and to verify how this factor could influence the performance of the system. The second motive is to study the differences in simulation between deterministic and stochastic approaches in the hierarchical system and to find out how this could have an effect on the activities of objects. The prey-predator population is used as the case study to analyze the former and the signal processing in the Ligand-Receptor Networks of protein TGF- β is used to analyze the latter.

Gillespie algorithm: Gillespie (2001; 2003) algorithm provides a method for the stochastic simulation of systems of bio-chemical reactions. The validity of the method is rigorously proved and it has been already successfully used to simulate various biochemical processes. As well as this, the Gillespie's algorithm is used in the implementation of stochastic calculus (Philips and Cardelli, 2009) and in its application to the modeling of biological systems (Priami *et al.*, 2001). Meanwhile, multi-compartmental Gillespie algorithm (Perez-Jimenez and Romero-Campero, 2006) is developed by taking into account the fact that, with respect to the original algorithm where only one volume is studied, in membrane computing where there are different regions or compartments, each one can be seen as a volume with its own set of rules, besides the application of a rule inside a compartment can also affect the content of another one; for example the application of a communication rule.

Gillespie algorithm proposed that at each time step, the chemical system is in exactly one state and to directly simulate the time evolution of the system. Basically, the algorithm determines the nature and occurrence of the next reaction, given that the system is in state s at time t .

Gillespie algorithm has been applied to many *in silico* biological simulation recently. Kastner and Fraser (2002) has applied the algorithm in simulation of Hoxcis-regulatory mechanisms. The simulation is successful in reproducing key features of the wild-type pattern of gene expression and *in silico* experiments yield results similar to that of *in vivo* experiments. Besides that, Kierzek *et al.* (2001) had applied the algorithm to simulation of LacZ gene expression and uncovered relationship between frequencies of transcription and translation initiation on random fluctuations in gene expression. McAdams and Arkin (1997) had also studied the transcript initiation and translation mechanism in the cellular regulatory network using the algorithm and concluded on several behavior of stochastic gene expression mechanisms. It

has also been used to simulate the Quorum Sensing (Romero-Campero and Perez-Jimenez, 2008).

The algorithm and its attributes are described in detail in DT Gillespie (2001; 2003) research.

Case studies: Two biological case studies, prey-predator population and signal processing in ligand-receptor network of protein TGF- β are chosen to model them by using membrane computing formalism and then evaluate them by Gillespie simulator.

Prey-predator population: The prey-predator population (Jones and Sleeman, 2003), also known as the Lotka-Volterra population, is frequently used to describe the dynamics of biological systems in which two species interact, one a predator and another one its prey. Primary example of a prey-predator system comprised a plant population and an herbivorous animal dependent on that plant for food. The predator species is totally dependent on the prey species as its only food supply. The prey species has an unlimited food supply and no threat to its growth other than the specific predator. If there were no predators, the prey species grows exponentially. But there are predators, which must account for a negative component in the prey growth rate. The assumptions for the model are the rate at which predators encounter prey is jointly proportional to the sizes of the two populations and a fixed proportion of encounters lead to the death of the prey.

The prey are assumed to have an unlimited food supply and to reproduce exponentially unless subject to predation. The rate of predation upon the prey is assumed to be proportional to the rate at which the predators and the prey meet. Therefore the rule over the prey can be interpreted as: The change in the prey's numbers is given by its own growth minus the rate at which it is preyed upon.

The predator rule represents the growth of the predator population. However, the predator population growth is not necessarily equal to the rate at which it consumes the prey. There is another rule to represent the natural death of the predators which is an exponential decay. Hence the equation represents the change in the predator population as the growth of the predator population, minus natural death.

ODE model of prey-predator: The prey-predator equations are modeled by a pair of first order, non-linear, differential equations used to describe the dynamics of biological systems in which two species interact, the predator and the prey. It is modeled in differential equations (Roberts *et al.*, 2009) as:

$$\frac{dx}{dt} = x(k_1 - k_2 y) \text{ and } \frac{dy}{dt} = -y(k_3 - k_4 x)$$

Where:

- y = The number of some predator
- x = The number of its prey
- dy/dt and dx/dt = Represents the growth of the two populations against time; t represents the time in seconds
- k₁, k₂, k₃ and k₄ = Parameters representing the interaction of the two species

The simulation using Scilab in Fig. 1 shows the oscillations of the prey-predator model simulated by ODE for 60 sec for x = 200 and y = 80 with kinetic constants k₁ = 1, k₂ = 0.01, k₃ = 0.05 and k₄ = 0.005. This result established the oscillation behavior of prey-predator model.

Signal processing in the ligand-receptor network of

TGF-β: Transforming Growth Factor Beta (TGF-β) is a type of protein that functions in cells (Villar *et al.*, 2006). In signal transduction some cells secrete TGF-β and also generate receptors for TGF-β. The TGF-β signal transduction pathway plays a central role in tissue homeostasis and morphogenesis. It transduces a variety of extracellular signals into intracellular transcriptional responses that control the excesses of cellular processes such as cell growth, migration, adhesion, apoptosis and differentiation. At the molecular level, complex signal transduction machinery integrates signals from the 42 known ligands of the TGF-β superfamily of proteins. The elements of this machinery incorporate the members of the two main receptor families called type I and type II receptors.

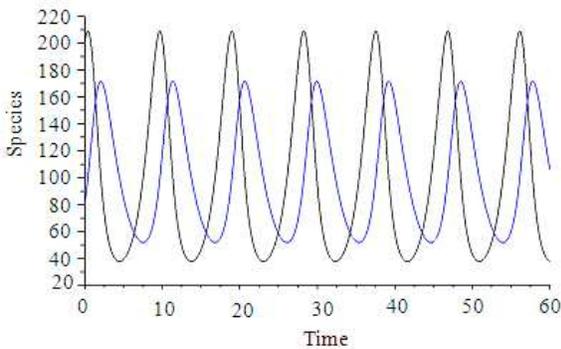


Fig. 1: Oscillation of the prey-predator model simulated by ODE

Each ligand induces the formation of a receptor complex with type I and type II receptors, which then signal through the channels. The capacity of most ligands to bind several type I and type II receptors lead to a complex ligand-receptor interaction network.

TGF-β has particular interest in cancer research. For instance, in epithelial cells, it suppresses cellular growth and its inactivation contributes to tumourigenesis. The versatility of the pathway in eliciting different types of behavior is perhaps best epitomized by the pervasive, rather paradoxical ability of TGF-β to change its function from suppressor to promoter of growth in epithelial cells during tumor progression. It has been suggested that TGF-β can suppress the growth of cells around the tumor, that it can shut down locally the immune system and that it can promote angiogenesis. All these paracrine effects would help the growth of the tumor *in vivo*, where it has to compete with neighboring cells.

ODE model of ligand-receptor network of TGF-β:

The model intends to study the signal processing potential of the ligand-receptor network and receptor trafficking (Villar *et al.*, 2006). The essential elements of this model are: Ligands induce the formation of receptor complexes with type I and type II receptors; Receptors and ligand-receptor complexes can be present in two spatially distinct compartments: Plasma membrane and internalized endosomes; The signaling activity is proportional to the number of ligand-receptor complexes that are present in the internalized endosomes; Receptors and ligand-receptor complexes are continuously internalized into endosomes and recycled back to the plasma membrane; Receptor degradation has a constitutive contribution, which is the same for free receptors and ligand-receptor complexes; Receptor degradation has a ligand-induced contribution, which affects only receptors that have been complexed with ligands.

In this model the concentration of ligand is denoted by [l]. The numbers of type I and type II receptor and ligand-receptor complexes in the plasma membrane are represented by [RI], [RII] and [IRIRII]. The numbers of internalized type I and type II receptor and ligand-receptor complexes in endosomes is represented by $\overline{[RI]}$, $\overline{[RII]}$ and $\overline{[IRIRII]}$. The other parameters are: k_a is the rate constant of ligand-receptor complex formation; p_{RI} and p_{RII} are the rates of receptor production; k_i, k_r, k_{cd} and k_{lid} are the internalization, recycling, constitutive degradation and ligand-induced degradation rate constants; α is the fraction of active receptors that are recycled back to the plasma membrane and can interact again with the ligand.

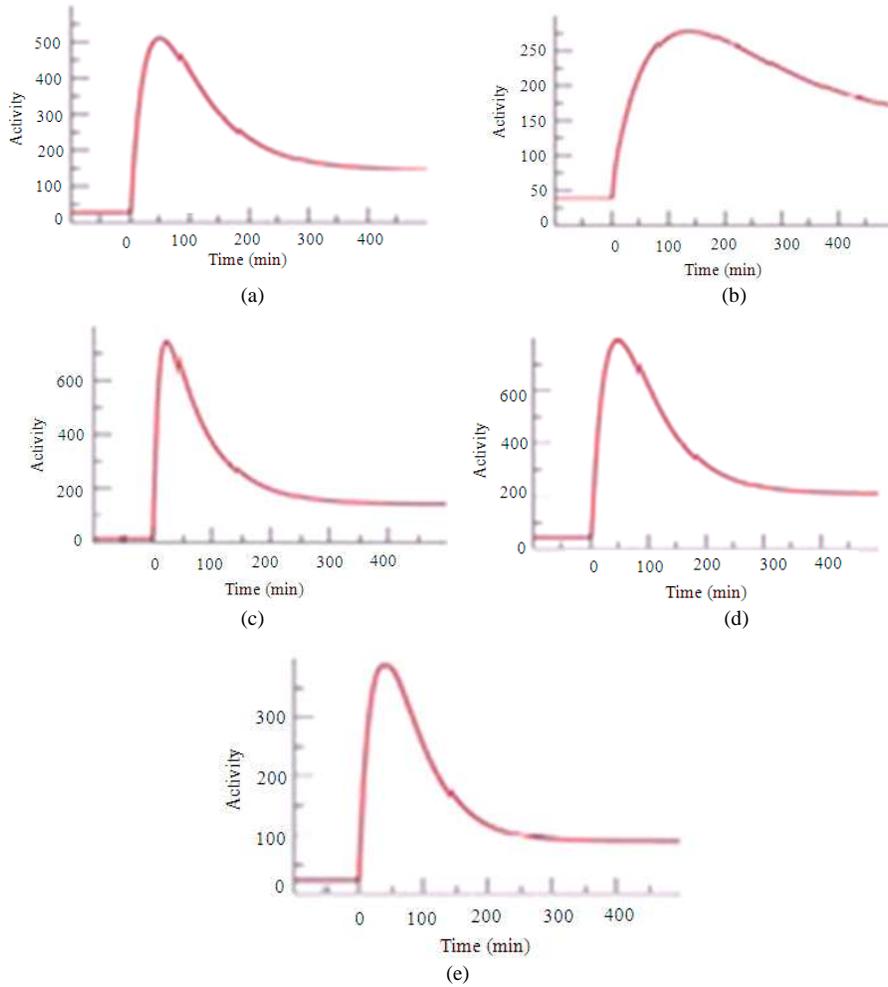


Fig. 2: Simulation of ODE model of TGF-β (Villar *et al.*, 2006)

The ODE model of signal processing in the TGF-β is represented as:

$$\begin{aligned} \frac{d}{dt}[\overline{IRIRII}] &= k_a[I][RI][RII] - (k_{cd} + k_{lid} + k_i)[\overline{IRIRII}] \\ \frac{d}{dt}[RI] &= p_{RI} - k_a[I][RI][RII] - (k_{cd} + k_i)[RI] + k_r[\overline{RI}] + \alpha k_r[\overline{IRIRII}] \\ \frac{d}{dt}[RII] &= p_{RII} - k_a[I][RI][RII] - (k_{cd} + k_i)[RII] + k_r[\overline{RII}] + \alpha k_r[\overline{IRIRII}] \\ \frac{d}{dt}[\overline{IRIRII}] &= k_i[\overline{IRIRII}] - k_r[\overline{IRIRII}] \\ \frac{d}{dt}[\overline{RI}] &= k_i[RI] - k_r[\overline{RI}] \\ \frac{d}{dt}[\overline{RII}] &= k_i[RII] - k_r[\overline{RII}] \end{aligned}$$

The signaling activity of the pathway is assumed to be proportional to the number of internalized ligand-receptor complexes, $[\overline{IRIRII}]$. The ODE model used to analyze how different parameters affect the behavior of the system: Fig. 2a shows behavior of the model for typical trafficking rates ($k_i = 1/3$, $k_r = 1/30$, $k_{cd} = 1/36$, $k_{lid} = 1/4$, $k_a = 1$, $\alpha = 1$, $p_{RI} = 1/8$, $p_{RII} = 1/4$); Fig. 2b-e show the behavior of the model with same parameter values as in 2(a), with the exception of : 2(b) ($k_i = 1/10$, $k_r = 1/100$), 2(c) ($k_i = 1$, $k_r = 1/10$), 2(d) ($k_{lid} = 0$, $\alpha = 0.5$) and 2(e) ($\alpha = 0.5$). The results are analyzed and described in detail in the Villar *et al.* (2006) research.

MATERIALS AND METHODS

Reactions and parameters: The two biological case studies are taken from research (Jones and Sleeman, 2003; Villar *et al.*, 2006) in which both case studies are

modeled by using ODE approach. For each case study, the objects, reactions and parameters involved in the biological processes are extracted from the ODE model.

Kinetic constants and initial multisets: The selection of initial multisets and the kinetic constants are required to determine the computation in membrane computing model. This basically is an attempt to gain similar simulation result as achieved with the ODE model of the biological system to subsequently analyze whether membrane computing model could preserve the biological characteristics of the biological system better than the ODE model. The initial multisets and kinetic constants extracted from the ODE model are taken as initial test cases with Gillespie simulator. Then these test cases are altered accordingly to determine the appropriate multisets and kinetic constants for membrane computing model. The kinetic constants for membrane computing model of prey-predator population are determined through simulation tests with Gillespie simulator. The best kinetic constants are chosen when the oscillation of prey-predator population is obtained. The chosen kinetic constants for k_1 , k_2 , k_3 are 10, 0.02 and 15, respectively, when the initial numbers of preys and predators are fixed at 1000 and 200, respectively. The kinetic constants for membrane computing model of TGF- β are similar to that of the ODE model except for the constant of ligand-receptor complex formation rate (k_a) and for ligand-induced degradation rate (k_{lid}). k_a is adjusted from 1-0.01 to accommodate the ligand concentrations changes from 3×10^{-5} to 0.01 at time 0. k_{lid} which should be 0 in the ODE model are adjusted to 0.01 in the membrane computing model to obtain peak activity of IRIRII in endosomes. The initial concentrations of RI and RII in the plasma membrane are selected through simulation test. The initial concentration for both RI and RII in plasma membranes is fixed at 1130 units. The initial concentration of IRIRII in endosomes is similar to the ODE model and it is 40 units.

Modeling: The objects, reactions and parameters extracted from ODE model for the case studies are used in modeling them using membrane computing formalism (Muniyandi and Abdullah, 2009).

Simulation: Multicompartment Gillespie simulator in C (Romero-Campero and Gheorghe, 2007) is used to simulate the membrane computing model. Firstly, the membrane computing model is converted into System Biology Markup Language (SBML) notations which describe the components of the biological system. Then, the simulator will specify the list of

compartments and the structural hierarchy and the initial amounts of the objects from the SBML notations. This is the initial state of the system and is given to the simulator to produce an evolution of the objects over simulation steps. The membrane computing simulation results using Gillespie algorithm are compared to the results of ODE approach.

RESULTS

Membrane computing model of prey-predator population: A model for prey-predator population is obtained by considering a membrane computing with a compartment contains rules describing the reactions between preys and predators. The Prey-Predator model (PP) is represented as:

$$PP = (V, \mu, \omega, R)$$

The objects are preys and predators represented as X and Y respectively. They are:

$$V = \{X, Y\}$$

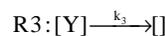
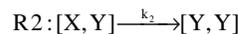
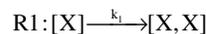
The initial multisets are:

$$\omega = \{nX, mY\}$$

where, n and m are integer multiplicities.

Since the system has single compartment, it only able to perform transformation of objects. Therefore, the transformation rule has the form: $R: [u] \xrightarrow{k} [v]$, where u, v are multisets in a compartment. k is a real number representing the kinetic constant, which represent the rate of reaction between objects.

The prey-predator population dynamics can be described by a simple set of rewriting rules (Manca, 2008):



R1, R2 and R3 are prey reproduction, predator reproduction and predator death rules, respectively.

Membrane computing model of ligand-receptor network of TGF- β : Ligand-Receptor network of TGF- β is two compartments system with 5 objects and 14 rules. The membrane system of Ligand Receptor Network of TGF- β (LRN) can be represented as:

$$LRN = (V, \mu, \omega_p, \omega_E, R_p, R_E)$$

The system contains two compartments that are Plasma membrane (P) and Endosome (E). Therefore its membrane structure can be represented as:

$$\mu = []_E]_P$$

The objects are receptors and complex of ligand and receptors. They are Receptor type I (RI), Receptor type II (RII) and ligand TGF- β . The ligand TGF- β formed complex with the receptors called ligand receptors complex (IRIRII). Two objects, D1 and D2 are introduced to accommodate rules that produce RI and RII respectively. The roles of these objects are merely to determine the continuous production of those receptors. Therefore the objects can be listed as follow:

$$V = \{RI, RII, IRIRII, D1, D2\}$$

The initial multisets are:

$$\omega_p = \{RI, RII, D1, D2\} \text{ and } \omega_E = \{IRIRII\}$$

The evolution rule has the form:

$$R_a : u[v]_a \xrightarrow{k} u'[v']_a$$

Where:

u, v, u', v' = Multisets and a is compartment
 k = A real number representing the kinetic constant

There are transformation as well as communication rules in LRN system. The rules used in the LRN are:

Ligand receptor complex formation:

$$R_{p1}: [RI, RII]_p \xrightarrow{k_1} [IRIRII]_p$$

Ligand receptor complex constitutive degradation:

$$R_{p2}: [IRIRII]_p \xrightarrow{k_2} []_p$$

Ligand independent complex degradation:

$$R_{p3}: [IRIRII]_p \xrightarrow{k_3} []_p$$

Ligand receptor complex internalization:

$$R_{p4}: IRIRII []_E \xrightarrow{k_4} [IRIRII]_E$$

RI synthesis:

$$R_{p5}: [D1]_p \xrightarrow{k_5} [D1, RI]_p$$

RI constitutive degradation:

$$R_{p6}: [RI]_p \xrightarrow{k_6} []_p$$

RI internalization:

$$R_{p7}: RI []_E \xrightarrow{k_7} [RI]_E$$

RI recycling:

$$R_{E1}: [RI]_E \xrightarrow{k_8} RI []_E$$

Ligand Receptor complex recycling:

$$R_{E2}: [IRIRII]_E \xrightarrow{k_9} IRIRII []_E$$

$$R_{p8}: [IRIRII]_p \xrightarrow{k_{10}} [RI, RII]_p$$

RII synthesis:

$$R_{p9}: [D2]_p \xrightarrow{k_{11}} [D2, RII]_p$$

RII constitutive degradation:

$$R_{p10}: [RII]_p \xrightarrow{k_{12}} []_p$$

RII internalization:

$$R_{p11}: RII []_E \xrightarrow{k_{13}} [RII]_E$$

RII recycling:

$$R_{E3}: [RII]_E \xrightarrow{k_{14}} RII []_E$$

Simulation of prey-predator population: The PP model of membrane computing is simulated with Gillespie simulator. Firstly, the information about the kinetic constants, initial objects in the population and the rules represented in the model are extracted into files respectively. The kinetic constants are obtained by considering the role of each rule in the system and its contribution to Gillespie algorithm. The kinetic constants k_1, k_2, k_3 are adjusted to 10, 0.02 and 15, respectively. The initial multiset is $\omega = \{1000X, 200Y\}$.

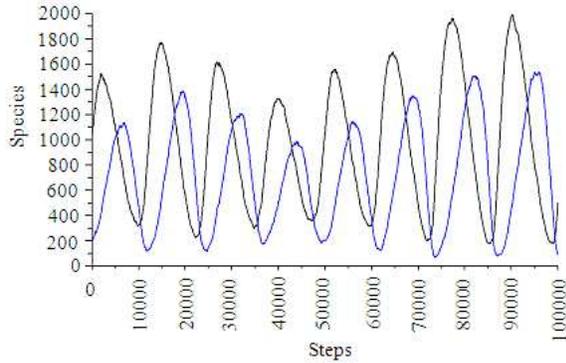


Fig. 3: Oscillation of the prey-predator model simulated with Gillespie algorithm ($k_1 = 10$, $k_2 = 0.02$, $k_3 = 15$)

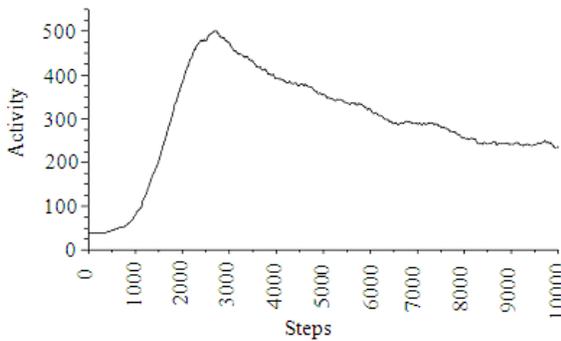


Fig. 4: Behavior of the model for typical trafficking rates

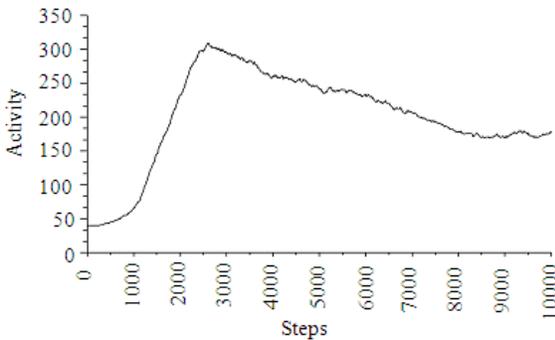


Fig. 5: Behavior of the model when rate constants for internalization and recycling are decreased

The simulation in Fig. 3 shows the oscillations of the prey-predator model simulated by Gillespie algorithm for 100,000 simulation steps. This result generated the oscillation behavior for the number of preys and predators in prey-predator model.

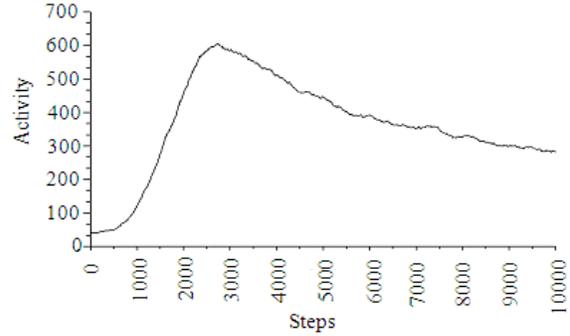


Fig. 6: Behavior of the model when rate constants for internalization and recycling are increased

Simulation of TGF- β : Membrane computing model of TGF- β is evaluated by using Gillespie algorithm. This experiment intended to analyze how different parameters affect the behavior of the TGF- β system by using multi-compartment Gillespie simulator. The signaling activity will peak or stop rising at certain steps according to the changes in the parameters. The parameters are extracted from the mathematical model build by Villar *et al.* (2006). The chosen initial multisets are, $\omega_p = \{1130RI, 1\ 130RII, 1D1, 1D2\}$ and $\omega_E = \{401RIRII\}$.

Figure 4 shows the behavior of the model for typical trafficking rates. In this model, internalization rate (k_i) represented by k_4 , k_7 and k_{13} , is 1/3; recycling rate (k_r) represented by k_8 , k_9 , k_{10} and k_{14} is 1/30; constitutive degradation rate (k_{cd}) represented by k_2 , k_6 and k_{12} is 1/36; ligand-induced degradation rate (k_{lid}) represented by k_3 is 1/4; complex formation rate (k_a) represented by k_1 is 0.01; synthesis rate (p_{RI} and p_{RII}) for k_5 and k_{11} is 8 and 4, respectively. The efficiency of recycling of active receptors rate, α is 1. The results show that signaling activity is peaking when the concentration of IRIRII in Endosome is around 500.

When the rate constants for internalization and recycling are decreased to 1/10 and 1/100 respectively, the peak of signaling activity is also decreased as shown in Fig. 5. The concentration of IRIRII in Endosome is around 300. Meanwhile, when rate constants for internalization and recycling are increased to 1 and 1/10, respectively, the peak of signaling activity is also increased as shown in Fig. 6. The concentration of IRIRII in Endosome is around 600.

The simulation in Fig. 7 and 8 show the behavior of the model when the efficiency of recycling of active receptors rate is decreased to 0.5. Figure 7 shows that when ligand-induced degradation rate is decreased to 0.01, the signaling activity is peaking when the concentration of IRIRII in Endosome is around 800.

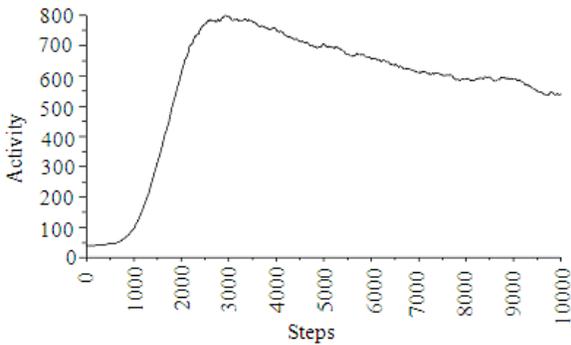


Fig. 7: Behavior of the model when rate constants for ligand-induced degradation and efficiency of recycling of active receptors are decreased

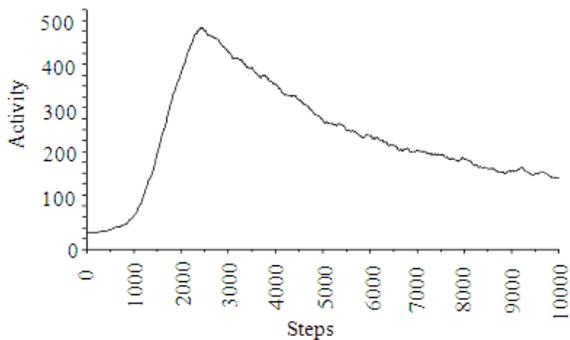


Fig. 8: Behavior of the model when recycling of active receptors is decreased

Meanwhile, with the same ligand-induced degradation rate as in Fig. 4, the signaling activity is peaking when the concentration of IRIRII in Endosome is around 500.

DISCUSSION

The investigations above demonstrates that, the PP model that described using differential equations is able to be formalized in membrane computing and this formalism could be used to produce similar results. This shows that Gillespie simulation strategy able to confine the membrane computing formalism that used to represent the dynamics of prey-predator population. Differential equations in PP are used to model kinetics of the reactions of two species. It would continuously vary the concentration of species in deterministic dynamics. In contrast, membrane computing takes into consideration the discrete character of the quantity of species in PP system by using rewriting rules. The inherent randomness in PP system is captured by using stochastic simulation strategy of Gillespie algorithm. Nevertheless, there are differences in performance between these approaches. The ODE simulation with

the ratio of initial objects is two predators to five preys, takes around 10 units time to complete a cycle in the oscillation. Meanwhile, the Gillespie simulation with a ratio of initial objects is one predator to five preys takes 15000 steps to complete a similar cycle. This shows that in the ODE approach, the large ratio in the mix of objects in the reaction and the deterministic feature make the reactions fast. And in the Gillespie simulation, more time are needed to measure the weight of each reaction based on the ratio in the mix of objects and to subsequently choose the appropriate reaction at each time step.

The simulation results of membrane computing model of TGF- β are compared to the simulation results of ODE model generated by Villar *et al.* (2006). The simulation of (a), (b), (c), (d) and (e) in Fig. 2 are compared to the simulation of membrane computing model in Fig. 4-8, respectively. The results show that approximately similar simulation results of ODE model could also be obtained using membrane computing model. The peak reached in the membrane computing model is almost similar to the ODE model. However, since objects are modeled as a global entity in the ODE model, the concentration of objects in each of the compartments could not be verified. On the contrary, the membrane computing model could determine the movement of objects from one compartment to another and with this the changes of concentration of objects in the specific compartments at each time step can be measured. Meanwhile, the ODE simulation takes around 200 time units to reach a peak and the Gillespie simulation takes around 2500 steps to reach the similar peak with same initial concentration of IRIRII in Endosome. This shows that more time needed to select a reaction in the stochastic approach at each time step. In the membrane computing model the numbers of simulation steps are almost similar to each of the investigation and it could not be adjusted as in the ODE model due to the limitations of Gillespie simulator which could not accommodate event objects. Event objects trigger the state changes in model when a specific event or condition is invoked. In this model, ligand can be considered as the event object in which the event at time 0 should change the ligand concentration from 3×10^{-5} to 0.01. However this element is not accommodated in the Gillespie simulator and due to this limitation the ligand concentration is fixed at 0.01 at all the time.

CONCLUSION

The experiments above show that Gillespie algorithm can capture the stochastic characteristics of

biological system represented by membrane computing. In the both case studies, the simulation of membrane computing model with Gillespie algorithm is able to produce the similar result as the ODE model. Moreover the membrane computing model is capable of preserving the stochastic elements of biological systems. This is not the case with the ODE which is deterministic in nature. This reinforce that the elements of biological system can be represented in a better way by using membrane computing model in order to be interpreted straightforwardly by biologist.

Since prey-predator population is a system with only a compartment and the objects are modeled as a global entity, the computation involved in membrane computing model are approximately similar to ODE model. Nonetheless, due to the stochastic characteristics of membrane computing model, more time is required to complete the simulation with membrane computing model as shown by prey-predator population. Meanwhile, the investigation of hierarchical system of two compartments in protein TGF- β shows the membrane computing model preserve the element and characteristics of hierarchical system which is ignored in the ODE model. In order to reduce the computation time in the simulation of membrane computing model due to its stochastic characteristics, the enhancement of the simulator to increase its performance should be dealt with. The limitation in Gillespie simulator that unable to capture features like event handling has contributed to the inability to accommodate the event as required by TGF- β . This is because the Gillespie simulator only able to capture the basic structure of biological systems. In the future research this limitations should be addressed to enhance the membrane computing simulator to capture other complex characteristics of biological system such as event handling.

REFERENCES

- Andrei, O., G. Ciobanu and D. Lucanu, 2005. Executable specifications of the P systems. Lecture Notes Comput. Sci., 3365: 127-146. <http://www.springerlink.com/content/8mmtwaj1jbyh5up2/fulltext.pdf>
- Arroyo, F., C. Luengo, A.V. Baranda and L.F. Mingo, 2003. A software simulation of transition P systems in Haskell. Lecture Notes Comput. Sci., 2597: 19-32. <http://portal.acm.org/citation.cfm?id=647270.721873>
- Baranda, A.B., J. Castellanos, F. Arroyo and R. Gonzalo, 2002. Towards an electronic implementation of membrane computing: A formal description of non-deterministic evolution in transition P systems. Lecture Notes Comput. Sci., 2340: 350-359. DOI: 10.1007/3-540-48017-X_33
- Ciobanu, G. and D. Paraschiv, 2002. Membrane software: A P system simulator. Fundamenta Inform., 49: 61-66. <http://psystems.disco.unimib.it/procwmc01/CIOBANU1.pdf>
- Ciobanu, G. and W. Guo, 2004. P systems running on a cluster of computers. Lecture Notes Comput. Sci., 2933: 123-139. DOI: 10.1007/b95207
- Cordon-Franco, A., M.A. Gutierrez-Naranjo, M.J. Perez-Jimenez and F. Sancho-Caparrini, 2004. A Prolog simulator for deterministic P systems with active membranes. New Gen. Comput., 22: 349-363. DOI: 10.1007/BF03037286
- Gillespie, D.T., 2001. Approximate accelerated stochastic simulation of chemically reacting systems. J. Chem. Phys., 115: 1716-1733. <http://users.soe.ucsc.edu/~msmangel/Gillespie01.pdf>
- Gillespie, D.T., 2003. Improved leap-size selection for accelerated stochastic simulation. J. Chem. Phys., 119: 8229-8234. <http://users.soe.ucsc.edu/~msmangel/GillespiePetzold03.pdf>
- Jones, D.S. and B.D. Sleeman, 2003. Differential Equations and Mathematical Biology. Chapman and Hall/CRC Press, London, UK., ISBN: 9781420083576, pp: 183.
- Kastner J.S. and S. Fraser, 2002. Modeling a hox gene network *in silico* using a stochastic simulation algorithm. Dev. Biol., 246: 122-131. PMID: 12027438
- Kierzek, A.M., J. Zaim and P. Zielenkiewicz, 2001. The effect of transcription and translation initiation frequencies on the stochastic fluctuations in prokaryotic gene expression. J. Biol. Chem., 276: 8165-8172. DOI: 10.1074/jbc.M006264200
- Manca, V., 2008. The metabolic algorithm for P systems: Principles and applications. Theor. Comput. Sci., 404: 142-155. DOI: 10.1016/j.tcs.2008.04.015
- McAdams, H.H. and A. Arkin, 1997. Stochastic mechanism in gene expression. Natl. Acad. Sci., 94: 814-819. PMID: 9023339.
- Muniyandi, R. and M.Z. Abdullah, 2009. Modeling of biological processes by using membrane computing formalism. Am. J. Applied Sci., 6: 1961-1969. DOI: 10.3844/ajassp.2009.1960.1968

- Noval, D.B., M.J. Perez-Jimenez and F. Sancho-Caparrini, 2003. A Mz Scheme implementation of transition P systems. *Lecture Notes Comput. Sci.*, 2597: 58-73. DOI: 10.1007/3-540-36490-0
- Paun, G., 1998. Computing with membranes. *J. Comput. Syst. Sci.*, 61: 108-143. DOI: 10.1006/jcss.1999.1693
- Paun, G., 2000. *Membrane Computing. An Introduction*. Springer-Verlag, Berlin, ISBN: 3540436014, pp: 1-11.
- Perez-Jimenez, M.J. and F.J. Romero-Campero, 2004. an efficient family of p systems for packing items into bins. *J. Uni. Com Sci.*, 10: 650-670. DOI: 10.3217/jucs-010-05-0650
- Perez-Jimenez, M.J. and F.J. Romero-Campero, 2006. P systems, a new computational modeling tool for systems biology. *Lecture Notes Comput. Sci.*, 4220: 176-197. DOI: 10.1007/11880646_8
- Pescini, D., D. Besozzi, G. Mauri, and C. Zandron, 2006. Analysis and simulation of dynamics in probabilistic P systems. *Lecture Notes Comput. Sci.*, 3892: 236-247. DOI: 10.1007/11753681_19
- Philips, A. and L. Cardelli, 2009. An Abstract Machine for the Stochastic Bioambient calculus. *Elect. Notes Theor. Comput. Sci.*, 227: 143-159. DOI: 10.1016/j.entcs.2008.12.109
- Priami, C., A. Regev, E. Shapiro and W. Silverman, 2001. Application of a Stochastic Name-Passing Calculus to Representation and Simulation of Molecular Processes. *Inform. Process. Lett.*, 80: 25-31. DOI: 10.1016/S0020-0190(01)00214-9
- Roberts, S., D. MacKinley and P. Humburg, 2009. Overview of ODE's for computational science. <http://comptlsci.anu.edu.au/Module-ODEs/ode-overview.pdf>
- Romero-Campero, F.J. and M. Gheorghe, 2007. Multicompartment Gillespie simulator in C. <http://www.dcs.shef.ac.uk/~marian/PSimulatorWeb/PSystemMF.htm>
- Romero-Campero, F.J. and M.J. Perez-Jimenez, 2008. A model of the quorum sensing system in vibrio fischeri using P systems *Artif. Life*, 14: 95-109. DOI: 10.1162/artl.2008.14.1.95
- Suzuki, Y. and H. Tanaka, 2000. On a LISP implementation of a class of P systems. *Roman. J. Inform. Sci. Technol.*, 3: 173-186. http://www.imt.ro/romjist/Volum3/Volum3_2/LISP.htm
- Villar, J.M.G., R. Jansen and C. Sander, 2006. Signal processing in the TGF- β superfamily ligand-receptor network. *PLoS Comput. Biol.*, 2: e3. DOI: 10.1371/journal.pcbi.0020003