Journal of Computer Science 6 (7): 669-678, 2010
ISSN 1549-3636
© 2010 Science Publications

Model Checking the Biological Model of Membrane Computing with Probabilistic Symbolic Model Checker by Using Two Biological Systems

[1]Ravie Chandren Muniyandi, [2]Abdullah Mohd. Zin and [1]Zarina Shukor
[1]School of Computer Science,
[2]School of Information Technology,
Faculty of Information Science and Technology,
University Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

Abstract: Problem statement: Membrane computing formalism has provided better modeling capabilities for biological systems in comparison to conventional mathematical models. Model checking could be used to reason about the biological system in detail and with precision by verifying formally whether membrane computing model meets the properties of the system. **Approach:** This study was carried to investigate the preservation of properties of two biological systems that had been modeled and simulated in membrane computing by a method of model checking using PRISM. The two biological systems were prey-predator population and signal processing in the legend-receptor networks of protein TGF-β. **Results:** The model checking of membrane computing model of the biological systems with five different properties showed that the properties of the biological systems could be preserved in the membrane computing model. **Conclusion:** Membrane computing model not only provides a better approach in representing and simulating a biological system but also able to sustain the basic properties of the system.

Key words: Membrane computing, model checking, biological system

INTRODUCTION

Membrane computing provides a hierarchical structure for molecular computation in which embraces play an essential role for objects to pass in a regulated fashion within and across the membranes. Since its inception in 1998, much research on theoretical aspects has been done to establish membrane computing computational power (Paun, 1998; 2000). In recent time, the research interest is more concentrated in applying the membrane computing formalism to solve real world problems. One of the attempts is using membrane computing capabilities in modeling biological systems. Studies of biological systems such as cells *in silico* have greatly reduced the need for expensive and prolonged lab experiments. Construction of a biological system into the membrane computing model provides a better understanding of the dynamics and functionality of the system. The biological description of membrane computing formalism has been utilized to characterize and preserve the elements in biological systems. The research in this line shows that biological system can be modeled better using membrane computing than the conventional methods using mathematical representation such as Ordinary Differential Equation (ODE) (Bernardini *et al*., 2006; Muniyandi and Abdullah, 2009; 2010). Simulation strategies based on membrane computing such as Gillespie Algorithm are also used to proof this claim.

Meanwhile, through simulations the operation of the model can be studied and consequently, the properties concerning the behavior of the biological system can be inferred as shown by Muniyandi and Abdullah (2010). However, although simulation outlines some key features, the preservations of other properties of model such as computability, understandability, extensibility and relevancy also need to be checked to make sure essential properties, dynamic behaviors, informal concepts and higher levels extensible capabilities of specific biological system are captured as well (Regev and Shapiro, 2004).

In comparison to deterministic models, stochastic models are more difficult to visualize and due to that it is necessary to reason about the systems in detail and with precision. Model checking is a verification procedure to verify formally whether a model meets the

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

properties stated in the specification. The system's specification is represented by using temporal logic formulas and symbolic algorithms are used to navigate the model to check if the specification is fulfilled or not.

In recent time with the rapid research developments in Systems Biology, the use of model checking for the analysis of biological models has attracted much attention. Some of the research are: The analysis of transcriptional regulation by LTL model checking (Barnat *et al.*, 2009); Performing statistical model checking using Bayesian sequential hypothesis Testing on biological systems (Sumit *et al.*, 2009); algorithmic algebraic model checking on metabolic networks (Mysore and Mishra, 2007); temporal logic analysis of Gene Networks (Batt *et al.*, 2007) and probabilistic model checking of complex biological pathways(Heath *et al.*, 2006; Kwiatkowska *et al.*, 2008).

There are also studies investigating the use of model checking for membrane computing models such as natural algebraic specification for the membrane computing (Andrei *et al.*, 2005) investigate various models of membrane computing to identify their model-checking problems (Dang *et al.*, 2005) and attempt to use PRISM (probabilistic and symbolic model checker) with a membrane computing model (Romero-Campero *et al.*, 2006).

In this study, we investigate whether the properties of two biological case studies modeled using membrane computing and simulated by Muniyandi and Abdullah (2010) are preserved by using the PRISM model checker. The two biological case studies are prey-predator population and signal processing in the legend-receptor Networks of protein TGF-β. These two case studies have been modeled in membrane computing formalism and simulated with simulation strategy of membrane computing with Gillespie Algorithm (Muniyandi and Abdullah, 2009; 2010).

MATERIALS AND METHODS

PRISM model checker: PRISM (Kwiatkowska *et al.*, 2002) is probabilistic model checker that represent a technique for formally verifying quantitative properties of a stochastic system. A model to be analyzed is specified in PRISM language, which is a simple, high level, state based language based on the Reactive Modules formalism (Alur and Henzinger, 1999). PRISM supports a probabilistic model based on Continuous Time Markov Chains (CTMC) and systems specifications through continuous stochastic logic for stochastic systems.

Case studies:

Prey-predator population: The primary example of a Prey-Predator population (Jones and Sleeman, 2003) is a system comprised of a plant population and an herbivorous animal dependent on that plant for food. In this case, the predator is totally dependent on the prey 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. Based on these behaviors, three rules are formulated in membrane computing as follows (k_1, k_2 and k_3 are kinetic constants):

$$R1:[X]\xrightarrow{k_1}[X,Y]$$

$$R2:[X,Y]\xrightarrow{k_2}[Y,Y]$$

$$R3:[Y]\xrightarrow{k_3}[]$$

Legend-receptor network of TGF-β: Transforming Growth Factor beta (TGF-β) (Villar *et al.*, 2006) is a type of protein that functions in cells. In signal transduction some cells secrete TGF-β and also generate receptors for TGF-β. The elements of this machinery incorporate the members of the two main receptor families called type I and type II receptors. Each legend induces the formation of a receptor complex with type I and type II receptors, which then signal through the channels. The capacity of most legends to bind several type I and type II receptors lead to a complex legend-receptor interaction network. Based on these behaviors, fourteen rules are formulated in membrane computing as follows ($k_1 \ldots k_{14}$ are kinetic constants).

Ligand receptor Complex Formation (CF):

$$R_p1:|RI,RII\,\underline{\bar{P}}|\xrightarrow{k_1}|RIRII\,\underline{\bar{P}}|$$

Ligand receptor complex Constitutive Degradation (CD1):

$$R_p2:|RIRII\,\underline{\bar{P}}|\xrightarrow{k_2}|\underline{\bar{P}}|$$

Ligand independent Complex Degradation (CD2):

$$R_p3:|RIRII\,\underline{\bar{P}}|\xrightarrow{k_3}|\underline{\bar{P}}|$$

Ligand receptor complex Internalization (I1):

$$R_p4:|IRIRII\,\underline{\bar{E}}|\xrightarrow{k_4}|RIRII\,\underline{\bar{E}}|$$

RI Synthesis (S1):

$$R_p5 :|\, D1\bar{\underline{P}}\,| \xrightarrow{k_5} |\, D1, RI\bar{\underline{P}}\,|$$

RI Constitutive Degradation (CD3):

$$R_p6 :|\, RI\bar{\underline{P}}\,| \xrightarrow{k_6} |\, \bar{\underline{P}}\,|$$

RI Internalization (I2):

$$R_p7 :|\, RI\bar{\underline{E}}\,| \xrightarrow{k_7} |\, RI\bar{\underline{E}}\,|$$

RI Recycling (RC1):

$$R_E1 :|\, RI\bar{\underline{E}}\,| \xrightarrow{k_8} |\, RI\bar{\underline{E}}\,|$$

Ligand Receptor Complex recycling (RC2):

$$R_E2 :|\, RIRII\bar{\underline{E}}\,| \xrightarrow{k_9} |\, lRIRII\bar{\underline{E}}\,|$$

$$R_P8 :|\, RIRII\bar{\underline{P}}\,| \xrightarrow{k_{10}} |\, RI, RII\bar{\underline{P}}\,|$$

RII Synthesis (S2):

$$R_P9 :|\, D2\bar{\underline{P}}\,| \xrightarrow{k_{11}} |\, D2, RII\bar{\underline{P}}\,|$$

RII Constitutive Degradation (CD4):

$$R_P10 :|\, RII\bar{\underline{P}}\,| \xrightarrow{k_{12}} |\, \bar{\underline{P}}\,|$$

RII Internalization (I3):

$$R_P11 :|\, RII\bar{\underline{E}}\,| \xrightarrow{k_{13}} |\, RII\bar{\underline{E}}\,|$$

RII Recycling (RC3):

$$R_E3 :|\, RII\bar{\underline{E}}\,| \xrightarrow{k_{14}} |\, RII\bar{\underline{E}}\,|$$

Properties: The properties for the Prey-Predator Population and TGF-β are obtained from the behavior of these systems elaborated in the respective research studies (Jones and Sleeman, 2003; Villar *et al*., 2006).

Properties of prey-predator population: Recurrence behavior of the system and the existence of equilibrium probability distribution maintain the stability of the system in the form of oscillations. To facilitate this behavior, the prey-predator system should preserve the following properties:

- The rules are selected stochastically based on the number of preys and predators at each time steps and the value of reaction constants to maintain the equilibrium of the system
- The number of preys and predators must not equal to 0 at any time steps
- The number of preys and predators become equal or intersect each other twice at each cycle of the system
- Percentage of increase or decrease of number of preys are higher most of the time steps than the number of predators
- Percentage of change between preys and predators are higher most of the time steps for preys than predators

Properties of TGF-β: The purpose of this model is to study the signal processing potential of the ligand-receptor network and receptor trafficking. The signaling activity is proportional to the number of legend-receptor complexes that are present in the internalized endosomes. In order to attain this behavior, the other essential properties required for this model are:

- Legends induce the formation of receptor complexes with type I and type II receptors
- Receptors and legend-receptor complexes can be present in two spatially distinct compartments: Plasma membrane and internalized endosomes
- Receptors and legend-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 legend-receptor complexes
- Receptor degradation has a legend-induced contribution, which affects only receptors that have been complexes with legends

The biological case studies modeled in membrane computing by Muniyandi and Abdullah (2010) are translated into PRISM formalism. This translation technique from membrane computing into PRISM applied in this research is proposed by (Romero-Campero *et al*., 2006). Then the model in PRISM is simulated and model checked with the properties for each of the case studies. PRISM is used to specify and to analyze properties based on rewards. Rewards are used to reason the behavior of the model in a certain fashion by measuring the probability as well as to identify a wide range of quantitative measures relating to modeling behavior. The data of the results generated by PRISM is exported into Scilab (Campbell *et al*., 2000) to be presented into a graph. Finally, the graph is analyzed to verify whether the identified properties are preserved or not.

RESULTS

Model checking of prey predator population: The membrane computing model of prey-predator population translated into PRISM formalism as follows:

```
ctmc

 // initial number of species
const int Prey_bound = Max;
const int Predator_bound = Max;

//Maximum number of Preys or Predators allowed in the//system
const int Max;

// base rates of reactions
const double k1 = 10;
const double k2 = 0.02;
const double k3 = 15;

// Prey-predator reactions
module PreyPredator

Prey : [0..Prey_bound] init 1000;
Predator : [0..Predator_bound] init 200;

[R1] Prey>0  &  Prey<Prey_bound  -> k1*Prey:
      (Prey' = Prey+1);
[R2] Prey>0 & Predator>0 &
      Predator<Predator_bound -> k2*Prey*Predator :
      (Prey'=Prey-1)& (Predator'=Predator+1);
[R3] Predator>0 & Predator<Predator_bound->
      k3*Predator : (Predator'=Predator-1);

endmodule
```

By using the concept of rewards, PRISM is used to specify and to analyze properties of Prey-Predator as follows:

Property 1: Rewards “reactions” is to analyze the stochastic behavior of the system to maintain the equilibrium. The rewarded value of 2, 4 and 6 refer to the selection of rules R1, R2 and R3 respectively at each time steps:

```
rewards “reactions”
    [R1] true: 2;
    [R2] true: 4;
    [R3] true: 6;
endrewards
```

The Fig. 1 shows four graphs indicating the selection of rules at four different periods of time steps based on the rewards. These graphs show that at each time step one of the three rules is selected stochastically. The patterns of selections differ for each period of time steps as shown by each graph. This means that the stochastic behavior of the system is maintained to make sure the stability and consistency of the system at each cycle of the oscillation.

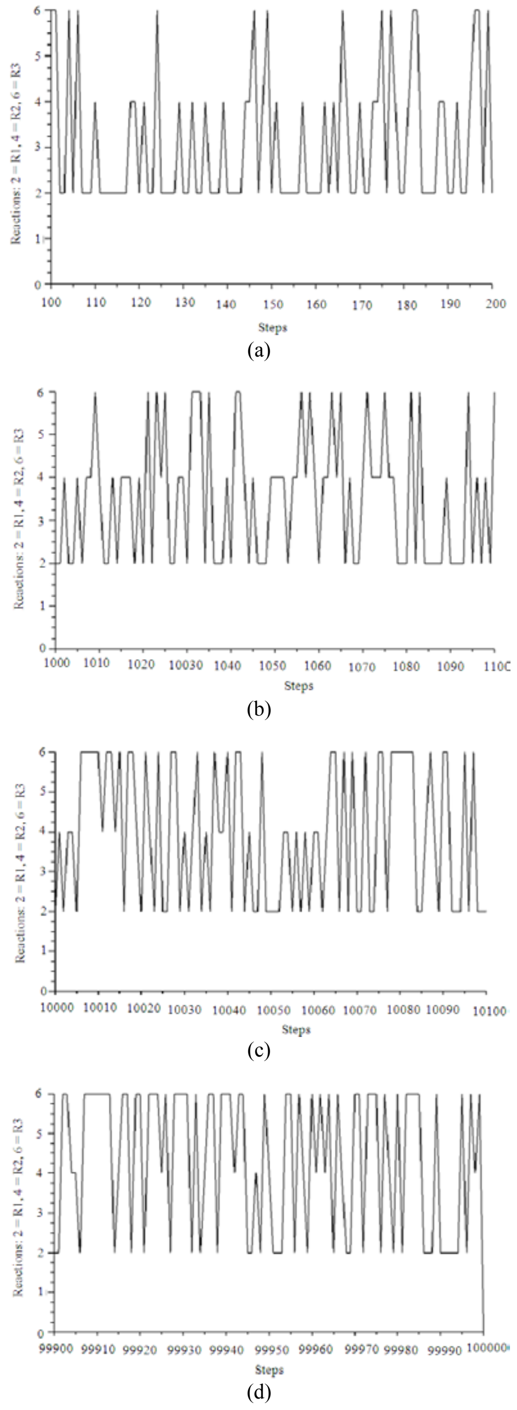


Fig. 1: Stochastic behavior of prey-predator model for four different time steps: (a) 100-200, (b) 1000-1100, (c) 10000-10100, (d) 99900-100000

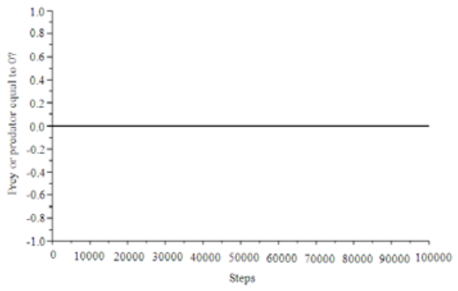


Fig. 2: Number of preys and predators greater than 0

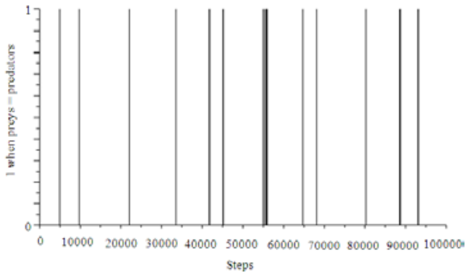


Fig. 3: Intersection between prey and predator

Property 2: Rewards "PreyPredatorgreater0" is to verify that the number of preys or predators is always greater than 0 throughout the period of oscillation. The rewarded value of 1, 2 and 3 are assigned to each of the following conditions respectively, when the condition is true at each time steps: Both prey and predator equal to 0; prey greater than 0 and predator equal to 0 and prey equal to 0 and predator greater than 0. If the three conditions are not met, then the rewarded value is 0:

```
rewards "PreyPredatorgreater0 "
            Prey = 0 & Predator = 0: 1;
            Prey>0 & Predator = 0: 2;
            Prey = 0 & Predator>0: 3;
endrewards
```

Figure 2 shows that not at once along the simulation steps of the system, the number of prey or predator has been equal to 0 based on the rewards. This result demonstrates that the behavior of the system is always consistent to make sure that the number of prey and predator must always greater than 0 to stabilize the system.

Property 3: Rewards "Intersection" is to determine the expected number of intersection at each cycle of the oscillation. Rewarded value of 1 is generated when the number of preys and predators are equal that is there is an intersection between prey and predator:

```
rewards "Intersection"
     Prey = Predator: 1;
endrewards
```

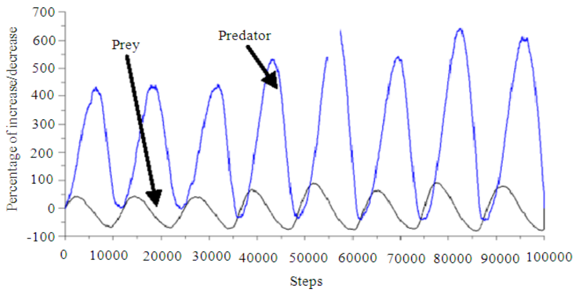


Fig. 4: Percentage of increase/decrease of preys and predators

Figure 3 demonstrates the time steps when number of prey is equal to number of predator generated by the rewards. The intersection between prey and predator occurs twice at each cycle in the oscillations when the prey and predator either keep decreasing or increasing. However the graph shows that the period of occurrence of one intersection to another is not similar. This is mainly due to the stochastic behavior of the system. However the pattern of intersection in each of the cycle in the oscillation is preserved to maintain the stability of the system.

Property 4: Rewards "PreyIncreaseDecrease" and "PredatorIncreaseDecrease" are to measure the percentage of increase or decrease of prey and predator from their respective initial amount, at each time step:

```
rewards "PreyIncreaseDecrease"
            true: ((Prey-1000)/1000)*100;
endrewards
rewards "PredatorIncreaseDecrease"
            true: ((Predator-200)/200)*100;
endrewards
```

Figure 4 illustrates the percentage of increase and decrease of prey and predator based on the rewards. The graph shows that the percentage of increase and decrease of predator is higher than the percentage of increase and decrease of prey. At the initial state, the number of prey is five times higher than the number of predator. The sharp increase of predator is to certain extent decrease the population of prey. Meanwhile, the sharp decrease of predator population gives some space for prey to increase its population to attain the initial level over again. The percentage of increase/decrease of prey and predator is almost similar at each of the cycle of the oscillation. This demonstrates that the equilibrium of prey-predator population has been preserved by maintaining the percentage of increase/decrease of prey and predator accordingly at each time step.

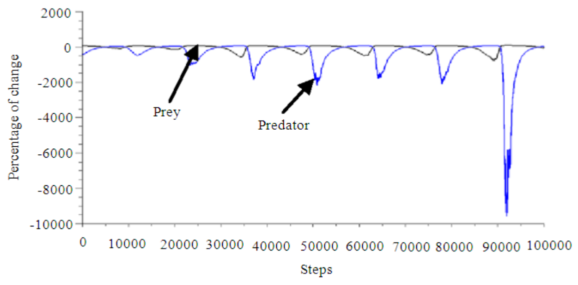


Fig. 5: Percentage of changes between preys and predators

Property 5: Rewards "PreyChanges" and "PredatorChanges" are to measure the percentage of differences between the amount of preys and predators in the population, at each time step:

```
rewards "PreyChanges"
    true: ((Prey-Predator)/Prey)*100;
endrewards

rewards "PredatorChanges"
    true: ((Predator-Prey)/Predator)*100;
endrewards
```

Figure 5 shows the percentage of changes of prey and predator compared to the opposite population as outlined in the rewards. This graph shows that at most of the time number of prey exceeds number of predator. But at certain period when number of predator is above the number of prey, prey is decreasing to control the increase of predator. When number of predator is decreasing, the number of prey is increasing. As shown in the Fig. 4, Prey Predator these results also show that the stability of the system is preserved by controlling the increase and decrease of the prey and predator accordingly at each time step.

Model checking of TGF-β: The membrane computing model of TGF-β translated into PRISM formalism as follows:

```
ctmc

// initial number of molecules
const int RI1_bound=2*N;
const int RI2_bound=N;
const int RII1_bound=2*N;
const int RII2_bound=N;
const int IRIRII1_bound=N;
const int IRIRII2_bound=2*N;

const int N;
// base rates of reactions
const double alpha =1;
const double R1 =0.01;
const double R2 =0.0277777778;
const double R3 =0.25;
const double R4 =0.3333333333333;
const double R5 =8.0;
const double R6 =0.0277777778;
const double R7 =0.3333333333333;
const double R8 =4.0;
const double R9 =0.0277777778;
const double R10=0.3333333333333;
const double R11=0.0333333333333333;
const double R12=0.0333333333333333;
const double R13=0.0333333333333333;

module plasma_membrane

RI1 : [0..RI1_bound] init 1130;
RII1 : [0..RII1_bound] init 1130;
IRIRII1 : [0..IRIRII1_bound] init 0;

[CF] RI1>0 & RII1>0 & IRIRII1<IRIRII1_bound ->
    R1*RI1*RII1*alpha: (RI1'=RI1-1) & (RII1'=RII1-
    1) & (IRIRII1'=IRIRII1+1);
[CD1] IRIRII1>0 ->R2*IRIRII1: (IRIRII1'= IRIRII1-1);
[CD2] IRIRII1>0 ->R3*IRIRII1: (IRIRII1'= IRIRII1-1);
[CD3] RI1>0 ->R6*RI1: (RI1'= RI1-1);
[CD4] RII1>0 ->R9*RII1: (RII1'= RII1-1);
[S1] RI1<RI1_bound ->R5: (RI1'= RI1+1);
[S2] RII1<RII1_bound ->R8: (RII1'= RII1+1);
[I1] RI1>0 ->R7*RI1: (RI1'= RI1-1);
[I2] RII1>0 ->R10*RII1: (RII1'= RII1-1);
[I3] IRIRII1>0 ->R4*IRIRII1: (IRIRII1'= IRIRII1-1);
[RC1] RI1<RI1_bound ->1: (RI1'= RI1+1);
[RC2] RI1<RI1_bound & RII1<RII1_bound ->1:
      (RI1'= RI1+1) & (RII1'= RII1+1);
[RC3] RII1<RII1_bound ->1: (RII1'= RII1+1);

endmodule

module endosome

RI2 : [0..RI2_bound] init 0;
RII2 : [0..RII2_bound] init 0;
IRIRII2 : [0..IRIRII2_bound] init 40;

[I1] RI2<RI2_bound ->1: (RI2'= RI2+1);
[I2] RII2<RII2_bound ->1: (RII2'= RII2+1);
[I3] IRIRII2<IRIRII2_bound ->1:
    (IRIRII2'= IRIRII2+1);
[RC1] RI2>0 ->R11*RI2: (RI2'= RI2-1);
[RC2] IRIRII2>0 ->R12*IRIRII2: (IRIRII2'= IRIRII2-
      1);
[RC3] RII2>0 ->R13*RII2: (RII2'= RII2-1);

endmodule
```

By using the concept of rewards, PRISM is used to specify and to analyze properties of TGF-â as follows:

Property 1: Rewards "CF" is to measure the process of legends inducement in the formation of receptor complexes (IRIRII1) with type I (RI) and type II (RII) receptors. This process is performed by reaction CF in the PRISM model and rewards "CF" measures the concentration of IRIRII1 as the product of this reaction activation at each time step:

```
rewards "CF"
    [CF] true:IRIRII1;
endrewards
```

Figure 6 shows the execution of reaction CF based on rewards "CF". At the initial stage this reaction is activated more regularly to meet the formation of IRIRII in plasma membrane. The increasing amount of IRIRII at this stage shows that internalization into endosomes is occurring consistently but slowly. This shows by the slow increase prey predator of IRIRII in endosomes as shown in the simulation done by Muniyandi and Abdullah (2010). Internalization is rapidly performed between time steps of 2000 and 3000 in which period the IRIRII in endosome achieve the peak. After this process, the formation of IRIRII in plasma membrane is stabilized with the combination of IRIRII recycling into plasma membrane and IRIRII internalization into endosomes.

Property 2: The following rewards are to measure receptors and ligand-receptor complexes availability in Plasma Membrane (PM) and Endosome (E):

```
rewards "RIAvailability_PM"
        true: RI1;
endrewards
rewards "RIAvailability_E"
        true: RI2;
endrewards
rewards "RIIAvailability_PM"
        true: RII1;
endrewards
rewards "RIIAvailability_E"
        true: RII2;
endrewards
rewards "IRIRIIAvailability_PM"
        true: IRIRII1;
endrewards
rewards "IRIRIIAvailability_E"
        true: IRIRII2;
endrewards
```

Figure 7 and 8 demonstrates the availability of receptors and legend-receptor complexes in plasma membrane and endosomes as outlined in the rewards. This result shows that the receptors and legend receptor complexes are always available in both compartments.

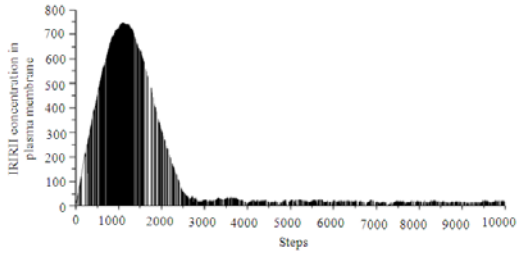


Fig. 6: The formation of IRIRII in plasma membrane

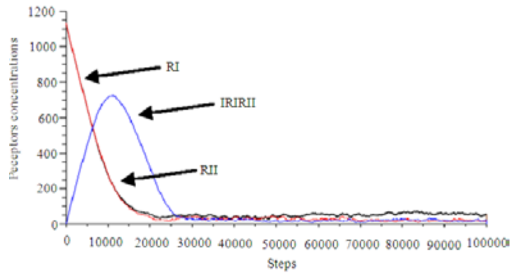


Fig. 7: Receptors and legend-receptor complexes in plasma membrane

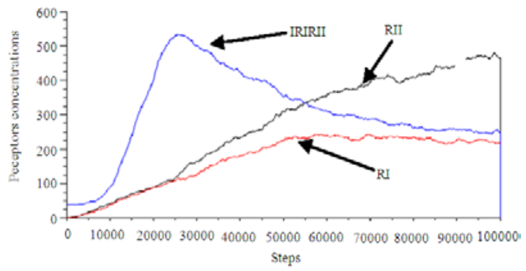


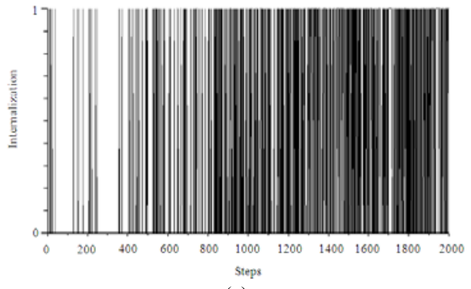
Fig. 8: Receptors and legend-receptor complexes in endosomes

However there is interaction between objects in these two compartments to generate IRIRII in endosomes to measure signal processing. This can be seen when RI and RII are internalized from plasma membrane, the concentration of these receptors is keep increasing in endosomes. So is for IRIRII until it achieves the peak. But after the peak, the concentration of receptors and legend receptor complexes are stabilized to a certain level with recycling and internalization activity.

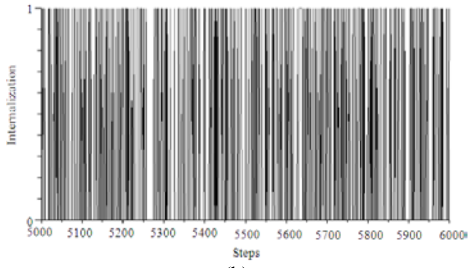
Property 3 and 4: Rewards "IN" is to measure the internalization of receptors and legend-receptor complexes into endosomes. The reactions I1, I2 and I3 involve in this process. Rewards "RC" is to measure the recycling back of receptors and legend-receptor complexes into the plasma membrane. The reactions RC1, RC2 and RC3 involve in this process:

```
rewards "IN"
      [I1] true: 1;
      [I2] true: 1;
      [I3] true: 1;
endrewards
rewards "RC"
      [RC1] true: 1;
      [RC2] true: 1;
      [RC3] true: 1;
endrewards
```

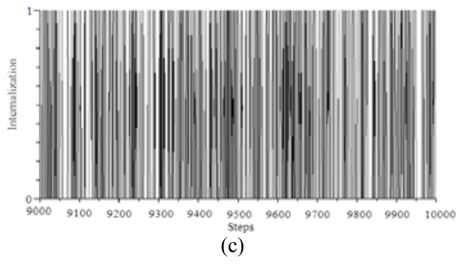
Figure 9 and 10 shows receptors and ligand-receptor complexes are continuously internalized into endosomes and recycled back to plasma membrane based on the rewards. This result shows that the internalization activity is slow at the initial stage before becoming intense at the period of achieving the peak for IRIRII in endosomes. Meanwhile there is hardly any recycling activity at the initial stage but this activity gain momentum after IRIRII in endosomes achieve the peak. At last both activities become constant to stabilize the system.



(a)



(b)

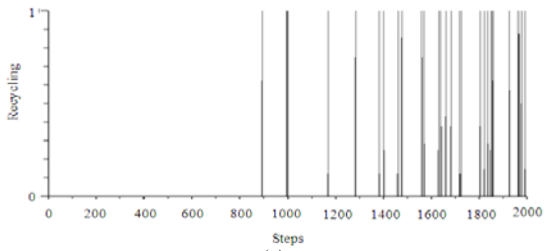


(c)

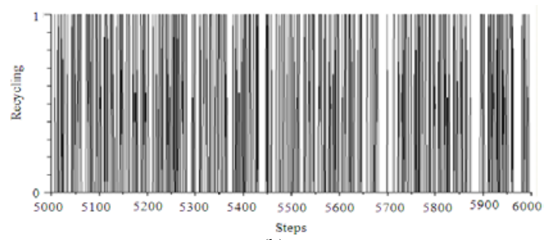
Fig. 9: Receptors and legend-receptor complexes internalization into endosomes for three different time steps; (a): 0-2000; (b): 5000-6000; (c): 9000-10000

Property 5: The following rewards are to measure receptor degradation. There are two purposes in these rewards. The first is to measure the constitutive contribution as result of this process for the receptors and legend-receptor complexes. This process is indicated by reactions CD1, CD2, CD3 and CD4. The second is to measure the legend-induced contribution, which affects only receptors that have been complexes with legends. This process is signified by reactions CD1 and CD2:

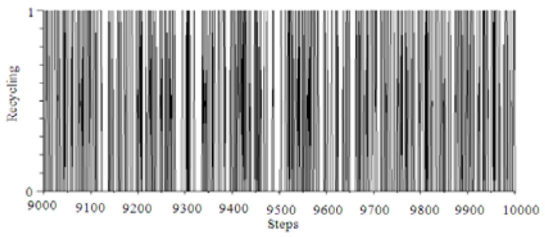
```
rewards "CD1"
      [CD1] true: 1;
endrewards
rewards "CD2"
      [CD2] true: 1;
endrewards
rewards "CD3"
      [CD3] true: 1;
endrewards
rewards "CD4"
      [CD4] true: 1;
endrewards
```



(a)



(b)



(c)

Fig. 10: Recycling of receptors and legend-receptor complexes into plasma membrane for three different time steps; (a): 0-2000; (b): 5000-6000; (c): 9000-10000

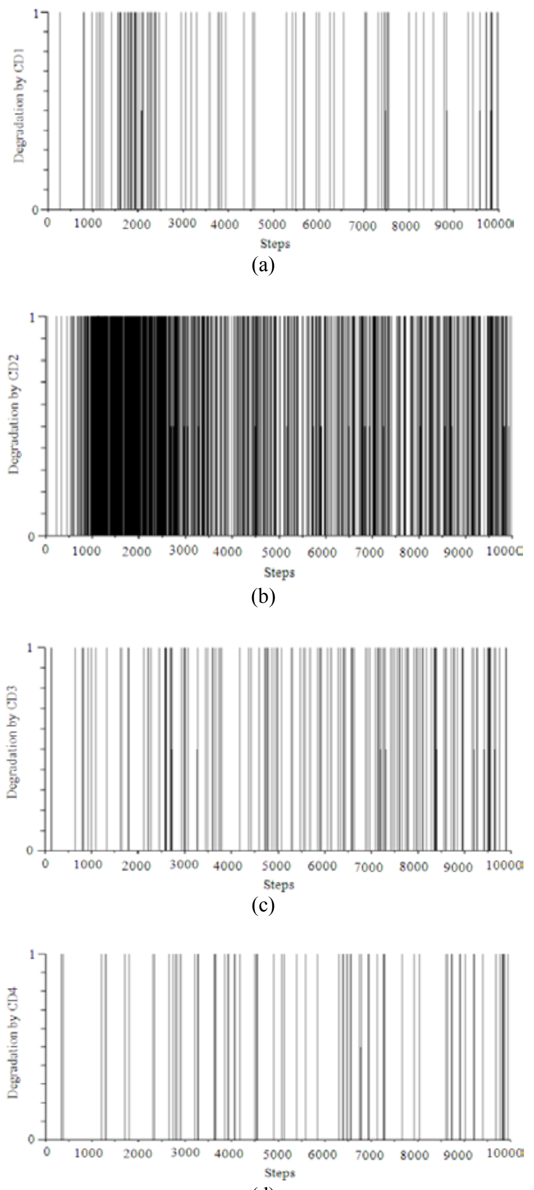


Fig. 11: Constitutive contribution and legend-induced contribution caused by receptors degradation for reaction: (a) CD1; (b): CD2; (c): CD3; (d): CD4

Figure 11 demonstrates the reactions that activate receptors degradation based on the rewards. The graph for CD1, CD3 and CD4 shows that receptor degradation has a constitutive contribution, which is the almost similar for the receptors and legend-receptor complexes. And graph CD2 shows that receptor degradation has a ligand-induced contribution that affects legend-receptor complexes.

DISCUSSION

Membrane computing has shown that it is capable of abstracting the structure and processes of biological system to represent them in a formal way without disregarding its biological characteristics. This allows the biological systems to be modeled in better way to counter some of the limitations in the conventional method based on ordinary differential equations. Moreover the membrane computing model is capable of preserving the stochastic elements of biological systems which is absent in the ODE model.

The results of the model checking of membrane computing models for two biological systems demonstrates that the properties of the system has been preserved in membrane computing model. The model checking with Prism shows that the properties of Prey-Predator Population have been preserved in the membrane computing model by maintaining the stability and consistency of the system. This means that the stochastic behavior of membrane computing has conserved the equilibrium of the oscillation of the Prey-Predator system by controlling the growth and reduction of the population of Preys and Predators accordingly. The model checking of Legend-Receptor Networks of protein TGF-β shows the formation, degradation, internalization, recycling and synthesis processes in the system have been activated according to the properties of the system in the membrane computing model. This enables the system to perform the signaling activity according to the number of ligand-receptor complexes in the internalized endosomes.

CONCLUSION

This study demonstrates that the non-determinism and stochastic behavior of membrane computing capable in preserving the properties of the biological system that had been modeled using deterministic approach of ODE. This reinforces that the membrane computing model not only provide better approach in representing and simulating biological system but also able to sustain the properties of the original model.

REFERENCES

Alur, R. and T.A. Henzinger, 1999. Reactive modules. Formal Methods Syst. Des., 15: 7-48. DOI: 10.1023/A: 1008739929481

Andrei, O., G. Ciobanu and D. Lucanu, 2005. Executable specifications of the P systems. Lecture Notes Comput. Sci., 3365: 127-146.

Barnat, J., L. Brim, I. Erna, S. Draan, J. Fabrikova and D. afranek, 2009. On algorithmic analysis of transcriptional regulation by LTL model checking. Theor. Comput. Sci., 410: 3128-3148. DOI: 10.1016/j.tcs.2009.02.017

Batt, G., C. Belta and R. Weiss, 2007. Temporal logic analysis of gene networks under parameter uncertainty. Proceeding of the 10th International Workshop on Hybrid Systems: Computation and Control, Apr. 3-5, Pisa, Italy, pp: 215-229. DOI: 10.1109/TAC.2007.911330

Bernardini, F., M. Gheorghe, N. Krasnogor, R.C. Muniyandi, M.J. Perez-Jimenez and F.J. Romero-Campero, 2006. On P systems as a modeling tool for biological systems. Lecture Notes Comput. Sci., 3850: 114-133. DOI: 10.1.1.144.5278

Campbell, S.L., J.P. Chancelier and R. Nikoukhah, 2000. Modeling and Simulation in Scilab/Scicos. 1st Edn., Springer, USA., ISBN: 10: 0-387-27802-8, pp: 313.

Dang, Z., O.H. Ibarra, C. Li and G. Xie, 2005. On model-checking of P systems. Lecture Notes Comput. Sci., 3699: 82-93. DOI: 10.1007/11560319

Heath, J., M. Kwiatkowska, G. Norman, D. parker and O. Tymchyshyn, 2006. Probabilistic model checking of complex biological pathways. Lecture Notes Comput. Sci., 4210: 32-47. DOI: 10.1007/11885191

Jones, D.S. and B.D. Sleeman, 2003. Differential Equations and Mathematical Biology. 1st Edn., Chapman and Hall/CRC Press, London, UK., ISBN: 9781420083576, pp: 183.

Kwiatkowska, M., G. Norman and D. Parker, 2002. PRISM: Probabilistic symbolic model checker. Lecture Notes Comput. Sci., 2324: 113-140. DOI: 10.1007/3-540-46029-2_13

Kwiatkowska, M., G. Norman and D. Parker, 2008. Using probabilistic model checking in systems biology. Assoc. Comput. Mach., 35: 14-21. DOI: 10.1145/1364644

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

Muniyandi, R. and M.Z. Abdullah, 2010. Experimenting the simulation strategy of membrane computing with Gillespie algorithm by using two biological case studies. J. Comput. Sci., 6: 525-535.

Mysore, V. and B. Mishra, 2007. Algorithmic algebraic model checking IV: Characterization of metabolic networks. Lecture Notes Comput. Sci., 4545: 170-184. DOI: 10.1007/978-3-540-73433-8_13

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. 1st Edn., Springer-Verlag, Berlin, ISBN: 3540436014, pp: 419.

Regev, A. and E. Shapiro, 2004. The π-Calculus as an Abstraction for Bimolecular Systems. In: Modeling in Molecular Biology, Ciobanu, G. and G. Rozenberg (Eds.). Springer, Berlin, ISBN: 978-3-540-40799-7, pp: 219-266.

Romero-Campero, F.S, M. Gheorghe, L. Bianco, D. Pescini, M.J. Perez-Jimenez and R. Ceterchi, 2006. Towards probabilistic model checking on P systems using PRISM. Lecture Notes Comput. Sci., 4361: 477-495. DOI: 10.1007/11963516_30

Sumit, K.J., M.C. Edmund, J.L. Christopher, A. Legay, A. Platzer and P. Zuliani, 2009. A Bayesian approach to model checking biological systems. Lecture Notes Comput. Sci., 5688: 218-234. DOI: 10.1007/978-3-642-03845-7_15

Villar, J.M.G., R. Jansen and C. Sander, 2006. Signal Processing in the TGF-β super family legend-receptor network. PLoS Comput. Biol., 2: 3. DOI: 10.1371/journal.pcbi.0020003