

Implementing Mitogen Activated Protein Kinases Cascade on Membrane Computing Using P-Lingua

Bareq Shaalan and Ravie Chandren Muniyandi

Center of Software Technology and Management, Faculty of Information Science and Technology,
University Kebangsaan Malaysia, 43000 Bangi, Selangor, Malaysia

Article history

Received: 26-06-2014

Revised: 08-07-2014

Accepted: 07-08-2014

Corresponding Author:

Bareq Shaalan

Center of Software Technology
and Management, Faculty of
Information Science and
Technology, University
Kebangsaan Malaysia, 43000
Bangi, Selangor, Malaysia
Email: ravie@ukm.edu.my

Abstract: The Mitogen Activated Protein Kinases (MAPK) cascade has been used as a case study for different computational software tools in both modelling and simulation of real-life problems. This study focuses on the development of new techniques for solving a MAPK cascade by using membrane computing. Membrane computing is an unconventional computational approach that provides a platform for modelling discrete system. This approach deals with parallel, distributed and non-deterministic computing models. P-Lingua is used to specify and analyze a wide range of quantitative properties and to offer a general syntactic framework that could be a formal standard for membrane computing. The model is simulated by using MeCoSim, a membrane computing simulator used to verify and validate the model. This study aimed to compare the use of the membrane computing approach for modelling the MAPK cascade with other models. The MAPK cascade, which is specified in P Lingua evaluated based on the simulation with MeCoSim. In general, we can say that the membrane computing model is better than previous models because it takes into account the changes that occur in the cell and because the problem is mainly a problem in the cells; therefore the model achieves the best results and reliability.

Keywords: Membrane Computing, P-Lingua, MeCoSim, MAPK Cascade

Introduction

Membrane computing is an area within computer science that seeks to discover new computational models for the study of biological cells, particularly the cellular membranes (Muniyandi and Zin, 2011). It is a sub-task of creating a cellular model and it deals with distributed and parallel computing models, processing multiple sets of symbolic objects in a localized manner. Thus, evolution rules allow evolving objects to be encapsulated into compartments defined by membranes.

The communications between compartments and with the environment play an essential role in the processes. The various types of membrane systems are known as P systems, after Gheorghe Păun, who first conceived the model (Păun, 2006). The approach of membrane computing used here is the one considered in the P-Lingua project, which is able to define P systems

within the active membrane model with division rules; this approach also provides software tools for compilation, simulation and debugging tasks.

Membrane computing is considered as an alternative to address the rules of the Mitogen Activated Protein Kinases (MAPK) cascade by taking into consideration its essential features that are of interest for biological applications. Membrane computing is a theoretical paradigm that abstracts from the structure and functionality of the living cell to create models and simulators of cellular phenomena (Muniyandi *et al.*, 2010); hence, it produces formal specifications defined in a language that are close to molecular biology and returns meaningful and easily interpreted information to biologists (Muniyandi and Zin, 2011).

Moreover, membrane computing could also represent the stochastic and nondeterministic characteristics of dynamical biological systems in a discrete way

(Muniyandi *et al.*, 2013). In this investigation a biological process in a MAPK cascade is analysed by modelling with membrane computing and the results are compared with the other approaches in order to determine the capability of membrane computing as a modelling tool for biological processes.

The outline of the paper is as follows. In the next section, the MAPK cascade is visualized as a three-tier cascade, structured in membrane computing by using its language (P-Lingua) (Diaz-Pernil *et al.*, 2009) and compared with two other models: A non-deterministic model (PRISM) (Kwiatkowska *et al.*, 2008) and a Deterministic model (ODE) (Pan and Mhatre, 2013). The conclusions from the current work are summarized in section 4 and this study aims to solve the problem of the MAPK cascade in membrane computing using P-Lingua as the specific programming language. The Gillespie Algorithm (Gillespie, 1977) or Stochastic Simulation Algorithm (SSA) is used in this programming language using stochastically by using direct way and then the MAPK cascade is simulated in MeCoSim, which is a general purpose software tool for simulating biological phenomena by means of P systems.

Materials and Methods

MAPK

The MAPK is involved in a pathway through which information is sent to the nucleus. It is one of the most important signalling pathways, playing a pivotal role in the molecular signalling that governs the growth, proliferation and survival of many cell types. The MAPK cascade consists of a MAPK Kinase Kinase (MAPKKK), a MAPK Kinase (MAPKK) and a MAPK. The cascade is initialized through the phosphorylation of MAPKKK, which then activates MAPKK through phosphorylation at two serine residues. This then activates MAPK through phosphorylation at threonine and tyrosine residues. The initialization of the pathway can be caused by a diverse set of stimuli including growth factors, neurotransmitters and cytokines. Figure 1 gives an overview of the structure of the pathway and Fig. 2 details the reactions that form the cascade (Huang and Ferrell, 1996).

Huang and Ferrell (1996) proposed the first model, where they revealed the ultra-sensitivity of the signalling cascade. Two models that showed a two-collision, distributive mechanisms for the activating dual phosphorylation of MAPK were proposed by (Orton *et al.*, 2014). They suggested that after phosphorylating the first site, MAPKK dissociates and then again rebinds to add a phosphate to the second

site. An increasing number of MAPK cascade models have been developed (Pan and Mhatre, 2013).

The motivation for the current study was twofold: The vast range of applications of the MAPK pathway and progress in modelling are affected by overpowering complexity rather than there being a simple modular concept of the signalling cascade. In standard molecular biotechnology, the MAPK pathway is perhaps the most rigorously studied pathway (Chang and Karin, 2001). The pathway is also vital, owing to its regulation of the growth, proliferation, differentiation and apoptosis (cell-death) of the major cell. The MAPK pathway can also be modelled to capture the *in vivo* metabolic complexities associated with abnormal cellular growth such as cancer (Ernst *et al.*, 2001). It may also act as a suitable drug target. It is therefore scientifically important and challenging to model this pathway and to get a better understanding of all these cellular machineries.

The ODE and Prism Models for MAPK

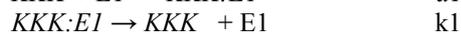
The general layout of the MAPK pathway consists of three kinases/phosphatase cycles built into a three-tiered cascade (Fig. 2a), consisting of a MAPK, which is activated via phosphorylation (Fig. 2b) by a MAPKK/MKK (MAPK kinase), which in turn is phosphorylated by a MAPKKK/MKKK (MAPKK kinase).

The rules 1-10 represent the reactions of the MAPK cascade, which are shown schematically in Fig. 1. They have used Goldbeter and Koshland's nomenclature for the rate constants: The letter *a* denotes association, *d* denotes dissociation without catalysis and *k* denotes product formation (Goldbeter and Koshland, 1981). KKK denotes MAPKKK, KK denotes MAPKK and K denotes MAPK.

The MAPK cascade is predicted to exhibit ultra-sensitivity, with the degree of ultra-sensitivity increasing as the cascade is descended.

This behaviour is robustly predicted for a wide range of assumed concentrations and K_m values (meaning concentrations of *a*, *d* and *k*) for the cascade enzymes and reactions, although the exact extent of the predicted ultra-sensitivity varies as the assumed values are varied.

1. MAPKKK activated by enzyme E1



2. MAPKKK deactivated by enzyme E2



$KKK_;E2 \rightarrow KKK + E2$	$k2$
3. MAPKK activated by MAPKKK	
$KK + KKK_ \rightarrow KK:KKK$	$a3$
$KK + KKK_ \leftarrow KK:KKK$	$d3$
$KK:KKK_ \rightarrow KK-P + KKK$	$k3$
4. MAPKK-P deactivated by MAPKK	
<i>phosphatase</i>	
$KK-P + KK-Ptase \rightarrow KK-P:KK$	$a4$
$KK-P + KK-Ptase \leftarrow KK-P:KK$	$d4$
$KK-P:KK-Ptase \rightarrow KK + KK$	$k4$
5. MAPKK-P activated by MAPKKK	
$KK-P + KKK_ \rightarrow KK-P:KKK$	$a5$
$KK-P + KKK_ \leftarrow KK-P:KKK$	$d5$
$KK-P:KKK_ \rightarrow KK-PP + KKK$	$k5$
6. MAPKK-PP deactivated by MAPKK	
<i>phosphatase</i>	
$KK-PP + KK-Ptase \rightarrow KK-PP:KK$	$a6$
$KK-PP + KK-Ptase \leftarrow KK-PP:KK$	$d6$
$KK-PP:KK-Ptase \rightarrow KK-P + KK$	$k6$
7. MAPK is activated by MAPKK-PP	
$K + KK-PP \rightarrow K:KK-PP$	$a7$
$K + KK-PP \leftarrow K:KK-PP$	$d7$
$K:KK-PP \rightarrow K-P + KK-PP$	$k7$
8. MAPK-P deactivated by MAPK	
<i>phosphatase</i>	
$K-P + K-Ptase \rightarrow K-P:K-Ptase$	$a8$
$K-P + K-Ptase \leftarrow K-P:K-Ptase$	$d8$
$K-P:K-Ptase \rightarrow K + K-Ptase$	$k8$
9. MAPK-P activated by MAPKK-PP	
$K-P + KK-PP \rightarrow K-P:KK-PP$	$a9$
$K-P + KK-PP \leftarrow K-P:KK-PP$	$d9$
$K-P:KK-PP \rightarrow K-PP + KK-PP$	$k9$
10. MAPK-P deactivated by MAPK	
<i>phosphatase</i>	
$K-PP + K-Ptase \rightarrow K-PP:K-Ptase$	$a10$
$K-PP + K-Ptase \leftarrow K-PP:K-Ptase$	$d10$
$K-PP:K-Ptase \rightarrow K-P + K-Ptase$	$k10$

where, A1-A10 = 1 and D1-D10 = 150 and K1-K10 = 150.

P-Lingua

P-Lingua is intended to become a standard for P systems definition and it could also consider different variants of P systems. P-Lingua 2.0 is presented as a framework to define cell-like P system models, including several algorithms to simulate P system computations for the supported models (from now on,

simulators), as well as different formats to represent P systems with associated parsers to translate from each other (Diaz-Pernil *et al.*, 2009). The supported P system models in P Lingua 2.0 are Transition P systems, Symport/Antiport P systems, P systems with active membranes with membrane division and membrane creation rules, Probabilistic P systems and Stochastic P systems. More details of those models can be found in the Eclipse Project [<http://www.eclipse.org>].

The purpose of using P Lingua as a programming language to define cellular machines is a concept in the development of applications for membrane computing that leads to standardization and has the following advantages (Diaz-Pernil *et al.*, 2009):

- Users can define cellular machines in a modular and a parametric way by using an easy-to-learn programming language
- It is possible to define libraries of modules that can be shared among users to facilitate the design of new programs
- This method of defining P systems decoupled from its applications and the same P-Lingua programs can be used in different software environments
- By using compiling tools, the P-Lingua programs translated into other file formats that can be interpreted by a large number of different applications
- Figure 3 illustrates the process of converting a MAPK cascade problem from an ODE model into implementation on a PRISM model as shown in the next sections and then representing it using membrane computing, which will be explained in the next section

MeCoSim

MeCoSim is a membrane computing simulator (Perez-Hurtado *et al.*, 2010) initially designed to enable the use of user-defined customized interfaces with inputs, outputs and charts adapted to each family of P systems. This makes it possible to enter data for different initial conditions or to instantiate different P systems of the family.

MeCoSim has been extended such that it can cover a more general set of applications by providing flexible and powerful methods of integrating various software applications and packages as MeCoSim plugins.

These kinds of plugins can be easily added to MeCoSim by setting appropriate parameters in a configuration file. Keeping in mind this architecture and the developed plugins, MeCoSim may provide a platform for the integration of different tools for the modelling, simulation, analysis, property extraction and verification of P systems.

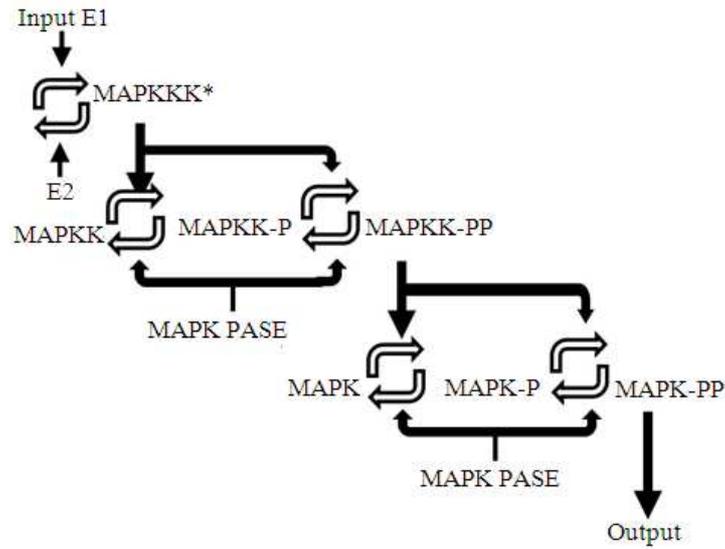


Fig. 1. MAPK cascade pathway (Huang and Ferrell, 1996)

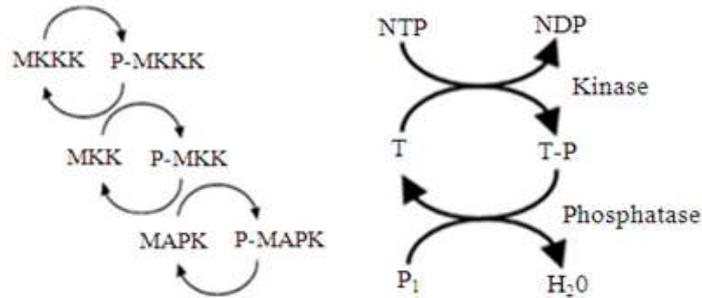


Fig. 2. Reactions in the formation of MAPK cascade (Huang and Ferrell, 1996)

ODE	Prism	M.C
Extract the biological elements and processes from the biological system modelled in ODE	Extract the biological elements and processes from the biological system modelled in Prism	Use membrane computing formalism to represent biological elements

Fig. 3. Framework of the steps for MAPK

The Membrane Computing Model for MAPK and P-Lingua Representation

The ODE model is converted into a discrete system by using rewriting rules to be modelled in membrane computing. And we convert the PRISM model by using the same rules used in PRISM model for solving MAPK are also used for membrane computing. The chemical substances, the structure of the system, the processes that interact between compartments and the parameters assigned to each process are extracted from the ODE model (Muniyandi and Abdullah, 2012). Before the system is represented in P-Lingua it converted into a discrete form by taking into consideration the differential equations and the following rewriting rules for some of the processes involved in the system.

In P systems with symport/antiport, computation is restricted to the synchronous movement of objects from one compartment into another. A configuration of the system is given by a finite multiset for each of the compartments; each compartment contains a finite number of objects, whereas the objects initially present in the environment are assumed to have infinite supply. Rules associated to the compartments and inspired by the protein mediated transport.

The degree of a system is the number of compartments it has while the weight of a system is a pair of numbers given by the maximum weights for the symport and antiport present in the system. If a system does not have any symport, then the maximum weight for symports is 0; similarly for antiports. If the symports in a system can have an unbounded weight (that is, they can move an unbounded number of objects), then the maximum weight for symports is denoted by *; similarly for antiports.

Computational steps consist of the application of a multiset of these rules, under the requirement of maximal parallelism: Such a multiset cannot be carried out if there is the possibility of performing a step that involves a strictly larger multiset of rules.

A computation is successful when starts in the initial configuration (specified as a multiset of objects for each membrane and an infinite supply of objects in the environment) and when ends in a halting configuration, a configuration where no rule is applicable. The result of a successful computation is the number of objects present in a designated membrane, the output membrane.

Our model is a stochastic model that works based on Gillespie algorithm according to nature of MAPK cascade. In this study, a cell-like membrane is used and we define a membrane with level 0 and there are

objects E1 and E2 in the membrane. These objects controlled the reactions with MAPK, MAPKK and MAPKKK in a phosphorylation and dephosphorylation.

The first step is to assign the initial amount (N) of MAPK, MAPKK and MAPKKK to 100.

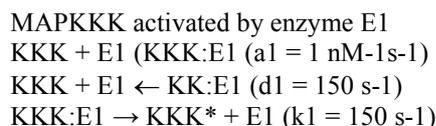
In additional we have constant reaction rates for element a from a1 to a10 and for the element d from d1 to d10 and for the element k from k1 to k10. The cell-like membranes of Mitogen Activated Protein Kinases (MAPK) cascade are represented as follow:

Membrane (0) = E1*E1-Lni where (E1[^] {E1-Ini}) initialize the E1-Ini copy of object E1 in membrane 0)

Membrane (0) = E2*E2-Ini; where (E2[^] {E2-Ini}) initialize the E2-Ini copy of object E2 in membrane 0)

With the Initial parameters for a, d and k where E1 and E2 is the enzymes for activation as shows on this example for some rules:

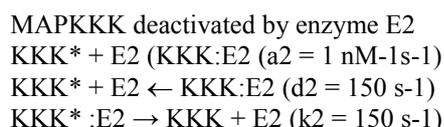
1. To activate enzyme E1 in PRISM by this rule:



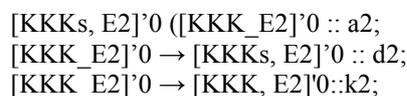
This rule converted to P-Lingua:



2. To deactivate enzyme E1 in PRISM by this rule:

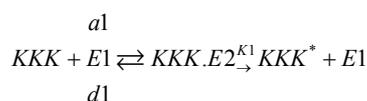


This rule converted to P-Lingua:

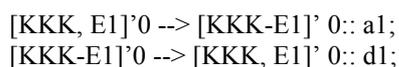


For example, object E1 for MAPKKK activator. This object takes part in two reactions as a reactant in Rule 1).

Rule 1:



Representation in P-Lingua is:



We used the same rule that used in PRISM model, but we converted this rule to work in membrane computing by using P-Lingua.

Results

In the reactions presented in Fig. 4 and 5, it is assumed that the phosphorylation of both MAPK and MAPKK occurs in two distributed steps. For example, when MAPK collides with its activator (MAPKK-PP), the first Phosphorylation (MAPK-P) occurs and the activator is released. The phosphorylated MAPK must then collide again with its activator for the second Phosphorylation (MAPK-PP) to occur. The deactivation of phosphorylated MAPK and MAPKK is caused by the corresponding phosphatase, while the activation and deactivation of MAPKKK is through

the enzymes E1 and E2 respectively. To simplify the presentation, in Fig. 8 we denote MAPK, MAPKK and MAPKKK by K, KK and KKK respectively.

By using the general cascade in MAPK and the same values as were previously used in PRISM to implement the MAPK cascade in membrane computing using P-Lingua. Then we implement the model with MeCoSim to compare it with the activation of three main elements in the cascade (MAPK, MAPKK and MAPKKK). The first activation in the cascade implemented in P-Lingua and simulated in MeCoSim, to obtain a new result, as shown in Fig. 4, for the first MAPK.

We implement the second activation in the cascade as shown in the Fig. 5. In the second activation in which MAPKK is activated in the same way using the same value for $N = 100$, we obtain the result shown in Fig. 5 for 60 steps.

In the third activation that active the MAPKKK and the same value of $N = 100$ that activated, we obtain the result shown in Fig. 6.

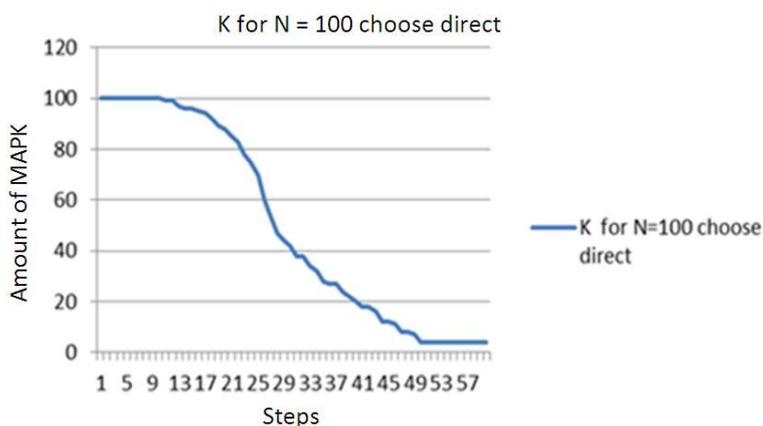


Fig. 4. Activation of MAPK

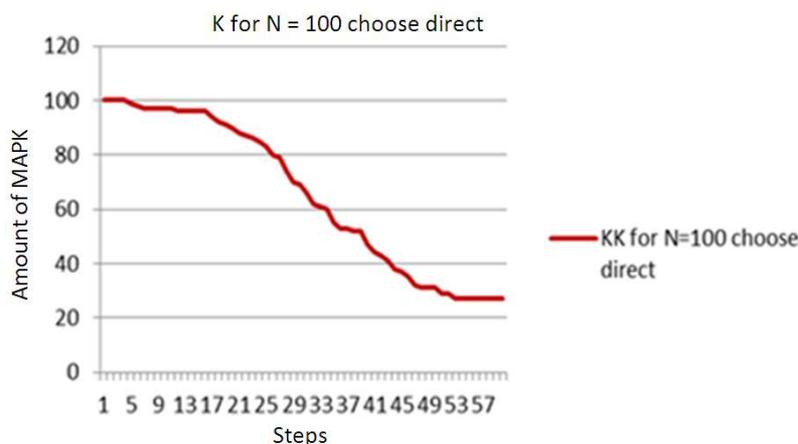


Fig. 5. Activation of MAPKK

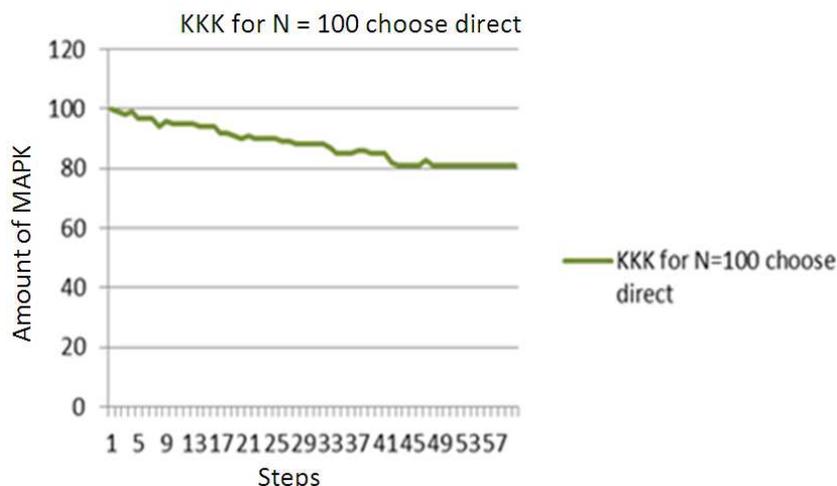


Fig. 6. Activation of MAPK

Table 1. Some of values of 20 steps on MeCoSim

Some of values of 20 steps on MeCoSim

Steps	MAPK	MAPKK	MAPKKK
1	100	100	100
2	100	100	99
3	100	100	98
4	100	100	99
5	100	99	97
6	100	98	97
7	100	97	97
8	100	97	94
9	100	97	96
10	100	97	95
11	99	97	95
12	99	96	95
13	97	96	95
14	96	96	94
15	96	96	94
16	95	96	94
17	94	94	92
18	92	92	92
19	89	91	91
20	88	90	90

The modelling of the MAPK cascade with membrane computing shows that modelling of biological systems with membrane computing provides a better option for symbolizing the structure and the processes involved in biological systems compared to the ODE approach. This is because we are able to distinguish the processes and the movement of objects within membrane in any cell that

has a MAPK cascade. The simulation results show that the membrane computing model provides results similar to those generated in the ODE model. However, with the membrane computing model, the non-deterministic and parallel executions of processes in a compartment are obvious compared to the deterministic approach of ODE.

Table 1 shows some of value from the P-Lingua model after simulating in MeCoSim.

Discussion

PRISM is a probabilistic model checker that represents a technique for formally verifying quantitative properties of a stochastic system (Huang and Ferrell, 1996). The MAPK cascade that is modelled in PRISM formalism is translated into the membrane computing. The technique for translation from PRISM into membrane computing that is applied in this study was proposed by (Kwiatkowska *et al.*, 2008).

When we compare the activation between the three activated steps for MAPK, MAPKK and MAPKKK, we can understand the difference between the activation in this cascade for MAPK, as shown below in Fig. 7.

Figure 8, we can see the difference in the activation for MAPK between three different approaches: The first approach used membrane computing to implement this cascade and has good results that are better than those of other approaches. The second approach implements the MAPK cascade using the PRISM model, which is checked by a probabilistic model checker. And the third approach used ODE representation, which was the first

approach to the MAPK cascade and we can see the values for reactions in Fig. 8.

A generalized and simplified MAP kinase cascade has been structured based on assumptions from past studies and was solved in P-Lingua, the programming language of membrane computing. MAPKKK, MAPKK and MAPK concentrations were expressed as a function of time. Assuming some arbitrary values and some values from existing studies, a plot was created between the concentrations and time. Although MAPKKK behaved in accordance with theoretical predictions, some anomalous behaviour was observed in the case of MAPKK and MAPK.

Further works can be carried out by incorporating Michaeli-Menten kinetics into the modelling and analysis of complex cascading simulations.

When analysing quantitative properties such as those listed above, it is often beneficial to study trends in these results as some parameters of the model (e.g., initial species concentrations or reaction rates) or of the property specification (e.g., a time bound) vary. Performing the analysis in this way is more likely to provide insight into the dynamics of the model or to identify interesting or anomalous behaviour. To illustrate this, Fig. 6 shows the results obtained with P-Lingua for the MAPK cascade case study.

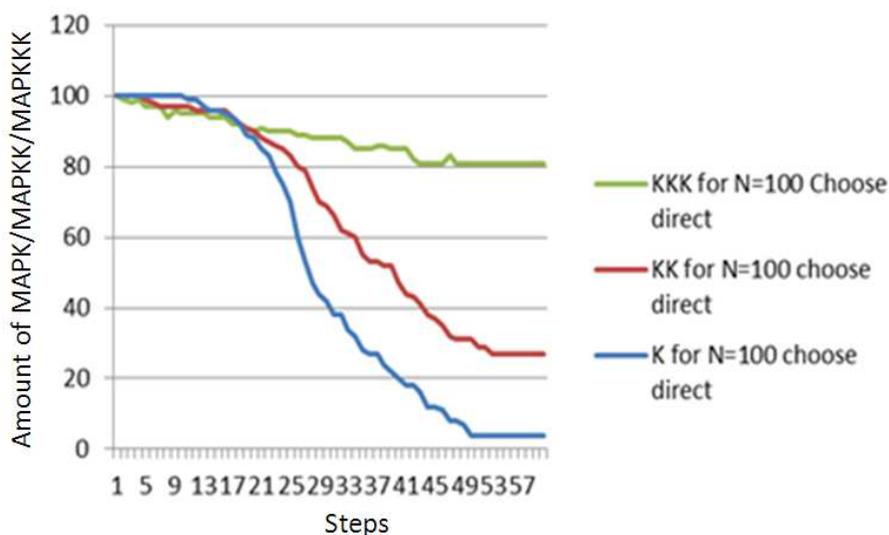


Fig. 7. Comparing the activations of MAPK, MAPKK and MAPKKK

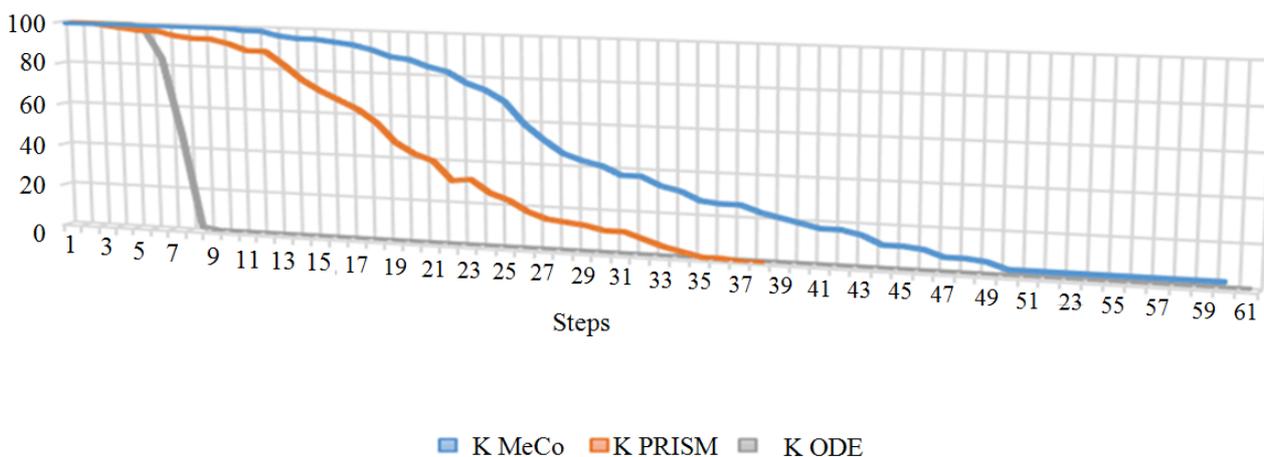


Fig. 8. Activation of MAPK with ODE, PRISM and MeCosim

The properties we consider are the expected percentage of activated MAPK at time instant t for the ODE model, the PRISM model and our work on membrane computing after simulation on MeCoSim. The expected number of reactions between MAPK up until time t and the expected time until all MAPK are activated in the same interval. We varied the initial quantities of MAPK for each model: Initially we supposed that there were N of each of these species and 100 of all remaining species in the cascade (the enzymes E1 and E2 and the phosphatases for MAPK and MAPKK). For the first two properties, the results presented in Fig. 7 demonstrate that, as the size of N grows, the percentage of MAPK that is activated increases and the time until all MAPK are activated decreases. We also observed that, as N increases, the behaviour of the membrane computing demonstrates the same behaviour as that presented in (Huang and Ferrell, 1996) (computed through ODEs and the reactions given in Fig. 2, where, in response to an external stimulus (E1), the cascade acts as a switch for the activation of MAPK). The biological case study has shown that biological systems could be modelled using membrane computing in a manner which maintains the physical structure and biological behaviour.

Conclusion

This approach overcomes some of the limitations of the conventional method based on ODE, which emphasizes the mathematical computation of the processes of the overall system, but at the expense of the system structure and the behaviors' of processes involved in a compartment. This is mainly because membrane computing is capable of representing those biological structures and processes in a formal way, respecting the salient biological characteristics (Muniyandi and Zin, 2014).

The case study has also validated the modelling capabilities of membrane systems regarding a specific MAPK cascade. The biological data concerning a particular biological system support the definition of a membrane computing model which has turned out to be capable of describing more natural conditions than those analyzed in laboratories by microbiologists.

Therefore the approach developed in this work of modelling a biological system represented by membrane computing using P-Lingua could provide guidelines for computational biologists to utilize membrane computing as a formalism for the modelling of biological systems.

Acknowledgment

This study has been supported in part by a grant of grant code FRGS/1/2012/SGOS/UKM/02/3 of the

University Kebangsaan Malaysia (National University of Malaysia).

References

- Chang, L. and M. Karin, 2001. Mammalian MAP kinase signalling cascades. *Nature*, 410: 37-40. DOI: 10.1038/35065000
- Diaz-Pernil, D., I. Pérez-Hurtado, M.J. Pérez-Jiménez and A. Riscos-Núñez, 2009. A p-lingua programming environment for membrane computing. *Membrane Comput.*, 5391: 187-203. DOI: 10.1007/978-3-540-95885-7_14
- Díaz-Pernil, D., P.H. Ignacio and M.J. Pérez-Jiménez, 2009. A P-lingua programming environment for membrane computing. *Membrane Comput.*, 5391: 187-203. DOI: 10.1007/978-3-540-95885-7_14
- Ernst, M., J. Cockrell, W.G. Griswold and D. Notkin, 2001. Dynamically discovering likely program invariants to support program evolution. *IEEE Trans. Software Eng.*, 27: 99-123. DOI: 10.1109/32.908957
- Gillespie, D.T., 1977. Exact stochastic simulation of coupled chemical reactions. *J. Phys. Chem.*, 81: 2340-2361. DOI: 10.1063/1.2710253
- Goldbeter, A. and D.E.J. Koshland, 1981. An amplified sensitivity arising from covalent modification in biological systems. *Proc. Natl. Acad. Sci.*, 78: 6840-6844. PMID: PMC349147
- Huang, C.Y. and J.E.J. Ferrell, 1996. Ultrasensitivity in the mitogen-activated protein kinase cascade. *Proc. Natl. Acad. Sci.*, 93: 10078-10083. PMID: 8816754
- Kwiatkowska, M., G. Norman and D. Parker, 2008. Using probabilistic model checking in systems biology. *SIGMETRICS Perform. Eval. Rev.*, 35 14-21. DOI: 10.1145/1364644.1364651
- Muniyandi, R.C. and M.Z. Abdullah, 2012. Membrane computing as a modeling tool for discrete systems. *J. Comput. Sci.*, 7: 1667-1673. DOI: 10.3844/jcssp.2011.1667.1673
- Muniyandi, R.C. and A. Mohd Zin, 2014. Modeling framework for membrane computing in biological systems: Evaluation with a case study. *J. Comput. Sci.*, 5: 137-143. DOI: 10.1016/j.jocs.2013.12.004
- Muniyandi, R.C. and A.M. Zin, 2011. Comparing membrane computing with ordinary differential equation in modeling a biological process in liver cell. *Proceedings of the IEEE 6th International Conference on, Bio-Inspired Computing: Theories and Applications*, Sept. 27-29, IEEE Xplore Press, Penang, pp: 278-283. DOI: 10.1109/BIC-TA.2011.68
- Muniyandi, R.C., A.M. Zin and J.W. Sanders, 2013. Converting differential-equation models of biological systems to membrane computing. *Biosystems*, 114: 219-226. PMID: 24120990

- Muniyandi, R.C., A.M. Zin, Z. Shukor, R.C. Muniyandi and A.M. Zin *et al.*, 2010. Model checking the biological model of membrane computing with probabilistic symbolic model checker by using two biological systems. *J. Comput. Sci.*, 7: 669-669. DOI: 10.3844/jcssp.2010.669.678
- Orton, R.J., O.E. Sturm, V. Vyshemirsky, M. Calder and D.R. Gilbert *et al.*, 2014. Computational modelling of the receptor-tyrosine-kinase-activated MAPK pathway. *Biochem. J.*, 392: 249-261. PMID: 16293107
- Pan, S. and S.E. Mhatre, 2013. Modeling the Mitogen Activated Protein (MAP)-kinase pathway using ordinary differential equations. *Comput. Biol. Bioinform.*, 1: 6-9. DOI: 10.11648/j.cbb.20130102.11
- Păun, G., 2006. Introduction to Membrane Computing. In: Applications of Membrane Computing, Ciobanu, G., M.J. Pérez-Jiménez and G. Păun (Eds.), Springer Berlin Heidelberg, ISBN-10: 978-3-540-25017-3, pp: 1-42.
- Perez-Hurtado, I., L. Valencia-Cabrera, M.J. Pérez-Jiménez, M.A. Colomer and A. Riscos-Núñez *et al.*, 2010. MeCoSim: A general purpose software tool for simulating biological phenomena by means of P systems. Proceedings of the IEEE 5th International Conference on Bio-Inspired Computing: Theories and Applications, Sept. 23-26, IEEE Xplore Press, Changsha, pp: 637-643. DOI: 10.1109/BICTA.2010.5645199