Adaptive Modularization of the MAPK Signaling Pathway Using the Multiagent Paradigm

Abbas Sarraf Shirazi[1], Sebastian von Mammen[1], and Christian Jacob[1,2]

[1] Dept. of Computer Science, Faculty of Science
[2] Dept. of Biochemistry & Molecular Biology, Faculty of Medicine
University of Calgary, Canada
asarrafs@ucalgary.ca, s.vonmammen@ucalgary.ca, cjacob@ucalgary.ca

Abstract. We utilize an agent-based approach to model the MAPK signaling pathway, in which we capture both individual and group behaviour of the biological entities inside the system. In an effort to adaptively reduce complexity of interactions among the simulated agents, we propose a bottom-up approach to find and group similar agents into a single module which will result in a reduction in the complexity of the system. Our proposed adaptive method of grouping and ungrouping captures the dynamics of the system by identifying and breaking modules adaptively as the simulation proceeds. Experimental results on our simulated MAPK signaling pathway show that our proposed method can be used to identify modules in both stable and periodic systems.

1 Introduction

A signaling pathway is a process by which a cell transfers information from its external receptors to a target inside [1]. It usually consists of a cascade of biochemical reactions carried out by enzymes. From a software engineering point of view, a signaling pathway description is similar to a UML diagram describing which components interact in the cascade. Playing a key role within the cell cycle, the Mitogen-Activated Protein Kinase (MAPK) pathway is one of the most documented signaling pathways in the literature.The MAPK pathway creates responses to extracellular stimuli and regulates cellular activities, such as gene expression, mitosis, differentiation, etc [2].

A multiagent system (MAS) can be composed of a number of agents interacting with their neighbours as well as their environment. This paradigm is a promising approach to model a biological system in which there are different entities that interact locally [3,4,5,6]. One of the key challenges associated with multiagent modeling is its high computational cost. Therefore, there should be a mechanism for efficient usage of computational resources. Modularization is such a mechanism in which the average behaviour of similar processes is learned, thus creating a higher-level algorithmic representation, which is then used instead of the original, more elementary processes. However, in cases when the behaviour

R. Schaefer et al. (Eds.): PPSN XI, Part II, LNCS 6239, pp. 401–410, 2010.

of agents changes over time, a static modularization cannot be used. Instead, the model should have the ability of adaptive modularization, in order to properly reflect the dynamics of the underlying system.

The goal of this work is to propose such a modularization method and demonstrate its effectiveness by example of the MAPK signaling pathway. To achieve this objective, there are various issues to be addressed. The first issue is how to group different agents into a module and learn their behaviour. Another issue is whether and how to break or integrate modules whenever the dynamics of the system is changed. Furthermore, the transition between different states of the model should be seamless. This research will shed light on how to build a smooth transition between various models in a complex and multiscale model and therefore will serve as the first step to multiscale modeling of biological systems using multiagent systems.

The remainder of this paper is organized as follows. Section 2 reviews related work in the field of multiagent modeling of biological systems and multiscale modeling. Section 3 presents the details of our proposed method. Section 4 reports on the experiments conducted to demonstrate the performance of the proposed method. Finally, concluding remarks are presented in Section 5.

2 Related Work

Amigoni and Schiaffonati [1] present a thorough analysis of multiagent-based simulation of biological systems. In particular, they discuss three different multiagent approaches to model the MAPK signaling pathway. The first approach [2], models every chemical reaction as agents, while the approach proposed in [7] defines a multiagent system in which each intracellular component is an agent that uses a blackboard mechanism to interact with other agents in the system. The third approach [8], models each molecular entity as an agent. In this model, a reaction is implemented as messages communicated among the agents.

A modularization approach for the MAPK signaling pathway is presented in [9]. It works by finding the node with the maximum number of neighbours in the biological interaction network. Further expansion of this node into a subgraph is called a module. To this end, it is assumed that the graph of the network is known beforehand and a static graph analysis is performed. Despite being a static and intuitive algorithm, it serves as a starting point for the modularization part of this research toward a multiscale model.

Another approach which is proposed by Papin *et al.* [10] tries to find modules in an unbiased fashion using mathematically based definitions. These authors reviewed three different approaches to calculate correlated reaction sets (Co-Sets). Co-Sets are groups of reactions in a network whose functional states are similar. Network-based pathways methods like elementary modes [11] and extreme pathways [12] aim to optimize a flux-balance equation by finding sets of similar nodes. Another method is referred to as the flux coupling finder, which also minimizes or maximizes the ratio between all pair-wise combinations of nodes [13]. Finally, a correlation coefficient is defined between each pair of nodes in the network based on their reaction fluxes [14].

As the technology advances, the prospect of making a multiscale model becomes more prominent. In [15], different issues and trends in multiscale modeling of complex biological systems are addressed. In [16], a software framework for multiscale model integration and simulation is proposed; however, no specific modeling techniques are described. There are a few physical multiscale models, e.g. CPM [17], and Synergetics [18]. However, as of yet, there is no universally adopted theoretical or computational framework for the assembly of multiscale biological models [19].

Bassingthwaighte *et al.* identify a systems approach for developing multiscale models which includes six steps [20]: (1) defining the model as its highest level of resolution, (2) designing reduced-form modules, (3) determining the range of validity of the reduced form modules, (4) monitoring the variables of the system, (5) replacing higher resolution models with reduced form modules, and finally, (6) validating the performance of the multiscale model against available real data. They further identify issues that must be addressed by any attempt to multiscale modeling. Examples of these issues are parameter identification of closed-loop systems, the identification of input-output delays, and the imposition of known constraints. Their work is among very few attempts to identify challenges ahead of multiscale modeling from a computer science perspective.

3 Adaptive Modularization in a Multiagent Environment

Modularization is the process of identifying modules within a network that are functionally similar. By replacing the behaviour of individual nodes with the behaviour of their enclosing module, the complexity of the network will be reduced. This way, a large network can be efficiently analyzed using a reduced number of nodes. Modularization is usually a static process in which modules are found before the simulation starts. Furthermore, most modularization approaches assume that the agent interaction graph is completely known as a whole. This assumption is restrictive, especially in the case of extended networks where the number of nodes is very large. Furthermore, analyzing the global graph is not scalable, since with the introduction of each new node the analysis must be performed again. As a result, we propose that the multiagent paradigm can be used to tackle the problem of scalability and also complexity of large graphs.

A multiagent system usually has no top-down control unit, which operates on the whole system. Agents cooperate or compete autonomously to perform various tasks. Contrary to traditional systems, a MAS agent only knows about its local interactions. Consequently, agents can form their local directed graph of interaction. Agents can cooperate and share their information (in this case, their interaction graph) with other agents. This way, they can form groups or modules in a bottom-up fashion.

In our proposed approach, we aim to find, integrate and break modules dynamically as the simulation proceeds. Based on the system dynamics, we expect our algorithm to find different modules that act together over a period of time. To this end, we must address several issues as described in [20]. How and when

Algorithm 1. Module Indentification

```
$m = current\_module$;
$Module\ new\_module$;
$Queue\ q$;
$q.Enqueue(m)$;
$new\_module.Add(m)$;
while $!q.empty()$ do
    $Module\ head = q.Dequeue()$;
    for all $Agent\ s$ in $head$ do
        for all $Agent\ t$ in $s.Neighbours()$ do
            if $|\rho_{st}| \geq \tau_{edge}$ then
                $new\_module.Add(t)$;
                $q.Enqueue(t)$;
            end if
        end for
    end for
end while
return $new\_module$;
```

Algorithm 2. Validity Monitoring

```
$m = current\_module$;
$needToBreak = false$;

for all $Agent\ s$ in $m$ do
    for all $Agent\ t$ in $s.Neighbours()$ do
        if $|\rho_{st} - \rho'_{st})| > \tau_{valid}$ then
            $needToBreak = true$;
            $break$;
        end if
    end for
end for

if $needToBreak$ then
    $simulation.remove(m)$;
    for all $Agent\ s$ in $m$ do
        $simulation.add(s)$;
    end for
end if
```

to integrate nodes to form a module, how to learn the behaviour of a module, and how to monitor the validity of modules are among the issues that we address in this section.

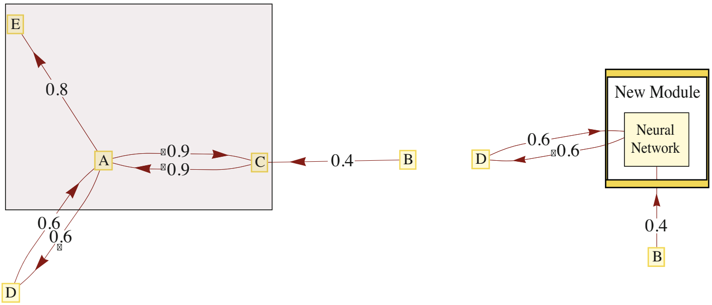
3.1 Creating Modules

In our system, agents are associated with an interaction graph as well as an interaction history for all their neighbours. The weight of an edge in their interaction graph is equal to their correlation coefficient with their neighbour. A correlation coefficient between two statistical variables indicates their linear dependancy. A zero correlation coefficient means that two variables are independent, while +1 or -1 shows highly correlated variables. The more two variables are correlated, the more similar their function is. In case there is a series of n measurements of agents s and t in the form of s_i and t_i, where $i = 1, 2, ..., N$, their correlation coefficient (ρ_{st}) is defined as follows:

$$\rho_{st} = \frac{\sum_{i=1}^{N}(s_i - \bar{s})(t_i - \bar{t})}{(n-1)\sigma_s\sigma_t} \tag{1}$$

where $\bar{s}$ and $\bar{t}$ are the mean values, and σ_s and σ_t are standard deviations of s and t, respectively.

Having a local weighted graph, each agent then periodically checks if its correlation coefficient with each neighbour is greater than some threshold (τ_{edge}). If so, they form an initial module and repeat this process to identify a cluster



(*a*) (*b*)

Fig. 1. Example of an interaction graph. The edges denote the correlation coefficients. (a) *Agent A*, *Agent C*, and *Agent E* form a module, (b) The new neighbours of this module are *Agent B* and *Agent D*.

of agents that are highly correlated (Algorithm 1). Fig. 1 shows an example in which *Agent A* finds *Agent C* and *Agent E*, and they form a module. The set of new neighbours is the union of all neighbours of the underlying nodes. Having formed such a module, the next step is to train this new module, so that it learns and imitates the group behaviour of its underlying nodes.

3.2 Learning the Group Behaviour

A module has to subsume the behaviour of its underlying nodes by abstracting from their behaviour. In other words, the new module has to replace its associated nodes and produce the same outputs as if there were individual agents in the system. The learning algorithm can employ neural networks, time series, or any other function approximation algorithm. No matter what learning algorithm is used, each node has to have an interaction history to be used during the learning phase. In our approach, we used a three-layer feed-forward neural network with back propagation learning algorithm [21] to train the network. This way, we also have control over the speed of learning.

The structure of the neural network is determined by its inputs, the number of nodes in the hidden layer, and the outputs. Since in our model, agents are not aware of their dependent agents (in fact, they only know about their outgoing edges), the output of the network should simply be all of the underlying nodes (in the example of Fig. 1, outputs are *Agent A*, *Agent C*, and *Agent E*). Regarding the input to the network, there are different design choices. The first one would be to assign external incoming edges and ignore internal connections (*Agent D* in Fig. 1). An alternative approach is to consider internal nodes as well. This way, the neural network has more meaningful sets of data to be trained with. As for the number of nodes in the hidden layer, we follow a simple rule-of-thumb and assign it to be the number of inputs + 2.

3.3 Monitoring the Validity of Modules

Once a module is found and trained, it subsumes the behaviour of its underlying nodes. Due to the dynamic behaviour of the system, at some point, the module might show invalid behaviours. To address this issue, we check the validity of each module periodically. Nonetheless, we need an indicator to compare the current and expected behaviour of the module. A heuristic indicator is the previous correlation coefficients of the underlying nodes before they form a module (ρ'_{st}). According to Algorithm 2, we compare the current correlation coefficients of the module to previous values for each individual node, if the difference is larger than some threshold, we consider the module invalid and consequently break it into its underlying nodes.

4 Experiments on the MAPK Signaling Pathway

Our proposed adaptive modularization approach can be employed in any system where there are different agents interacting locally. Signal transduction pathways are such ideal candidates, as for most of them there is quantized data available. In general, a signal transduction pathway starts with an external stimulus in a cascade of biochemical processes, which in turn results in a change of state in a cell. In the MAPK signaling pathway [22], a hypothetical enzyme E1 stimulates the cell and results in an increase in production of MAPK-PP enzyme (Fig. 2(a)). In another model [23], a negative feedback loop causes sustained oscillations in the production of MAPK-PP (Fig. 2(b)).

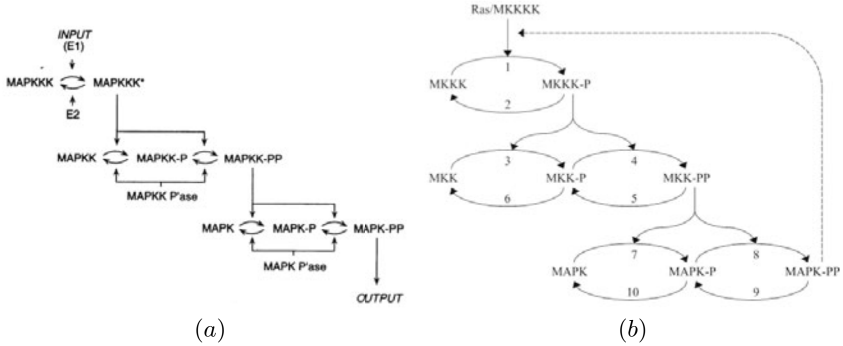


Fig. 2. (a) The MAPK signaling pathway (from [22]), and (b) The MAPK signaling pathway with a negative feedback (from [23])

4.1 Agent-Based Model of the MAPK Signaling Pathway

Contrary to the differential equation-based approach discussed above, in our model each substance is considered to be an independent entity which is loosely

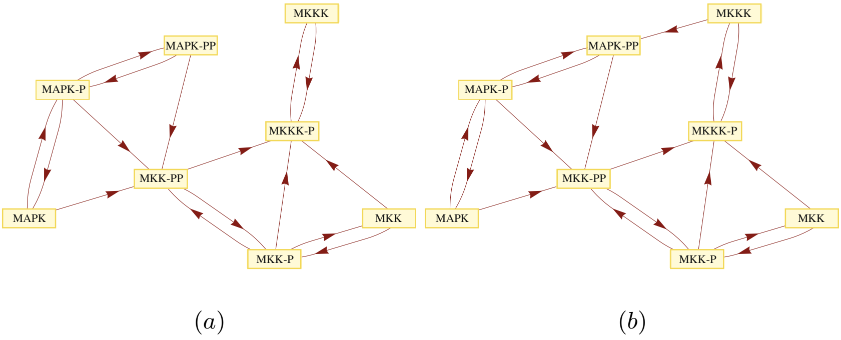


Fig. 3. Agent graphs for the MAPK signaling pathways of Fig. 2

defined as an agent. For each agent, the interaction graph defines its relations with the substances that appear in its update formula[1]. Fig. 3 shows the complete interaction graph for the signaling pathways of Fig. 2.

To validate the performance of the adaptive modularization, we conducted a series of experiments on both MAPK models. Essentially, there are five parameters in our algorithm which are summarized in Table 1. We let the system run in its normal mode for some time (t_{wait}) and then start looking for modules within a time interval (Δ_{find}). t_{wait} is important in that the system has to reach a rather stable condition before the modularization algorithm starts to work. We keep monitoring the system also in predefined intervals ($\Delta_{monitor}$). A module is valid as long as its correlation coefficients with its neighbours do not vary too much with regards to those of individual agents (τ_{valid}). Finally, to integrate nodes and find modules, the value of an edge in the interaction graph should be greater than some threshold (τ_{edge}). τ_{valid} and τ_{edge} have been found through trial and error. A more detailed exploration of the parameter spaces will be undertaken in our future work.

Fig. 4(a) shows the result of applying our approach to the first model (Fig. 2(a)) in terms of the number of modules. Initially, each agent is its own module. The identification of modules starts after $t = 1200$. The process of construction and deconstruction of modules results in the emergence of a periodic pattern. The reason is that whenever a module is broken, all of its underlying nodes start to work as individual agents again. Naturally, when a module contains a larger number of nodes, the probability of that module to become invalid is higher. In other words, since there is no hierarchical learning, after an all-encompassing single module is created and it breaks, there are again eight individual modules (one for each agent) in the system. As this modularization/demodularization process continues, a periodic pattern appears as illustrated in Fig. 4(a). Fig. 4(b) shows that the final concentration successfully resembles that of the PDE solver.

[1] The complete set of update equations can be found in [22] and [23].

Table 1. Model Parameters

Parameter Name	Symbol	Value in Experiment 1	Value in Experiment 2
Delay before finding modules	t_{wait}	1200	1500
Modules finding interval	Δ_{find}	300	300
Monitoring interval	$\Delta_{monitor}$	20	20
Validity Threshold	τ_{valid}	0.1	0.1
Edge Threshold	τ_{edge}	0.95	0.7

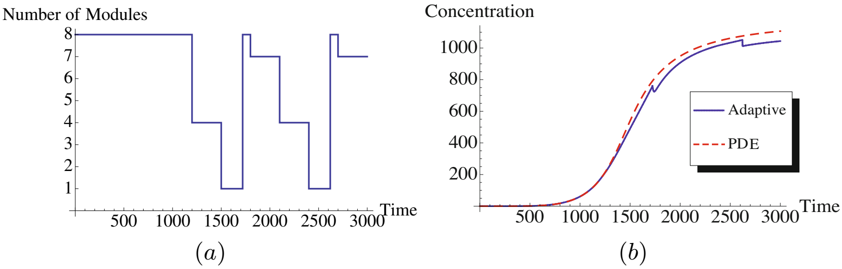


Fig. 4. Adaptive modularization results for the first MAPK pathway model of Fig. 2(a). (a) Number of agents, (b) Concentration of MAPK-PP.

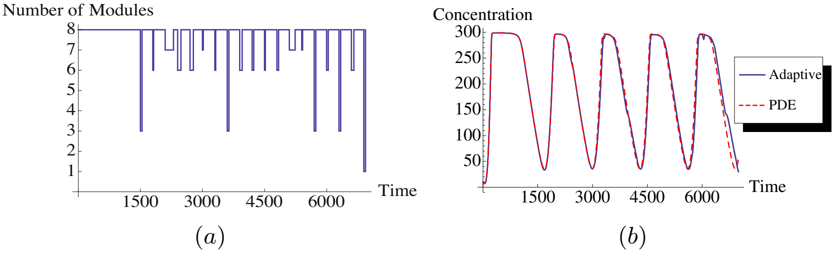


Fig. 5. Adaptive modularization results for the second MAPK pathway model of Fig. 2(b). (a) Number of agents, (b) Concentration of MAPK-PP.

Fig. 5 shows the result of adaptive modularization for the second MAPK pathway. Since this model is periodic, the adaptive modularization algorithm successively finds, trains, and breaks modules over time. The number of spikes in Fig. 5(a) shows that the validity period of a composite module is not long enough. The reason is that the correlation coefficient is a linear indicator which varies from -1 to +1 over a periodic signal. This variation makes a module in a periodic system invalid. This result suggests that we have to look for other parameters when we have a nonlinear system with feedback. Nevertheless, it is still more reliable than if modules were broken at random.

5 Conclusion and Future Works

In this paper, we introduced a bottom-up method to reduce the complexity of a multiagent system simulating the MAPK signaling pathway by adaptive modularization and demodularization. Although we have shown that this approach works very well for this specific example, we believe that our module composition and decomposition algorithm can be applied to a wide range of other multiagent systems. In particular, individual agents share their interaction graph to build a higher-level module which subsumes their behaviour. After a new module is formed, it learns the behaviour of its underlying nodes using a feed-forward neural network. To monitor the validity of a module, the values of any edge in its interaction graph is checked – at defined intervals – and compared against the previous values of its nodes. A module is broken if the difference between the current and previous value of an edge is greater than some threshold.

We use correlation coefficients to determine the edge value in the agent's interaction graphs. Although this indicator is mathematically sound, it does not capture the nonlinear dependance between agents. Looking for other nonlinear indicators seems to be a promising approach. This work is among very few attempts to find an algorithmic framework to address the complexity reduction in an agent-based system and can serve as the first step to address the reduction of complexity in highly complex systems.

References

1. Amigoni, F., Schiaffonati, V.: Multiagent-based simulation in biology a critical analysis. Model-Based Reasoning in Science, Technology, and Medicine 64, 179–191 (2007)
2. Desmeulles, G., Querrec, G., Redou, P., Kerdelo, S., Misery, L., Rodin, V., Tisseau, J.: The virtual reality applied to biology understanding: The in virtuo experimentation. Expert Syst. Appl. 30(1), 82–92 (2006)
3. Hoar, R., Penner, J., Jacob, C.: Transcription and evolution of a virtual bacteria culture. In: IEEE Congress on Evolutionary Computation, Canberra, Australia. IEEE Press, Los Alamitos (2003)
4. Jacob, C., Burleigh, I.: Genetic programming inside a cell. In: Yu, T., Riolo, R.L., Worzel, B. (eds.) Genetic Programming Theory and Practice III, pp. 191–206. Springer, Heidelberg (2006)
5. Jacob, C., Barbasiewicz, A., Tsui, G.: Swarms and genes: Exploring λ-switch gene regulation through swarm intelligence. In: CEC 2006, IEEE Congress on Evolutionary Computation, Vancouver, BC, Canada (2006)
6. Jacob, C., Burleigh, I.: Biomolecular swarms: An agent-based model of the lactose operon. Natural Computing 3(4), 361–376 (2004)
7. Gonzalez, P.P., Cardenas, M., Camacho, D., Franyuti, A., Rosas, O., Lagunez-Otero, J.: Cellulat: an agent-based intracellular signalling model. Biosystems 68(2-3), 171–185 (2003)
8. Khan, S., Makkena, R., McGeary, F., Decker, K., Gillis, W., Schmidt, C.: A multiagent system for the quantitative simulation of biological networks, pp. 385–392 (2003)

9. Nayak, L., De, R.K.: An algorithm for modularization of mapk and calcium signaling pathways: comparative analysis among different species. J. Biomed. Inform. 40(6), 726–749 (2007)
10. Papin, J.A., Reed, J.L., Palsson, B.O.: Hierarchical thinking in network biology: the unbiased modularization of biochemical networks. Trends Biochem. Sci. 29(12), 641–647 (2004)
11. Schuster, S., Dandekar, T., Fell, D.A.: Detection of elementary flux modes in biochemical networks: a promising tool for pathway analysis and metabolic engineering. Trends Biotechnol. 17(2), 53–60 (1999)
12. Schilling, C.H., Letscher, D., Palsson, B.O.: Theory for the systemic definition of metabolic pathways and their use in interpreting metabolic function from a pathway-oriented perspective. J. Theor. Biol. 203(3), 229–248 (2000)
13. Burgard, A.P., Nikolaev, E.V., Schilling, C.H., Maranas, C.D.: Flux coupling analysis of genome-scale metabolic network reconstructions. Genome. Res. 14(2), 301–312 (2004)
14. Price, N.D., Schellenberger, J., Palsson, B.O.: Uniform sampling of steady-state flux spaces: means to design experiments and to interpret enzymopathies. Biophys. J. 87(4), 2172–2186 (2004)
15. Martins, M., Ferreira Jr., S.C., Vilela, M.: Multiscale models for biological systems. Current Opinion in Colloid & Interface Science 15(1-2), 18–23 (2010)
16. Erson, E.Z., Cavuşoğlu, M.C.: A software framework for multiscale and multilevel physiological model integration and simulation. In: Conf. Proc. IEEE Eng. Med. Biol. Soc., vol. 2008, pp. 5449–5453 (2008)
17. Merks, R.M., Glazier, J.A.: A cell-centered approach to developmental biology. Physica A: Statistical Mechanics and its Applications 352(1), 113–130 (2005)
18. Haken, H.: Synergetics: an introduction: monequilibrium phase transitions and self-organization in physics, chemistry and biology / Hermann Haken. Springer, Berlin (1977)
19. Walker, D.C., Southgate, J.: The virtual cell–a candidate co-ordinator for 'middle-out' modelling of biological systems. Briefings in Bioinformatics 10(4), 450–461 (2009)
20. Bassingthwaighte, J., Chizeck, H., Atlas, L.: Strategies and tactics in multiscale modeling of cell-to-organ systems. Proceedings of the IEEE 94(4), 819–831 (2006)
21. Haykin, S.: Neural Networks: A Comprehensive Foundation. Macmillan, New York (1994)
22. Huang, C.Y., Ferrell, J.E.: Ultrasensitivity in the mitogen-activated protein kinase cascade. Proc. Natl. Acad. Sci. USA 93(19), 10078–10083 (1996)
23. Kholodenko, B.N.: Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. Eur. J. Biochem. 267(6), 1583–1588 (2000)