PhD Dissertation

International Doctorate School in Information and

Communication Technologies

DIT - University of Trento

COMPOSITIONAL MODELING OF BIOLOGICAL SYSTEMS

Judit Zámborszky

Advisor:

Prof. Corrado Priami

Università degli Studi di Trento

Co-Advisor:

Attila Csikasz-Nagy, PhD

The Microsoft Research - University of Trento Centre for Computational and Systems Biology

December 2010

Abstract

Molecular interactions are wired in a fascinating way resulting in complex behavior of biological systems. Theoretical modeling provides us a useful framework for understanding the dynamics and the function of such networks. The complexity of the biological systems calls for conceptual tools that manage the combinatorial explosion of the set of possible interactions. A suitable conceptual tool to attack complexity is compositionality, already successfully used in the process algebra field to model computer systems. We rely on the BlenX programming language, originated by the beta-binders process calculus, to specify and simulate high-level descriptions of biological circuits. Gillespie’s stochastic simulation algorithm applied for BlenX simulations requires the decomposition of phenomenological functions into basic elementary reactions. Systematic unpacking of complex reaction mechanisms into BlenX templates is shown. The estimation/derivation of missing parameters and the challenges emerging from compositional model building in stochastic process algebras are discussed. A biological example on circadian clock is presented as a case study of modeling.

Keywords

compositionality, process algebra, nonlinearity, templates, circadian clock, positive feedback

Contents

1. Introduction

1.1 The Context

Life is difficult in all sense. Systems biology research focuses on understanding how organisms carry out their function, resulting in a sophistically orchestrated system that is self-sustained, reproductive and responds to both internal and external changes. Like everything else, biological systems can also "go wrong". The growing knowledge of the underlying mechanisms contributes to novelties in application in healthcare and medical biotechnology and brings biological - and biology related - research to the front to help understanding living organisms.

Biological systems are extremely complex structures performing several crucial properties of life. The concept of cells, as the *functional units of life*, was established by the mid-nineteenth century - by Matthias Jakob Schleiden and Theodor Schwann [1] - and each day an enormous amount of new biological data is produced, still we are far from a detailed understanding how an organism, a population or even a cell functions. The physiological properties of a biological system can be observed by various microscopy techniques, the information coding DNA can be sequenced and the molecular interactions might be also detected, yet our knowledge about the mechanisms describing the observed behavior is incomplete. Theoretical models can assist molecular biologists to find a better understanding of cell physiology by revealing the dynamical behavior of the system and also by investigating complex interactions of regulatory molecules.

The pioneers of a novel field - called systems biology [2-4] - were able to proof that there is a need of a comprehensive system-level approach to handle complexity in biological research. This multidisciplinary field originates from molecular biology [5], the science of curious biologists asking the question: *What is in the black box of cells?* The discoveries of the 20th century explored the basic molecular components of the cell that made manipulation of the elements and behavior of cells and organisms possible. Even before opening this imaginary *box*, theoreticians were able to describe biological systems by the help of simple models, proposing a different - abstract - approach to handle biological questions. Later the genome revolution resulted in much more details of biological elements and led to a boom also in the field of informatics. Application of computers had a large effect on molecular biology research as bioinformatics emerged [6]. Its main goal was the creation and maintenance of databases, algorithms, computational and statistical techniques to store biological information. After exploring several molecular details about parts of the system, scientists still had to face the problem of complexity. In order to understand how biological systems achieve their tasks, molecular biologists moved from managing the collected data towards the question: *How does the box actually function?* Bioinformatics gave birth to computational biology [7], concentrating on asking biological questions that could be solved with novel computational techniques. Finally, biology, physics, computer science, systems theory and mathematics have joined to propel a research that provides tools for the analysis of biological studies in a systematic way; this is what we call *systems biology*.

It is no more a question that in order to really dig out how enormously large networks operate adequate computational models and tools are highly required to address biological problems. With them, it is possible to analyze, simulate, understand and make predictions of a complex system. I would like to emphasize that personally I believe the real strength of computational modeling is not replacing wet biology, rather is providing a tool to evaluate the behavior of the system, to understand the basic mechanisms driving them, to identify the key components and form experimentally testable predictions. None the laboratory work nor the theoretical effort can be substituted by the other. The main contribution of computational biology to biological research is to develop algorithms for modeling. Several computational modeling approaches exist [8], such as logical discrete models (e.g. Boolean), agent-based models, continuous models (e.g. deterministic or stochastic ordinary differential equations (ODEs)) or discrete Monte Carlo simulations (e.g. process algebras, Petri nets). Various tools support the work of theoretical modeling (e.g. COPASI [9], JigCell [10], XPPAUT [11], VCell [12], SBML [13], BioAmbient [14], BIOCHAM [15] and others). The chosen model and the applied software depend on our extant knowledge about the system and on the knowledge we would like to gain from investigating it.

Stochastic approaches are becoming more and more sophisticated as novel experimental techniques - such as quantitative flow cytometry [16] and fluorescence microscopy [17] - provide single level measurements of cell physiology. While the average behavior of a cell population has been described by continuous modeling approaches (e.g. with Ordinary Differential Equations, ODEs) [18] from a long time, single cells are analyzed in a stochastic framework as fluctuations may have a significant effect on the physiology of the cell [19]. The influence of noise also in gene expression and signal transduction processes have been shown to be important by both theoretical and experimental approaches [20-22].

Process algebras were introduced in early 1980s by Milner [23], as specification languages for concurrent processes, namely of computational entities executing their tasks in parallel and able to synchronize over certain kinds of activities. Some examples of concurrent computing systems include communication networks, air traffic controllers, and industrial plant control systems. The abstraction provided by process algebras was shown to be successful in modeling several scenarios from life sciences (e.g. biology, including transcriptional circuits, metabolic pathways and signal transduction networks) [24]. A biology oriented programming language (BlenX) [25] inspired by process calculi (specifically Beta-binders [26]) is one of the progressive stochastic modeling approaches. Beta Workbench (BWB) [27] defines and implements the BlenX programming language and has been designed for biology from the beginning. As the field of executable biology [28] and algorithmic systems biology [29] gather ground in computational biology, development of conceptual tools that manage the combinatorial explosion of the set of possible interactions and that attack complexity through compositionality is becoming more and more important.

1.2 The Problem

Systematic approach for biological research requires an idea of an inferential model, a considered wiring diagram describing a system of studies and unanswered crucial questions. Additionally, adequate computational tools and curious students are also important in the research process. During my university studies, as a beginner “systems biologist”, I had to face several problems that aroused my interest in computational work.

The most difficult problem I had was how to choose from an enormous number of modeling tools that support biological questions. Decisions are hard to take, thus I simply started with the classical methods in order to address the interconnectivity of oscillatory systems (namely the cell cycle [30] and the circadian rhythms [31]). The classical tool for analyzing such crucial biological systems was definitely the apparatus of ordinary different equations (ODEs) [32]. Due to the fact that early experimental techniques have provided an opportunity to investigate only a population of cells, researchers turned to deterministic approaches to describe the behavior of an average phenotype (for instance the growth of a culture of cells). Ordinary differential equations illustrate dynamic processes with a set of equations for the change of continuous variables, usually the concentration of molecules. Based on the known or hypothesized molecular interactions in a system, one could write biochemical reactions that describe the kinetics of the network. The kinetic aspect concerning the rate equations for elementary reactions (reactions that proceeds through only one transition state) is generally defined by the law of mass action kinetics; stating that the rate of an elementary reaction step is proportional to the product of the concentrations of the participating molecules. There are also non-elementary functions that have been empirically developed [33]. These abstractions simplify the system leading to a decrease in the required computational power for calculation. Furthermore, modelers often turn to these phenomenological functions to describe the observed behavior of a system without knowing all its details, such as multistep reactions are often assumed to happen at the same time in cooperative reaction schemes [34].

After becoming familiar with the chosen biological system, one should actually start to realize - in the classical way - equations or - in computational modeling - the lines of codes. There are two approaches in modeling concerning its construction [7]: (1) building the system up starting from its basic elements (called as bottom-up approach) or (2) constructing the network based on our observation of the biological system without the complete knowledge of the details (referred to as top-down). Extension of a model is even trickier in some cases. Initiatives for generalizing different languages in one tool have been proposed to be a solution for model composition (e.g. SBML [35]) but it still remains under development.

The concept of connecting meaningful parts of a system into a larger model where their meaning remains still is called compositionality. It is a principle of languages. Words constitute sentences where the meaning of the complex expression is determined by the meanings of the subparts and the rules used to combine them. Process calculi tools [36] offer compositionality and formal description of interactions, communications and synchronizations between concurrent elements of a system. Lack of modularity in ODE systems - composed of equations that are denotational - makes their use to become more and more complicated with the increasing size of the model. Process calculi tools provide the introduction of an interaction simply by adding or changing a single rule instead of modifying a large number of equations or lines of the code. Rules can be encoded and the computer can execute them. Process calculus is a novel, promising approach also to model noise in biological systems with the integration of stochasticity. By joining the Centre for Computational and Systems Biology in Trento (Italy), I have started to work on a particular process calculus language, BlenX [25]. In computational biology, development of conceptual tools that manage the possible interactions and complexity through compositionality is a crucial task. The BlenX programming language has been specifically designed for biology from the beginning, thus it provides a formal stochastic framework for modeling biological systems.

The current version of the BlenX language applies Gillespie's stochastic simulation algorithm (SSA) [37] to follow the time evolution of the system. The assumptions of the algorithm require rate equations to be elementary reactions that are defined by the law of mass action kinetics. However, there are several non-elementary functions in biology that apply assumptions for describing an observed property of the reaction [38]. The frequent use of these nonlinear terms, such as Michaelis-Menten kinetics or the Hill function, creates a gap between classical - deterministic - models and the stochastic simulations implemented within the BlenX framework making composition of existing models difficult within the tool. The need of a bridge linking different modeling approaches has been recognized previously and few initiatives have already been proposed on the computational field [39-43], although the studies mentioned ahead paid less attention to the problem that arises in stochastic simulations applying Gillespie's stochastic simulation algorithm [37]. Gillespie's method assumes the exponential distribution of random variables describing the occurrence of the elementary reaction steps. Complex rate functions do not satisfy this assumption and however the application of nonlinear reaction rates is allowed in BlenX, they require the proof and the verification of their use in all modeling case studies.

The size and complexity of many biological systems results in a difficult process of modeling. Exploring principles and frequently occurring submodules - often referred to as motifs [33,44,45] - is a well-known approach to contribute to a higher level understanding of complex networks. Modularity and compositionality (i.e. the possibility of defining the whole system starting from the definition of its subcomponents) are key features of process calculi languages. Those offer an easier way of systematical modeling, although one also might find difficulties of presenting a real systematic way of model composition within those programming languages. For instance, the shift from the classical methods towards computational models is asking for the import of previous models and their implementation into an e.g. stochastic framework, such as the BlenX language. The current version of the BlenX language requires core computational knowledge and it is not an intuitive tool for theoretical biologists. Furthermore, listing the frequently used rules takes long time if the user is asked to build his model from the bottom each time. BlenX is asking for a higher level program design for the modeling process in order to be suitable for building BlenX models from the bottom-up in an easier way or for composing existing models for investigating larger networks. Exploring principles and frequently occurring submodules is a well-known approach to contribute to a sophisticated way of encoding programs. The BlenX language is suitable to describe biological systems in an exact way with elementary reaction steps and execute the code with Gillespie's stochastic simulation algorithm however the complex terms often used in ODE systems create a gap between these modeling approaches. We present a solution for these problems, summarized in the subsequent chapter.

1.3 The Solution Presented in This Thesis

Systematic "unpacking" of complex reaction mechanisms (description of the complex rate functions with intermediate steps) into BlenX templates is shown in this study. We have chosen frequently used and biologically relevant motifs that offer specific properties to the models in deterministic approaches. They had been implemented into the BlenX language as computational templates. To study the stochastic effects in nonlinear biochemical reactions, we should first describe the complex rate functions with elementary steps. Decomposition of the motifs into single reactions is shown. These submodules offer a systematic modeling framework though the compositional behavior of BlenX. In this thesis a novel modular approach to process calculus is presented. Adaptation of a software engineering style structuring techniques is shown as predefined computational templates are coded and present a biologically relevant library of important motifs providing a higher level compositionality of the BlenX language.

Additional to the proposed solutions for the composition of the computational models in BlenX, we also dealt with queries in biological systems and tested the computational templates presented in this work. Furthermore, the following biological questions were also approached from the modeling perspective. Recently, the circadian clocks [31] have been shown to gate cell cycle transitions [30]. The biochemical interactions [46] interconnecting these two oscillatory systems lead to the question what properties could arise from this relationship and what the importance of clocks could be in cell cycle regulation. The daily rhythms have been presented to be influenced by ionizing radiation treatment exploiting novel insights into cell cycle related circadian functions. In order to explore these queries, we built models and investigated the systems mentioned ahead [47,48]. Our approach reveals hypothesis of experimentally observed behaviors and provides a detailed analysis of them, and compositional modeling with the proposed BlenX extension is also presented as a feature for analysing complex regulatory networks.

The goal of this thesis is to realize a library consisting of predefined, biologically relevant submodules. The library offers a higher level compositionality with the BlenX language allowing the use of biologically relevant modules in building highly complex models. It is a novel design methodology for computational systems biology providing extensions and merge of models that are available in various modeling frameworks.

1.4 The Structure of the Thesis

The thesis starts with an overview of the research field in Chapter 2 (State of the Art), focusing on the process calculi languages developed for biology. After introducing the concept of compositionality in Chapter 2.5, we present the BlenX language in Chapter 2.6. From Chapter 2.7 to 2.10 the reader is introduced to the problems appearing on the field of stochastic simulation of complex reaction rates. In the Chapters from 2.11 to 2.13 we present the basics for the biological systems of our interest. Chapter 3 deals with the problem in details, to which the solution is proposed in Chapter 4. Results are shown in Chapter 5, introducing and explaining the properties and the realization of the templates within the BlenX language that is important in modeling stochastic biological circuits. Finally, the thesis ends with the Conclusion session summarizing the novelties of the work.

1.5 Related Publications

<u>Book chapters</u>:

Zámborszky J and Csikász-Nagy A (2010) Modeling the cell cycle. in *Mathematical Physiology* section, (eds. De Gaetano A. & Palumbo P.), In series *UNESCO Encyclopedia of Life Support Systems,* in press

Zámborszky J (2010), Cell Cycle Transitions, G2/M, in *Encyclopedia of Systems Biology*, (eds. Werner Dubitzky, Olaf Wolkenhauer, Kwang-Hyun Cho, Hiroki Yokota), *Springer*, in press.

Csikász-Nagy A, Palmisano A and **Zámborszky J.** (2010), Molecular network dynamics of cell cycle control: transitions to *Start* and *Finish*, in *Cell Cycle Synchronization: Methods and Protocols* (volume ed: Gáspár Bánfalvi), in series *Methods in Molecular Biology* (ed. John M. Walker), in press.

<u>Journal articles</u>:

Hong CI, **Zámborszky J**, Csikász-Nagy A (2009), Minimum Criteria for DNA Damage-Induced Phase Advances in Circadian Rhythms, *PLoS Comp Biol. 5(5):e1000384*. cover article of the 2009 May issue.

Impact factor: 5.759	**ISI citations: 1**	**G.s. citations: 1**

Zámborszky J, Hong CI, Csikász-Nagy A (2007), Computational analysis of mammalian cell division gated by a circadian clock: quantized cell cycles and cell size control. *J Biol Rhythms. 22(6):542-53.*

Impact factor: 4.633	**ISI citations: 9**	**G.s. citations: 15**

<u>Peer reviewed conference proceedings</u>:

Zámborszky J and Priami C (2010), BlenX-based compositional modeling of complex reaction mechanisms, in *Electronic Proceedings in Theoretical Computer Science (EPTCS).19:85-102.*

International conferences:

Zámborszky J and Priami C (2010), BlenX-based compositional modeling of complex reaction mechanisms, *3rd Workshop 'From Biology To Concurrency and back'*, Paphos, Cyprus. (oral presentation)

Romanel A, Ballarini P, Jordán F, Larcher R, Lecca P, Mazza T, Mura I, Palmisano A, Sedwards S, **Zámborszky J**, Csikász-Nagy A, Tradigo G (2009) Analyzing the effect of noise on various models of Circadian Clock and Cell Cycle coupling. *International Modeling Competition at Leibniz-Zentrum für Informatik, Germany's Schloss (Dagstuhl), 1st Prize.*

Zamborszky J, Hong CI, Csikasz-Nagy A (2009), Effects of DNA damage response on circadian rhythms, *XI. Congress of The European Biological Rhythms Society*, Strasbourg, France. (poster presentation)

Zamborszky J, Hong CI, Csikasz-Nagy A (2009), Computational analysis of effects of DNA damage response on circadian rhythms, *FEBS-SysBio2009*, Alpbach, Austria. (oral and poster presentation)

Hong CI, **Zamborszky J**, Csikasz-Nagy A (2008), Computational analysis of effects of DNA damage response on circadian rhythms, *ICSB2008*, Gothenborg, Sweden. (oral and poster presentation)

Zamborszky J (2008), G1/S transition and size control in budding yeast cells, *Computational and Systems Biology Course* at CoSBi, Trento, Italy. (poster presentation)

Zamborszky J, Hong CI, Csikasz-Nagy A (2007), Connection Between the Cell Cycle and the Circadian Rhythm in Mammalian Cells, *32nd FEBS Congress (Molecular Machines)*, Vienna, Austria. (poster presentation)

Zamborszky J, Hong CI, Csikasz-Nagy A (2007) Connection Between the Cell Cycle and the Circadian Rhythm in Mammalian Cells, *7th Young Scientist Forum (Molecular Networks)*, Vienna, Austria. (oral presentation)

Zamborszky J, Hong CI, Csikasz-Nagy A (2007), Coupling between the cell cycle and the circadian rhythm in mammalian cells, *Modeling Complex Biological Systems in the Context of Genomics*, Evry, France. (poster presentation)

2 State of the Art

2.1 Systems Biology: an interdisciplinary approach

Systems biology is a relatively new discipline of biological sciences that focuses on understanding complex systems by studying them as a whole [3]. Researchers have realized that finding the elements (e.g. molecules) of biological systems and knowing their properties will not let us understand how their intertwined reactions give rise to their function unless we analyze them as a complex network. But instead of only looking at a complicated picture and trying to guess, scientists turn into a more sophisticated analysis. They switch on their computers and build models to gain more information and produce interesting predictions that might be tested experimentally. Through the union of theoretical and experimental biology, we hope to find novel results that may also have an impact on healthcare or drug discovery. Quantitative modeling requires the contribution of an interdisciplinary field including biologists, mathematicians, physicists, engineers and computer scientists. The cycle of systems biology research is the following: (1) one should propose a question based on previous experimental results. Thus, we collect the pieces of the puzzle we would like to play with (2) the next step is the model building process that is based on our previous knowledge (3) then analysis of the model (4) and we answer questions or make predictions (5) test the hypothesis, thus we gain additional knowledge of the system (6) and refine the model and start the cycle again.

In these days, enormous amount of tools are accessible, although it was not always like that. Even before digital computers became available, theoretical models were solved and simulated on analog machines [49,50]. The problem of biological self-organization - how steady state systems can create structures, oscillations and waves - was always a challenge. One of the first theoretical examples of complex behavior emerging in a biological system of was proposed by Max Delbrück in the 40s [51]. His concept of bistability (the possibility of being in either one of two different states under the same conditions) was used to explain differentiation: how cells of identical genotypes (the coded genetic sequence) grown in identical environments can result in different phenotypes (the physical manifestation). Later Jacque Monod, Francois Jacob [52], Rene Thomas [53] and others [54-57] formalized the requirements for positive and negative feedback loops and their findings have shown the relation between the feedback structure of a system and the biological phenomena of homeostasis and differentiation [58]. In the 50s, one of the first mathematical models of the field brought Nobel Prize to Alan Lloyd Hodgkin and Andrew Fielding Huxley [59]. They carried out a series of measurements and used complex mathematical models to propose how impulses are formed along the axon of neuronal cells. It was a breaking result as the molecular details of that system was unknown that time and they were able to publish novel findings thanks to their unique approach. They became the pioneers of systems biology and brought the focus on the potential power in theoretical biology. At the same time, Turing started to study the phenomena so called symmetry breaking [60]. Others showed how non-linearly interacting chemical processes develop complex behavior, such as oscillations [61,62]. Chemical oscillators of the mixture of some reactants became the first and classical example of non-equilibrium thermodynamics [63] and the tools developed to analyze chemical reactions (in the field of theoretical physical chemistry) appeared to be helpful in biological research as well. In the following years, Denis Noble developed the first computer model of the heart pacemaker [64]; and from the 60's, we can find several fascinating discoveries of some of the rules that determine the observed physiology of cells [65]. The large amount of data produced after the birth of functional genomics was asking for both data storage and the comprehension of the role of biological molecules. Thus, informatics has kept up with these requirements and as a result, bioinformatics and computational modeling have emerged [5].

The increasing number of details led to more complex kinetic models which gave way advanced computational methods (first for ODEs) and abstraction of complex reactions (e.g. enzyme kinetics) as well. Several crucial qualitative properties of biological systems have been explained by dynamical systems theory and deterministic approaches [8,66]. However, other simulation formalisms have been also developed, e.g. Petri nets [67,68], transformational grammars [69] and process algebras [70]. An initiative to reduce the size of the possible states of a system is based on logical modeling with Boolean algebras [71-75]. The birth of novel computer science approaches brought novel analytic techniques into systems biology. For instance, model checking has evolved from testing safety requirements in hardware or software systems to check and verify if a biological model matches some specific conditions [76,77]. As novel experimental techniques have been developed, the need of modeling few numbers of molecules or a single cell behavior turned up. Stochastic models describe events occurring probabilistically. Randomness is present and the states of components are expressed by probabilistic distributions rather than by unique values.

Biological systems are composed of molecules acting in chemical reactions and they create a complex network resulting in proper physiological function. These well-orchestrated circuits describe common regulatory modules [78]. Their function cannot be easily predicted by studying the properties of the isolated components. Rather we try to understand their design principles as a whole. Modular approaches are inevitable tools to reveal the task of biological structures. Computational systems biology can contribute into this work. In the subsequent, an introduction into computer science inspired formalisms is presented. These methods have been successfully used to model biological systems and reveal novel insights in future systems biology research as well.

2.2 Computational systems biology

Ordinary differential equations (ODEs) are the most widespread and classical formalisms to model dynamical systems in science. They represent a mathematical description of biochemical reactions with rate equations. ODEs are simulated through numerical integration methods. They are mainly applied to describe population dynamics within a deterministic framework, however stochastic extensions are also in use. When stochastic effects are important, other computational structures, such as continuous-time Markov chains (CTMCs) [37], are also available for modeling biological systems [73-75].

We distinguish computational and mathematical models because their basic concepts differ. Computational models function as executable algorithms and not just simply solvable equations. The comparison of the two points of view has been recently summarized in [28] and in [29]. Execution means that we can predict the flow of control between molecules and reactions making novel analytic techniques available (e.g. model checking, analysis of the causality relation among the events, etc.), while ODEs describe only the outcome of the system through its evolution over time.

Abstractions of computer science entered the field of systems biology with a class of formal languages that enable elegant and precise description of biological interactions. Hereinafter, the reader will be introduced to the various computational tools used in the field. The list is enormous, thus only the, let's say, most popular ones are mentioned in this study.

2.3 Computer science formalisms in systems biology

Abstract computer languages often form a class of generalized approaches towards modeling larger biological systems with ignoring their details. These initiatives allow only the study of common properties of networks but suffer from limited predictive power. The models created within these computational frameworks are usually top-down systems providing abstractions for uncovered biological interactions.

Directed graphs are one of the modeling concepts that are widely used for gene regulatory networks characterizing the system through its elements (called as vertices or nodes) and with a set of ordered pairs (arcs, directed edges, or arrows). Although graphs are simple representations of biological systems, they are applied for predict unrevealed paths between elements and a variety of clustering algorithms have been used to group together the components with similar temporal expression patterns providing high degree of organization and the better understand of genetic networks [51,79].

Bayesian networks [80] describe the structure of a system by a directed acyclic graph. It is a probabilistic graphical model that encodes the Markov assumption. The Bayesian network approach applies statistical analysis for investigating stochastic aspects and noisy measurements. It represents a set of random variables and their conditional dependences. They are used when incomplete knowledge is available about the system. The simple and intuitive representation of such models might be restrictive for dynamical systems. Generalizations - like dynamical Bayesian networks - have been also presented with which feedback relations can be modeled [81].

Boolean networks [82] are further examples of the formalisms applying generalization and simplification in order to reduce complexity of biological systems. These discrete dynamical networks consist of Boolean variables whose state is determined by other variables and they can exist in two states: active (on, 1) or inactive (off, 0). Interactions between elements are represented by Boolean functions. One particular type of Boolean networks, the *cellular au-*

tomata, is a popular approach to model Boolean variables whose state is determined by its spatial neighbors. It can be applied for Genetic Algorithms (GA) whose rules have given rise to sophisticated emergent computational strategies [83]. In simple cases, the attractors and their basins of attraction in the state space are calculated by hand, but for larger systems computer programs are applied. The reduction approach of the Boolean networks employs strong simplifying assumptions on the structure and dynamics of a biological system, thus it allows also large regulatory networks to be analyzed in an efficient way. For instance, intermediate steps are neglected. Also, transitions between the activation states of the elements are assumed to occur synchronously. Thus, certain properties of the system may not be predicted with this tool and there might be situations in which these abstractions are not appropriate and other methods are required. *Generalized logical method* [84-86] is an extension of Boolean networks. Their formalisms are equivalent, however the general logical method allows variables to have more than two values and that transitions between states may occur asynchronously.

Basic Petri nets [67,68,87] are directed bipartite graphs used for describing distributed systems. In the 1960s, Petri Nets have been developed for modeling systems in a formal way. They have an exact mathematical definition of their execution semantics and a well-equipped mathematical theory for process analysis. Nodes represent either places (signified by bars) or transitions (signified by circles). Directed arcs represent the trajectories. They run from a place to a transition or vice versa, but never between places or between transitions. Basic Petri nets have been also applied to biological modeling [88-93]. Places may contain a nonnegative integer number of tokens. Transitions can fire whenever there is a token at the start of all input arcs; when it fires, it consumes these tokens, and places tokens at the end of all output arcs. Transitions can also be equipped with rates, giving rise to stochastic Petri nets (SPN) [68] or colors (Coloured Petri Nets (CPNs)) [94] providing higher levels of abstraction by allowing tokens be marked. The usage of Petri nets is intuitive, but except some special initiatives [95-97] they lack modularity.

Rules can be abstract representations of one or more reactions, thus rule-based languages are also specialized to biological modeling. For instance, *BioNetGen* [98] is a language designed for generating a biochemical network of a set of reactions or a basic Petri net from an abstract, rule-based description.

In contrast to BioNetGen, *Biochemical Abstract Machine BIOCHAM* [15] is a software environment that makes compartmentalization available and the implementation of the type inference tool provides an analytic method for biochemical models through the mathematical formalization of abstractions of the systems. It is composed of rules containing variables for modification site states, for atomic species names and for complexes. Furthermore, it is a language compatible with other initiatives (SBML) and it is equipped with several simulators (Boolean, differential and stochastic). The tool infers kinetic parameters from temporal logic constraints.

Realizing the need for modular rule-based modeling techniques and supportive tools, *Little b* employs a notion of rules [99] at a similar level of abstraction to those of BioNetGen but in a modular way.

Process algebras are abstract calculi originally to specify and formally reason about concurrent computer systems. In the last decades this formal approach have been used extensively providing an additional representation of complex systems and have been also applied for biological modeling. In the following section, an introduction into the evolution of process calculi is presented.

2.4 Process algebras

The theory of simultaneously executed and interacting computations, called concurrency theory, is an active field of computer science. Some examples include communication networks, air traffic controllers, and industrial plant control systems. Process calculi languages are members of a family of computational approaches created to model concurrent systems in a formal way, permitting also the use of algebraic laws to manipulate process descriptions. They provide a high-level representation of interactions, communications, and synchronizations between processes. Their programming structure uses parallel composition of communicating sequential processes that can be executed more efficiently on multiple core units. A variety of process calculi languages originate from Robert Milner's Calculus of Communicating Systems (CCS) [23] and Sir Charles Antony Richard Hoare's Communicating Sequential Processes (CSP) [100]. Thereinafter, I will present the common properties of process calculi languages and some representative of the approach.

In a process calculus abstraction processes are playing a key role in a rules-driven description [101]. They are often symbolized as entities, agents, boxes or others, depending on the specific language and their collection defines the system. Entities execute computations parallel and they interact through channels. They have an internal state and the interaction capabilities of the processes are defined by names. As a result of a communication, the states of the computational units or the affinity of their interaction might change. Primitives are combined by some operators (prefix, parallel composition, choice, restriction, relabeling and the null agent). These syntax-driven rules are automatically implemented providing novel insights into applications for modeling. In contrast to classical methods (ODEs) which are denotational, process algebra descriptions offer a formal way to execute complex systems through their operational semantics.

In the 1990s several stochastic extensions have been emerged for various process algebras when random variables were added into the system characterizing the duration of the computations (actions). In most cases, these random variables are exponentially distributed and rates of actions are introduced for quantifying the models. The systems equipped with a stochastic semantics are associated with a continuous-time Markov Chain (CTMC) [102] where the system remains in the current state for some random amount of time and then step to a different state. The future states of the process depend only upon the present state, giving rise to the Markov property.

Pi-calculus [103] (continuous [104] or stochastic [105]) evolved from Milner's CCS and allows complementary actions to occur and also name-passing is possible, thus the communicating processes can exchange names over channels resulting in novel interaction capabilities. This extension enables modeling of mobile entities.

CCS, the pi-calculus, and all the calculi derived from them provide a well-understood formal mathematical theory and a number of associated tools for verification and analysis; however, previously they lacked biological representations. The application of pi-calculus in the field of biology began in the 20th century, with the work of Aviv Regev, Ehud Shapiro and Corrado Priami [70]. The abstraction of processes communicating and acting parallel has brought novel insights into the application in the field of biology as well. *Biochemical pi-calculus* [24] inherited the process algebra description of a system, thus biological entities are abstracted as processes that interact through channels defined by channel names and co-names. Reactions are represented as actions. Molecules are present as computations. Various application of biochemical pi-calculus have been shown (BIOSPI [24], SPIM [106], etc). In their approach proteins are mobile processes, the ‘messages’ that molecules exchange affect their behavior and the sites of the proteins are named as communication channels. Proteins send and receive messages and protein-protein interactions occur. The system is governed by rules and proteins can also form complexes. Derivation of the pi-calculus is still a minimalist (using small collection of primitives and operators) rule-based language but being appropriate for modeling living system.

Several other process calculi have been proposed to model biological systems. *Performance Evaluation Process Algebra (PEPA)* [107] is also a formal language designed for concurrent programming and later has fruitfully been applied for modeling biological systems. PEPA (or more likely its extension, Bio-PEPA [43]) allows the users to quantitatively model and analyze large pathways. Several techniques can be applied within PEPA, for instance it can be combined with the probabilistic model checker PRISM [108]. It has been mostly used for describing, simulating and analyzing signaling pathways [109]. *CCS-R* is a variant of CCS with the extension for managing reversibility in biology. Reactions are presented as binary synchronized communications, similar to pi-calculus.

Kappa-calculus [110] has a language specialized in encoding proteins that are modeled by an identification name and by two multisets of domains. The first set of domains is visible, while the other contains hidden domains. The two basic primitives of the language are complexation and activation representing protein interactions occurring in cells.

Certain initiatives within the process calculi focus on modeling biological structures, compartements or membranes. For instance, *BioAmbients* [14] have been evolved from Mobile Ambients [111] and provide abstractions for biological compartments. Ambients can be organized in hierarchical way and entities interact through communications in the bound places. This calculus is a suitable tool for representing localization (the movement) and compartmentalization of molecules. *Brane Calculi* is another representative for a computational abstraction inspired by biology and applied for modeling biological systems [112]. It focuses on biological membranes, which are both containers and active entities. Brane Calculi primitives provide properties for membranes such as merge, split, shift or action. Directed actions of membranes are present in Projective Brane Calculus [113], as an extension of its progenitor.

Beta-binders [114] are extensions of pi-calculus and their abstraction is based on the idea of representing bio-processes as boxes equipped with sites (beta-binders). Beta-binders introduced the concept of compatibility [116] meaning that upon communication the types of the interfaces have to be compatible, but actions and co-actions are not required to match precisely to fire. In addition, they are enriched with specific events, such as split or join of boxes and hiding or unhiding or exposing binders. A stochastic extension of Beta-binders for quantitative experiments has been presented in [115]. *BlenX* [117] is a progeny of Beta-binders where Beta Workbench [27] provides a non-deterministic kernel of such models. The BlenX programming language is described in details in Chapter 2.6.

Biological systems’ properties	Beta-binders representation
Entities (mRNA, DNA, proteins, etc.)	Boxes
Interaction capabilities (protein domain)	Binders
Complex formation and dissociation	Binding creation and deletion
Interactions (modifications)	Communications
Dynamics	State change

Table 1: Representation of biological systems with Beta-binders.

2.5 Compositionality, a challenge in systems biology

“Anything that deserves to be called a language must contain meaningful expressions built up from other meaningful expressions.” by Zoltán Gendler Szabó [118].

The opportunity of the construction of a system starting from the definition of submodules is called compositionality and it is a common property of languages in general. Programming languages are combined of basic primitives. Compositional modeling was originally implemented as a framework for constructing adequate device models with the composition of physical devices (Device Modeling Environment, DME [119]) that emerged from the principles of a compositional modeling language (CML) [120]. Several initiatives have already investigated the question of biological models in a systematic way. Computational approaches have already focused on the definition of common motifs in different research fields providing novel insights into methods that may enhance the difficult, error-prone and time-consuming process of model composition. All proposals approach the problem from a different point of view and they add novel meaning to the original question into the model-building process. In the following section, the reader is introduced to the methods and research directions emerged on the field of computational biology concerning compositionality.

One of the solutions to ease the modeling process was proposed by Falkenhainer and Forbus [119], namely the authors suggested a creation of a collection - a library - of physical model fragments. A few computational tools using classical approaches (e.g. ODEs) - thus non-process algebra-based approaches - have already defined libraries of frequently used motifs in the field of biology (e.g. COPASI [9]). These libraries contain abstract biochemical reactions applying several assumptions. Concentrating on signaling networks, Saez-Rodriguez defined submodules for creating modularization based on network theory within the tool *ProMoT* [121]. In [122], the authors developed a method for the design of genetic circuits with composable parts. Each part is modeled independently by the ordinary differential equations (ODE) formalism and integrated into the software ProMoT (Process Modeling Tool). They realized a ‘drag and drop’ tool for genetic circuits. Furthermore, *SBMLsqueezer* [123] facilitates modeling via automated equation generation, overtaking the highly error-prone process of manual assign of kinetic equations to the biological systems. This approach provides an automatic derivation of the kinetic equations starting from the stoichiometric relation between the reagents visualized on a diagram. SBMLsqueezer helps to simplify the modeling process and it applies complex rate functions within the deterministic framework with compound mathematical terms.

Compositionality is mainly discussed as one of the key features of process calculus tools that enable the composition of processes through basic primitives of the language. A suitable conceptual tool to attack complexity has already been successfully used in the process algebra field to model biological systems as an issue of model-construction from elementary reactions with the basic operators by Blossey et al. [40]. The authors presented an approach for constructin*g* dynamic models for the simulation of gene regulatory networks from simple computational elements, called “gene gates”. These gates define an input/output relationship corresponding to binding states and the modularity of the approach creates another level of description for biological systems. The properties of each gate are defined by a set of abstract kinetic reactions (e.g. through Boolean expressions) that reduce the complexity of a model. Michael Pedersen's definition of minimal flows in Petri nets can be also applied as an example for modular rule-based modeling approach in the Petri net-based Calculus of Biochemical Systems (CBS) [124]. Being able to carry out analyses in a compositional way allows much larger models to be handled efficiently. However, the methods described above disregard the crucial nonlinear behavior originating from complex reaction schemes in biological systems.

One remarkable research direction of model-composition lays behind the idea of the translation of a modeling approach to another. The need for composing different languages within one framework and the combination of different mathematical representations (ODE, CTMC, etc...) is a crucial property for systematic modeling of large biological networks. Bortolussi and Policriti [125] defined a syntactic procedure that translates programs written in stochastic Concurrent Constraint Programming (sCCP) into a set of Ordinary Differential Equations (ODEs), and viceversa. Jane Hillston and co-workers established similar connection between the ODEs and process calculus approaches [126,127] . Furthermore, they [128] have focused on generating an aggregated CTMC in a compositional way, tackling with the state space explosion and with the implementation of an efficient algorithm that recognizes symmetries and avoids unnecessary computation within the PEPA Workbench. Their publication demonstrates how compositionality may be exploited to reduce the state space of the CTMC in the PEPA framework. The methods mentioned ahead all focus on language translation, but they disregard the problem of implementing the frequently used complex mathematical expressions of deterministic models.

Hybrid methods have been proposed to solve the impermeability between different approaches. Bockmayr's hybrid concurrent constraint programming is an example of a highly expressive, compositional language with a well-defined semantics [129]. A proposal for connect classical approaches and rule-based languages has been also shown in Biochemical Abstract Machines (BIOCHAM) [15]. This tool achieves simulations and it queries the model in temporal logic. Biochemical systems described by differential equations are handled in a hybrid framework using time discretization methods, and it is combined with Boolean models. The language is able to represent multi-molecular complexes and localization of proteins (compartments where species are). In addition, Hermanns and Herzog constructed large Generalized Stochastic Petri Nets (GSPN) by hierarchical composition of smaller components in real hardware and software systems inspired by process algebraic operators [130]. The Bionet [131] tool is a novel approach for biological pathway modeling based on hybrid intelligent systems (fuzzy logic, neural nets, genetic algorithms, and statistical analysis) for the computational exploration of new drug targets.

The need of a bridge linking different modeling approaches has been recognized previously and as it was mentioned, few initiatives have already been proposed on the field [39-43]. However, the studies described above paid less attention to the specific problem of the presence of nonlinear functions within stochastic models. The direct use of complex rate functions has been implemented currently into the BlenX language (in Chapter 2.10.1) requiring proof and verification of their use in all modeling case studies. In order to solve the problem of the presence of non-elementary reactions that are assumed to be elementary in Gillespie's stochastic simulation algorithm, systematic unpacking of often used nonlinear terms into single-step reactions are shown in this thesis. Challenges arise from the disappearing nonlinear behavior upon the decomposition of some complex terms often used in biological modeling [132]. This thesis will focus on an improvement of compositionality in the BlenX language by definition of frequently used submodules. First, the evolution and the description of the process calculus language BlenX are shown.

2.6 The BlenX programming language for biology

BlenX (Biology encoding language) [117] is a stochastic programming language for modeling biological systems in a formal way. It was inspired by Beta-binders [114] and it has been designed for biology from the beginning. It offers a high-level description of interactions, communications, and synchronizations between molecules or processes. BlenX also offers an opportunity to define algebraic laws and as other process calculi, it provides a formal specifications of concurrent systems (the molecules of a biological system) executing their tasks in parallel and able to synchronize with each other. BlenX models define the possible properties of the various elements of the encoded systems. Calculi contain syntax-driven rules, the so-called operational semantics [101] that can be automatically implemented in the Beta Workbench framework (BWB [27]).

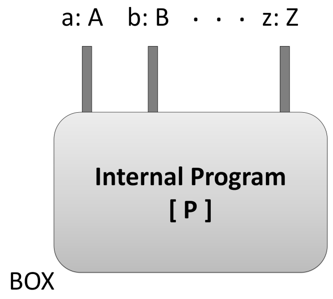


Figure 1: Graphical notation for abstract biological entities in BlenX.

BlenX allows the user to create *boxes* to represent biological species. Boxes have well-defined interaction sites (called *binders*) and internal structure (*behavior*). The box shown on Figure 1 is defined by its internal process, *P*, and types *A*, *B*,…, *Z*. The types discriminate among possible and prohibited interactions based on compatibility [116].

The declaration for boxes containing the rules that encode the entity is

let Box : bproc = #(a:0,A), #(b:0,B),…, #(z:0,Z) [P];

The behavior of a biological system is given by the ordered sequence of actions and reactions (complementary actions or simply coactions) that the program can perform leading to the biochemical interactions between the elements. Actions for instance can occur when binders "sense" signals (receive an input) and propagate signals (send an output) and the internal structure codifies for the mechanism that transforms an input signal into the change of the box (e.g. activation (*unhide* or *expose*), deactivation (*hide*) or changing the type of a binder (*ch*)). To denote such a chain of events, the *action prefix operator* is used, which is written as an infix dot (*a!().P*). Signals are sent over a channel named *a* (*a!*) or waiting for a reply over a channel named *b* (*b?*). Operators in Table 2 (e.g. sequentialization, parallel composition, name declaration, recursion, bang and deadlock operator) are made up to compose elementary actions over distributed channels. Parallel composition (denoted by the infix operator "|", as in *P* | *Q*) allows the description of processes that may run independently in parallel. The process *P* + *Q* behaves either as *P* or as *Q* and the selection of one process discards the other forever. To represent a situation, where the process is unable to perform action or co-action, the *nil* (deadlock) operator is used. Replication operator (*rep*) is a typical operator of process calculi that ensures the process sends a signal each time it is needed, allowing infinite behaviors to happen. Events specify statements to be executed with a rate and/or when some conditions are satisfied. Boxes are able to born (when a *new* box is synthesized) and to die (when a box is *delete*d) as biological entities. Boxes are merged or split upon different conditions (*join* and *split* events). They can also form complexes through their binders and dissociate depending on the state of the overall system.

parallel composition of processes	*P | Q*
choice	*P + Q*
sequentialization of interactions	*a().P*
specification of which channels to use for sending and receiving data	*b?().P* or *b!().P*
recursion or process replication	*rep P*
deadlock	*Nil*

Table 2: Basic operators of process calculus languages.

Following the BlenX metaphor, we look at genes, proteins, and other biological entities (depending on the level of abstraction) as independent processes that can communicate and interact with each other. These interactions between the communicating entities give raise to the complex network of biochemical reactions taking place inside an organism. The effect of an interaction between the components can change the future behavior of the whole system. Simulations of BlenX are based on Gillespie's stochastic algorithm [117]. In addition to model execution, various other methods have been proposed to analyze pi-calculus models (e.g. causality and concurrency analysis [133] or model checking [76]). In the following, the BlenX's features supporting the modeling process of biological systems are summarized.

2.6.1 Complexes

The boxes of BlenX are able to form (or break down) complexes through their binders. The typed interfaces represent their interaction capabilities. The boxes bind over their interfaces with certain sorts enabling specific reactions to occur (through the creation of a link that only they can use) (Figure 2). The affinities of binders to form complexes are declared in a separate file in the following way:

(A1 , A2 , rate (k) , rate (l) , rate (m));

where *A1* and *A2* are two binder types that are capable to form complexes. The first rate (*k*) is referred to as the association rate, the second (*l*) is the dissociation rate and the third one (*m*) is the rate of communication firing upon binding. Thus, bindings between interfaces enable processes to communicate and actions to occur (Section 2.6.2). For instance, biological systems often form intermediates (e.g. enzyme-substrate complexes) which allow a catalytic step to occur during the substrate is turned into a product.

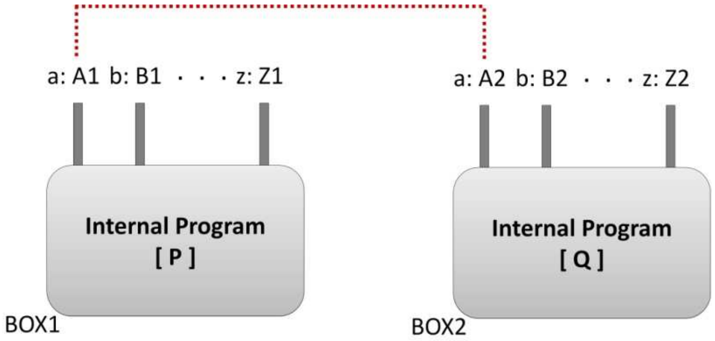


Figure 2: Graphical representation of boxes forming complexes. Binding of BOX1 and BOX2 might occur through their binders named *a* with the types *A1* and *A2*, respectively.

2.6.2 Communications

Processes can perform actions when the primitive *b()!.P* sends a signal - enclosing the process *P* - through the interface (*b,B1*), while, a box with a primitive *b()?.Q* waits for a signal on the binder (*b,B2*). When communication happens, the two boxes synchronize each other and execute the sequential process in their internal behavior (Figure 3). Thus, boxes can execute an action theirselves and for example, with the primitive *change*, they transform theirselves into the products.

In the previous section (Section 2.6.1) biochemical reactions were introduced through complex formations and the following modifications. However, the expressive power of BlenX also enables communications to happen without formed links. If the rates *k* and *l* (used in Section 2.6.1) are both zero, but *m* has a value greater than zero, then binding and unbinding are not contemplated for the pair of types and the boxes exposing them can communicate without the need of first creating a link. The boxes indeed can use an intra-communication without creating an intermediate complex.

Intra-communications occur on perfectly symmetric input/output pairs that share the same subject, while inter-communication can occur between primitives that have different subjects provided that their binder identifiers are compatible. This new concept of communication is a special extension of the language for modeling biological systems where interactions occur based on their affinity and molecules can react with several reactants in the same context.

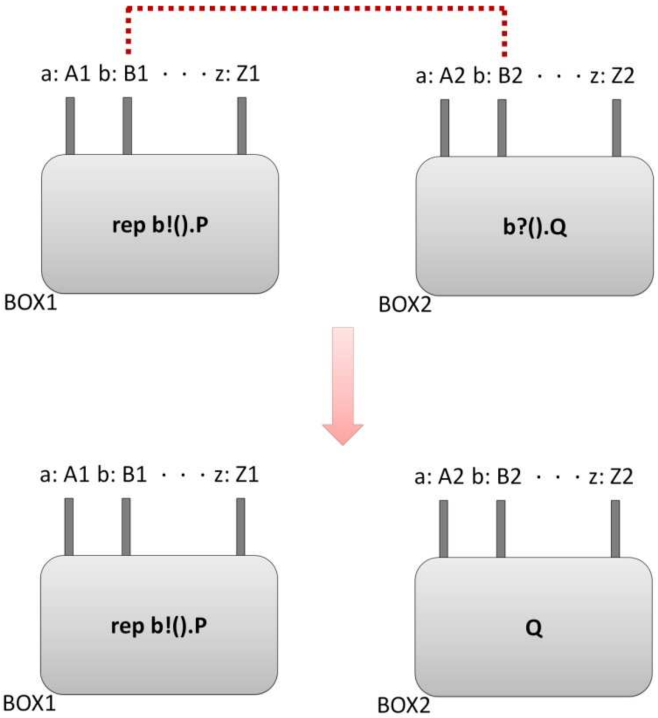


Figure 3: An example for a communication-driven reaction in BlenX.

2.6.3 Events

BlenX offers an abstract description of biochemical reactions through events. Events provide a solution for modeling non-elementary steps as well. They encode rewriting rules that substitute a set of boxes with another set of boxes. The abstraction of events allows the users to define reactions without specifying complex formation or details - intermediate steps - of the complex reaction. For instance, complex formation of the boxes can be modeled as the substitution of the two components into a third box, representing their dimer (*join* event in

Table 3). The *split* event provides dissociation of the complex into the single elements or even into modified molecules. In the later case a multi-step reaction is described by an abstract single step. Synthesis and degradation of the boxes (*new* and *delete* events, respectively) can be also coded with events. The use of events can be combined with real complex formation rules and with communications.

$Box1 \overset{k}{\rightarrow} Box2 + Box3$	*when(Box1::rate(k)) split(Box2,Box3);*
$Box1 + Box2 \overset{k}{\rightarrow} Box3$	*when(Box1,Box2::rate(k)) join(Box3);*
$\overset{k}{\rightarrow} Box$	*when(Box::rate(k)) new(1);*
$Box \overset{k}{\rightarrow}$	*when(Box::rate(k)) delete(1);*

Table 3: Representation of events in BlenX.

2.6.4 Conditions

The execution of primitives may depend on specific conditions. In this case, processes are fired by checking the state of the box. Conditions allow the definition of general rules valid for a biological entity. For instance, the state of the molecule specified by the type of the binders can lead to the execution of an action.

The condition

pproc = if (not (a , bound)) then b?().nil endif ;

will let the binder *b* receive signals if the binder *a* is not bound to another box. Thes rules-driven and component-based descriptions of biological networks offer a novel computational systems biology approach that differs from the classical, equation based modeling tools.

2.6.5 Conditional events

In order to allow the user to perturb the system through events, conditional events are implemented in the language. For instance, events can be written in a time dependent way, e.g. accounting for an event of *Box* deletion at the time point 100 such as

when (Box : time = 100.0 : inf) delete (1) ;

2.7 Compositionality with BlenX

Compositionality is a key feature of process algebra tools. As the size of the revealed interaction network increases, the modelers wish to link the discovered interconnected subsystems together, there is an urgent call for a framework that supports extension of models in a cumulative manner. Introduction of rule-based and process algebra modeling was a departure from classical dynamical approaches, such as ordinary differential equations (ODEs). ODEs require the explicit report of all interactions that occur in time, while rules-driven approaches can be used to generate biochemical reactions. If we would like to extend the reaction scheme with a novel role of an element in a model described through ordinary differential equations, we do not only need to add new equations, but we also have to modify the existing ones. While process calculi models rely on modeling with the concept of compositionality.

BlenX inherited the basic properties of process calculi, such as the key primitive that distinguish the process calculi from sequential models of computation. Parallel composition of processes (see the detailed description ahead) makes compositionality a crucial feature of biological modeling in BlenX. Parallel composition of two processes *P* and *Q* are written as *P* | *Q* and it allows computation in *P* and *Q* to proceed simultaneously and independently, and it also allows interactions to occur. Model composition and extension are proposed to be easier with BlenX than with classical modeling methods. However, there are several initiatives that improve the compositionality of ODE systems based on some building blocks (e.g. the collection of reactions in COPASI [9]), while the current representation of BlenX language remains a tool for only experts in the computer science field with programming skills. Compositionality on the other hand means that a model can be built and analyzed by dividing it into smaller submodels that are easier to understand. It is a crucial property and enables the construction of large systems. Compositionality focuses on the basic operations and complex behavior that the model can perform. It is surely the key issue needed for systematic biological modeling to become effective. It allows to fix the building bricks of systems and to enlarge models by composition without large changes in the description of the subsystems that are already available.

Pre-defined modules and the collection of frequently used complex reaction schemes provided in ODE systems ease modeling within the deterministic framework, but cannot be applied under the stochastic simulations of BlenX as there is no direct translation of ODE models and the stochastic simulations applied in BWB are often found to be inappropriate for complex reaction schemes (discussed in Chapter 3). Thus, there is a need for a solution of a tool that inherits the properties of extendable process calculus approaches and provides building bricks that makes the modeling process and analysis of the system more effective. In the subsequent, I present the stochastic simulation algorithm used for BlenX models and after that I point out the problems we have to deal with compositional modeling within the currently available BlenX language.

2.8 Stochastic simulations with BlenX

To analyze the system after the model building process, we follow its dynamics over time starting from a set of initial conditions. The deterministic modeling approach regards the time evolution of chemical reactions as a continuous, predictable process that is governed by a set of coupled reaction-rate equations, also referred to as the system’s kinetic description. Several experimental studies showed the importance of noise in biological systems [19-22,134] bringing stochastic simulation techniques into the focus of theoretical biology. The stochastic approach to chemical kinetics was first described by Delbrück in the ’40s [51] and Novick and Weiner [135] showed that at the low inducer concentrations used in their experiments the population of cells consists essentially of individual bacteria that are either making enzyme at full rate or not making it at all. The basic assumptions of stochastic reactions are that a chemical reaction occurs when two (or more) molecules of the right type collide in an appropriate way, and that these collisions in a system of molecules in thermal equilibrium are random [37]. Noise in biology is usually represented by assuming that the time evolution of the system is a random-walk process which is governed by a single differential-difference equation called the chemical master equation (CME). Monte Carlo procedure is used to numerically simulate the time evolution of a given species. The most famous method to simulate a set of chemical reactions is the stochastic simulation algorithm (SSA) of Gillespie [37]. The BlenX framework also applies the method presented in the following section.

2.9 Gillespie's stochastic simulation algorithm

In the '70s, Gillespie published an exact solution to generate the stochastic time-evolution of a biochemical system as a random-walk process [37]. It probably became the most popular computational method for stochastic simulations in systems biology. Gillespie developed a variant of Monte Carlo simulations assuming that the system is well-stirred and molecules are randomly distributed. In this way, the produced exact numerical calculation within the framework of the stochastic formulation is a relatively simple digital computer algorithm that correctly accounts for the inherent fluctuations and correlations that are necessarily ignored in the deterministic formulation. It describes the transition of a system from one state to another through changes of the probability of being in a certain state. BlenX refers to an efficient variant of the stochastic Gillespie's algorithm for simulations [27].

In Gillespie's approximation, the reactants of the system ($S_1, S_2, \ldots, S_n$) are randomly distributed in a fix well-stirred volume (V) and they collide in a random manner, assuming that molecules are in thermal equilibrium. Initial quantities of the reactants are defined as $X_1(t), X_2(t)\ldots X_n(t)$ at $t=0$. Not every collision results in a reaction. Depending on the state of the molecules, the collisions form a stochastic Markov process characterized by "collision probabilities per unit time". The theory assumes that nonreactive molecule collisions occur much more frequently than successful ones. The chemical reactions are described by stoichiometric equations. Suppose that the species interact via m reaction channels ($R_1, R_2,\ldots,R_m$) and that these reactions occur with individual propensities ($c_1, c_2,\ldots,c_m$). The average probability of a molecular pair (X_1 and X_2) that will react according to a reaction R_j in the next infinitesimal time interval ($t, t+dt$) equals to $c_j \cdot dt$; where c_j is the stochastic reaction rate constant (or basal rate). To define when the next reaction occurs and which reaction will occur, Gillespie calculated a combinatorial function h_j that specify the number of all possible reactant combinations for reaction R_j. The probability that an R_j reaction will occur in the interval ($t, t+dt$), given that the system is in the state ($X_1, \ldots, X_n$) at time t is

$h_j \cdot c_j \cdot dt = a_j \cdot dt$ (a_j is denoted with propensity value). To numerically simulate the stochastic time evolution of a biochemical system, Gillespie presented an exact solution, called exact stochastic simulation. The reaction probability density function (*P(T,j)*) provides the probability that, given the state (X_l,. . .,X_n) at time *t*, the next reaction (*Rj*) in *V* will occur in the infinitesimal time interval (*t, t +dT*) and the analytical expression now comes to the set of random pairs whose probability distribution equals to $P(T, m) = a_m \cdot e^{(-a_0 T)}$. The calculation assumes that reactions are elementary steps (involve one or two reactants), thus there is no reaction occurring in the time interval (*t, t+T*).

$$P(T, m) = \begin{Bmatrix} a_m \cdot e^{(-a_0 T)} \; if \; 0 \le T < \infty \; and \; m = 1, \ldots, M \\ 0 \; otherwise \end{Bmatrix}$$

where

$$a_m \equiv h_m \cdot c_m$$

and

$$a_0 \equiv \sum_{i=1}^{M} a_i \equiv \sum_{i=1}^{M} h_i \cdot c_i$$

The stochastic simulation algorithm generates two random numbers (r_1, r_2) that defines the pairs of *T* and *j* according to the probability density function:

$$T = (\frac{1}{a_0}) \cdot \ln (\frac{1}{r_1})$$

and

$$\sum_{i=1}^{j-1} a_i < r_2 \cdot a_0 \le \sum_{i=1}^{j} a_i$$

Note that in every state of the system the time to the next occurrence of reaction R_j is a random variable following a negative exponential distribution. The validity of this fundamental hypothesis of Gillespie has been queried by Ivan Mura [136] with a simple mathematical argument on complex reactions including multiple elementary reactions. Meaning that even if time to the next occurrence of each elementary reaction follows a negative exponential distribution, the time to the occurrence of the abstract reaction will not have exponential distribution that Gillespie's SSA assumes. This problem emerges in bio-inspired process calculi that are equipped only with exponential distributions. Note that there has been a proposal for BlenX including the use of general distributions investigated in special biological cases [137], although other solutions are still needed that enable compositional modeling of complex reactions in a sophisticated way.

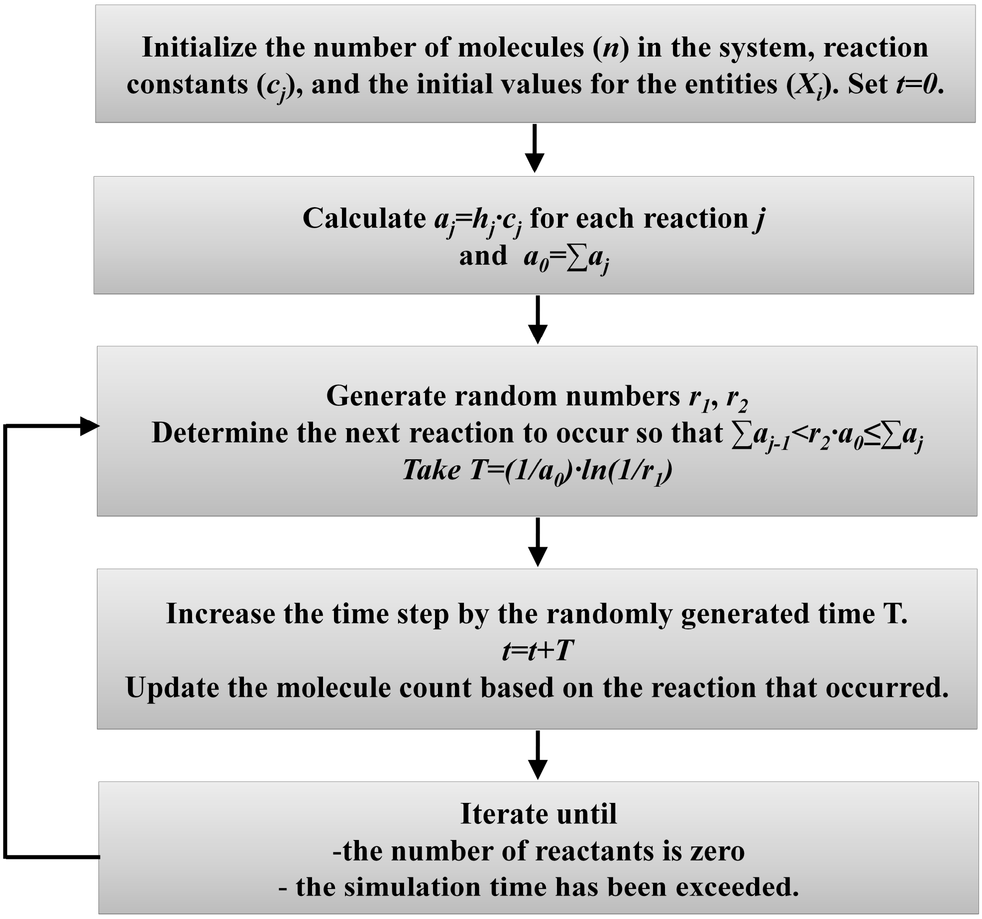

Initialize the number of molecules (*n*) in the system, reaction constants (c_j), and the initial values for the entities (X_i). Set *t=0*.
Calculate $a_j=h_j{\cdot}c_j$ for each reaction *j* and $a_0=\sum a_j$
Generate random numbers r_1, r_2
Determine the next reaction to occur so that $\sum a_{j-1}<r_2{\cdot}a_0\leq\sum a_j$
Take $T=(1/a_0){\cdot}ln(1/r_1)$
Increase the time step by the randomly generated time T.
$t=t+T$
Update the molecule count based on the reaction that occurred.
Iterate until
-the number of reactants is zero
- the simulation time has been exceeded.

Figure 4: Gillespie’s stochastic simulation algorithm.

2.10 Reaction rates in BlenX

In biochemical reaction kinetics, the rate of a reaction - the speed at which the concentration of reactants or products change - is defined by the law of mass action kinetics assuming that the system is homogeneous and chemical reaction rates of elementary reactions (that proceed through only one transition state) are proportional to the concentrations of the reactants.

An example for a chemical equation is: $A \overset{k}{\rightarrow} C$

where *A* is the reactant and *C* is the product, while *k* defines the reaction rate constant.

The deterministic, kinetic description of the reaction characterized by the velocity of product formation is

$$\frac{dC}{dt} = k(T) \cdot [A]$$

where *k(T)* is the reaction rate constant that changes with the temperature and *[A]* stays for the concentration of molecules *A*. The units of the rate coefficient depend on the global order of the specific reaction. In this first order example (where one reactant is converted into a product) the rate constant's unit equals to *[1/time]*. In case of a second order reaction (e.g. when $A + B \rightarrow C$) it equals to *[(1/(numbers of molecules per unit volume) per time)]* unit.

Reaction rate constants are crucial properties of biochemical systems as they quantify the speed of each reaction. Rate constants could be measured in some cases but it is often a missing property of biological models. The parameters of the system are mostly derived from measurements or estimated through computational parameter inference algorithms. Several methods have been introduced to assist different computational approaches (*PET* [138], *SBML-PET* [139], *KInfer* [140], etc).

In case of reaction rates having exponential distribution and when the model has finite number of states, the BlenX program gives rise to a continuous-time Markov chain (CTMC) [37]. Evolution of the model is generated by Monte Carlo sampling methods and the transition between states is labeled with the stochastic reaction rate. Intrinsic noise in the model is implemented by the usage of random numbers. In this way, BlenX models are executed through Gillespie's stochastic simulation algorithm (SSA) in which rate constants are defined as specific probability rate constants. In general, when the molecules of the system collide in an appropriate way, the SSA calculates the occurrence of reactions in thermal equilibrium that take place in a random manner. Thus, in a stochastic framework, a reaction probability density function (reaction probability per time unit) is used to compute the probability of an action to occur that depends on the stochastic rate constants and the number of molecules present in the system (Section 2.9). For each reaction channel *Rj* the propensity function is defined as

$$a_m \equiv h_m \cdot c_m$$

such that a_m is also called as the *actual rate* and h_m is the number of distinct reactant combinations for reaction R_m and c_m is a constant (called *base rate*) depending on physical properties of the reactants. The way of computing the combinations and the actual stochastic rate varies with the type of the reactions. The stochastic rate constants usually can be derived from the widely used (and measured) deterministic rate constants through a conversion factor [37]. From the practical point of view, this conversion is straightforward and depends only on one factor, although from the theoretical point of view the difference between the two constants is much more complicated (discussed in [37]).

The conversion of the deterministic reaction rates into stochastic ones is implemented in two steps. First, the concentrations of the deterministic system are translated into molecule numbers through a scalar constant α that depends on the volume of the system (*V*):

$$\alpha = 1/(N_A \cdot V)$$

where N_A is Avogadro's number, a scaling factor between macroscopic and microscopic systems expressing the number of elementary entities per mole of substance. It has the value

$$6.022 \cdot 10^{23} mol^{-1}$$

Transformation from concentrations into the number of molecules is carried out as

$$N = n \cdot N_A = c \cdot V \cdot N_A = \frac{c}{1/(N_A \cdot V)} = \frac{c}{\alpha}$$

where *c* is concentration with the molar concentration unit called *molarity* (*mol/liter*) and *N* is the number of molecules with the unit of 'number' that we will denote with a number sign (#).

reaction	**order of reaction**	**reaction rates (deterministic)** $(\boldsymbol{concentration\ unit/time})$ **unit**	**stochastic reaction rates** $(\#/\boldsymbol{time})$ **unit**
$\rightarrow A$	zero	k $(concentration\ unit/time)$	$^{k}/_{\alpha}$ $\#/time$
$A \rightarrow B$	first	$k \cdot [A]$ $(1/time) \cdot [concentration\ unit]$	$k \cdot \|A\|$ $\left(\frac{1}{time}\right) \cdot \|\#\|$
$A + B \rightarrow C$	second	$k \cdot [A] \cdot [B]$ $(1/(time \cdot concentration\ unit))$ $\cdot [concentration\ unit]^2$	$k \cdot \alpha \cdot \|A\| \cdot \|B\|$ $(1/(time \cdot \#)) \cdot \|\#\|^2$

Table 4: Conversion of deterministic reaction rate constants into stochastic reaction rate constants. α is inversely proportional to the volume *V*.

In the stochastic interpretation we have to note that in case of multimerization, the stochastic probability of a reaction to occur differs not only in a conversion factor from the deterministic case. Thus, if the reaction R_j has a scheme $\mathrm{A}_1 + \mathrm{A}_2 + \ldots + \mathrm{A}_\mathrm{n} \rightarrow \mathrm{C}$ with only *n* number of a single reactant forming complexes, the reaction rate in the stochastic case is described via a combinatorial function instead of a multiplication:

$$k \cdot \frac{n \cdot (n-1)}{2}$$

This makes the term describing the reaction in the deterministic framework different from the stochastic one.

In addition to the rate constants described by the mass action kinetic law, we find several other popular characteristic of experimentally observed phenomena. These reaction schemes rely on approximations and they are defined by a complex mathematical term. The introduction into nonlinear rate equations is presented in the following section, while detailed description of some concrete examples are shown later (in Chapter 5).

2.10.1 Rates of non-elementary reactions

In case of reactions that occur in a single step (elementary), experimental measurements reveal a simple linear relation between the reactants and the reaction rate. However, biological systems are much more complex than that. There are several biochemical mechanisms where researchers have observed multi-step reaction or the nonlinear response of the rates to changes in the concentration of reactants. We can say that most biologically observable reactions are not elementary. In fact, most reactions take place by a complicated set of steps. Thus, the kinetics of the reaction - and the rate law - may not simply depend on one rate constant, and may not have a simple order.

A famous example of such behavior was the observation that the reaction rates of some enzyme catalyzed reactions can be described with a saturation curve. Michaelis and Menten approximated the velocity of these enzymatic reaction schemes with a nonlinear function named after them [141]. Their assumptions became the landmark of a simple way for modeling enzymatic reactions and the use of their approximation also ease the determination of the arising constants from measured data. Furthermore most biological reactions are driven by enzymes, so this description can be used to couple enzymatic reactions in a biological regulatory network.

Some enzymes provide a sigmoidal response curve indicating cooperative binding of substrates to the active sites. This behavior is common in multimeric enzymes with several interacting active sites. The first very famous example was the binding of oxygen ligands onto haemoglobin in a cooperative manner altering the affinity of the other active sites for substrate molecules. The Hill equation is applied for modeling these reaction schemes assuming simultaneous binding of the ligands [142].

Besides the reactions mentioned above enormous type of biochemical scenarios exist. For instance, certain hypotheses were built upon the assumptions of earlier works, such as interconnected enzymatic reactions or different inhibition mechanisms (concerning multi-substrate reactions, inhibition and activation mechanisms, allostery, ligand and receptor interactions, scaffold proteins, etc) [143,144]. Nonlinear reactions may lead to more complex behaviors playing crucial roles in biological systems [38]. Positive feedback provides networks (e.g. signaling cascades) the potential for bistability and relaxation oscillations. On the other hand, negative feedback can bring about adaptation and robustness to parameter variations within the feedback loop [145].

The current expressive power of BlenX allows the characterization of elementary reactions and also permits complex reactions to occur in a single step via *events* (Section 2.6.3). The nonlinear terms are characterized and coded by the user during the model-building process. Interconnected regulatory loops and nonlinear reaction terms together might give rise to more complex behavior, such as oscillators [38]. One of the crucial cyclic systems in eukaryotic cells (with real nucleus) is the recurring division of cells. It is a fundamental sequence of events that cells must proceed to keep reproducing and is controlled by a complex molecular machinery containing intricate molecular mechanisms. One of the early success stories of mathematical biology includes cell cycle regulation [18]. Through the description and analysis of the network, theoreticians predicted several dynamical properties and unknown components of the system that were later experimentally verified. Moreover, lately these computational and theoretical approaches got more and more incorporated in the main stream cell cycle research. In order to follow the later discussion about this biological system, an introduction into the cell division cycle is given in the subsequent section.

2.11 The cell division cycle

Cells perform a sequence of coordinated events (referred to as ‘cell cycle’) that result in self-reproduction [30]. The major processes of the cell cycle are quite much the same in all eukaryotic cells (with real nucleus). During these events a cell must properly replicate its hereditary material (DNA) in the S-phase and separate the two copies into two daughter nuclei during mitosis (M-phase). Cells need to to double all their other components (proteins, ribosomes, RNAs, phospholipid bilayers, carbohydrates, metabolic machinery, etc.) during a cycle and usually the doubling time of the cytoplasm takes longer; hence temporal gaps (G1 and G2) are inserted in the cell division cycle between S-phase and M-phase in order to keep the size of the two daughter cells similar to that of the mother.

The major events (DNA replication and division) are tightly regulated and events are checked at several points. These ‘surveillance mechanisms’ are called *checkpoints* [146]. With the assist of a sophisticated network of interactive molecules, cells regulate and monitor the progress through the cell cycle. They check if an earlier event has been properly executed before proceeding to the subsequent step. Newborn cells are in G1-phase with unreplicated chromosomes and start the cell division cycle by monitoring the internal and external conditions if they are favorable for a round of events. Cells must grow to a critical size before they can commit to chromosome replication and division to guarantee the balance between the cell growth and the DNA cycle. When the internal and external conditions are favorable, cells make the decision to start a round of the cell cycle. They prepare the materials to get ready to the crucial events of cell cycle with the correct timing. G1-phase can be separated into two functionally different parts. The frontier between early and late G1-phase is called the *restriction point* in higher organisms [147] or *START* in yeasts [148]. At this point a cell commits itself to the whole process. The decision is irreversible; once DNA-synthesis begins, it goes to completion and eventually the cell will finish it even if conditions are getting worse in the meantime. Irreversibility is ensured by interconnected regulatory feedback loops building up a complex machinery of interacting entities [149]. During the process of *DNA replication*, sister chromatids are produced and ‘glued’ together by specific proteins, called cohesins [150]. Accuracy of S-phase events is crucial for producing healthy and viable daughter cells, thus the synthesis is permanently checked and repair mechanisms guard the correct DNA replication. G2-phase is inserted to ensure that DNA replication is properly finished and cells have grown to an appropriate size before mitosis. G2/M transition can happen only after these requirements are matched. Events during *mitosis* are critical for proper distribution of DNA between the two daughter cells [151]. Mitosis has several subphases: during prophase, replicated chromosomes condense into compact structures, in metaphase these condensed chromosomes are aligned on the center of the cell with the help of mitotic spindles. When all chromosomes are aligned, the so called *FINISH* transition (or meta-anaphase transition) is induced: the cohesions, that hold the two sister chromatids together, are destroyed allowing the chromosomes to be pulled to the opposite poles of the cell. After distributing the DNA content in telophase, the daughter nuclei form and eventually the two daughter cells separate during cytokinesis.

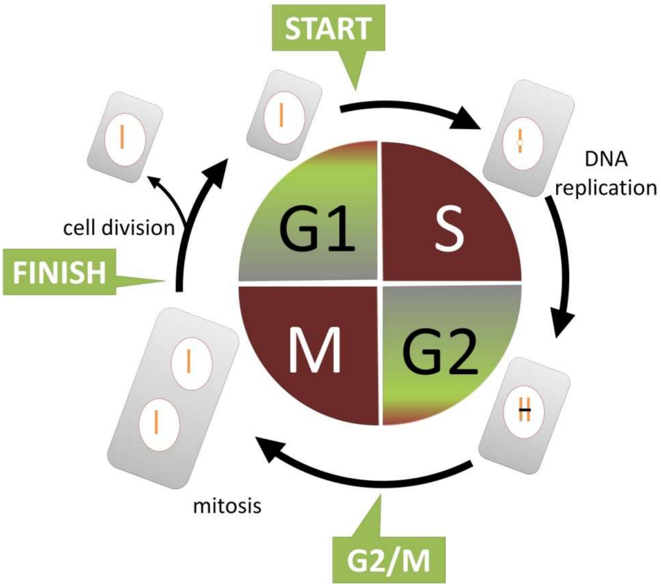


Figure 5: The cell cycle phases (G1, S, G2 and M) and the checkpoints (START, G2/M, FINISH).

2.11.1 Molecular mechanisms of the cell cycle control

The proper order of cell cycle events is controlled by a complex regulatory network of interacting macromolecules that control the cell cycle transitions. Systematic analysis of cell cycle mutants in the 70s by Lee Hartwell [152] and Paul Nurse [153] led to the discovery of the key regulator of the cell cycle (CDK) that works in a complex with a cyclically appearing molecule (cyclin), what was discovered by Tim Hunt [154]. These three researchers received the Nobel Prize in 2001 for their breakthrough results in understanding cell cycle regulation. After their discoveries, several cell cycle regulators and their functions have been identified that helped us to better understand the crucial regulatory steps of the cell cycle.

By now we know that active CDK proteins are bound to their regulatory cyclin partner that helps substrate recognition. CDK/cyclin complexes initiate events of the cell cycle by phosphorylating specific protein targets. They are fundamental kinases and are regulated by (1) controlling the availability of cyclins, (2) covalent modification of the complex by inhibitors and activators (3) and the CDK subunit is inhibited by phosphoryation and CDK might be sequestered to a stoichiometric inhibitor (CKI, for Cyclin-dependent Kinase Inhibitor) as well. CDK molecules are constantly present in excess, thus their level is not controlling their activity.

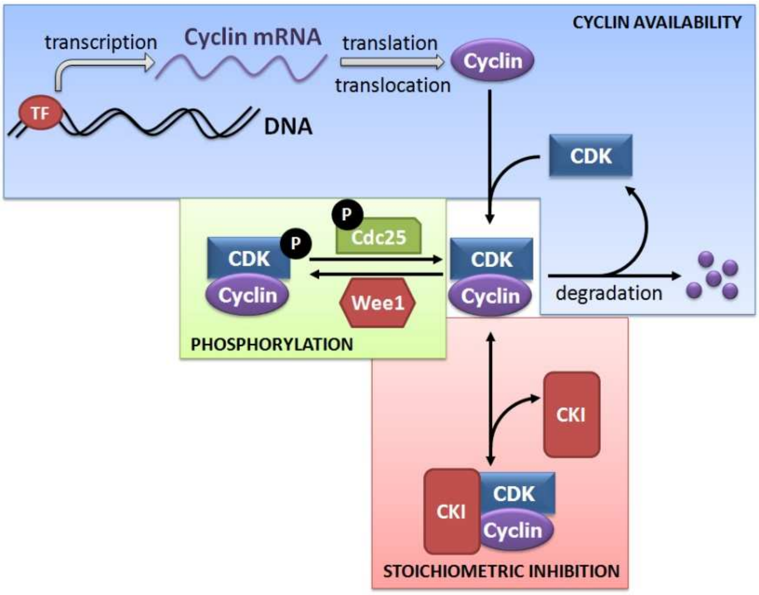


Figure 6: Control mechanisms of the core cell cycle component, the CDK/cyclin complex on different levels: availability of the regulatory partner (cyclin) of CDK is modified through transcriptional or translational control (TF = transcription factors enhancing mRNA synthesis of the gene encoding cyclin); the phosphorylation state of the CDK/cyclin complex is regulated through different kinases (e.g. Wee1) and phosphatases; the stoichiometric inhibitor (CKI) also inactivates the CDK/cyclin complex. The figure is adapted from [155].

The logic of cell division cycle is conserved in all eukaryotic cells: interconnected feedback loops ensure the order and the irreversibility of the cycle. In G1-phase, CDK activity is low due to the missing cyclin partners (e.g. D-, E-, A- and B-type cyclins), most of which are inhibited and rapidly degraded during this period. At the transition from G1- to S-phase (called START in yeast or restriction point in higher eukaryotes), cells make a decision of whether start a round of cell cycle or not. In order to make the best choice, cells sense both external (e.g. the presence of growth factors or nutrients) and internal (e.g. the size is large enough, mitosis is properly finished) conditions. The G1 cyclins (Cyclin D) are bound to CDKs (CDK4 and CDK6) and initiate the phosphorylation and, with it, the inactivation of the retinoblastoma protein (Rb). Rb's main role is to inhibit the transcription factor E2F of certain cyclin moldecules (Cyclin E and Cyclin A). These cyclins combine with a Cdk2 and help the total inactivation of Rb. Thus, there is a positive feedback loop in the regulation of transition from G1 to S-phase, with Cyclin E and Cyclin A inhibiting the inhibitor (Rb) of their transcriptional activator (E2F). CDK/Cyclin A cannot be fully active after Cyclin A is transcribed, since the CDK inhibitor (CKI) protein keeps this complex inactive as long as CKI is not phosphorylated and degraded. This phosphorylation also depends on CDK/cyclin complexes, adding a second positive feedback loop to the system. When CDK/Cyclin A gets fully active, it phosphorylates the proteins that regulate the unwinding of chromosomal ‘origins of replication’ (specific nucleotide sequences, where DNA replication can start) and induces DNA replication. After passing proper DNA replication, cells continue to grow in G2-phase and they check if the DNA is properly copied and intact. Cells also detect when they reach a critical size before to proceed. The G2/M transition and entry into mitosis is triggered by the activity of CDK in combination with B-type cyclins. In G2-phase, the CDK molecule is phosphorylated (thus inactivated) by an inhibitory kinase, called Wee1. At the G2/M transition, a phosphatase (Cdc25) removes the phosphate group from the inactivated CDK/Cyclin B complex, resulting in an increase in the activity of CDK/Cyclin B. CDK/Cyclin B feeds back and phosphorylates both Wee1 kinase and Cdc25 phosphatase. This modification inactivates the inhibitory kinase (Wee1) and activates the phosphatase (Cdc25) providing two positive feedback regulations making the increase in CDK/Cyclin B activity real sharp at the G2/M transition. The antagonism between Wee1 and MPF is defined as a ‘double-negative feedback’ between the two enzymes as a sum providing a positive autocatalytic effect on CDK activity. The positive and double-negative feedbacks act synergistically to create a bistable system with two qualitatively different states: a G2 state (inactive CDK/Cyclin B) and an M-phase (active CDK/Cyclin B). Problems in DNA replication or DNA damage can delay the G2/M transition by keeping Cdc25 inactive and/or Wee1 active. This ensures that CDK/Cyclin B activity stays low because of the inhibitory phosphorylation by Wee1. This control mechanism helps to avoid the segregation of damaged chromosomes during mitosis. If DNA is intact and replicated, CDK/Cyclin B activity turns on its positive feedback loops and the highly active form initiates mitosis. During mitosis, the separation of sister chromatids happens in a well-organized way. All sister chromatids have to be segregated at the same time to avoid any daughter cells to receive more or less chromosomes than the other. The cohesin molecules, that hold the chromatids together, can be destroyed only after all chromosomes are properly attached to the mitotic spindles that will pull them apart. At FINISH, a group of proteins make up the anaphase-promoting complex (APC), which with a partner (Cdh1 and Cdc20) helps to induce both cohesin and Cyclin B degradation. All Cyclin B is destroyed by the end of the cell cycle, resulting drop in CDK activity that triggers the separation of daughter nuclei and induces the division of the daughter cells (cytokinesis) that brings the cells back to G1-phase.

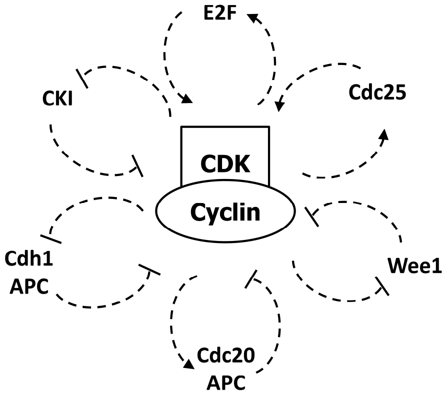


Figure 7: Regulatory feedback loops controlling the cell division cycle. Arrows represent activation, |- are inhibitory effects.

2.11.2 Modeling the events of the cell cycle

The cell cycle, being a periodic process, was in the interest of mathematical modeling rather from the beginning. Even before the molecular regulators of the cell cycle were known, mathematical models of the system had been already formulated. As the molecular details of the underlying regulatory network were revealed, models became more and more sophisticated. Indeed cell cycle has been one of the pioneering examples of systems biology approaches, where experiments and mathematical modeling have guided each other. Thanks to these efforts now we are able to better understand the dynamics of the cell cycle regulation and to explain how the oscillations appear in different cell types and what roles positive and negative feedbacks play in cell cycle regulation. Different modeling methods were used to attack these questions at different levels of complexity. Abstract logical models of the skeleton network, differential equations of the regulatory modules and stochastic models of some key control points all attacked cell cycle as an important biological example.

From the 1960’s we can find mathematical models that nicely explain some key aspects of cell cycle regulation from phenomenological observations on cell size and cell cycle time distributions. The discovery of chemical oscillators (BZ reactions) and the classical studies of non-equilibrium thermodynamics [156] provoked widespread interest in the 1970’s and gave huge contributions to research on theoretical physical chemistry and to mathematical biology. Researchers investigated biological oscillators, from calcium oscillations to circadian clocks, including the oscillations that drive cell division cycle [157]. As some data on the key regulator of cell cycle (CDK) were found by Nurse and others (see above), theoreticians started to create models to understand how the CDK/cyclin complex can regulate cell cycle events. Further experiments on yeasts and frog eggs produced a molecular description of the proposed protein interaction network, inspiring further mathematical analysis. The success story of joint work of theoreticians and experimentalists created a great interest for systems biology research.

The earliest efforts on the mammalian systems were the investigations of some particular modules of cell cycle. For instance, the DNA damage regulation in G2-phase of mammalian cells was tackled in details by Aguda, by modeling Cdc2, Wee1, Cdc25 and the DNA damage signal transduction pathway [158]. The regulation of the restriction point at the G1/S transition was also modeled by various groups [159,160]. The existence of this ‘point of no return’ in G1-phase has been first described in 1974 as the point, where mammalian cells decide whether they enter cell cycle or halt in a dormant G0 state [161]. A quantitative experimental characterization of this phenomenon was carried out by Zetterberg and Larsson [147], providing great data for mathematical modeling. Novak and Tyson modeled these experimentally tested physiological responses [160]. Their model relies on their earlier work on yeast cell cycle [162], which was extended with interactions describing the effects of retinoblastoma protein on global cell growth and on the synthesis of early/G1 cyclins (Cyclin A, E and D). Malfunction of the regulation of the restriction point might lead to cancer, so understanding this system by mathematical modeling is a very active field. Toettcher et al. [163] extended previous models by Csikasz-Nagy et al. [164] with DNA damage checkpoint mechanisms and the apoptotic pathway to get the so far most realistic mammalian cell cycle model.

There are also further simplified approaches to investigate mammalian cell cycles, when a few key events are spelled out in more details, but the rest of the cell cycle is greatly simplified. For instance, Pfeuty et al. [165] modeled cell fate determination by a simplified description of mammalian G1-phase. Different pathways regulating G1 arrest, growth, division and apoptosis were linked to each other and the four attractor states (G0 arrest, G1 arrest, S-phase and cell death) were simulated in order to recapitulate the simple rules that underlie the connection between input signals and cell states.

Noise can notably affect biological systems. While fluctuations in the average behavior of a cell population can be described by deterministic ODE models, the answer changes a lot on single cell level [19-21]. Stochastic modeling approaches are getting more and more popular because they provide opportunity to analyze single cells and find relevant results in co-operation with novel experimental techniques, such as quantitative flow cytometry [17] and fluorescence microscopy [166]. Researchers have been already using stochastic simulation techniques for modeling cell division cycle regulation. Modification of deterministic systems is a popular way for introducing noise into ODEs. Langevin-type equations have already led to novel results [163,167]. The need and the use of the exact stochastic simulation algorithm was shown for smaller systems [168], but the complexity of nonlinear multistep reactions makes the manipulation of large models difficult. This is one of the goals of this thesis, to overcome this barrier

Simplification by logical modeling has been proposed to overcome the problem of complexity. Logical modeling has a long tradition in biology and recently some applications to cell cycle research also appeared [74]. These models are based on Boolean algebra, where the activity of each component is represented by two states: ON and OFF, providing a method which is computationally less expensive. Behaviors that originate from the topology of the system have be nicely investigated in [75,169] . Recently, Davidich and Bornholdt worked out how to convert ODE models to Boolean to promote conversion between modeling formalism [73], while Faure and Thieffry compared the structure of currently existing logical cell cycle models [74]. An advantage of logical models is that they reduce the size of the possible state space, thus they permit the use of some analysis methods that work only for smaller systems.

Some other modeling concepts expanded from computer science towards biological systems and cell cycle modeling. Rule-based techniques [170] and process algebras [92] were built to handle combinatorial complexity caused by complex formation and various protein modifications.

In the next section we move to another interesting oscillatory system which has been in the focus of theoretical studies from the beginning.

2.12 The daily rhythm of living organisms

Additionally to the cyclic event of reproduction - growth and division - occurring in all eukaryotes (organisms with real cell nucleus), there is another important biological oscillator that has been attracted the attention of a large number of scientists working on different fields from mathematicians, physicians, engineers and biologists to computer scientists. The daily recurrence of activity and rest is a common property in everyone’s life. Early experiments revealed the existence of an endogenous clock regulating several periodic patterns occurring every 24 hours. The name of the biological clock is derived from latin ‘dies’ (day) and ‘circa’ (about). Circadian rhythms are observed from cyanobacteria to humans [171,172] and their importance is well recognized also in human physiology. Misregulation in circadian rhythms may lead to different conditions such as depression, familial advanced sleep phase syndrome (FASPS), delayed sleep phase syndrome (DSPS), or insomnia, which largely impact our society [173]. Increasing number of research focuses on studying these systems as recent findings indicate higher incidents of cancer in clock defective individuals [174,175] and chronic jet-lag is associated with higher mortality rate in aged mice as well as faster growth of tumor [176,177] .

Circadian rhythms originate from individual cells equipped with a molecular oscillator. In mammals, the pacemaker of circadian rhythms resides in the head, more specifically in the hypothalamic suprachiasmatic nucleus (SCN) [178]. However, it is generally accepted that most cells (not just SCN neurons) have a circadian machinery; thus, there are numerous peripheral oscillators (e.g. liver, muscle, lung, and even other parts of the brain) ticking the time for crucial biological functions, such as sleep-wake cycles, hormone secretion, blood pressure, mental performance or our mood [179]. In constant (*free-run*) conditions these cell autonomous clocks sustain a rhythm about 24h and in normal conditions, the periodic pattern of environmental cues (e.g. light-dark or temperature cycles) synchronize the clocks. The period is relatively invariant (e.g. over temperature), although it is able to phase-shift upon a stimulus from external cues (light, temperature or even ionizing radiation). This property allows organisms to adapt efficiently to the external environment. For example, a person traveling east to Europe from the U.S. will experience a jet-lag in the process to adapt advanced phase. Even a brief pulse of light may cause phase advances or delays depending on the timing and influence of the pulse [180].

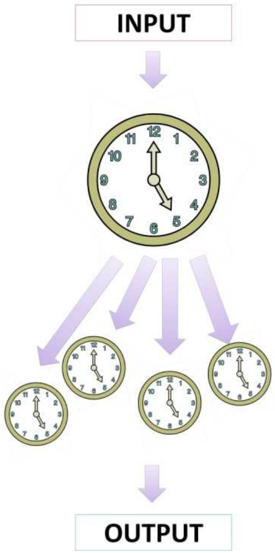


Figure 8: Hyerarchy between different molecular clocks interacting with each other. Rhythmic input signals (light-dark cycles, temperature cycles, etc) entrain the endogenous molecular system (SCN, large clock on figure) to the environmental cues. The core clock of mammals (SCN) influence the peripheral clocks (liver, fibroblasts, etc). Several phyisiology properties follow daily pattern (output); for instance, blood pressure, sleep-wake cycles, mental performance, mood, hormonal level.

The experimental research of the circadian clock has been started in 1729, when the French astronomer DeMairan discovered that leaf movements in plants show a 24h rhythmicity in constant darkness [181]. It took more decades to approve the existence of an internal and actively regulated biological clock in plants [182] and another hundred years in animals [183]. It is believed that a complex hierarchy exists between different (peripheral and SCN) clocks interacting with each other, although the one of the "mysteries" of circadian rhythms that people were interested in first was understanding the mechanism of a self-sustain, entrainable and robust oscillatory system (Figure 8). To our current knowledge, at the molecular level, the rhythms of the circadian clock are controlled by a negative feedback loop that is interconnected with several other positive and negative loops [171,184-186]. The molecular bases of this self-sustained system have been revealed with the finding of the period (per) gene in fruit fly (*Drosophila melanogaster*) in 1971 [187] and the frequency (frq) gene in a mold (*Neurospora crassa*) in 1973 [188]. The pieces of the clock's mechanism have been described one by one [171]. Most of the genes encoding proteins involved in the mechanism of circadian rhythms have been found simply by screening techniques.

Researchers have found that however the elements of the circadian clock are not necessarely sequence homologs in different organisms, they play similar role and the logic of the clock is conserved from bacteria to human. The nature of an oscillation is based on the idea of a system that moves away from equilibrium before returning. To achieve this, the product of a molecular process (negative elements) feeds back to slow down the rate of the product formation. This negative feedback loop is moved away from equilibrium with the help of delay mechanisms or additional (positive) feedback loops. The molecular mechanism underlying the daily cycles are described in the subsequent chapter.

2.12.1 The mammalian circadian clock

Figure 9 shows a schematic circadian oscillator relying on a simple transcriptional-translational feedback loop (TTL). The core regulatory loop consists of specific transcription factors (positive elements) activating the synthesis of several genes. This clock controlled transcription factor is a heterodimer of BMAL1 and CLOCK proteins in mammalian cells. They form the active complex via their specific protein-protein-binding (PAS) domains. Then the BMAL1/CLOCK dimer binds to the promoter region, onto the domains called E boxes, of certain genes. Among these rhythmically expressed coding sequences, we find some negative regulator of the clock. In mammalian systems, to our current knowledge, there are three period (Per1, Per2 and Per3) and two chryptochrome (Cry1 and Cry2) genes within the core negative feedback regulatory loop of circadian rhythms. The functional differences among these elements are still unclear. However, we know that after their transcripts (mRNA of Pers and Crys) are translated, thus generate clock proteins (PERs and CRYs) they all bear a negative role in the feedback loop. PERs and CRYs in complex block the clock gene’s activation, thus downregulate their own transcription, closing a negative feedback loop.

However, our knowledge about the complete regulation of circadian clock still remains partial, we are aware of several additional control loops of the clock. It is evident that the transcriptional-translational loop (TTL) presented above is not enough to generate a long - 24h - pattern, therefore post-transcriptional and post-translational modifications of circadian components occur, resulting in a more complex, but robust system. The large number of feedback regulations is crucial for the intact function of the clock [189].

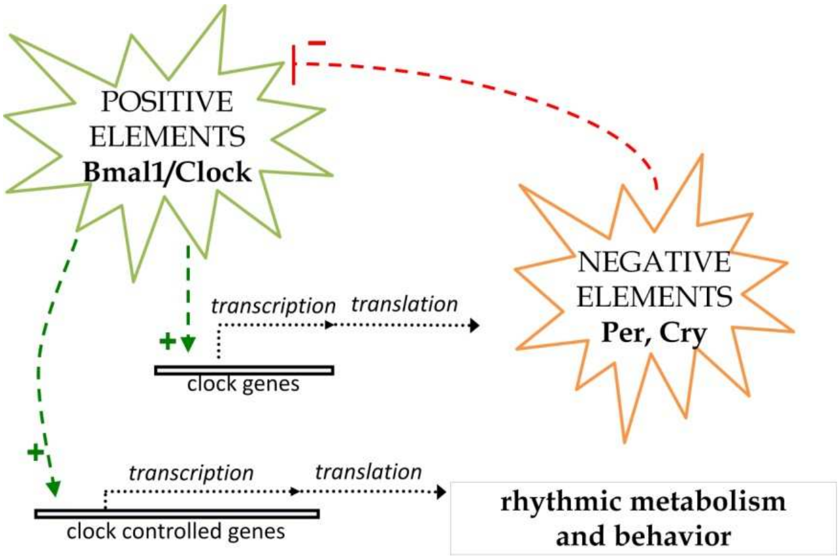


Figure 9: TTL negative feedback loop generating oscillations in circadian clocks.

One of the regulatory loops interconnected to the core negative feedback loop is an inhibitory branch that the BMAL1/CLOCK heterodimers generate through the activation of the so-called orphan nuclear receptor gene Rev-Erbα [190]. The REVERBα protein, in response, represses Bmal1 transcription by acting through Rev-Erb/ROR response elements in its promoter. Moreover, the PER/CRY complex enhances indirectly Bmal1 expression by binding to BMAL1/CLOCK and thereby reducing the transcription of the Rev-Erbα gene. Results reported by Sato et al. [191] show that the transactivator RORA acts coordinately with REV-ERB α and that they compete on the same promoter element driving the rhythm in Bmal1 transcription. This finding defines another feedback loop in mammals. In additional to the orchestrated control of the expression of clock transcription factors, post-translational modifications are also present. Recent evidence points to a clock based entirely on post-translational modifications in cyanobacteria [192]. Several reports [193] indicated that the joint activity of kinases and phosphatases (e.g., casein kinase 2 (CK2), casein kinase 1 (CK1), protein phosphatase 2A (PP2A) and protein phosphatase 1 (PP1)) regulate the phosphorylation and/or stability and/or the nuclear transport of the negative elements (PERs and CRYs). These relatively slow processes appear to be critical for creating a sufficiently long delay to support a 24 h rhythm. Besides phosphorylation events, other protein modifications (acetylation, deacetylation, ubiquitination, etc) occur within the clock. Recent findings of a complex picture showed that mouse CLOCK has histone acetyltransferase (HAT) activity that is required for rhythmic expression of core clock and output genes [194], however it also acetylates its partner, BMAL1 [195]. BMAL1 is deacetylated rhythmically by SIRT

[196] and it is also rhythmically SUMOylated on a highly conserved lysine residue in a CLOCK-dependent manner [197].

The mammalian circadian clocks' dynamical properties - such as the endogenous oscillations with an approximately 24h period, the entrainment to external environmental changes, temperature compensation and synchronization of multiple clocks - require systematic research. How the regulatory feedback loops are interconnected ensuring a robust rhythm and an interactive, entrainable system at the same time is an attractive query asked by modelers. Furthermore, the way how a biological system including biochemical reactions is able to compensate the changes of the temperature still remains a mystery. The picture is incomplete and its detailed research appears to be important as our endogenous self-sustained clock regulates a large number of physiological functions contributing to a healthy life.

2.12.2 The *in silico* clock: modeling the circadian rhythms

Several studies of mathematical modeling and systems approaches helped further understanding of the circadian rhythms in various organisms. Biological clocks were always excellent models for theoretical work [198]. Until recently, little was known about the underlying mechanisms of clock. Without this knowledge, researchers performed modeling with a top-down approach, such as the early attempts of molecular biology. They simulated and fit the model to the properties (phenotypes) they observed in experiments. In order to gain knowledge about the system, environmental stimuli (e.g. light or temperature or later DNA damaging drugs as well) were used to perturb this timekeeping machinery. Therefore, the first models were lacking the description of interactions between molecular components, only the underlying principles of the clock have been illustrated with the help of theoretical work (e.g. Aschoff's rule [199]). Early predictions were achieved with theories borrowed from physics, for instance using Van der Pol equations derived for an electrical oscillator serving for modeling the response of human circadian oscillations to light [200]. A morning oscillator model, proposed by Daan and Pittendrigh [201], contains two variables representing the phases of oscillators. Kronauer's model had physiological interpretation of light's effect on the human circadian pacemaker, and Borbély's two process model - for the regulation of alertness and sleep-wake dynamics - had also a great impact on the field [198]. There exist several abstract models including only a delay with feedback, or some other details without completeness [202,203]. Vilar and co-authors presented a study of a simple circadian model resistant to noise revealing a possible scenario of a robust oscillation driven only by two elements [204]. Brian C. Goodwin’s oscillator [205], which connects three components into a negative feedback loop, is still broadly incorporated into kinetic models of biological rhythms after four decades (e.g. [206]). Arthur Winfree had an approach to describe circadian clocks as limit cycles [207] and he predicted that a critical pulse, given at just the right phase and with just the right strength, would collapse this limit cycle and the system would become arrhythmic. This prediction has been verified experimentally later [208].

After finding some molecules of the circadian clock, more detailed models have been built that governed by a set of kinetic equations [209]. These models are considered as bottom-up approach as biochemical reactions are described by biophysical laws (e.g. mass action kinetics). The first circadian model following this approach was done by Albert Goldbeter [209]. He showed how 24h oscillations could be generated simply by a transcription/translation-phosphorylation feedback loop involving the PER protein (a negative element of the clock). He used standard expressions for transcription (Hill-function), translation (linear relation), and phosphorylation (Michaelis-Menten kinetics) to describe the system. As more molecular details have been identified, more refined models have been developed [210-217].

Beyond using differential equations - both deterministic and stochastic [218] - the non-classical techniques (process algebra, rule-based, etc.) have also been introduced to model the circadian clock [219,220]. They exploit a simplified - and abstract - picture of the known regulations and care less about the details.

2.13 The interconnected cell cycle and circadian rhythm

The cell cycle is a series of fundamental actions being able to respond to changes in the environment. Various effects (stress, light, temperature, etc.) can influence the progression through cell division cycles as well as several drugs, external signaling molecules and metabolites could also affect cell proliferation. Cells in order to carry out a proper function are interlocked with several other pathways. The harmonious progress of the cell cycle and the circadian rhythms is necessary for the well-being of organisms as malfunctions in the cell cycle and/or clock can lead to tumorigenesis [175,221]. Earlier studies from the late 1950s to the 1980s indicate that cell divisions in Euglena, Tetrahymena, and Gonyaulax occur only at particular times of the circadian cycle [222,223]. Gated cell division cycle is also observed in some cyanobacteria, with average doubling times less than 24h [224]. The first molecular link between the cell division cycle and the circadian clock was found by Matsuo and his colleagues [225]. A cell cycle regulator, Wee1, is directly regulated by clock components via Wee1's E-box elements in mammalian cells. The Wee1 protein is known to phosphorylate the CDK/Cyclin B complex in cell cycle and inhibits the cells' entry into mitosis. Intrigued by these results, recently, several groups presented coupled theoretical models of the mammalian cell cycle and circadian clock through this transcriptional link. Laurence Calzone and Sylvain Soliman have investigated the effects of interconnected circadian and cell cycle model systems [170]. They focused on the synchronization of the two oscillators within the parameter space and they identified the conditions of the entrainment. Altinok et al. [226] used an automaton model for the cell cycle to assess the toxicity of various circadian patterns of anticancer drug delivery. The gating of cell cycle events by a circadian clock model was analyzed in details by Kang et al. [227] with a systems biology approach. Furthermore, we also showed crucial properties arising from interconnected cell cycle and circadian oscillations and with our detailed model we proposed novel insights in the size control of mammalian cells [47] (see details in Chapter 5.8.1).

Biological model systems assess several assumptions through the complex rate functions describing multi-step reactions, making the extension of the model including noise difficult. Both circadian and cell cycle contain transcriptional regulation where low number of molecules is present and enzymatic reactions where assumptions cannot be applied throughout the whole simulation time. By joining the Centre for Computational and Systems Biology in Trento, I focused on improving compositionality in process algebra tools, particularly in BlenX. At the same time, biological questions aroused and we also concentrated on analyzing an additional molecular connection link between the circadian clock and the cell cycle under a deterministic framework [48]. Recently, a novel link has been found that the phase of the circadian clock of *Neurospora crassa* [228] and Rat-1 fibroblasts (a type of mammalian cells) [229] is shifted upon DNA damage caused by ionizing radiation or radiomimetic drugs. Experiments revealed that a checkpoint kinase (PRD-4 in *Neurospora* and CHK2 in mammals), bearing crucial role in cell cycle regulation, phosphorylates and targets the core clock protein's (FRQ in *Neurospora* and PER2 in mammals) degradation. This conditional link creates a bidirectional interaction between the cell cycle and the circadian clock.

3 The Problem in Details

After the numerous important discoveries were found with the help of theoretical biology models, we still have a lot to work on. The increasing power of the newest experimental techniques will enable us to model a single cell's and even a single molecule's behavior, which will require computational tools that can handle these problems. Better and better measurements on RNA and protein levels provide enormous amount of data that should be fitted by future models. As the models grow, they will get more specialized by simulating specific cell types, while others will get even broader by connecting several networks (e.g. cell cycle, metabolic cycle, circadian clock, etc.) together. This later goal of interlinked systems is greatly accelerated by the collection and standardization of computational methods. Still we need some improvement in modeling formalisms, simulation techniques and model analysis to achieve the knowledge of the whole picture of life.

BlenX allows a rule-based process calculi modeling method that contributes to systems biology research. A BlenX program is made of an optional declaration file for the user-defined constants and functions, a binder definition file that associates unique identifiers to binders of entities used by the program and a program file that contains the program structure. "Boxes" represent the interacting biological entities (proteins, genes, etc.) and contain an internal program (or internal behavior) describing their possible activities and a set of typed interfaces describing their interaction capabilities. Sequential and parallel composition of processes; definition of events and actions provide the backbone of a BlenX model. Composition of possible conditions leading to reactions that might occur is the first challenge that the modelers should deal with. A key innovative aspect of BlenX is the ability to model the biochemical reactions between components simply by listing their affinity and without the need of programming all the possible interactions. The BlenX framework allows the user to build systems by fixing each reaction of the network (also called as bottom-up approach) or gives opportunity to handle abstractions as well (such as a top-down approach). After specifying the system, the BlenX program is executed with the Gillespie stochastic simulation algorithm (SSA) [37]. The reactions occurring in the system are defined by rate dependent functions that are crucial for the reaction propensities of the stochastic model. Rate functions are associated to actions and events of boxes, and those rates can be determined by the mass action kinetic law.

One crucial point of biological models built upon mathematical formalisms is the additional presence of the complex mathematical functions (e.g. Michaelis-Menten kinetics [141], Hill function [142], etc.) that have been empirically developed through several assumptions in order to provide an abstraction due to the lack of model parameters. These abstractions simplify the system leading to a decrease in the required computational power for calculation. Furthermore, modelers often turn to these phenomenological functions to describe the observed behavior of a system without knowing all its details, such as multi-step reactions are often assumed to happen at the same time in cooperative reaction schemes [34]. Experimental measurements are becoming more and more sophisticated. Data on elementary steps are asking for a technique which describes single reactions in a modular manner and they require tools that are able to transform abstract mechanisms into elementary ones. BlenX is one of the promising computational languages with a feature of compositionality for modeling biological systems. Complex rate functions raise several problems in stochastic process algebra approaches.

Among the current research problems arising in process calculi, one crucial point is the expressivity of the calculus and the challenge to improve the compositionality offered by process-theoretic tools in biology. Compositionality is believed to be one key advantage of formal languages. The capability of easily composed models lies within the tool, although there have been only a few research on bringing it closer to perfection [230,231].

During the stochastic process algebra composition of biological models the nonlinear description of biological phenomena raises the following problems: (1) complex rate functions are not appropriate for the guidance of stochastic Gillespie method assuming only elementary steps in the system. (2) Interpretation of nonlinear terms is currently available in the BlenX framework although it highly limits its compositionality. The hidden elements of these functions might be necessary to be expressed for the extension of the model. (3) Assumptions of the complex terms are often found to be inaccurate in a larger system. Thus, conversion of nonlinear terms to elementary steps often leads to the disappearance of crucial nonlinear behavior of the large system. (4) Current interpretation of BlenX models requires computer science knowledge from the users. Furthermore, we need some biologically important test cases to investiage these problems and the method we propose in this work.

3.1 The Gillespie method and nonlinear functions

Molecules undergo random collisions resulting in discrete biochemical interactions. The BlenX framework applies Gillespie's exact algorithm to simulate chemical or biochemical systems of reactions in a stochastic manner (discussed in Chapter 2.9). One limitation of the algorithm is that it considers only elementary reactions, while biological models often deal with nonlinear terms in the deterministic framework. Nonlinearity is known to serve oscillations in several periodic biological systems [38] or multsistability in others [145], giving an important role for these mathematical formulas in simple models. The problem of complex rate functions as stochastic rate constants has already discussed by several authors [136,232-234]. Following the assumptions of Gillespie's hypothesis, we specify that each reaction time is a random variable following a negative exponential distribution with rate equal to the value of the propensity function that cannot be applied in case of nonlinear reaction rates in the system. Most models ignore the inquiry of the validity of this assumption.

Previous work concerning the use of general kinetic laws in process algebras and formal methods was presented in [39,235]. Within the BlenX framework, generally distributed reaction times have been also implemented recently [137]. It provides choices of the reaction time distribution for the stochastic simulation algorithm of Gillespie. In this way, abstracted rate laws can be handled stochastically that leads to a better quantitative tool for matching wet-lab experiments and in-silico results. This aggregation and level of abstraction lay above the elementary reactions. Another initiative to approach complex reactions with simple abstractions in BlenX has been studied [233]. The idea taken from the application of web-service transactions have been used to extend the tool with the representation of multiple-reactant multiple-product reactions with elementary reactions as if it were atomic. Atomicity is summarized as all or nothing, reducing the model but it may lose nonlinearity as a property of the biological model. A novel approach is needed within the BlenX language for a systematic and proper way of model composition.

3.2 Compositionality with complex rate functions

Living organisms are governed by several complex systems of interactions among genes, proteins and other molecules. These systems contain effectors (activators/enhancers or inhibitors/repressors) of the reactions whose communication results in different regulatory loops (feedback and feed-forward loops). Interconnection of these structures may lead to oscillations, acceleration, pulse generation or bistability that are frequently seen features of biological systems [38,145,236,237]. Theoretical models address general concepts of dynamical systems and find principle design of networks that are crucial in particular behaviors (multistability, hysteresis, oscillations, irreversibility, etc.) [38].

Compositionality is a crucial feature of a computational language and it signifies the possibility of defining a large system starting from the definition of its subcomponents. Systems biology calls for modeling languages that can be built up in a systematic way. Several biological models are proposed each day making compositionality to be one of the most important key features of process algebra. Compositionality has been addressed as an issue of model-construction from elementary reactions with the basic operators [238,239], as the translation of one approach to another [125], or as the combination of different types of models (ODEs with process algebras, Boolean, hybrid models) [42,240], but the compositionality of complex rate functions has attracted less attention. When a model has to fulfill several assumptions applied in phenomenological modules, the freedom of compositionality is reduced. The hidden parts of the modules may contain important linkage between the networks that are chosen to be merged together. Stochastic computational modeling is asking for a tool that supports network composition in an adequate and user-friendly way.

3.3 Nonlinearity in biology and in computational models

Biological systems are highly nonlinear with numerous interacting molecules. In order to represent a nonlinear behavior of these networks, complex mathematical functions are used to describe the response of a system to a particular signal. Mathematical models are usually composed of variables (molecules, signals, etc) and operators (algebraic laws, functions, rules, conditions, etc.). If all the operators exhibit linearity, the mathematical model is defined as linear. This is often the case in assuming elementary reactions described by mass action kinetics where the rate of a reaction is proportional to the activity of the reagents. For more complex schemes, such as enzyme kinetics, protein modifications, transport mechanisms, etc., there are well-defined complex formulas to describe the observed physiological output. These simplifying equations are often used by modelers. For instance, the assumption of the Michaelis-Menten kinetics [141] makes description of enzymatic reactions simple and computation of the term requires less details of the system. The Goldbeter-Koshland switch [241] and the Hill-function [142] provide sigmoidal signal-response curves that are highly sensitive to changes in signals around the threshold level giving rise to an ultrasensitive property. Sigmoid responses are used to generate switch-like (binary) decisions [134] and have been shown to be able to filter out noise or delay responses of the system [242]. Furthermore, nonlinearity originating from ultrasensitivity can create oscillations in combination with negative feedback loops [243].

Several authors have reinvestigated the application of the approximations of complex reaction schemes in different scenarios. In case of Goldbeter-Koshland's switch Bluthgen et al. [244] have shown that high enzyme concentration can modify the response of the MAPK cascade [243] and make oscillations disappear under these conditions. Ciliberto et al. [132] have analyzed the total quasi-steady-state assumption of the same reaction scheme in details and proposed a novel approximation of the scenario which is found to be more appropriate for metabolic networks than the assumptions of the original kinetics. Additionally, Berg et al. [245] have pointed out the differences between macroscopic and average behavior analyzing the effect of noise on an ultrasensitive system. Under the stochastic framework the approach of complex and simplified reaction terms applying assumptions may not be a good solution for modeling biological systems.

When non-elementary reactions occur and compound mathematical formulas are used in modeling, the direct translation of mathematical terms into the stochastic context is a well-liked approach. Usage of these general functions for calculating the rate of a reaction is also possible in BlenX [117,246]. However, these implementations have been pointed out by several authors to be incompatible for some cases [136,232,233,247-249], thus modelers have to pay particular attention to the assumptions they apply. Stochastic modeling of complex functions is only an approximation and assumptions have to be handled globally. Thus the BlenX framework calls for a semi-automatic method of describing these complex rate functions with intermediate steps (we refer as an "unpacking" mechanism) not only owing to ease the compositional programming process, but to provide a correct (and generalized) way of stochastic simulations.

3.4 A programming language for computational systems biology

One of the goals of Beta Workbench (BWB) that has been set to target is a design of a programming language that facilitates and engineers the model-building process of biological systems at different level of abstractions [250]. BlenX inherited the properties of BWB that has been developed for modeling, analyzing and simulating biological networks. The usage of the language - at this stage - requires computer science knowledge that biologist might lack. Furthermore, the large variety of process algebra primitives provides different representation of the same biological schema but with diverse efficiency of compositionality. Extension of nonlinear models is currently not straightforward in BlenX (as it is also challenging in other approaches) and depending on the expression of a reaction with the process algebra language, creates a problem in compositionality.

4 The Proposed Approach

Men always search for common patterns that are simple to understand and that might help them to comprehend complex features. Mathematicians, physicists and also biologists have tried to find order in “chaos”. As systems biology is one of the research fields concentrating on understanding life and how biological systems function; the common motifs of molecular networks are in the heart of focus of interest [44]. Several essential biological network structures have been published in biology [33,44,56,145,236] and in computer science [39,239,251] as well. These repetitive regulatory patterns carry special and general properties that are crucial for the proper function of the overall biological system.

Templates are constructs representing basic structures, motifs. Templates are applied in programming as re-usable codes that allow the paste of different parameters without changing the structure of the template. The concept of patterns and modular programming originates from an idea of an architect, Christopher Alexander, who had a great impact in the research of programming languages and software design [252]. Comprehensive biological models are difficult to be constructed reaction by reaction, thus it is worth realizing general building blocks that can serve the initiative of systematic modeling process. Process calculi - therefore also BlenX - provide a programming environment that could support systematic and formal composition of large models, although the modelers do not profit from this property as the current model building method is difficult in case of larger systems. We have to start the composition from the very “bottom” (with the basic primitives) that is time-consuming and contains many repetitive and error-prone steps. It is believed that the modeling technique of BlenX should support an additional, higher level composition than what is available now. By creating and using pre-defined templates stored in a library, computational models might be easier composed and the users are more capable of profiting from the compositionality feature of process calculi tools on a larger scale. The collection of biological functions has already existed in non-process-algebras based modeling tools (e.g. COPASI [9]), but it is still a missing feature of BlenX. Realization of different templates leads to interesting and unsolved problems of computer science. A library of modules should influence the program design in a positive manner as in all the fields affected by computer technology. With a motif library, biologists and modelers could pass the current limitations of BlenX.

Nonlinearity might be challenging to code in the current BlenX representation. Stochastic solutions require elementary steps, while unpacking the complex mathematical terms may lead to disappearance of nonlinear behavior (discussed in Chapter 3). In stochastic simulators - such as Beta Workbench (BWB) - decomposition of mathematical terms into the underlying elementary reactions (with BlenX primitives (*actions*, *events*, *complex formation*, etc.)) seems to be important, although usually it is not straightforward. If the specific reaction schemes reproducing nonlinear properties (e.g sigmoidal curves, switches, etc.) are available, users can build models in BlenX in a simple and adequate way, by choosing the particular motif from the library.

Our study has to represent a method with which compositionality is carried out in a farseeing way. For instance, a single protein might participate in many reaction systems, thus definition of a module that can be easily composed into a larger model is not straightforward. These terms have to be extendable for later use. Depending on the level of abstraction (that is defined by our knowledge), the desired behavior can be achieved in different ways thanks to the rich opportunity of the language. We present an analysis for compositionality in BlenX starting from the basic primitives to the high-level, complex templates. The possibility of using different basic BlenX primitives gives us freedom of abstraction, although the users need to bear computational skills. Based on the predefined structural requirements and frequency of basic motifs, elements of library can be selected. These building blocks would improve the modeling process with BlenX as modelers are able to reuse frequently found biological structures and compose large models. Realization of these templates requires both biological and computer science approach. Motifs should be biologically relevant and computationally effective at the same time. The proposed approach may open novel questions and provide guidance on future improvement of compositionality.

Properties of the library elements are further investigated in this work. Stepping towards complexity, higher level submodels are also composed. Finally, we focus on presenting interesting biological case studies and questions that are answered with computational modeling approaches. Possible research directions in finding parameters for these motifs are also shown, but I would like to emphasize that parameter estimation is out of the scope of this work. The development of the library is presented hereinafter.

5 Results

5.1 The basic primitives

Model composition is the first challenge that the modelers should deal with. A key innovative aspect of process calculi tools is the ability to model the reactions between components simply by listing their affinities. The BlenX framework allows the user to build systems by specifying each reaction of the network (also called as bottom-up approach) or gives an opportunity to handle abstractions as well (such as a top-down approach).

Molecules are represented by interactive boxes that are able to synthesize, to degrade or to react with each other. Basic BlenX primitives apply mass action kinetics which describes elementary steps with one or more chemical species reacting in a single step and with a single transition state. The simplest scenario in biochemical reactions is when a molecule (*A*) is synthesized with a constant rate (*k*):

when(A::rate(k)) new(1);

Degradation with a rate (*k*) of the entity (*A*) is also described with an event:

when(A::rate(k)) delete(1);

A is modified with a specific reaction rate (*k*) resulting in a molecule (*B*):

when(A::rate(k)) split(B,Nil);

where *Nil* represents an empty box that degrade with an infinite rate.

Activation of a synthesis by a signal (*S*) can also occur in a linear way

when(S::rate(k)) split(S,A);

where the amount of the effector (S) does not change during the reaction.

Furthermore, boxes are also able to form complexes or change their internal behavior. In the sequential sections, we would like to introduce higher level of compositionality that modules are collected into a template library for reuse.

5.2 The Michaelis-Menten formula for enzymatic reactions: a hyperbolic response curve

Most of the biochemical reactions require catalytic molecules (enzymes) which increase the rate of a particular reaction. In enzymatic reactions, the molecules at the beginning of the process are called substrates (S), and the enzymes (E) selectively convert them into products (P). The kinetic description of such systems was expressed by L. Michaelis and M.L. Menten [141]. The derived equation of their results (referred to as Michaelis-Menten kinetics) is widely used in biological modeling.

The scheme of a one-substrate-one-product reaction (with one active site) is

$$E + S \overset{k_1,k_2}{\longleftrightarrow} ES \overset{k_3}{\rightarrow} E + P$$

The catalytic step is supposed to be irreversible and the rates of the reactions are given by the law of mass action. Reaction rates are summarized in Table 5.

Association rate of the enzyme-substrate complex	$\boldsymbol{k_1 \cdot [E] \cdot [S]}$
Dissociation rate of the enzyme-substrate complex	$k_2 \cdot [ES]$
Production rate of P	$k_3 \cdot [ES]$

Table 5 Steps of the enzymatic reaction. k_1, k_2 and k_3 are the rate constants of the reactions; *[E]*, *[S]* and *[ES]* represent the concentration of the enzyme, the substrate and the enzyme-substrate complex, respectively.

As enzymes are specific to their substrates and the Michaelis-Menten term assumes that the formation of the enzyme-substrate complex (ES) is relatively fast, the equilibrium is reached rapidly and the production of P becomes the rate-limiting step in the overall system. Therefore the *ES* complex is stable, meaning that the change of its concentration approaches zero (referred to as quasi-steady-state assumption (QSSA)). Another important assumption is that the concentration of the substrate highly exceeds the one of the enzyme (*[S] >> [Etot]*). When a critical substrate concentration is reached, the enzyme is saturated and an additional amount of substrate will not influence the velocity of the reaction; it is already maximal (v_{max}). If the last reaction is assumed to be irreversible and all the previously mentioned statements are valid, the rate of the substrate turnover to product can be estimated as

$$v = v_{max} \cdot \frac{[S]}{K_m + [S]}$$

where

$$v_{max} = k_3 \cdot [E_{tot}]$$

$$K_m = \frac{k_3 + k_2}{k_1}$$

and

$$[E_{tot}] = [E] + [ES]$$

The Michaelis-Menten equation provides a complex rate function assuming a single reaction step:

$$E + S \overset{v}{\rightarrow} E + P$$

with the reaction rate described previously:

$$v = v_{max} \cdot \frac{[S]}{K_m + [S]}$$

The Michaelis-Menten rate law is often found to be a good approximation to describe enzymatic reactions. Furthermore, it by-passes the problem of rarely available rate constants as the key parameters (v_{max} and K_m) of a Michaelis-Menten reaction might be easily determined from measured data through linear graphical representations (e.g. Lineweaver–Burk plot, Hanes–Woolf plot, Eadie–Hofstee diagram) [253] or by nonlinear regression methods [254].

In the next subsection, we provide a brief description of how to code enzymatic reactions in BlenX with elementary steps and we give a hint how to search for unknown parameters in the Michaelis-Menten module.

5.2.1 The Michaelis-Menten reaction scheme

The use of Gillespie's stochastic algorithm requires elementary steps instead of complex rate functions in a model (discussed in 1.2 The Problem session). Decomposition of the Michaelis-Menten rate law into elementary reactions may lead to crucial changes in a larger system's global behavior as nonlinearity may disappear if assumptions are inconsistent about enzyme-substrate complexes [132,168]. Compositional model building should carefully handle the enzyme molecules hidden in the quasi-steady-state assumption (QSSA).

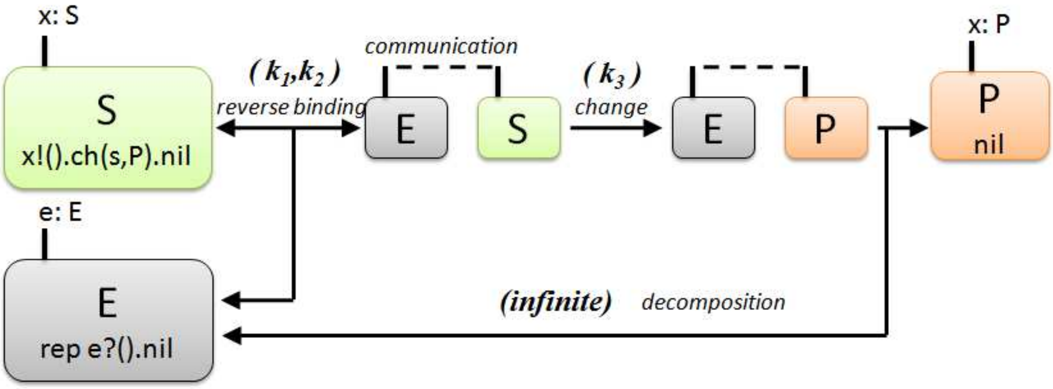


Figure 10: BlenX representation of the Michaelis-Menten kinetics.

The Michaelis-Menten module can be implemented easily into the BlenX language as the binding of the substrate and the enzyme is described as complex formation through specific binding sites of the boxes representing proteins. On Figure 10, the types *S* and *E* are compatible and equipped with complexation and decomplexation rates. After complex-formation, *S* and *E* communicate and the internal behavior of the substrate box is changed into the behavior of the product (the *ch(x,P)* action modifies the type of the binder *x* into *P*). The new product has binding affinity no more to the enzyme, thus an abrupt dissociation (“decomplexation” with infinite reaction rate constant) occurs to release the enzyme *E*.

The possible reactions of the system are described as the followings:

```
let P : bproc = #(s:0,P) [ nil ];
let S : bproc = #(s:0,S) [ s!().ch(s,P).nil ];
let E1 : bproc = #(e1:0,E1) [ rep e1?().nil ];
```

Complex formation, dissociation rates and the catalytic rate are provided through the types identifying the molecules in the reaction:

```
{ S,P,E1 } %% {  (S,E1,rate(k11),rate(k11r),rate(k21)), (P,E1,0,inf,0) }
```

5.2.2 Finding parameters for the Michaelis-Menten module

Decomposition of the nonlinear term into elementary reactions calls for the definition of rate constants of each step. Enzyme catalysis is the rate limiting reaction, thus the maximum velocity of product formation (also called as turnover) is given by the amount of available enzymes in the system and the particular rate constant. If we would like to implement the complex term defined by the Michaelis-Menten formula into a stochastic process calculus framework in order to carry out compositionality within e.g. BlenX, the parameter of the last reaction (k_3) is easily obtained from the known v_{max} and the total enzyme concentration:

$$k_3 = \frac{v_{max}}{[E_{tot}]}$$

The dissociation rate constant of the *ES* complex (k_2) is the following:

$$k_2 = \frac{k_1}{K_m} - k_3$$

and obtained from the Michealis-Menten constant. It is supposed to be low as the *ES* complex is assumed to be stable. The rate constants of the reversible complex formation (k_1 and k_2) can be chosen among several combinations by ensuring that the association rate is larger than the dissociation rate of the *ES* complex. Furthermore, we know that the catalytic step is the rate limiting, thus k_1 is chosen to be much larger than k_3. In this simple example, our choice determines the time of the simulation, thus values of the rate constants have to be carefully selected. In isolated systems we can scale down the constants easily in order to speed up the simulation. However the rates of the reversible complex formation cannot be limiting in a larger model.

The proper rate constants describing our compound Michaelis-Menten module have been selected by taking the minimum amount of substrate during the reaction and setting the initial (total) concentration of enzyme to $S_{min}/0.1$. As a consequence, we get

$$k_1 = k_3 \cdot 1000 = \frac{v_{max}}{[E_{tot}]} \cdot 1000$$

and

$$k_2 = K_m \cdot k_1 - k_3$$

Note that the selection of feasible parameters must lead to a positive value of k_2; and the total concentration of the enzyme has to be globally lower than the substrate with a large extent. Different approaches may be available to estimate the parameters of this module in a larger system. For instance, deterministic simulations can give us a guess of the minimum values of the substrate concentration during the simulation or more sophisticated methods may be also available for determination of the missing parameters from experimental measurements [140]. Optimization of the execution time with a rate of complex formation that is fast enough is necessary, thus equilibrium is reached rapidly but the simulations remain computationally cheap.

5.2.3 Simulation results

The complex term of the Michaelis-Menten kinetic is handled as a single step within Gillespie's stochastic simulation algorithm. The time of the next occurrence of each reaction is assumed to follow negative exponential distribution. The approximation of this abstraction has been shown to be good enough if the assumptions applied in the Michaelis-Menten term are valid [232]. Namely, if the amount of substrate molecules excess the number of enzymes in the system and the enzymes are quickly saturated by their substrates, the reaction exists in a quasi-steady state. The authors claim that the approximation of the Michaelis-Menten reaction reduce model complexity and are found to be appropriate in stochastic models upon valid assumption, although in case the assumptions described ahead are not valid, decomposition of the module leads to disappearance of nonlinearity. When the exact measurements for the parameters of the modeled system is unknown, but nonlinearity is observed or hypothesized, decomposition of the complex rate equation is required to provide the properties assumed for the module. Thus, we decided to analyze the different solutions for the decomposition process in BlenX and we compared two stochastic models, one with the complex rate function and another with an exact solution of the Michaelis-Menten kinetics.

To model molecular fluctuations, a probabilistic model of the biochemical dynamics is executed where the number of the reactants of every elementary step introduces noise into the system. This property might modify the result of the simulation as stochasticity becomes a crucial behavior of the biological system. We believe that the BlenX language should offer an alternative solution for complex reaction schemes within a stochastic framework providing a compositional tool for modeling biological systems.

First, we converted the concentrations of the deterministic system into molecule numbers through a transformation on the parameters using a scalar constant α defined as $1/(N_A \cdot 10^{-6} \cdot V)$, where N_A is the Avogadro number and V is the volume of the modeled system (discussed in details in 2.10 session (Reaction rates in BlenX)). The number of molecules influence the noise, thus during the simulations α sets the level of stochasticity depending on the size (volume) of the system. Then we set the models to different initial conditions for the substrate and run 200 simulations for each initials. The rates of product formation have been derived from the simulation results and these values are plotted over the initial amount of substrate molecules. It provides a saturation curve of the Michaelis-Menten kinetics (Figure 11).

Initial amount of Substrate [#]	*Rate of reaction and standard deviation with the exact solution (elementary steps)*	*Rate of reaction and standard deviation with the complex reaction*
30	5.630 ± 1.873	4.455 ± 2.130
60	10.450 ± 3.125	8.410 ± 2.648
120	17.395 ± 3.502	13.895 ± 3.503
180	22.010 ± 4.783	19.850 ± 3.645
240	27.170 ± 4.966	24.520 ± 5.011
300	30.490 ± 4.960	27.460 ± 5.077
360	31.575 ± 5.219	30.790 ± 5.552
420	34.980 ± 5.105	33.875 ± 6.445
480	36.880 ± 6.132	35.410 ± 6.516
540	39.360 ± 5.873	36.795 ± 6.422
600	41.035 ± 6.491	38.405 ± 6.191
1200	47.985 ± 5.990	46.510 ± 6.415
3000	53.865 ± 7.075	53.725 ± 7.473
6000	57.845 ± 8.212	56.405 ± 7.227

Table 6: Average value of the rate of reaction and the standard deviation of 200 stochastic simulations.

Calculation of the parameters in this case is based on the assumptions shown previously. K_m is set to *300#* (# refers to the unit of number of molecules) and v_{max} is *60#/min*. The total enzyme amount ($|E_{tot}|$) is set to *60#* molecules and the Michaelis-Menten constants define k_3 and the ratio of k_1 to k_2. The chosen parameters should also satisfy the assumption that the value of k_1 ($dm^3/(min \cdot mol)$) is much larger than the value of k_2 ($1/min$). This condition may be suited by different rates of k_1 and k_2, although these options only influence the speed of the reaction (and our simulation), but does not change the result (data not shown).

Parameter names	**Parameter values**	**Parameter units**
k_1	200 α	1/(min #)
k_2	99	1/min
k_3	1	1/min
K_m	0.5 α	#
v_{max}	0.1 α	#/min
E	0.1 α	#

Table 7: Parameters for the Michaelis-Menten module. α is set to 0.00167 during the simulations shown on Figure 11.

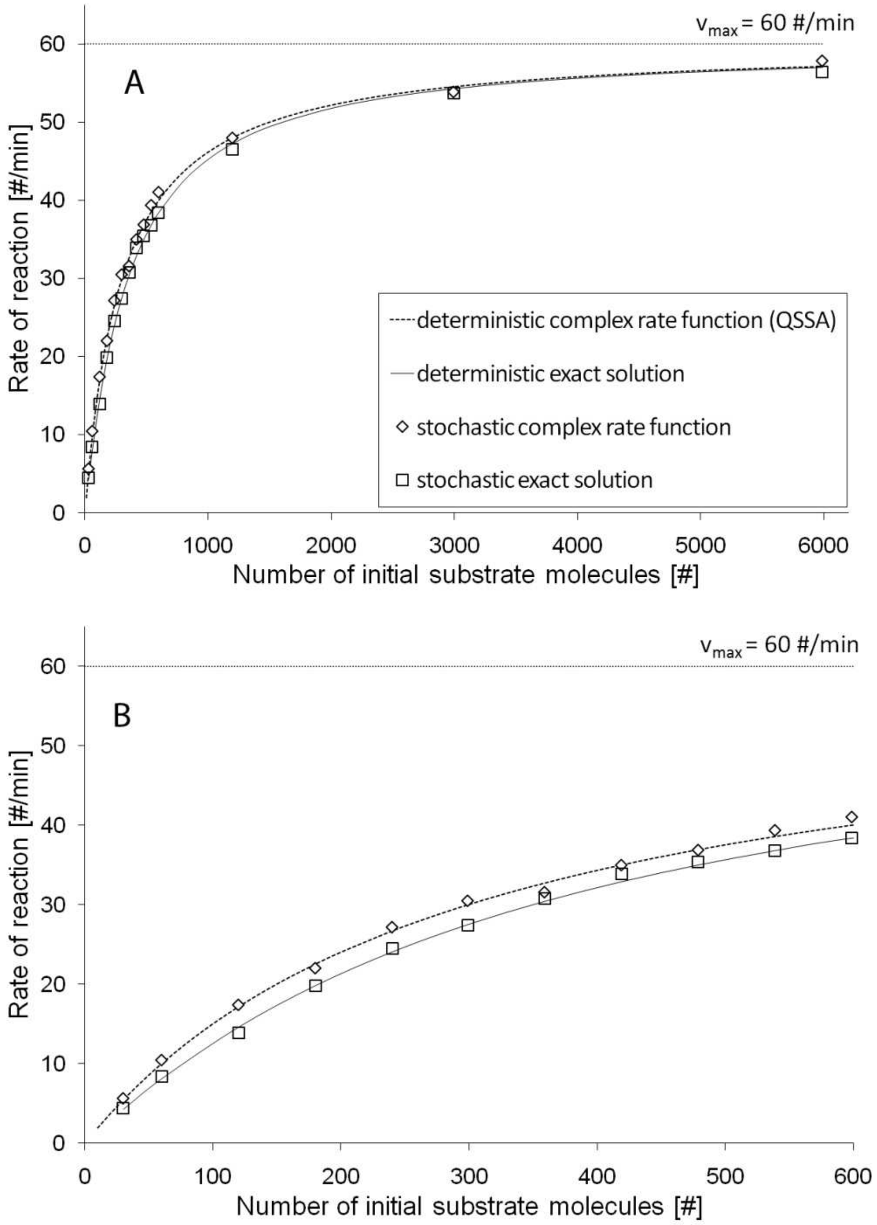


Figure 11: Simulation results of the stochastic BlenX model fit the deterministic saturation curve well (A). On (B), at low amount of initial substrate molecules the original assumptions of Michaelis and Menten do not match and the exact solution diverges from the complex function. This is a property of both the deterministic and the stochastic case. Each point is an average value of 200 run with an α=0.00167 defining the level of noise in the system. Standard deviations are summarized in Table 6.

We compared the deterministic and the stochastic simulations' results executed with the "unpacked" and the complex modules with a parameter set shown in Table 7. The module built up from a complex reaction and the one with elementary reactions shows us a good accordance with each other and also with the deterministic scheme (Figure 11). Simulation results of the BlenX model fits the deterministic saturation curve well, although when the original assumptions of Michaelis and Menten do not match, the exact solution diverges from the complex function as it has been shown previously by others [232]. When the enzyme is in excess to the substrate, the solution of the unpacked model differ greatly from the packed version as the assumption made for the QSSA is not more valid for the system (Figure 11 and Figure 12). This is one limitation of the compound function; however, it is also a limitation of the deterministic simulations.

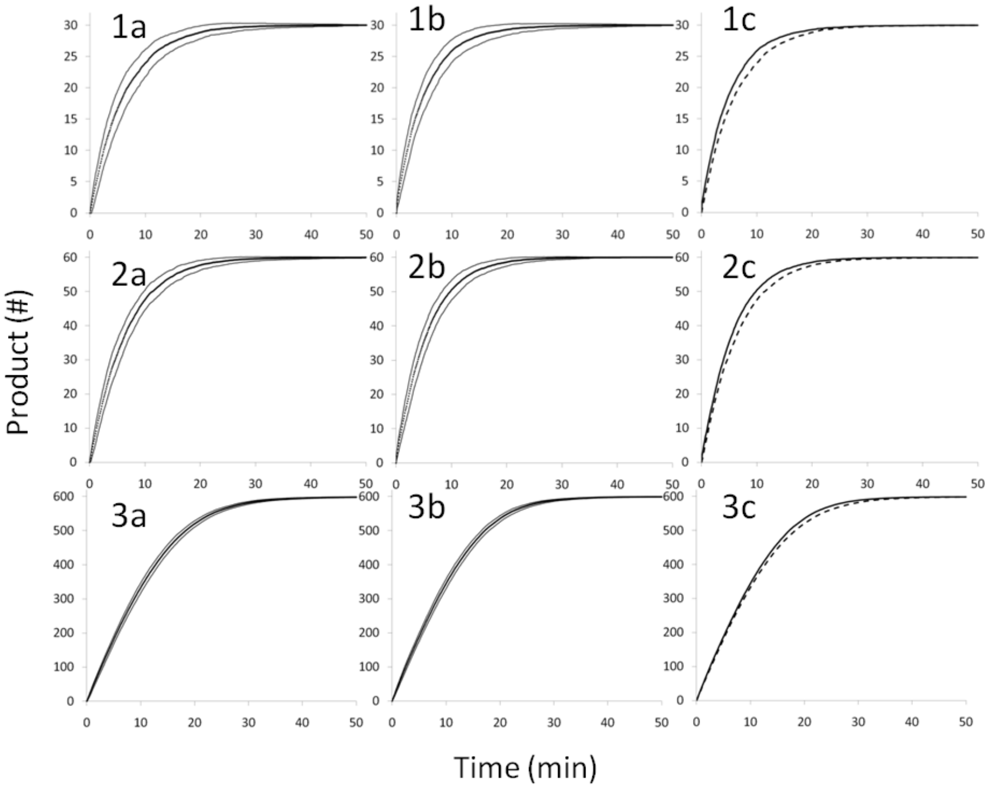


Figure 12: Simulation for different number of initial substrates: 1a-c: *S=30#*; 2a-c: *S=60#*; 3a-c: *S=600#*. Average number of product is plotted for each time step. 1-3a: stochastic unpacked versions (black lines) shown with standard deviation (grey lines) from the mean. 1-3b: complex rate functions with K_m*=300#*, v_{max}*=60#/min* where standard deviation is plotted. 1-3c: comparison of the average of the unpacked (dashed) and complex (solid) stochastic simulation results executed with BlenX. $\alpha = 0.00167$.

Arkin and Rao assumed [232] that the reactions are isolated and the amount of enzyme is fixed - but in complex networks this assumption seems to be weak. Enzyme concentration has to be much less than the substrate concentration and in e.g. oscillatory systems the substrate amount changes over time. In those cases, the minimum value of the substrate has to provide the base of the calculation. Simulation runs with complex rate functions may offer an initial guess for an appropriate enzyme value that satisfies the assumptions made for unpacked nonlinear terms.

We would like to emphasize that decomposition of the Michaelis-Menten kinetics is not always necessary, but in a compositional modeling framework it has to be available (as a part of a library). Assumptions have to be checked during the execution and the decomposition might be especially useful for further extension of the model. For instance, when an inhibitor of the enzyme is present or two substrates of the same enzyme are introduced, details of the complex reactions have to be elucidated. Hidden details of the assumptions might become a limitation of the currently available transformation of the deterministic models into the stochastic framework. Furthermore, we are aware of that the quasi-steady state assumption of enzymatic reactions in complex models might be violated. In those cases appropriate decomposition of the reaction into elementary steps is crucial. Parameter check or parameter estimation is possible within the CoSBi Lab platform [255], thus assumptions of the approximation or the algorithm can be monitored during the runs.

In this section, we provided a description of a template for enzyme kinetics in BlenX with a parameter search based on basic mathematical calculus. Implementation of the template-library into the CoSBi Lab platform [256] might automatize the method of parameter estimation as it contains inference tools (KInfer [140]). We only presented a hint of how parameters might be achieved.

5.3 Hill kinetics of cooperativity

Cooperativity is a phenomenon displayed by enzymes and receptors that have multiple binding sites and their affinities are modified upon the binding of a ligand. The classical example for such a behavior is the increased affinity of hemoglobin's four binding sites for oxygen when the first oxygen molecule binds [142]. Cooperativity frequently occurs in biological systems, most transcription factors are also composed of several repeated protein subunits. Often, full activity of these regulatory complexes is only reached when multiple subunits can bind to the target. Hill function provides a useful phenomenological equation approximating this cooperative process. In biological models, Hill sigmoidal response curves are commonly used to substitute multiple reaction steps with one term. The intermediate complexes are hidden in the Hill equation making the details of the system unavailable. This is a convenient way to handle unclear scenarios in biology, although the average of a nonlinear function (e.g. Hill function) is generally found to differ from the function of the average [249]. The proper usage of these mathematical functions in larger models is crucial. The Hill equation assumes that *n* molecules of an entity (e.g. ligand) bind to a scaffold (e.g. receptor) simultaneously [142] and intermediate states do not occur. This is physically possible only if the number of ligands is equal to 1 (*n=1*), but in most cases this approximation is far from reality.

The Hill equation assumes simultaneous binding of *X* molecules to *Y*, thus the reaction is often described as

$$Y + nX \overset{kf,kb}{\longleftrightarrow} YX_n$$

where k_f is the rate constant of the forward reaction and k_b is of the backward. At equilibrium, the ratio of bound to total receptors is given by the Hill equation

$$F_{Hill} = \frac{Bound}{Total} = \frac{YX_n}{Y + YX_n} = \frac{X^n}{X^n + J^n}$$

where the dissociation constant is $J = k_b/k_f$ and *n* provides the number of ligands. The steepness of the transition of the sigmoidal curve depends on the number of ligands (*n*) and *J* provides the number of ligands at which half of the receptors (*Y*) are bound Figure 13.

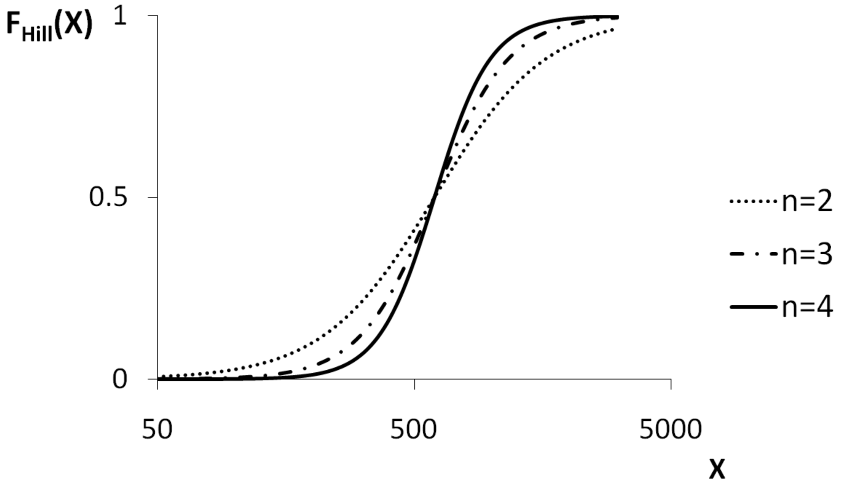


Figure 13: Analytic calculation of the Hill function for different Hill coefficients (*n*=2,3,4). When *F_{Hill}(X)* is 0.5, *J* equals to 600#.

Gene expression is known to be a particularly complex and noisy task [19-21]. Transcriptional regulation is often characterized by a sigmoid Hill function, where nonlinearity arises from the assumption that the transcription factor forms multimers before binding to the DNA (shown in Figure 14). This property creates an abrupt switch from one state (where transcription is “off”) to another state (where transcription is “on”). The detailed mechanism behind the observed behavior is still unclear, but there are several hypothesis and models available [34,257,258]. Although it is difficult to choose one particular model scenario [259], we decided to investigate a simplified model of positive cooperativity that captures the requirements of containing only elementary reactions but still maintaining the sigmoidal property of the module.

5.3.1 Decomposition of the module - transcriptional regulation case

Transcriptional factors (abbreviated as *TFs*) often form multimers when binding to DNA and inducing transcription [260,261]. This cooperative effect creats sigmoidal response of the system to the change of the active transcriptional factor amount. Such coooperativity widely occurs and the Hill equation is a good approximation of the underlying mechanisms, although it assumes simultaneous binding of the TFs to the promoter region that is far from the realistic picture. As intermediate states appear during the reaction, sequential binding of the transcription factor to the promoter has been considered for this study. The following scheme (on Figure 14) approximates the Hill function when the intermediate state (*TF2*) does not accumulate and positive cooperativity is present. Several other interactions are plausible [34], but are not investigated in this study for the sake of simplicity.

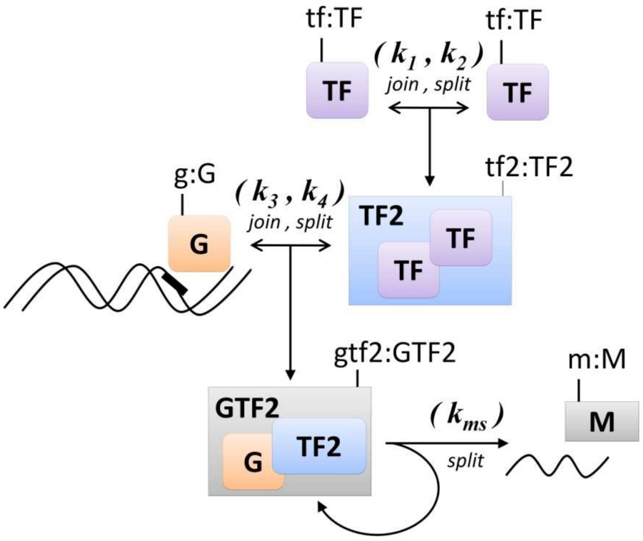


Figure 14: Transcriptional regulation: Sequential binding of transcriptional factors (*TF*s) occurs then a homodimer (*TF2*) sits onto the promoter of a gene (*G*) and transcription results in messenger RNAs (*mRNA*s). Reactions are described through mass action kinetics with rate constants k_1, k_2, k_3, k_4 and k_{ms}.

BlenX offers a formal and efficient definition of join and split events of boxes as complex formation and complex dissociation occur in biological systems. Transcription factors (*TFs*) form multimers (in this example *TF2*) through a *join* event, enhancing the affinity of its binding onto the gene's promoter region (*G*). Dissociation of the complexes occurs through *split* events:

when(TF,TF::rate(k1)) join(TF2);
when(TF2::rate(k2)) split(TF,TF);
when(TF2,G::rate(k3)) join(GTF2)
when(GTF2::rate(k4)) split(G,TF2);

Positive cooperativity results in a low dissociation constant of the *GTF2* trimer. The joined complex is able to transcribe mRNAs (*M*s) of the gene through a split event that creates *M* boxes with the release of the active trimer.

when(GTF2::rate(kms)) split(GTF2,M);

Binding processes are assumed to be reversible. Elementary reactions of the system are summarized in Table 8. The model contains the following *boxes*:

```
let G : bproc = #(g:0,G)            [ nil ];
let TF : bproc = #(tf,TF)           [ nil ];
let TF2 : bproc = #(tf2,TF2)             [ nil ];
let GTF2 : bproc = #(gtf2,GTF2)          [ nil ];
let M : bproc = #(m:0,M)                 [ nil ];
```

Description	**Reactions**	**Rate constants**	**Units**
Homodimerization of *TF*s	$2TF \rightarrow TF2$	k_1	*1/(min#)*
Dissociation of *TF2*	$TF2 \rightarrow 2TF$	k_2	*1/min*
Formation of an active complex	$TF2 + G \rightarrow GTF2$	k_3	*1/(min#)*
Dissociation of the active complex	$GTF2 \rightarrow TF2 + G$	k_4	*1/min*
Synthesis of the *mRNA*	$GTF2 \rightarrow TF2 + M$	k_{ms}	*1/min*
Degradation of the *mRNA*	$M \rightarrow$	k_{md}	*1/min*

Table 8: Reactions for Hill kinetics for the requirement of at least 2 ligands (TFs).

Sigmoidal curves are often measured in experiments providing specific Hill constants (such as n, *J*) to the Hill function. The Hill curve describing e.g. a transcriptional regulation scheme is not the proper way to apply stochastic calculations. Elementary steps of the sequential binding scheme contains four rate constants (k_1, k_2, k_3, k_4) that have to be determined from the constants n and *J*. Derivation of the missing parameters is calculated from the mass action kinetics describing the system. At equilibrium the intermediate complexes are assumed to be stable, thus

$$\frac{TF2}{TF^2} = \frac{k_1}{k_2}$$

and

$$\frac{GTF2}{G \cdot TF2} = \frac{k_3}{k_4}$$

As the total amount of gene promoters do not change, *G*=*G_{tot}*-*GTF2* leads to the term

$$\frac{GTF2}{G_{tot}} = \frac{TF^2}{\frac{k_2 \cdot k_4}{k_1 \cdot k_3} + TF^2}$$

that is identical to a Hill function

$$\frac{Bound}{Total} = \frac{X^n}{X^n + J^n}$$

Note that n=2 and $J^2 = (k_2 \cdot k_4)/(k_1 \cdot k_3)$. The rate of transcription of M is

$$k_{ms} \cdot \frac{TF^2}{J^2 + TF^2} = k_{ms} \cdot GTF2 \cdot G_{tot}$$

where G_{tot} is a constant equals to 1# in this example.

The Hill coefficient (n) and the dissociation constant (J) determine only the relation of the four rate constants ($J^2 = (k_2 \cdot k_4)/(k_1 \cdot k_3)$), but different values may satisfy the same reaction scheme. We analyzed several choices of k_1, k_2, k_3, k_4 suiting the constraint described above, and we compare the results through the response coefficient (R) of the sigmoidal curve [241]. The response coefficient has been shown to allow us to measure the steepness of the transition of sigmoidal responses, such as in a Goldbeter-Koshland zero-order ultrasensitive switch (see below). The R coefficient is defined as $S_{0.9}/S_{0.1}$, the ratio of the signal (substrate) amount required to give 90% saturation relative to the amount required to give 10% saturation [257]. In our example, the complex function is defined as the dissociation constant (J) is equal to 600# and the Hill coefficient (n) is 2. The reaction rate in this case equals to

$$\frac{TF^2}{600^2 + TF^2}$$

and the value of the response coefficient R equals to 9.

5.3.2 Simulation results

We chose eight different sets of parameters (Table 9) satisfying the relation

$$J^2 = (k_2 \cdot k_4)/(k_1 \cdot k_3)$$

and run with each set for long time. We measured the time average value of the bound form (*GTF2*) in case of a few initial amounts of *TF*s and calculated the response coefficient (*R*) and the actual Hill coefficient (n') for each set of parameters [241]. The derived Hill coefficient equals to

$$n' = \log 81/\log \mathrm{R}$$

The derived dissociation constant (*J'*) is calculated from the simulation points. The root mean square error of the fit to the simulation results and the simulation point's error to the theoretical Hill function curve are also shown in Table 9.

If we compare the results of the complex function and the unpacked module, we see that when the assumption for $K_1 >> K_2$ is valid, the decomposed module gives a good fit to the theoretical Hill curve (Figure 15). Our results agree with the observation of [34] that for simple sequential binding schemes the only condition under which the Hill coefficient does accurately estimate the number of binding sites is when marked positive cooperativity is present. Furthermore, our analysis points toward, that the larger the difference between the dissociation constants K_1 and K_2, the better the fit (e.g. compare set 1 to set 2). We also see that the derived Hill coefficient is not necessarily equal to the number of binding sites on the gene.

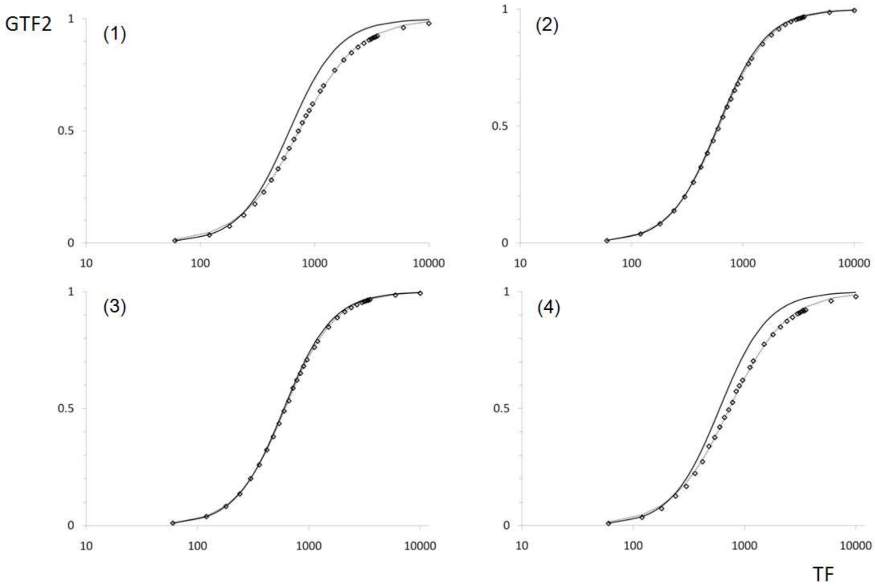


Figure 15: The best fits (set 1,2,3,4) satisfy the assumption of positive cooperativity as *K1 >> K2* and provides a good fit to the theoretical complex function. Data points are calculated as the time average of the GTF2 value of the simulation results at a given amount of TF with different parameter sets (see above). Solid curves show the analytic calculation of the Hill function with *J*=600# and *n*=2. All the plots are on a logarithmic x-axis.

Biological systems respond to signals in a dynamic way, although the underlying mechanisms depend on single molecules that are expressed and coordinated in a stochastic way. Recently, single cell measurements led to key hypothesis how independently fluctuating elements can be orchestrated in a well-organized manner. Cai et al [262] found that transcription of molecules happen in a burst-like manner, creating probabilistic transcription initiations that generate downstream transcription. Furthermore, the system seems to be controlled by frequency modulation instead of changes in transcription-factor concentrations. This is an elegant explanation for the large variety in the levels of expression or the concentration of transcription factors resulting in a particular response mechanism. Our simulation results also show behavior coincident to frequency modulation theory [259,262] where we see dense bursts of active transcription factors (*GTF2*) that results burst-like transcription giving similar downstream results that occur in a concentration dependent transcription in deterministic models (Figure 16). The rates of complex formations provide a frequency of transcription that determines the abundance of the mRNA (*M*). The different parameter sets define how often the active complex (*GTF2*) is present. This transcription is random, as it was discussed previously by Cai et al. [262]. Furthermore, our findings are in accordance with a recent publication investigating molecular noise of transcriptional event within the cell cycle regulation network. Csikasz-Nagy and Mura [263] showed that few steps of gestation and senescence of mRNA are enough to give a good match for both the measured half-lives and variability of cell cycle-statistics. Their result suggests that the complex process of transcription can be more accurately approximated by multi-step linear processes. Our analysis shows accordance to previous publications and also provides guidance to parameter search. The parameter set 2 - that assumes larger dissociation constant for the *TF* dimerization step ($K_1 >> K_2$) - provides better fit to the Hill function than the simulations with the parameter set 8. When the number of initial transcription factor is small (100#), the frequent bursts of *GTF2* activation give similar response than the complex Hill function assuming sigmoid reaction. The proper rhythm of stochastic *GTF2* activity peaks is able to increase the amount of *M* in the expected and less noisy manner. In case of the parameter set 8, the transcription is more frequently “switched off” and cannot provide the same amount of transcript (*M*) as one approximated complex Hill term do. As the number of transcription factors is increased (Figure 16 D and H), we see that the effect of frequent bursts becomes less important as the regularly activated *GTF2* cannot compensate the lack of a continuously active complex.

	k_1	k_2	k_3	k_4	$K_1=k_2/k_1$	$K_2=k_4/k_3$	$J=\sqrt[2]{K_1\cdot K_2}$	n'	J'	Error of the fit of the estimated Hill function	R	Error of the fit of the theoretical Hill function
units	1/(min#)	1/min	1/(min#)	1/min	#	#	#	-	#	-	-	-
set0	-	-	-	-	-	-	1/α	2	600	0	9	0
set1	1α	10	1000α	100	10	0.1	1/α	1.67	729	0.007511	13.89	0.066913
set2	1α	100	1000α	10	100	0.01	1/α	1.95	612	0.001737	9.52	0.008401
set3	10α	1000	100α	1	100	0.01	1/α	1.95	612	0.002859	9.52	0.600704
set4	100α	1000	10α	1	100	0.01	1/α	1.68	731	0.009201	13.68	0.709089
set5	10α	1	100α	1000	0.1	10	1/α	1.05	12418	0.002033	65.71	0.600837
set6	1α	1	10α	1000	0.01	100	1/α	1	124191	0.000234	81.00	0.709229
set7	1000α	100	1α	10	0.1	10	1/α	1.06	12248	0.001434	63.16	0.600704
set8	1000α	10	1α	100	0.01	100	1/α	1.03	110937	0.000255	71.27	0.709089

Table 9: Multiple simulation results on the module of the Hill kinetics. Set 0 refers to the deterministic version of the complex Hill function. Set 1-set 8 are different sets of parameters for the “unpacked” module.

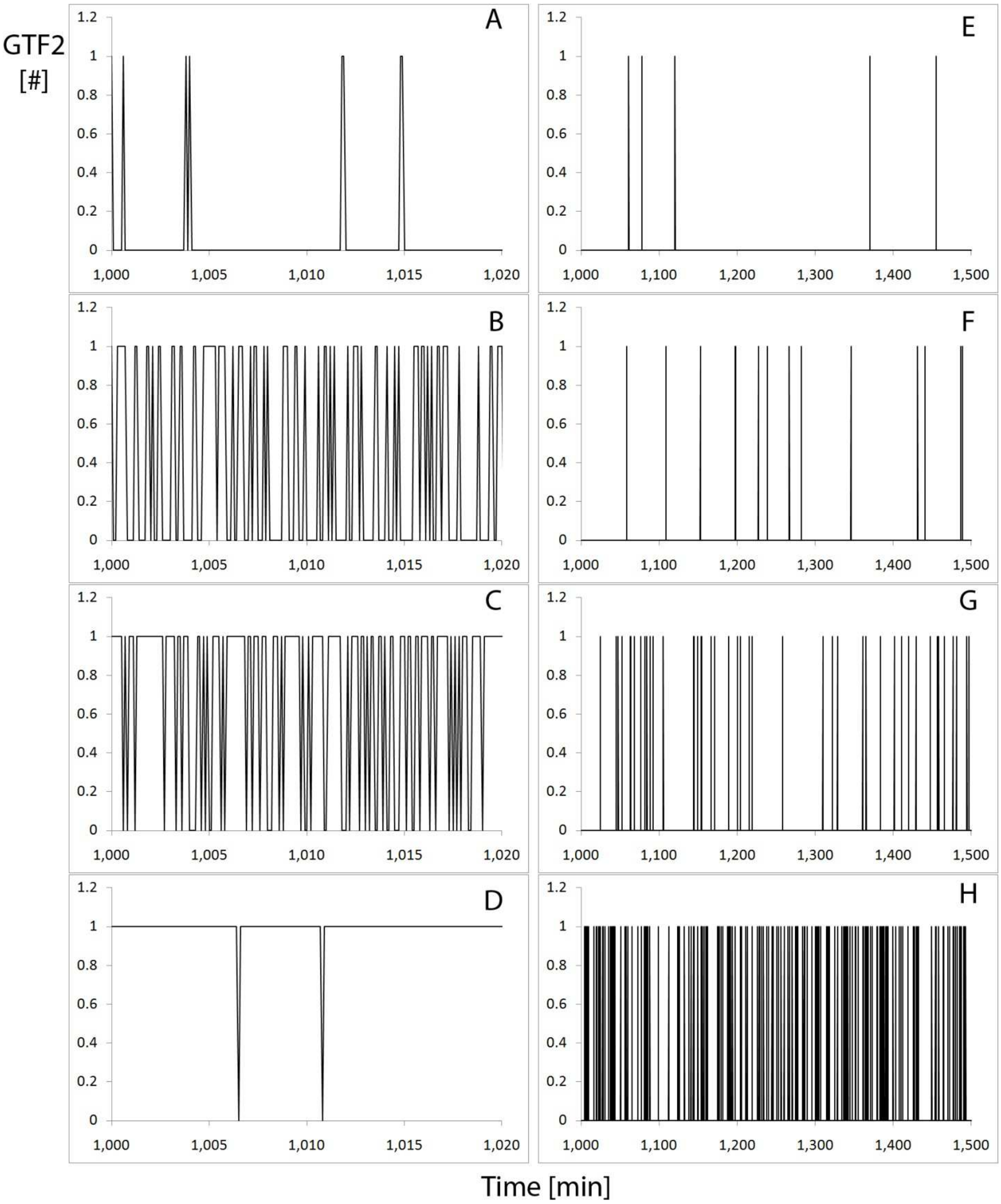


Figure 16: Simulation results of the unpacked Hill function. The frequency of *GTF2* activation defines the speed of the reaction. (A-D): Simulation results for the parameter set 2 and with different value of the *TF*s (A:*TF*=100#, B: *TF*=500#, C: *TF*=1000#, D: *TF*=5000#).(E-H): Stochastic simulations for the parameter set 8 (E:*TF*=100#, F: *TF*=500#, G: *TF*=1000#, H: *TF*=5000#). Note that the time scale of the two columns differs. In case of set 2 bursts are more frequent than in the case of set 8.

In order to present more in details simulation of the chosen two parameter sets (set 2 and 8) are shown on Figure 17. When the amount of *TF* (signal) is low, effective transcription -

producing sufficient amount of mRNAs - occurs if transcriptional bursts are more frequent. As the level of initial signal is increased, the transcription machinery is mostly active (with a *GTF2* value equals to 1). When there are higher amount of molecules in the system, the frequency of bursts becomes lower.

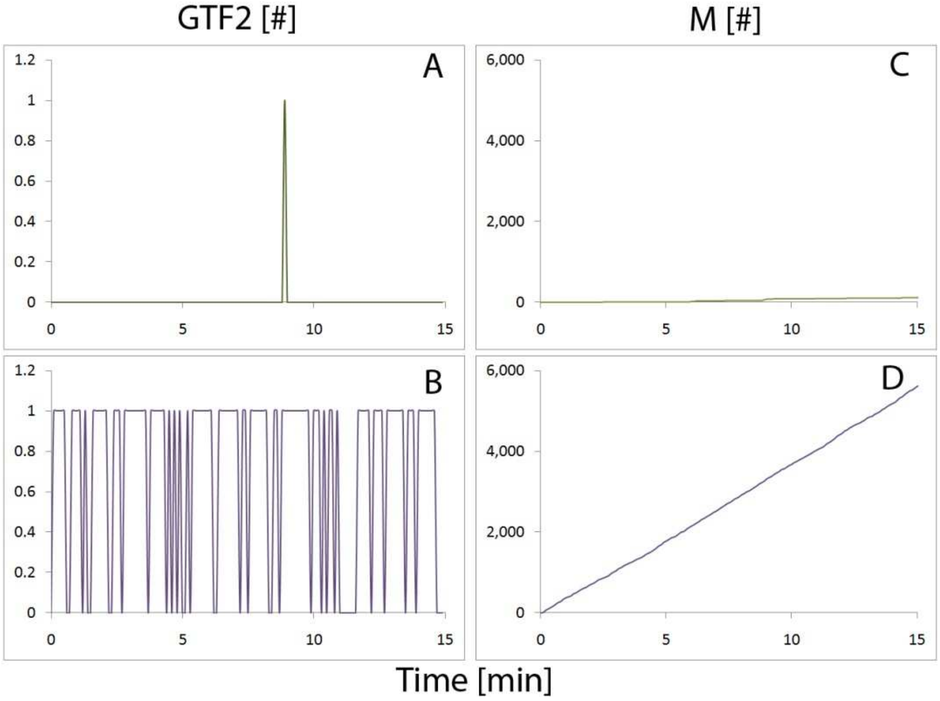


Figure 17: Simulation: with *TF*=1000#; set8 (A, C) and set2 (B, D) response differs. Set8 cannot provide a response that follows the behavior of a compound Hill function with particular assumptions. The frequency of bursts is not sufficient enough for describing the synthesis of M in the desired manner.

The module presented above states for a representation of the complex Hill term described by elementary steps and modeling the transcription of a gene explicitly. It can be easily dragged and dropped into a larger model as it is shown later.

5.4 The Goldbeter-Koshland switch

In the 1980s, Albert Goldbeter and Daniel E. Koshland introduced an elegant description of interacting oppozing enzymes acting on a substrate that switches between two forms (representing a kinase-phospatase pair woring by adding and removing phosphate groups or a GTPase, GAP pair addig or removing GTP's) providing an 'ultrasensitive' behavior to the system [241]. Biological systems must respond to stimuli rapidly and this often happens by phosphorylation events (e.g. metabolic pathways). The terms "ultrasensitivity" have been defined to indicate cases in which the sensitivity is greater or less than that to be expected from the standard hyperbolic, Michaelis-Menten response. In these cases "turning on or off" the activity of the elements of a pathway responds fast to relatively small changes in effectors' concentrations.

The scheme of cooperative binding is one way to increases the sensitivity of a system (discussed in the 5.3 session (Hill kinetics)). However, there are other modes to generate sigmoidal response. An enzyme may be present at more than one step in a pathway (like in the glycogen cascade [264]). Furthermore, in covalent modification schemes (like phosphorylation [265]) when one or more of the converter enzymes operate in the "zero-order" region, ultrasensitivity is observed. In this case, the enzyme activity is saturated with respect to the protein substrate and the velocity of the reaction is independent of changes in the substrate quantity. A nonlinear function describing a reaction presented ahead is referred to as the Goldbeter-Koshland switch [241] and it has been shown to be able to generate sensitivity equivalent to cooperative enzymes with high Hill coefficients [266].

5.4.1 The Goldbeter-Koshland module, as a composition study

The Goldbeter-Koshland function is our first example of a higher composition in this study, as - roughly speaking - two Michaelis-Menten templates built together can serve as one Goldbeter-Koshland motif. By defining a template for interacting enzymes (the scenario described above) we end up with another possible way to implement non-linearity into our BlenX model that behaves with a sigmoidal response, but contains only elementary steps. It is a commonly used building block of biological pathways - such as protein cascades -, thus it is worth to store this template in our library.

Imagine a protein with two forms: active (*P*) and inactive (P_{mod}) (it is a common motif that frequently occurs in biological systems). If the activation and inactivation are carried out by two different enzymes resulting in covalent modifications (e.g. phosphorylation or dephosphorylation) the system can be described by two, Michaelis-Menten-type of enzymatic reactions (Figure 18). If the same assumptions are valid as for the Michaelis-Menten term - when the enzymes (E_1 and E_2) are saturated by the substrates (*P* and P_{mod}) - and our simulation stays at the zero order regime, the response of the system will provide an ultrasensitive switch even to slight changes.

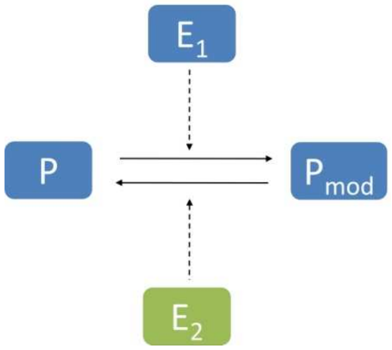


Figure 18: Schematic wiring diagram of covalent modification of *P* through the enzymes E_1 and E_2, acting on *P* in an opposite way.

The steady-state behavior of this module is expressed mathematically as a nonlinear function:

$$0 = \frac{k_1 \cdot E_1 \cdot (P_{total} - P_{mod})}{J_1 + (P_{total} - P_{mod})} - \frac{k_2 \cdot E_2 \cdot P_{mod}}{J_2 + P_{mod}}$$

where J_1 and J_2 are the Michaelis-Menten constants of the enzymatic reactions.

Golbeter and Koshland proposed a mathematical relationship for steady state solution of the system in the zero-order regime:

$$\frac{P_{mod,st.st}}{P_{total}} = G\left(k_1 \cdot E_1, k_2 \cdot E_2, \frac{J_1}{P_{total}}, \frac{J_2}{P_{total}}\right) =$$

$$= \frac{2 \cdot k_1 \cdot E_1 \cdot \frac{J_2}{P_{total}}}{B + \sqrt{B^2 - 4(k_2 \cdot E_2 - k_1 \cdot E_1) \cdot k_1 \cdot E_1 \cdot \frac{J_2}{P_{total}}}}$$

where

$$B = k_2 \cdot E_2 - k_1 \cdot E_1 + \frac{J_1}{P_{total}} \cdot k_2 \cdot E_2 + \frac{J_2}{P_{total}} \cdot k_1 \cdot E_1$$

The Goldbeter-Koshland function takes values between 0 and 1 (for the fraction of modified and unmodified proteins) and has a sigmoidal behavior. The smaller parameters J_1 and J_2 provide the steeper the function. The switch is ultrasensitive only in the zero order regime, although [267] showed that the sigmoidal response of the function is restored also in the first order regime if we introduce an additional cooperative mechanism of the phosphorylation reaction. In this study we do not touch that case.

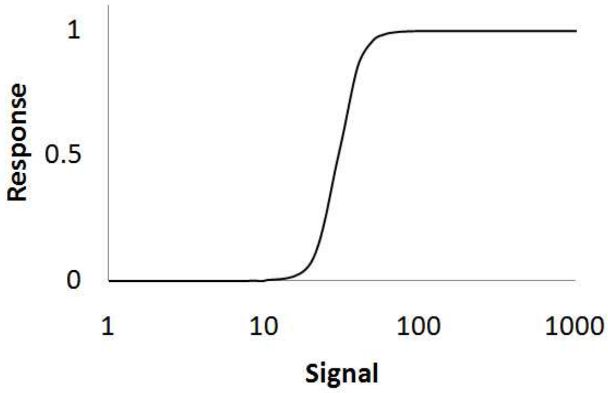


Figure 19: Schematic representation of a Goldbeter-Koshland ultrasensitive switch. In our example the *signal* represents the enzymes (E_1 or E_2) modifying the substrates, changing their *response* (the amount of P or P_{mod}).

5.4.2 Elementary steps of a Goldbeter-Koshland module

The Goldbeter-Koshland module is composed of two Michaelis-Menten reactions. P is modified by E_1 through a complex formation step. P_{mod}, the modified form of P is recovered through binding to E_2. Depending on the initial amounts of the molecules, the system can exist in two states: (1) either P is active (form P) due to the presence of E_2 or (2) P_{mod} level is higher than the amount of P.

Due to the variety of BlenX expressions, there is more than one solution to achieve the same reaction networks. This flexibility of the program-code makes modeling a creative job. Although the current programming method in BlenX assumes that users have programming skills and they construct models in the most beneficial way. The users might have to think in advance to the subsequent use - possible extension - of the model. The efficiency of composition differs in the variety of solutions providing and easier or more difficult way of compositionality in the future.

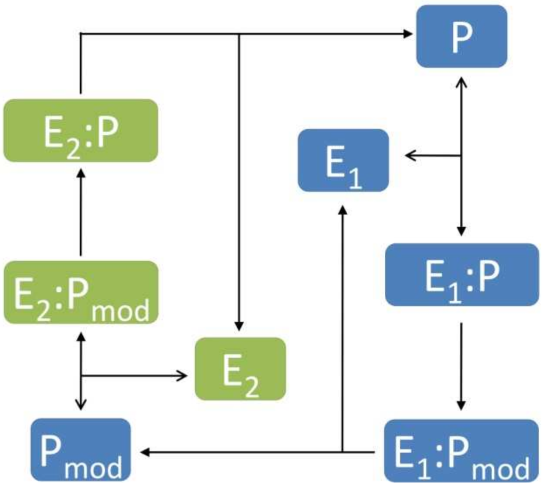


Figure 20: Reversible enzyme modification is described with elementary steps. Enzyme-substrate complexes (E_2:P_{mod} and E_1:P) are present during the reaction.

In the followings, we will show possible process algebra modules for the same Goldbeter-Koshland behavior exploiting the expressive power of BlenX and proposing an optimal solution for compositional modeling.

5.4.3 A simple solution for the decomposition of the Goldbeter-Koshland switch

Probably the most intuitive way - for biologists - of modeling the interconnected enzymatic reactions with BlenX is the use of events, especially if someone approaches the problem starting from the description of the biochemical reactions that present in the system. Events represent rules that are able to perform the substitution of boxes with other boxes. All the possible complexes in the system are defined explicitly at the beginning through the join and split operations. Their use is essential in modeling the dynamics of networks in which the use of the communication primitives are not intuitive.

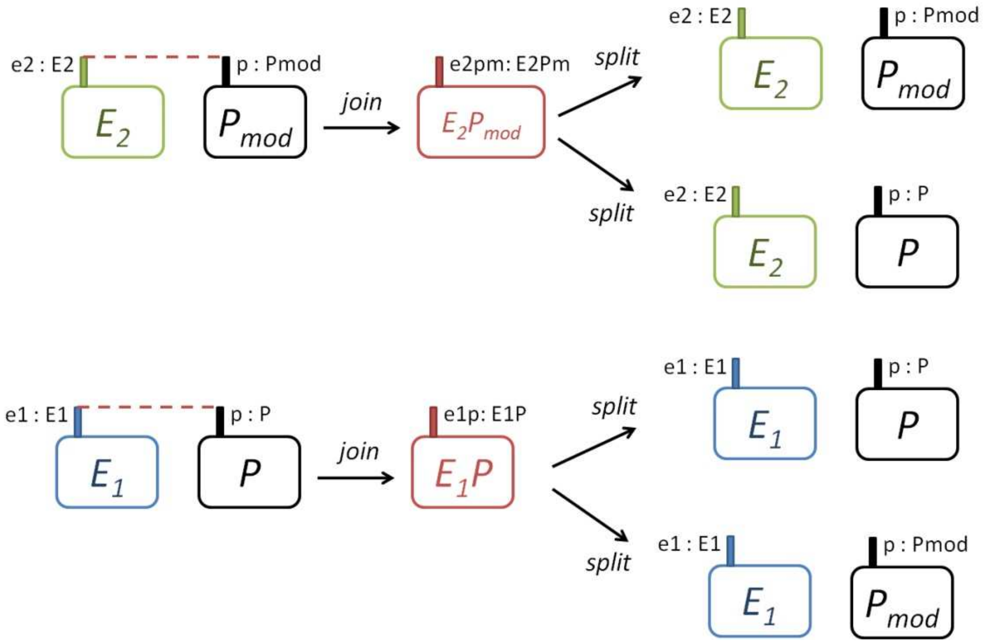


Figure 21: BlenX representation of enzymatic reactions through events. The substrates are prone to form complexes with the enzyme.

In case of a description of the switch with events, the definitions of the protein (*P*), the modified protein (*P_{mod}*) and the active enzymes (*E_1* and *E_2*) do not include internal processes and communication actions:

```
let P :          bproc = #(p:0,P)          [ nil ];
let Pmod :       bproc = #(p:0,Pmod)       [ nil ];
let E1 :         bproc = #(e1:0,E1)        [ nil ];
let E2 :         bproc = #(e2:0,E2)        [ nil ];
```

We also define the “intermediate” complexes explicitly in the model similar to the classical approach of modeling:

```
let E1_P :       bproc = #(e1p:0,E1P)      [ nil ];
let E2_Pmod :    bproc = #(e2pm:0,E2Pm)    [ nil ];
```

Chemical reactions, in this example, represent rules for complex formation and dissociation of enzymes and substrates by introducing the enzyme-substrate complexes. This solution is the most intuitive explanation of biochemical reactions and probably the closest to the classical modeling methods, making translation and composition of deterministic solutions into the process calculus framework. The type of events (*join*) substitutes the box of the enzyme (*E1* or *E2*) and the substrate (*P* or *Pmod*) with one box, the enzyme-substrate complex (*E1_P* or *E2_Pmod*):

```
when    (E1,P::rate(e1_1))          join    (E1_P);
when    (Pmod,E2::rate(e2_1))       join    (E2_Pmod);
```

The other event (*split*) create the opposite, it substitute one box (*E1_P* and *E2_Pmod*) with the enzyme and the substrate:

```
when    (E1_P::rate(e1_2))          split   (E1,P);
when    (E2_Pmod::rate(e2_2))       split   (E2,Pmod);
```

or the enzyme and the product of the reaction:

```
when    (E1_P::rate(e1_3))          split   (E1,Pmod);
when    (E2_Pmod::rate(e2_3))       split   (E2,P);
```

We tested three different sets of parameters for possible ratios among the stochastic rate constants of the elementary steps. All three cases satisfy the assumption of the saturated enzymes that are present in the system with less number of molecules than the substrates. Initial conditions are set for analyzing the reversibility of the ultrasensitive switch. No hysteresis is seen as expected (Figure 22). The sigmoidal response coefficients can be also calculated as

$$R = \frac{81(K_1 + 0.1)(K_2 + 0.1)}{(K_1 + 0.9)(K_2 + 0.9)}$$

where $K_1 = {}^{J_1}/_{P_{total}}$ and $K_2 = {}^{J_2}/_{P_{total}}$

R = 2.02 for all cases which corresponds to a Hill function with a coefficient equals to n=6.25.

rate constants	units	set 1	set 2	set 3
e1_1	$1/(\mathrm{min} \cdot \#)$	$1 \cdot \alpha$	$1 \cdot \alpha$	$1 \cdot \alpha$
e1_2	$1/\mathrm{min}$	0.045	0.05	0.005
e1_3	$1/\mathrm{min}$	0.005	0.05	0.045
e2_1	$1/(\mathrm{min} \cdot \#)$	$1 \cdot \alpha$	$1 \cdot \alpha$	$1 \cdot \alpha$
e2_2	$1/\mathrm{min}$	0.045	0.05	0.005
e2_3	$1/\mathrm{min}$	0.005	0.05	0.045
J1	#	$0.05/\alpha$	$0.05/\alpha$	$0.05/\alpha$
J2	#	$0.05/\alpha$	$0.05/\alpha$	$0.05/\alpha$

Table 10: Parameter sets used for the simulations satisfying the Goldbeter-Koshland assumptions. α is a conversion factor for the stochastic rate constants (α=0.00167).

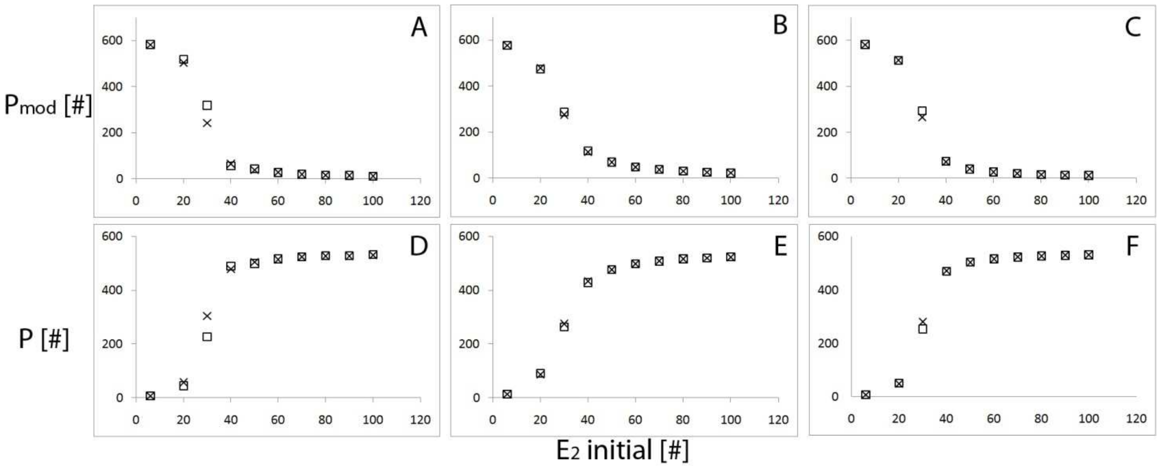


Figure 22: Stochastic simulations started with different initial conditions and analyzing the steady state of the system (P_{mod} and P are steady state values of the model calculated from the time average after reaching equilibrium). E1 value is 30#. (A and D): Parameter set 1. (B and E): Parameter set 2. (C and F): Parameter set 3. Squares sign for the runs with initial conditions *P*=0#, *Pmod*=600#. Crosses indicate the steady state values of the runs with initial condition *P*=600# and *Pmod*=0#. We see no hysteresis. Data points are calculated from time averages of the steady state values of multiple simulations. A single run of these cases are shown on Figure 23.

Although in all cases the system tends to reach an equal state, the time evolution of the reaction strongly depends on the parameters. The Goldbeter-Koshland function assumes rapid equilibrium similar to the Michaelis-Menten relation. However, when the assumptions of the Michaelis-Menten are valid (the conditions *e1_2 >> e1_3* and *e2_2 >> e2_3* apply) the runs do not satisfy the behavior of a Goldbeter-Koshland switch (Figure 23). The parameter set 2 and 3 matches the solution of the complex rate function better than the parameter set 1. In the first set, the transient behavior lasts long which brings the system outside of the zero-order regime where the change of the amount of the enzyme-substrate complex cannot be neglected. Although the value of the Michaelis-Menten constants are low and the substrate of each reaction saturates the enzyme, we see that the rate limiting step in the second and third cases is the dissociation of the complex instead of the catalytic step (in the parameter set 1). We see that the Michaelis-Menten assumptions might be incorrect for non-isolated systems, such as the Goldbeter-Koshland switch, thus implementation of a reusable module makes the check of assumptions available and can avoid misinterpretation of changes arising from composition of a larger model built upon the assumptions of the Michaelis-Menten kinetics of enzymatic reactions.

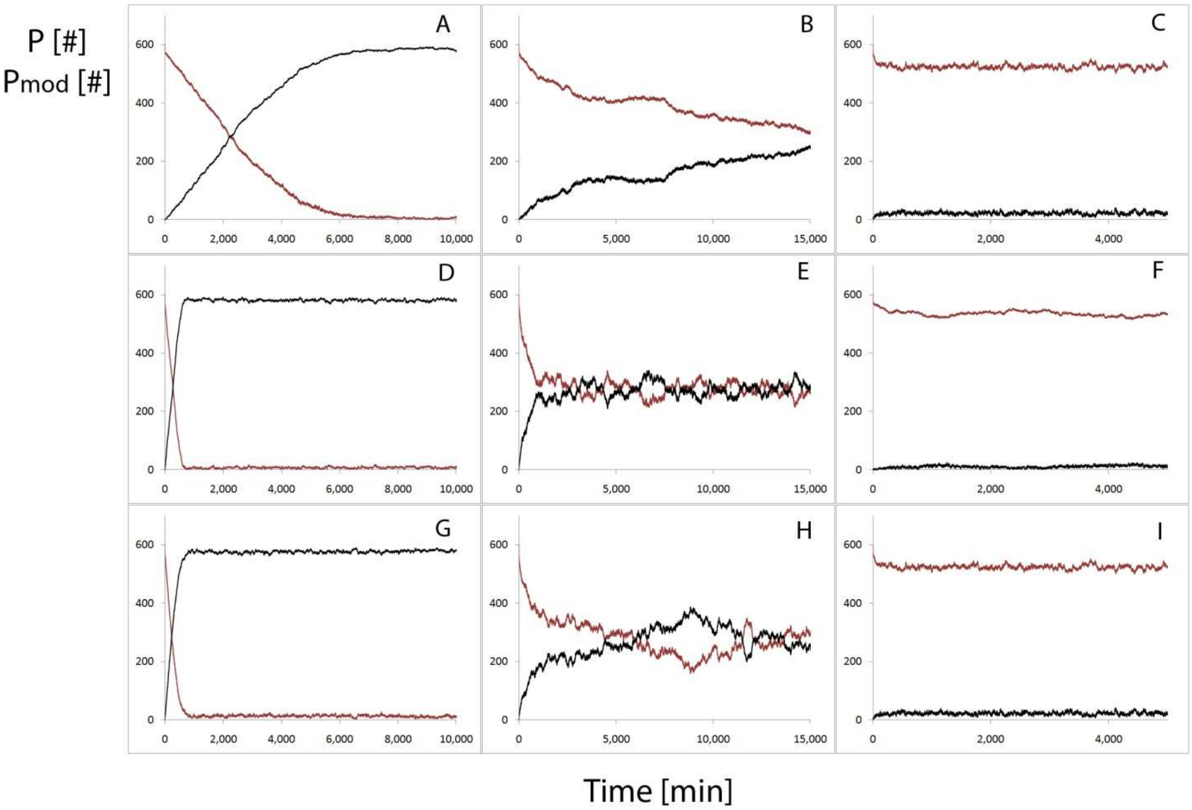


Figure 23: Time evolution of the reaction scheme. (A-C): simulations with the parameter set 1. (D-F): simulations with the parameter set 2. (G-I): sample simulations with the parameter set 3. (A,D,G): initial value of E_2=6#; (B,E,H): E_2=30#; (C,F,I): E_2=100#; (A-I): initial amount of P=600# (red curve) and P_{mod}=0# (black curve).

On Figure 23 simulation results of different parameter sets shows the expected variance in the level of noise depending on the initial number of the enzyme *E2*. If we take a look at the signal-response curves on Figure 22 and then to Figure 23, we notice that at smaller or at larger number of *E2* the noise is reduced compared to the *E2* values closer to the inflexion point of the sigmoidal curve (Figure 23 B, E, H).

Compositionality can be carried out easily in this case as all the elementary steps and complexes are present in the module from the beginning, thus another module can be interconnected even through the intermediate complexes. However, the definition of the initial values for the enzyme-substrate complexes is a crucial step. By default they are assumed to be zero as the complex reaction term assuming Michaelis-Menten kinetics does not include the initial (and later) steady state values, thus we lack the knowledge about the intermediate species when we want to transform this nonlinear function into elementary steps. As Goldbeter-Koshland's switch states that the system is in equilibrium during the ultrasensitive change, initial amounts for the intermediate complexes should be chosen to be equal to their steady state values. Calculation of those states can be implemented into the tool as an automatic feature. BlenX can propose initial values of the intermediate complexes assuming equilibrium for a given initial set of the model defined by the user. Calculations are based on:

$$|ES|_{st.st.} = \frac{e1}{e2 + e3} \cdot |E|_{init} \cdot |S|_{init}$$

$$|E|_{init} = |E|_{total} - |ES|_{st.st.}$$

$$|S|_{init} = |S|_{total} - |ES|_{st.st.}$$

In the subsequent Chapter we show another solution for encoding the Goldbeter-Koshland motif.

5.4.4 Conditions driven simulations for the Goldbeter-Koshland study

BlenX allows a novel and more elegant way to design a program through coding the probable reactions with conditions. These conditions check the state of the boxes' interfaces, allowing the change of their internal behavior depending on their actual configurations.

In case of a Goldebeter-Koshland switch enzymes and substrates form complexes. With the conditions it is possible to guard reactions though checking the state of the boxes if they are available for binding and firing the action of changing their behavior.

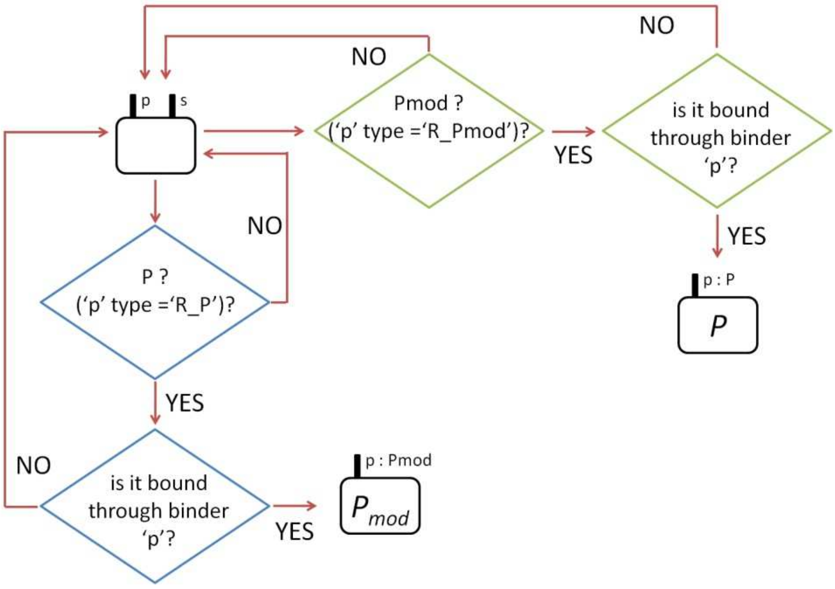


Figure 24: Representation of the conditions driven execution of the reaction.

The following rules define the state of the molecules (boxes).

(1) If the binder 'p' belongs to the box *P* (the type of the binder is equivalent of 'R_P') and if it is 'bound' (it forms a complex with another molecule), change the type of its binders (to 'R_Pmod') resulting in a transition to the box P_{mod} because the binder types and the box's internal behavior define the actual identity of the box. Then check sequentially if the binder 'p' belongs to the box P_{mod} (its type is 'R_Pmod') and if this 'p' binder is not bound. If the conditions are both met, the program fires the next process ('*proc_check1*'), stepping onto the second sequence of conditions (2).

let proc_P_Pmod :

pproc = if ((p,bound) and (p,R_P)) then ch(rate(e1_3),s,Pmod).ch(p,R_Pmod).proc_check1 endif;

let proc_check1 :

pproc = if ((not (p,bound)) and (p,R_Pmod)) then x?().nil endif;

(2) If the binder 'p' belongs to the box P_{mod} (the type of the binder is equivalent of 'R_Pmod') and it is bound (it forms a complex with another molecule), change the type of

its binders (to ‘R_P’) resulting in a transition to the box ‘P’. After this, sequentially check if the binder ‘p’ belongs to the box *P* (its type is ‘R_P’) and if this ‘p’ binder is not bound. By stepping back into the first condition (1), we assure the persistent check of the states of the molecules.

```
let proc_check2 :
pproc = if ((not (p,bound)) and (p,R_P)) then x?().nil endif;
```

```
let proc_Pmod_P :
pproc = if ((p,bound) and (p,R_Pmod)) then ch(rate(e2_3),s,P).ch(p,R_P).proc_check2 endif;
```

```
let choice :
pproc = if ((p,bound) and (p,R_P)) then ch(rate(e1_3),s,Pmod).ch(p,R_Pmod).proc_check1
endif + if ((p,bound) and (p,R_Pmod)) then ch(rate(e2_3),s,P).ch(p,R_P).proc_check2 endif;
```

The binding affinities are specified in a separate file and describe all the possible roles of a molecule in a particular system:

```
{ R_Pmod, R_P, Pmod, P, E1, E2 } %%
{   (R_P,E1, rate(e1_1), rate(e1_2), 0), (R_Pmod,E1, 0, inf, 0),
    (R_Pmod,E2, rate(e2_1), rate(e2_2), 0), (R_P,E2, 0, inf, 0)         }
```

The binders and the internal behavior of the boxes are set as the followings.

```
let P :                 bproc = #(p:0,R_P), #(s:0,P)
                                [ choice | rep x!().choice ];
let Pmod :              bproc = #(p:0,R_Pmod), #(s:0,Pmod)
                                [ choice | rep x!().choice ];
let E1 :          bproc = #(e1:0,E1)                      [ nil ];
let E2 :          bproc = #(e2:0,E2)                      [ nil ];
```

Please note that in this example two different binders are specified for *P* and P_{mod}. Extension of this module that does not modify the current code of the reactants might be carried out by simply introducing additional species into the model and by implementing the additional rules and the novel internal properties specifying the new reactions. However, one should be careful with the extension of this module as change in the described boxes may lead to an error execution of the conditional processes. As we modify the boxes, the conditions have to be updated as well. In this simple case boxes are identified by their internal behavior being equal (in case of the *P* forms) realized with the help of the *choice* operator. Thus, the rules provide a structure where only binder types should be changed in order to transform one box to another. When we would like to extend the module by composing into another system, similar structure of conditions and choice operators have to be implemented. However, it is also automatically doable. To reduce the size of the model, the presented template is appropriate and compositionality can be carried out with it. The enzyme-substrate complexes are automatically emerging properties of the system defined by the rules. We do not need to specify initial conditions for those species. It is important to observe that compositionality of BlenX is improved by the implementation of predefined modules as rule-driven policies have to be fixed from the beginning and templates are crutches for modeling. These solutions can be used in several additional situations.

5.4.5 A compact solution with conditions

The same module can be realized by using only one binder of *P* and of P_{mod}. It provides a shorter code and faster simulations than the previous case. The logic of the module is the same; conditions are defined for describing the possible states of the boxes. Reactants have only one binder specifying their activity and this very primitive interface will changes due to complex formation and communications through that single binder. Further compositionality may be more difficult as the box code its activity and binding capacity in the same type of one binder which can be much more complicated in real biological systems. It reduces our choice of modifying either of the functions of the molecule. Changes in the affinity of binding will automatically modify the previous condition, creating unambiguous code for the system.

```
let proc_check1 : pproc = if ((not (p,bound)) and (p,Pmod)) then x?().nil endif;
let proc_check2 : pproc = if ((not (p,bound)) and (p,P)) then x?().nil endif;

let choice :
pproc = if ((p,bound) and (p,P)) then ch(rate(e1_3),p,Pmod).proc_check1 endif + if
((p,bound) and (p,Pmod)) then ch(rate(e2_3),p,P).proc_check2 endif;

let Pmod :    bproc = #(p:0,Pmod)       [ choice | rep x!().choice ];
let P :       bproc = #(p:0,P)          [ choice | rep x!().choice ];
let E1 :      bproc = #(e1:0,E1)        [ nil ];
let E2 :      bproc = #(e2:0,E2)        [ nil ];
```

5.4.6 Different binders implemented for different functions

In the sequential, two binders are signed to two different functions: one is responsible for the activity of the protein *P* and the other is for the activity of the protein P_{mod}. Hiding and showing the binders define the actual identity of the box if it is *P* or P_{mod}.

```
let proc_Pmod_P :
pproc = if (not (p1,bound)) then ch(p1,Pmod_ACTIVE).hide(p1).x!().nil endif;

let proc_P_Pmod:
pproc = if (not (p2,bound)) then ch(p2,P_ACTIVE).hide(p2).x!().nil endif;

let choice :
pproc = if (p2,bound) then unhide(rate(e1_3),p1).ch(p2,NOT).proc_P_Pmod endif + if
(p1,bound) then unhide(rate(e2_3),p2).ch(p1,NOT).proc_Pmod_P endif;

let Pmod : bproc = #(p1,Pmod_ACTIVE), #h(p2,P_ACTIVE)
                   [ choice | rep x?().choice ];
let P :    bproc = #h(p1,Pmod_ACTIVE), #(p2,P_ACTIVE)
                   [ choice | rep x?().choice ];
let E1 : bproc = #(e1:0,E1)      [ nil ];
let E2 : bproc = #(e2:0,E2)        [ nil ];

{ Pmod_ACTIVE, P_ACTIVE, E1, E2, NOT } %%
{   (Pmod_ACTIVE,E2, rate(e2_1), rate(e2_2), 0),      (NOT,E2, 0, inf, 0),
    (P_ACTIVE,E1, rate(e1_1), rate(e1_2), 0),  (NOT,E1, 0, inf, 0)    }
```

We can assign additional binders for further functions of the molecules. We have to pay attention to the symmetry of the proteins, meaning that if we change the properties of one box, we should change the other one to ensure the program denote for the appropriate box after an action fired. We can also define the possible states of the molecule referring to different interfaces and internal behaviors and transfer these temporary boxes as we wish. However, it leads to a larger program with slower simulations.

5.4.7 Realization of the Goldbeter-Koshland module through communications

A communication process triggers the change of the p binder's type, resulting in the conversion of the molecule from P_{mod} to P,

```
let proc_Pmod_P : pproc = p?().ch(p,P).p_rep!();
```

Communication process that triggers change of the binder's type (P → Pmod)

```
let proc_P_Pmod_P : pproc = p?().ch(p,Pmod).proc_Pmod_P;
```

The molecules are represented as

```
let Pmod :bproc = #(p:0,Pmod)  [ rep p_rep?().proc_P_Pmod_P  |  proc_Pmod_P ];
let P :      bproc = #(p:0,P)        [ rep p_rep?().proc_P_Pmod_P  |  proc_P_Pmod_P ];
let E1 :     bproc = #(e1:0,E1)     [ rep e1!().nil ];
let E2 :     bproc = #(e2:0,E2)     [ rep e2!().nil ];

{Pmod, P, E1, E2}%%
{   (P, E1, rate(e1_1), rate(e1_2), rate(e1_3)),   (Pmod, E1, 0, inf, 0),
```

(Pmod, E2, rate(e2_1), rate(e2_2), rate(e2_3)), (P, E2, 0, inf, 0) }

This solution is composable in parallel to another system automatically if we ensure the symmetry that was mentioned previously. The problem of using only one binder raises the same question of differences between binding sites and active sites. Our choice will depend on the level of abstraction we would like to apply. During the model building process, with the availability of different representation of a simple module, we can realize the case that fits most our idea or simply substitute with a different solution if further extension of the model requires it.

5.4.8 Comparison of the different solutions

In order to choose between the possible interpretations of the same module, comparison is carried out in this session.

The different size of the modules results in different time for the simulations. To test the speed of the solutions of the program with our current simulator (Beta Workbench Simulator), we measured the time of each run. This application passes three text files (input) to the compiler; and translates these files into a runtime representation; and simulates the algorithm through a stochastic simulator engine.

After measuring the time with a timer for 25 run, we result that the usage of simple events is the fastest (in Chapter 5.4.3). Every solution provides the same dynamics for the system, but with different abstractions and different time for simulation. Realization of the template with events offers an easy way of compositionality in case we are interested in introducing novel links through the complexes explicitly encoded in the algorithm. However, the initial conditions are required to be calculated for each simulation, thus we wil have to compare the required simulation and calculation time after implementation of the template into the tool.

Paragraph number	*5.4.3*	*5.4.4*	*5.4.5*	*5.4.6*	*5.4.7*
minimum time [s]	0.468	3.65	2.73	4.212	4.773
maximum time [s]	0.717	4.461	3.88	4.929	6.271
average User Time [s]	0.58176	3.87156	3.09244	4.47936	5.41216

Table 11: The average simulation time of the previously presented BlenX templates. Initial values are *Pmod*=0#; *E1*=11976#; *E2*=1198#; *P*=2994#. Times are shown in the unit of seconds. 25 runs have been measured in each case.

5.5 The positive feedback loop

The previous example presents a scenario when a small module can create an abrupt switch between two states: (1) low amount of *P* and high amount of P_{mod} or (2) lower amount of P_{mod} and higher amount of *P*. In this case, the system responds fast to changes by switching between these two states, although this module does not have a "memory". Not like hysteresis which is a phenomenon that has history-dependence and the transitions between the two steady states are discontinuous. The system's activation threshold differs from the deactivation threshold (Figure 25). In other words, the effects of the input to the system are experienced with a certain delay in time. Time delay is a sort of memory as protein synthesis rate at the present time depends on protein concentration over some time in the past. Hysteresis is a well-described property in thermodynamics and it is also observed in biological networks.

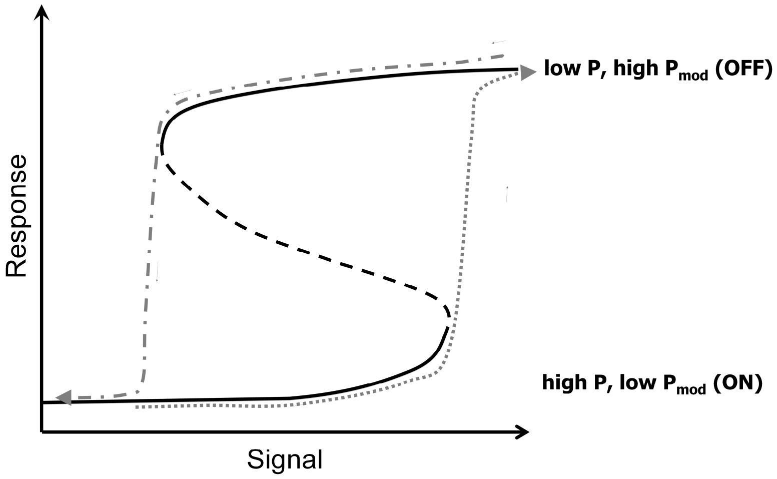


Figure 25: Hysteresis curve.

Bistability in biological systems has been defined as the property when the same system can be in either of two alternative stable steady states under identical chemical conditions [134,268,269]. It is an important biological network structure that was first found in the lactose utilization pathway of bacteria [135] and recently it was also synthetically engineered in a gene regulatory system [270]. Positive feedback and bistability are common themes in theories of cellular memory [57], differentiation [271] and in the study of programmed cell death (apoptosis) [272]. The importance of hysteretic - and bistable - systems in biology refers to switch behavior in a dynamical system creating irreversible changes of different states (e.g. cell cycle states [273]). It was proposed by Novak and Tyson that bistable switches are created by positive feedback loops involving two activation or two inhibitor steps within the cell cycle regulatory network as well [274]. Their prediction that the irreversibility of cell cycle transitions is based on a traverse around a hysteresis loop was also confirmed experimentally by two different groups [275,276]. The two main type of positive feedback loops are shown in the following.

5.5.1 Positive feedback generated by mutual antagonism

By coupling two Goldbeter-Koshland switches, we obtain another interesting behavior observed in biological systems, called mutual antagonism. When two components inhibit each other, they result in a so called “double-negative” - thus a positive - feedback loop. One example for such case may be the interaction between two enzymes reducing their activity through phosphorylation events. These mutual inhibitors provide a switch response of the system that can toggle between two states.

In our example we refer to two opposing proteins as *X* and *Y*. They are able to bind together then they modify and inactivate each other (Figure 26). In protein interaction networks this enzymatic reaction often ends with different phosphorylation states (here we refer to *XP* of *X* and *YP* of *Y*). We adopted an example from [168] where the inactive *X* is degraded, while the *YP* can be dephosphorylated (activated) through an additional enzyme (*A*).

One classical example for such systems is found in the research of the regulation of cell division cycle. Toggle switches are extremely important features because cells make decisions between two sequential phases. When cells commit to separate their previously replicated hereditary material (in mitosis), the activity of a key regulatory kinase complex (MPF also referred to as CDK/Cyclin B) is required to be "switched on". The transition occurs only at a proper time because before that point the regulatory kinase complex (CDK/Cyclin B) is inactivated through its phosphorylation by the protein kinase Wee1. In addition to this negative effect, CDK/Cyclin B also inhibits Wee1. Thus, if the activity of CDK/Cyclin B is increased, it is able to overtake the negative effect of Wee1 and can switch on itself indirectly. This double-negative relation results in a positive feedback on Cyclin B, creating a bistable toggle switch property.

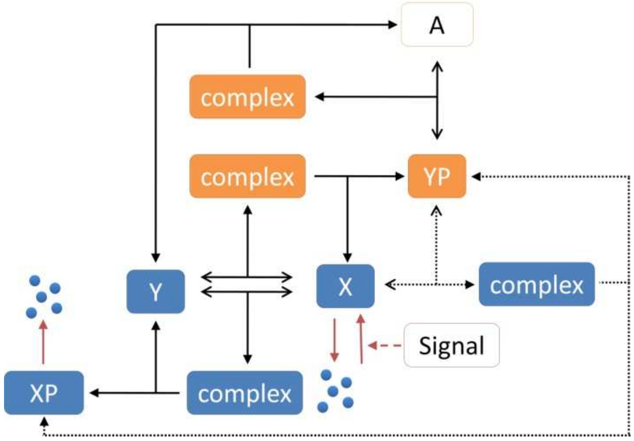


Figure 26: Positive feedback generated by mutual antagonism. X and Y are two inhibitors modifying each other through enzymatic reactions.

Ciliberto et al [132] showed that in protein interaction networks, the assumptions for the two Goldbeter-Koshland modules tend to be invalid when they are coupled into a larger system. The improper application of such a complex reaction scheme without valid assumptions may lead to the loss of bistability if we unpack the Goldbeter-Koshland functions into elementary reactions. Bistability is restored by allowing one inactive form to possess some activity and phosphorylate its substrate (see below). In the subsequent section we present the BlenX template encoding a mutual inhibition mechanism in a formal way.

5.5.1.1 The BlenX template encoding mutual antagonism

The motif of a double-negative feedback regulation is realized in BlenX through several activation and inactivation steps that are fired depending on the state of the boxes. As process calculi tools provide the opportunity to define general conditions that might be used for molecules with similar behavior, model-construction is easy and the size of the BlenX model stays relatively small.

First, we define the possible interactive sites of the molecules. As the module contains proteins with dual roles performing as enzymes or substrates, they all have two binders representing the activity (ENZ meaning enzyme activity) and the binding site (SUB referring to as substrate). All the SUB and ENZ sites can exist in two forms: active (available) or inactive (hidden). The conditions representing activation and inactivation steps are shown in the following paragraph. We would like to ensure that when the site which is waiting for modification is bound to its enzyme, the box cannot possess other (enzymatic) activity. Thus, the interface of the box is changed depending on its condition and the enzymatic site is hidden from any reaction:

```
let inactivate_ENZ : pproc = ( if (substrate, bound) then hide(enzyme).nil endif );
let inactivate_SUB : pproc = ( if (enzyme, bound) then hide(substrate).nil endif );
```

Furthermore, we would like to ensure that when the substrate site of the box dissociates from the enzyme, the catalytic activity of the box is restored through the reactivation of the enzymatic site:

```
let activate_ENZ :
  pproc = (if (not(substrate, bound)) then unhide(enzyme).rec_enzyme!().nil endif );
let activate_SUB :
  pproc = (if (not(enzyme, bound)) then unhide(substrate).rec_substrate!().nil endif );
```

These are all general properties (conditions) of biological systems and they can be easily composed into a larger model in BlenX. They ensure the exclusion of the presence of multimers that may not be realistic in a biological system where the binding-induced conformational change of the enzyme can modify its regulatory sites. These rules also reduce the complexity of the model system and the number of possible but unnecessary states.

To retain the bistability in the module, we assume that the inactive Y (*YP*) enzyme is also able to change the behavior of the molecule *X*; however, its activity is much lower than the active *Y* [132]. We included a third binding site for *X* that participates in the reaction triggered by *YP*. Activation of the binder responsible for enzymatic activity (*enzyme*) occurs when the box exists as a single molecule and its binders representing the substrate affinity (*substrate* and *substrate_YPX*) are not occupied. The conditions describing the activation and inactivation of the molecule *X* are:

```
let activate_ENZ_X :
  pproc = (if (not(substrate, bound) and not(substrate_YPX, bound)) then unhide(enzyme).rec_ENZ!().nil endif);

let activate_SUB_X :
  pproc = (if (not(enzyme, bound) and not(substrate_YPX, bound)) then unhide(substrate).rec_SUB!().nil endif);

let activate_SUB_X_YP :
  pproc = (if (not(enzyme, bound) and not(substrate, bound)) then unhide(substrate_YPX).rec_ENZ_X!().nil endif);

let inactivate_ENZ_X :
  pproc = ( if ((substrate, bound) or (substrate_YPX, bound)) then hide(enzyme).nil endif);

let inactivate_SUB_X :
  pproc = ( if ((enzyme, bound) or (substrate_YPX, bound)) then hide(substrate).nil endif);

let inactivate_SUB_X_YP :
  pproc = ( if ((enzyme, bound) or (substrate, bound)) then hide(substrate_YPX).nil endif );
```

The variables are characterized by their binding sites - one representing their enzymatic activity and one waiting for communications - and their internal course of actions executed as parallel processes. One possible sequence of step is shown on Figure 27. In case of the box *X*, an additional communication site has been defined in order to restore bistability:

```
let X :     bproc = #(substrate:0,X_sub), #(enzyme:0,X_enz), #(substrate_YPX:0,X)
      [     inactivate_ENZ_X     | activate_ENZ_X |
            rep rec_ENZ?().(inactivate_ENZ_X | activate_ENZ_X ) |
            inactivate_SUB_X | activate_SUB_X |
            rep rec_SUB?().(inactivate_SUB_X | activate_SUB_X ) |
            activate_SUB_X_YP | activate_SUB_X_YP |
            rep rec_ENZ_X?().(inactivate_SUB_X_YP | activate_SUB_X_YP )|
            rep enzyme?().nil |
            substrate!().ch(substrate, YX).nil |
            substrate_YPX!().ch(substrate_YPX, YPX).nil                 ];
```

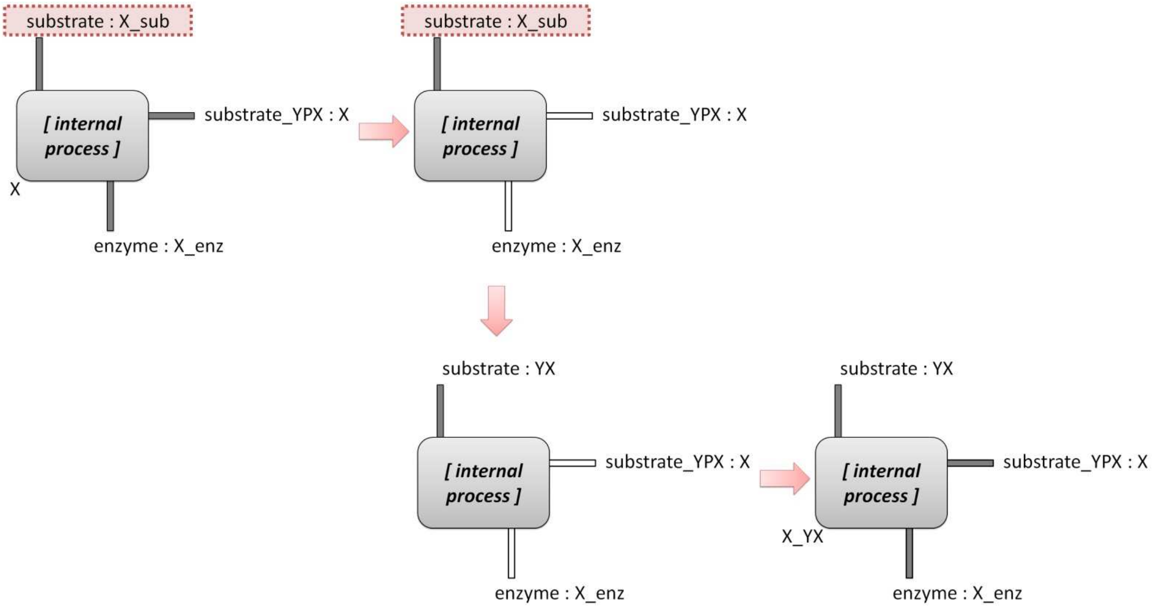


Figure 27: A possible sequence of reactions of the box *X*. When the binder with the type *X_sub* (red highlight) creates a link with a molecule, the two other binders of *X* become unavailable (hidden, white bars). A change of the binder's type occurs and dissociation of the regulatory molecule from the modified binder fires the condition of unhiding the binders.

The other species in the module are encoded with similar logic, but with only two binders:

```
let Y :     bproc = #(substrate:0,Y_sub), #(enzyme:0,Y_enz)
  [         inactivate_ENZ | activate_ENZ |
            rep rec_enzyme?().(inactivate_ENZ | activate_ENZ ) |
            inactivate_SUB | activate_SUB |
            rep rec_substrate?().(inactivate_SUB | activate_SUB ) |
            rep enzyme?().nil |
            substrate!().ch(substrate, XY).nil                        ];

let XP :  bproc = #(substrate:0,XP_sub), #(enzyme:0,XP_enz)
  [       inactivate_ENZ | activate_ENZ |
            rep rec_enzyme?().(inactivate_ENZ | activate_ENZ ) |
            inactivate_SUB | activate_SUB |
            rep rec_substrate?().(inactivate_SUB | activate_SUB ) |
            rep enzyme?().nil |
            substrate!().ch(substrate, XP).nil                ];

let YP :  bproc = #(substrate:0,YP_sub), #(enzyme:0,YP_enz)
  [       inactivate_ENZ | activate_ENZ |
            rep rec_enzyme?().(inactivate_ENZ | activate_ENZ ) |
            inactivate_SUB | activate_SUB |
            rep rec_substrate?().(inactivate_SUB | activate_SUB ) |
            rep enzyme?().nil |
            substrate!().ch(substrate, AYP).nil        ];
```

Intermediate complexes are also present in the model. Their definition is necessary to assign further functions to the modified forms of the molecules within a larger system. We show an example of a temporary box which originates from *Y* and which immediately turns to the - the modified form of *Y* (*YP*) that is inactive and degraded in this example (the reader finds the entire template with additional boxes in Appendix A):

```
let Y_XY :        bproc = #(substrate:0,XY), #(enzyme:0,Y_enz)
  [       inactivate_ENZ | activate_ENZ |
          rep rec_enzyme?().(inactivate_ENZ | activate_ENZ ) |
          inactivate_SUB | activate_SUB |
          rep rec_substrate?().(inactivate_SUB | activate_SUB ) |
          rep enzyme?().nil          ];
```

The box *S* (signal) modifies *E2*, while *A* is an unspecified enzyme reactivating the inactive *Y* (*YP*):

```
let S : bproc = #(s:0,S)    [ rep s?().nil ];
let A : bproc = #(a:0,A)   [ rep a?().nil ];
```

The catalytic steps of each reaction are defined through BlenX events such as:

```
when(Y_XY::inf) split(YP,Nil);
```

Synthesis and degradation of the *X* molecule are included for the further analysis of the model:

```
when(S::rate(k1)) split(S,X);
when(X::rate(k2p)) delete(1);
```

Types represent the interaction capabilities of the boxes and are listed in Appendix A.

Note that the way of composition can be automatic as extra binders may be added and the list of conditions can be extended as well. Each molecule that is able to perform as an enzyme or as a substrate can be described the same way, making composition of these elements easy. Listing the properties of the components makes compositionality a feature of process calculus based programming languages designed for biology and the general description of the enzymes acting similarly reduce the size of a larger model during extension.

The general description of an enzyme performing also as a substrate is supported with the following conditions:

```
let inactivate_ENZ :
  pproc = ( if (substrate, bound) then hide(enzyme).nil endif );
let inactivate_SUB :
  pproc = ( if (enzyme, bound) then hide(substrate).nil endif );
let activate_ENZ :
  pproc = (if (not(substrate, bound)) then unhide(enzyme).rec_enzyme!().nil endif );
let activate_SUB :
  pproc = (if (not(enzyme, bound)) then unhide(substrate).rec_substrate!().nil endif );
```

Binders can be added easily and processes can be composed parallel. We see that this property of the pi-calculus processes enable the representation of molecules with several independent functional domains. If the molecule *X* contains *n* independent binding or phosphorylation sites, each state of the molecule reflects the set of activities in which the site can participate. In contrast, with the ODE approach the modeler would be required to write kinetic equations for all modification states of X which grow exponentially with the number of independent sites. While the number of reactions needed to describe *X* in BlenX grows linearly with the number of sites. Furthermore, the BlenX framework as a member of process algebra initiatives, allows the formation of heteromers resulting in a larger and more complex picture of the system that would be realized in a difficult way with ODEs. If needed, with the help of conditions we can exclude the occurrence of certain bindings, thus we can limit our network depending on our modeling goal. Realization of the templates and the code of general rules assigned to common behavior of biological systems easen the modeling process and provide a more systematic way of building biological networks in silico.

In the subsequent we show the analysis of this module and how it behaves in a stochastic environment.

5.5.1.2 Computational simulations for the mutual antagonism module

Bifurcation analysis is one of the leading techniques for analyzing classical modeling tools, ODEs. The system’s properties can be presented in an elegant way by showing how it responds to the increase or decrease of a key signal. In this particular antagonistic module a bistable region is present where the same system is able to perform in two distinguishable states depending on the initial conditions. In a discrete model the global properties should not vary, thus after running stochastic simulations with the same parameter set but with different initial conditions, we have to be able to see a bistable region if nonlinearity remained in our BlenX module. The parameters used in this example have been adapted from [168].

In more details, to plot a bifurcation diagram, we changed the synthesis rate of E_2 by changing the value of the signal (*S*) and then we measured the time-average of *P* and P_{mod} after they had reached the steady state during the simulation. We can clearly see the region of bistability on Figure 28, in accordance with the experimental measurements (Figure 29).

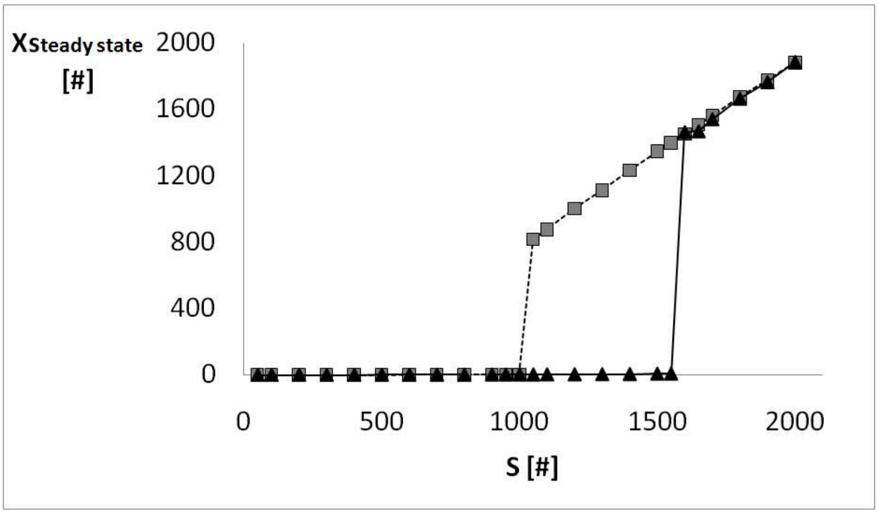


Figure 28: Two distinguishable stable steady states are seen. Bistability characterizes the system between the *S* values of 1000# and 1600#. Steady state values are calculated as the time average of the molecule after reaching steady state. α=0.00167.

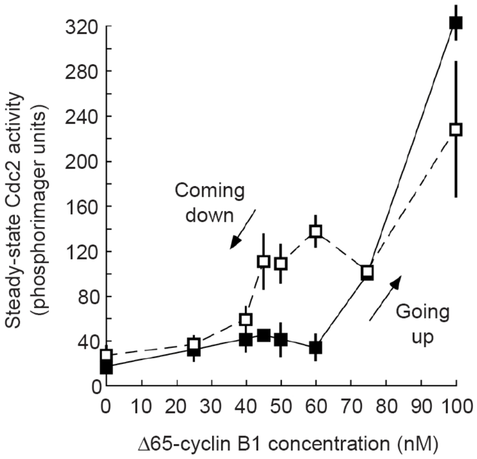


Figure 29: Experimental measurements of existence of bistability in cell cycle regulation by Pomerening et al. [276].

One question we might ask is that when we implement a mutual inhibition module into a stochastic model, how noise would affect our simulation results. In order to investigate the influence of stochasticity in our BlenX module, we reduced the number of molecules playing role in this isolated system (by reducing the value of the α conversion factor). When it includes only a small number of enzymes, the large internal noise arising from the stochastic simulations is able to disrupt bistability (Figure 30). We cannot see the hysteretic property on the diagram of Figure 30.

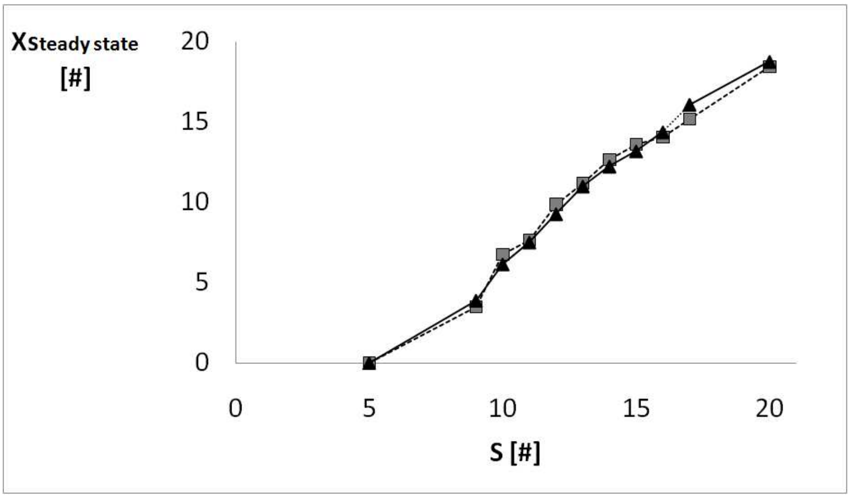


Figure 30: Steady state values of X over S with large noise. Initial conditions are *X*=6#, *Y*=6#, *YP*=0#, *A*=6# for black triangular; and *X*=0#, *Y*=0#, *YP*=6#, *A*=6# for grey squares. Steady state values are caluclated from time averages of the simulations. α=0.167.

It has been shown by [168] that the unpacked version of the module containing only elementary steps shorten the range of the bistable regime. Steuer et al. [167] showed how noise is able to trigger oscillations in a cell cycle model. In our example we observed that shortening the range of the multistate regime by unpacking the complex reaction terms may lead to the fusion of the bifurcation points in stochastic simulations. Vanishing multistability is a crucial consequence of highly stochastic systems, thus modelers should pay attention to it. Even if the deterministic module is able to behave as an irreversible switch due to its hysteretic property, noise may influence this behavior leading to different results from the deterministic module including several complex rate functions. Stochasticity may trigger the system to bounce from one to the other state and back if the size of the bistable region is not large enough. On Figure 31 we see that within the bistable region noise influence the properties of the system. The increase of noise leads to disappearance of the hysteretic behavior and decrease the robustness of the switch. In general cells tend to overcome this phenomenon in real systems by increasing the number of molecules present in this regulatory module.

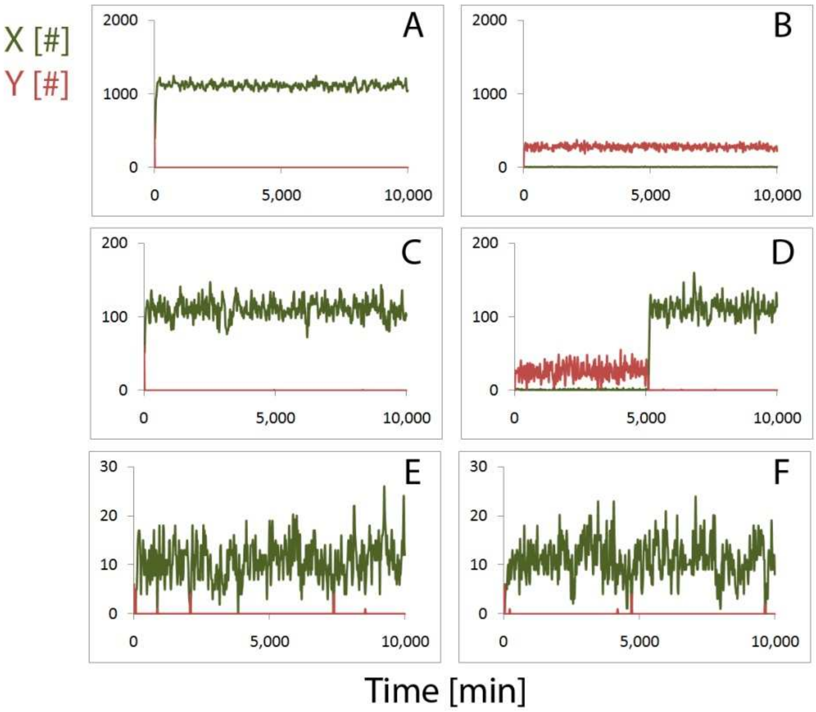


Figure 31: Time evolution of the bistable system depends on the amount of noise. α is varied in order to modify the number of molecules present in the module. (A-B) α=0.167, (C-D) α=0.0167, (E-F) α=0.00167. (A) initial *X* and *Y* equal 600#, (C) *X* and *Y* equal 60#, (E) *X* and *Y* equal 6#, (B,D,F) *X* and *Y* equal 0#. The amount of signal (*S*) equals to 13#, 130# and 1300# respectively.

Besides the positive feedback regulation emerging from a double inhibitory relation cells also perform mutual active regulation (Figure 32). In the following section we show a BlenX realization of this type of positive feedback.

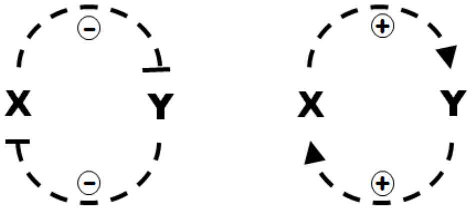


Figure 32: Positive feedback regulations. The diagram on the left hand side represents a mutual antagonism (or double-negative feedback loop) and the other on the right represents mutual activation.

5.5.2 Positive feedback loop generated by mutual activation mechanisms

Positive feedback regulation may occur when two enzymes mutually inactivate or activate each other. The realization of the mutual activation module can be composed of two Goldbeter-Koshland functions as in the previous example. This regulatory motif is important when both enzymes are inactive at the initial step. In cell cycle we can think about the example when CDK/Cyclin B (named and referred to as CYCB in the subsequent) and its phosphatase (CDC25P), are both inactive before mitosis. During G2-phase CYCB is inhibited by a kinase (WEE1), while CDC25 is inactive due to another enzyme. Cells in order to enter mitosis need to increase the amount of active CYCB (CYCB form). Slight increase in the CYCB level triggers activation of the CDC25 phosphatase which in response removes the inhibitory phosphate group from CYCBP. The more increased amount of active CYCB form increases the active CDC25P creating an abrupt change in the level of both proteins (Figure 33). This double activation switch cells into mitosis due to the sudden increase in CYCB activity.

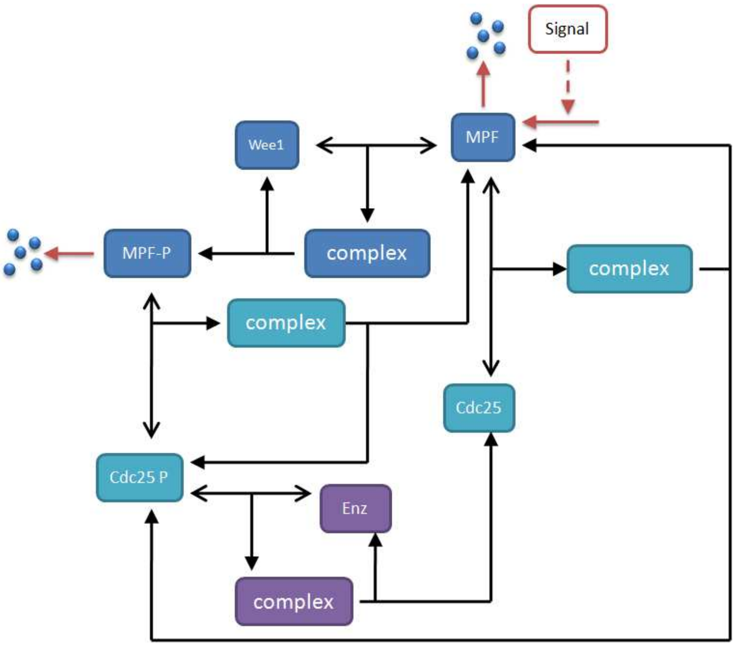


Figure 33: Wiring diagram of a mutual activation mechanism.

Enzymatic reactions are described in three steps, based on the enzyme-substrate reversible complex formation and a catalytic step. Each interacting enzyme-substrate pair performs a Goldbeter-Koshland module. The rules guiding the model are similar to the ones used in the mutual antagonism system. Conditions are activating and inactivating the binders of the boxes.

```
let activate_ENZ :
        pproc = (if (not(substrate, bound)) then unhide(enzyme).rec_ENZ!().nil endif );
let activate_SUB :
        pproc = (if (not(enzyme, bound)) then unhide(substrate).rec_SUB!().nil endif );
let inactivate_ENZ :
        pproc = ( if (substrate, bound) then hide(enzyme).nil endif );
let inactivate_SUB :
pproc = ( if (enzyme, bound) then hide(substrate).nil endif );
```

CYCB acts as a substrate of the inhibitory kinase (WEE1) and also as an enzyme activating its partner, CDC25P:

```
let CYCB : bproc =      #(substrate:0,CYCB_sub),
                        #(enzyme:0,CYCB_enz_CDC25)
   [      inactivate_ENZ | activate_ENZ |
          rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
          inactivate_SUB | activate_SUB |
          rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
          rep enzyme?().nil |
          substrate!().ch(substrate, CYCB_sub_mod).nil      ];
```

In this module WEE1 kinase has no affinity to other reactions than inhibiting CYCB.

```
let WEE1 : bproc = #(enzyme:0,WEE1_enz)
   [       rep enzyme?().nil ];
```

CDC25 has an inactive form assigned as CDC25 and an active form called CDC25P. The active CDC25 can function both as an enzyme and a substrate. The inactive form (CDC25) has no enzymatic activity.

```
let CDC25 :       bproc = #(substrate:0,CDC25_sub)
   [       substrate!().ch(substrate, CDC25_sub_mod).nil     ];
let CDC25P : bproc = #(enzyme:0,CDC25P_enz), #(substrate:0,CDC25P_sub)
   [      inactivate_ENZ | activate_ENZ |
          rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
          inactivate_SUB | activate_SUB |
          rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
          rep enzyme?().nil |
          substrate!().ch(substrate, CDC25P_sub_mod).nil                    ];
```

An undefined enzyme keeps CDC25 inactive.

```
let ENZYME1 : bproc = #(enzyme:0,ENZYME1_enz)
  [ rep enzyme?().nil ];
```

A signal is introduced that increases the amount of the enzyme CYCB to trigger the switch after CYCB reaches an activity level that can activate CDC25:

```
let SIGNAL :    bproc = #(s:0,S)      [ nil ];
when(SIGNAL::rate(s)) split(SIGNAL,CYCB);
```

Degradation of CYCB and CYCBP equilibriates their synthesis:

```
when(CYCB::rate(deg)) delete(1);
```

Temporal species are defined to assign additional functions for the modified boxes and their binders (Appendix A). The molecular links are defined in the binder definition file (Appendix A). Binders are associated to each other to allow enzymatic reactions to happen.

parameters and units	*parameter values*	*parameters and units*	*parameter values*
$k1\ [1/(min \cdot \#)]$	$1 \cdot \alpha$	$m1\ [1/(min \cdot \#)]$	$0.05 \cdot \alpha$
$k1r\ [1/min]$	0.1	$m1r\ [1/min]$	0.01
$k2\ [1/min]$	0.5	$m2\ [1/min]$	0.5
$l1\ [1/(min \cdot \#)]$	$0.01 \cdot \alpha$	$n1\ [1/(min \cdot \#)]$	$0.1 \cdot \alpha$
$l1r\ [1/min]$	0.01	$n1r\ [1/min]$	0.05
$l2\ [1/min]$	0.05	$n2\ [1/min]$	0.05
$s\ [1/min]$	0.1		
$deg\ [1/min]$	0.01		

Table 12: Reaction rate constant values used in simulation on Figure 34. Initial values are ENZYME1=100#, CDC25=200#, CYCBP=500#, WEE1=100#, SIGNAL=50#, CYCB=CDC25P=0#.

Simulation result of the module shows the dynamic of the system (Figure 34). When CYCB level increases CDC25 turns into an active form. This change amplifies the effect of CYCB as CDC25P is able to restore its activity.

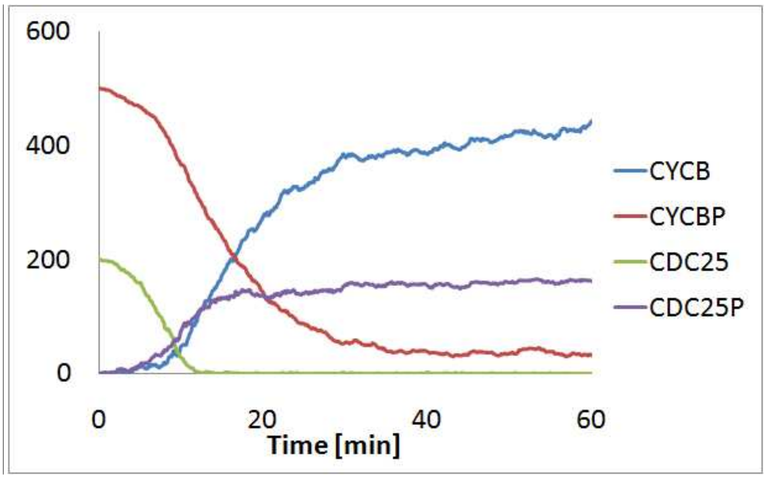


Figure 34: Simulation of the mutual activation module. The increase of CYCB switch CDC25 (inactive from) into CDC25P (active form). Active CDC25P activates CYCBP resulting in CYCB. α=0.0167.

This module can be used in various scenarios when amplification of a signal is realized through enzymatic reactions with mutual activation relation among the elements of the module. In biology it plays major role in developmental processes [277] or in apoptosis [272,278].

In the next session, we show a compositional case study for modeling the G2/M transition of the cell cycle. The example contains interconnected mutual antagonism and mutual activation.

5.6 An example: G2/M transition during cell cycle

Alternation of the cell cycle phases is tightly controlled through the interaction of a large set of regulatory proteins. Activity of the cyclin-dependent kinases (CDKs) is responsible for most basic cell cycle processes and its role is crucial in cell cycle progression [279]. One method for change the activity of CDK is to modify its phosphorylation state. The G2/M transition is one of the key irreversible switches in cell division cycle triggered by active CDK in combination with a cycling subunit (Cyclin B). During the G2-phase of the cell cycle, CDK activity is kept low with inhibitory phosphorylation events that are carried out by the Wee1 kinase. As the activity of CDK/Cyclin B increases in time, it activates its helper phosphatase Cdc25 which in return removes the inhibitory phosphate groups from CDK. The sudden increase of CDK activity triggers inactivation of Wee1 kinase by its phosphorylation. The active CDK generates mitotic events (e.g. chromosome condensation, nuclear envelope breakdown, chromatid segregation, assembly of mitotic spindle).

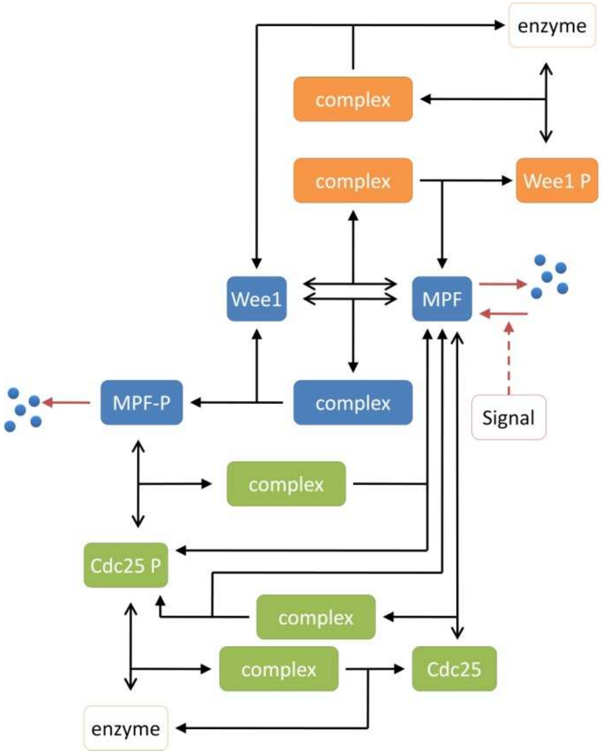


Figure 35: Wiring diagram of the G2/M transition module

The sequence of these events is realized through several feedback regulations. The antagonistic relation of CDK/Cyclin B (CYCB) and the Wee1 (WEE1) kinase can be modeled as a double-negative feedback relation. The mutual activation of the CDK/Cyclin B and the Cdc25 (CDC25), together with the positive feedback loop described ahead act synergistically and provide an abrupt increase in the activity of CDK making the G2/M transition an irreversible, switch-like event. Let us show how the previously presented template can be extended in order to model the cell cycle transition from the G2-phase into the M-phase.

5.6.1 Composition of the G2/M transition with BlenX

Each elementary step of the module might be coded in the BlenX framework starting from basic primitives. We go further and show a more elegant way of using BlenX templates for modeling a complex motif including a mutual inhibition module extended with a positive feedback regulation. It is an example for a higher level composition of the predefined templates.

The mutual antagonistic relation between CDK/Cyclin B (referred to as CYCB) and Wee1 is modeled as described in Chapter 5.5.1.1 and we extended it with a mutual activation module of the interaction of CYCB and CDC25 (Chapter 5.5.2).

The general conditions that regulate the activity of certain binders are defined the same way as in the mutual inhibitor and mutual activation modules:

```
let activate_ENZ :
        pproc = (if (not(substrate, bound)) then unhide(enzyme).rec_ENZ!().nil endif );
let activate_SUB :
        pproc = (if (not(enzyme, bound)) then unhide(substrate).rec_SUB!().nil endif );
let inactivate_ENZ :
        pproc = ( if (substrate, bound) then hide(enzyme).nil endif );
let inactivate_SUB :
        pproc = ( if (enzyme, bound) then hide(substrate).nil endif );
```

In this model there are three binders featured with different functions of the CYCB molecule. One of them represents the bindig site of the CYCB (for its inhibitor kinase WEE1 or its activator phosphatase CDC25), while the other two binders are responsible for the enzymatic activity of CYCB acting on WEE1 and CDC25 independently. If we compare, we see that the CYCB protein does not have the same substrate role as in the mutual inhibition module, but possesses an enzymatic activity creating a connection into the positive limb of the motif. In the code the processes and conditions related to the previous unnecessary role of the box have been erased and novel rules have been added parallel to the already existing module. Note that in case of this composed module, we do not need to apply the assumptions that we did in Chapter 5.5.1.1. In that module, in order to restore bistability in the mutual antagonism module, the inactive enzyme has a minimal enzymatic activity. In case of this larger module, complexity of the system solves the problem arising from disappearing nonlinearity in elementary steps. Interconnected feedback loops can increase robustness of the system and restore nonlinear dynamics in this example.

In the following, we highlight the parallel composition of the elements of the system. The black color stays for the mutual inhibition, while the blue text shows the extensions emerging from the mutual activation module.

```
let activate_ENZ_CYCB :
        pproc = (if (not(substrate, bound)
        and not(enzyme_CYCBCDC25, bound))
        then unhide(enzyme).rec_ENZ!().nil endif);

let activate_SUB_CYCB :
         pproc = (if (not(enzyme, bound)
        and not(enzyme_ CYCBCDC25, bound))
        then unhide(substrate).rec_SUB!().nil endif);

let activate_ENZ_CYCB_CDC25 :
        pproc = (if (not(enzyme, bound)
        and not(substrate, bound))
        then unhide(enzyme_CYCBCDC25).rec_ENZ_CDC25!().nil endif);

let inactivate_ENZ_CYCB :
        pproc = ( if ((substrate, bound)
        or (enzyme_CYCBCDC25, bound))
        then hide(enzyme).nil endif );

let inactivate_SUB_CYCB :
        pproc = ( if ((enzyme, bound)
        or (enzyme_ CYCBCDC25, bound))
        then hide(substrate).nil endif );

let inactivate_ENZ_CYCB_CDC25 :
        pproc = ( if ((enzyme, bound)
        or (substrate, bound))
        then hide(enzyme_ CYCBCDC25).nil endif );
```

We assign three different functions into CYCB. It acts as an inhibitory enzyme of its ‘enemy’ (the WEE1 kinase); and CYCB acts also as an enzyme activating CDC25 and last, it is a substrate of other enzymatic reactions triggered by CDC25P in this example. Three binders are present for these roles.

```
let CYCB : bproc =      #(enzyme:0,CYCB_enz),
                        #(substrate:0,CYCB_sub),
                        #(enzyme_ CYCBCDC25:0,CYCB_enz_CDC25)
   [      inactivate_ENZ_CYCB | activate_ENZ_CYCB |
          rep rec_ENZ?().(inactivate_ENZ_CYCB | activate_ENZ_CYCB ) |
          inactivate_SUB_CYCB | activate_SUB_CYCB |
          rep rec_SUB?().(inactivate_SUB_CYCB | activate_SUB_CYCB ) |
          inactivate_ENZ_CYCB_CDC25 | activate_ENZ_CYCB_CDC25 |
          rep rec_ENZ_CDC25?().(inactivate_ENZ_CYCB_CDC25 |
          activate_ENZ_CYCB_CDC25 ) |
          rep enzyme?().nil |
          substrate!().ch(substrate, CYCB_sub_mod).nil |
          rep enzyme_ CYCBCDC25?().nil                    ];

let CYCBP : bproc = #(substrate:0,CYCBP_sub)
   [      substrate!().ch(substrate, CYCBP_sub_mod).nil     ];

let WEE1 : bproc =      #(enzyme:0,WEE1_enz),
                        #(substrate:0,WEE1_sub)
   [      inactivate_ENZ | activate_ENZ |
          rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
          inactivate_SUB | activate_SUB |
          rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
          rep enzyme?().nil |
          substrate!().ch(substrate, WEE1_sub_mod).nil            ];
```

```
let WEE1P : bproc =     #(substrate:0,WEE1P_sub)
   [       substrate!().ch(substrate, WEE1P_sub_mod).nil      ];

let CDC25 :      bproc = #(substrate:0,CDC25_sub)
        [   substrate!().ch(substrate, CDC25_sub_mod).nil     ];

let CDC25P : bproc = #(enzyme:0,CDC25P_enz), #(substrate:0,CDC25P_sub)
   [       inactivate_ENZ | activate_ENZ |
           rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
           inactivate_SUB | activate_SUB |
           rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
           rep enzyme?().nil |
           substrate!().ch(substrate, CDC25P_sub_mod).nil                  ];

let ENZYME1 : bproc = #(enzyme:0,ENZYME1_enz)
  [ rep enzyme?().nil ];

let ENZYME2 : bproc = #(enzyme:0,ENZYME2_enz)
  [ rep enzyme?().nil ];
```

Binders are specified as:

```
{  WEE1_enz, WEE1_sub, WEE1_sub_mod,
   CYCB_enz, CYCB_sub, CYCB_sub_mod, CYCB_enz_CDC25,
    CDC25P_enz, CDC25P_sub, CDC25P_sub_mod,
   ENZYME1_enz,
   ENZYME2_enz,
   CDC25_sub, CDC25_sub_mod,
   WEE1P_sub, WEE1P_sub_mod,
   CYCBP_sub, CYCBP_sub_mod      } %%
{  (CYCB_enz_CDC25, CDC25_sub, rate(a1), rate(a1r), rate(a2)),
   (CYCB_enz_CDC25, CDC25_sub_mod, 0, inf , 0 ),
```

(CYCB_enz, WEE1_sub, rate(b1), rate(b1r), rate(b2)),
(CYCB_enz, WEE1_sub_mod, 0, inf , 0),
(ENZYME1_enz, CDC25P_sub, rate(c1), rate(c1r), rate(c2)),
(ENZYME1_enz, CDC25P_sub_mod, 0, inf , 0),
(CDC25P_enz, CDC25P_sub,rate(d1), rate(d1r), rate(d2)),
(CDC25P_enz, CYCBP_sub_mod, 0, inf , 0),
(WEE1_enz, CYCB_sub, rate(e1), rate(e1r),rate(e2)),
(WEE1_enz, CYCB_sub_mod, 0, inf , 0),
(ENZYME2_enz, WEE1P_sub, rate(f1), rate(f1r) ,rate(f2)),
(ENZYME2_enz, WEE1P_sub_mod, 0, inf ,0) *}*

Temporal species are the events assigned to them are defined in Appendix A.

We run stochastic simulations with the parameter set shown in Table 13.

	association rate constants *units:* $[dm^3/(\# \cdot min)]$		dissociation rate constants *units:* $[1/min]$		catalytic step *units:* $[1/min]$	
CycB (enzyme) : Cdc25 (substrate)	*a1*	0.2α	*a1r*	*0.02*	*a2*	*4*
CycB (enzyme) : Wee1 (substrate)	*b1*	$5\ \alpha$	*b1r*	*10.6*	*b2*	*0.4*
E1 (enzyme) : Cdc25P (substrate)	*c1*	$0.1\ \alpha$	*c1r*	*0.1*	*c2*	*20*
Cdc25P (enzyme) : CycB (substrate)	*d1*	$0.0009\ \alpha$	*d1r*	*0.005*	*d2*	*0.085*
Wee1 (enzyme) : CycB (substrate)	*e1*	$0.1\ \alpha$	*e1r*	*0.05*	*e2*	*0.05*
E2 (enzyme) : Wee1P (substrate)	*f1*	$0.1\ \alpha$	*f1r*	*0.01*	*f2*	*2*

Table 13: Parameter set for the simulation of G2/M transition. The values of the rate constants are taken from [132] and converted into stochastic rates through a conversion factor α=0.0167.

We compared the simulation results with the deterministic exact solution of the module. The stochastic template fits well the solution of the deterministic case even with larger noise present in the system (Figure 36). The usage and interconnecting the predefined templates are easy in this way. Parallel composition of processes and extension of binders and rules can be automatized within a software tool.

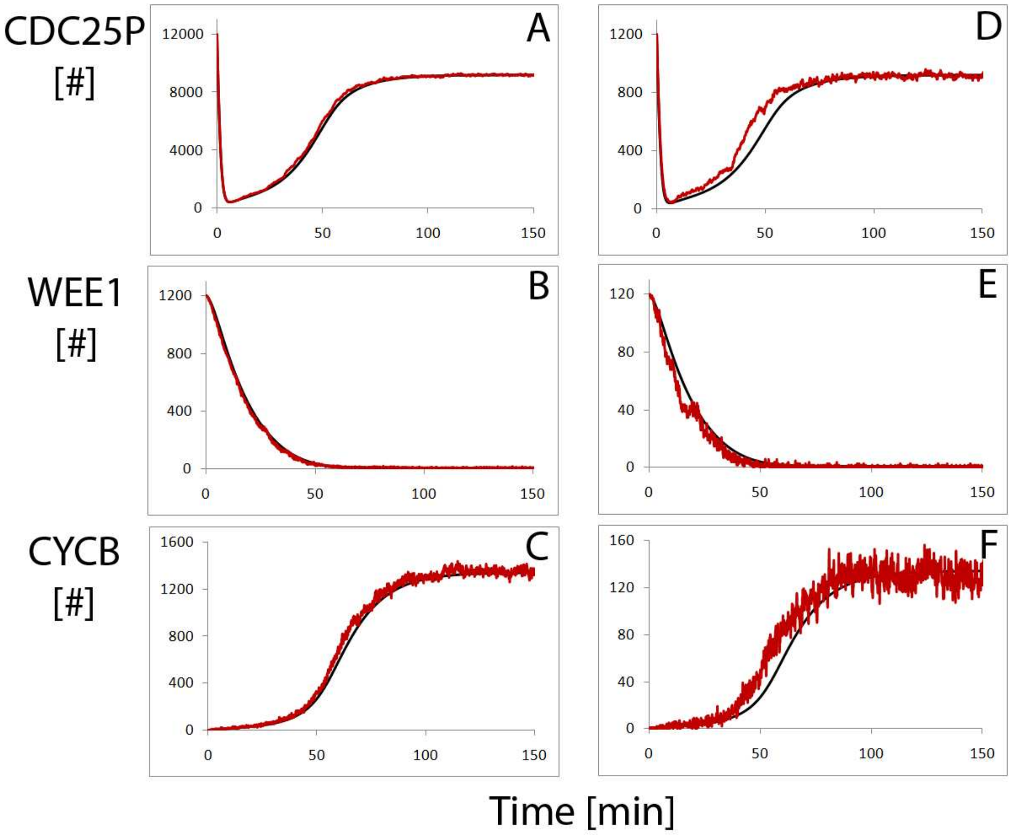


Figure 36: Stochastic simulation of the G2/M module is plotted in red, while the deterministic exact solution is signed by black lines. (A-C) time curves of a model with higher α (α=0.000167) representing less noise. (D-F) time cirves with larger noise (α=0.00167).

5.7 Perspectives

On the line of the work presented here, much more biological motifs could be implemented into the template library in the future. One interesting example might be the composition of negative feedback regulations that provide a more complex representation of biological oscillators. Negative feedback is often used to stabilize dynamic systems as it is able to perform adaptation (such a way as incoherent feed-forward loops) [280,281]. Furthermore, it has been shown having role in noise-reduction in complex systems. We decided to investigate one particular case during this work, when periodic behavior arise due to a negative feedback loop including a time delay or other interconnected feedback regulations.

Automatic use of the templates is possible thanks to the basic structure of the language with rules-driven reactions and parallel composition of the processes. Realization and implementation into a higher-level interface (such as CoSBi Lab [255]) is a matter of software development and out of scope of this work. The library presented here gives an opportunity for realization of a user-friendly, high-level process calculus modeling tool for biology on top of the BlenX language and Gillespie's SSA algorithm.

5.8 A circadian clock study

Circadian clocks [181] ensure the daily rhythms of several biological functions. This endogenous system is based on a negative feedback loop producing a time delayed downregulation of transcriptional events. In addition to the well-studied, but still unclear, features of the molecular clock (such as entrainment, robust oscillations and temperature compensation), recently, a novel molecular link has been discovered by Matsuo and co-workers [225]. They found that circadian rhythms regulate the daily expression of a certain cell cycle component (Wee1 kinase) being instrumental in the irreversible G2/M transition, inhibiting cell's entry into mitosis depending on the time of the day.

A systematic approach to such a biological network may lead us to unexpected discoveries and verification or contradiction of hypotheses as well. In the subsequent, our results achieved by modeling this 24h oscillatory system [47,48,282] will be presented.

5.8.1 Modeling cell division cycles gated by the circadian clock

After the first molecular evidence for the coupling between circadian and cell cycles has been revealed [225], we decided to investigate the importance of this connection with a systematic approach. We adapted Bela Novak and John J Tyson's deterministic cell cycle model [160] that focuses on the regulation of the restriction point of mammalian cells. However, their model lacks the detailed representation of the G2/M transition (see description in

Chapter 5.6), thus we exploited its extension with a mutual antagonistic switch control mechanism.

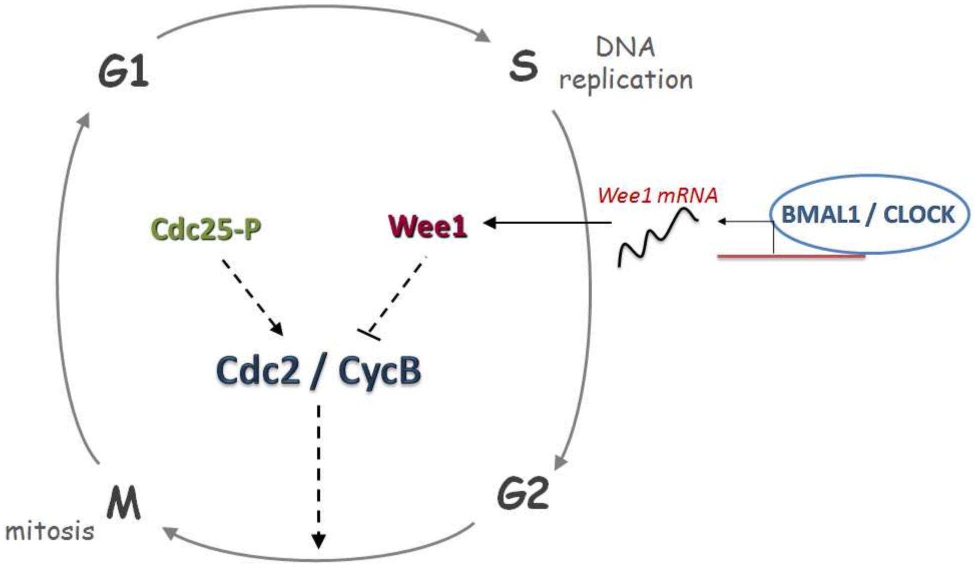


Figure 37: The G2/M transition of the cell cycle is gated by the circadian clock through the time-dependent transcription of the Wee1 gene. Every 24h BMAL1/CLOCK transcription factor dimer triggers the expression of Wee1. Wee1 protein inhibits (dashed |-- line) the signal of the entry into mitosis (the CDK/Cyclin B called Cdc2/CycB in mammals). On the other hand, the Cdc25P phosphatase activates the Cdc2/CycB (dashed arrow). The intertwined two positive feedback loops ensure the precise and irreversible G2/M transition of the cycle.

The first molecular link between these two cyclic processes resides in the direct regulation of the Wee1 kinase by the core clock transcription factors, BMAL1 and CLOCK (also called Clk) in mammals. The importance and the effect of this interconnected oscillatory system is unclear, thus among others [170,226], we also decided to analyze the systems through computational modeling [47].

During our work, the modeling purpose was not to address a comprehensive mammalian circadian rhythm model. For the sake of simplicity, we had a minimal oscillator that generates an endogenous cycle enforcing a periodic influence on the cell cycle. Hence, we built a simplified version of a 4-variable mammalian circadian clock model that consists of transcription factors (BMAL1 and CLOCK), clock messages (mPer or mCry mRNA), clock proteins (PER or CRY), and a dimer complex of clock proteins (Figure 38). For the simplicity of the model and because we lack the functional differences between the core clock proteins, we assume that PER and CRY are the same species. Therefore, PER/PER represents combinations of PER/PER, PER/CRY, and CRY/CRY dimers. The crucial structure of a negative feedback regulation originates from a transcriptional-translational control loop (TTCL) consisting of the transcription factors BMAL1 and CLOCK that form heterodimers (BMAL1/CLOCK) and triggers transcription of the core clock genes (Per and Cry). After translation and complex formation, the activated clock proteins (PER/PER) downregulate their own transcription through binding into the BMAL1/CLOCK complex and closing the negative feedback loop of the system. Furthermore, in our model we assume that the PER dimers (CP2) are more stable than the single proteins, which introduces an additional auto-catalytic positive feedback into the model [283]. This simplified circadian clock system has been connected to the mammalian cell cycle network and analyzed within the deterministic and stochastic framework [47].

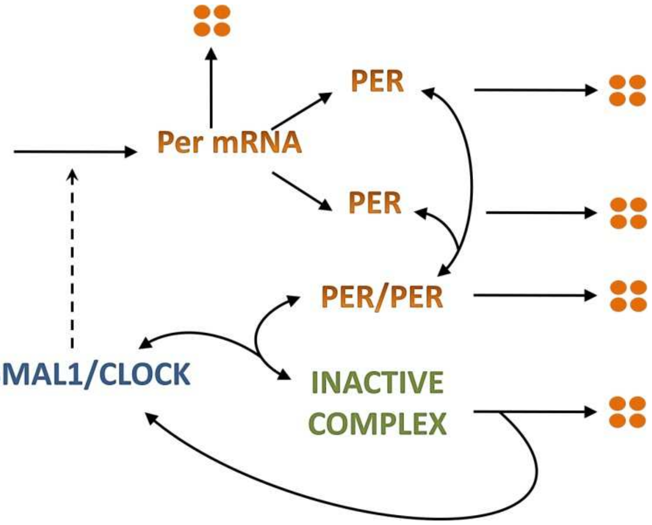


Figure 38: Simplified circadian clock model.

Simulations show robust endogenous oscillations with a period of 24 h (Figure 39).

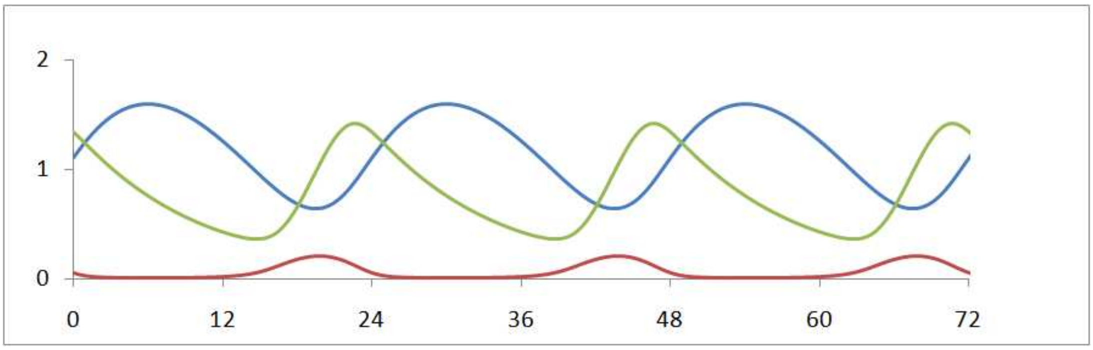


Figure 39: Deterministic simulation results of minimal circadian rhythm model. Blue curve indicates the total concentration of clock proteins (present in the CP, CP2 and IC forms); red curve shows the concentration of active transcription factors (TF) and the green curve states for the messenger molecule (M) in the model.

For stochastic simulations, we first introduced noise into the cell cycle regulatory equations by rewriting the cell cycle model as Langevin-type equations with multiplicative noise [167,284]:

$$\frac{d}{dt}x_i = f_i[\dots] + w_i(t) \cdot \sqrt{2 \cdot D_i \cdot x_i}$$

where *fi*[. . .] means the original deterministic equation, *wi(t)* is Gaussian white noise with 0 mean and unit variance, and *Di* is the noise amplitude. For simplicity, we kept the noise amplitude constant (0.005) for all variables. This number was set by matching the coefficient of variation (CV) of simulated uncoupled cell cycle length (at mass doubling time (MDT) = 24 h) to experimentally observed CV = 10% [285]. As a first assumption, we did not introduce stochasticity in the circadian clock module because its sensitivity to noise may not reflect a truly robust clock mechanism, being an overly simplified version of a clock model.

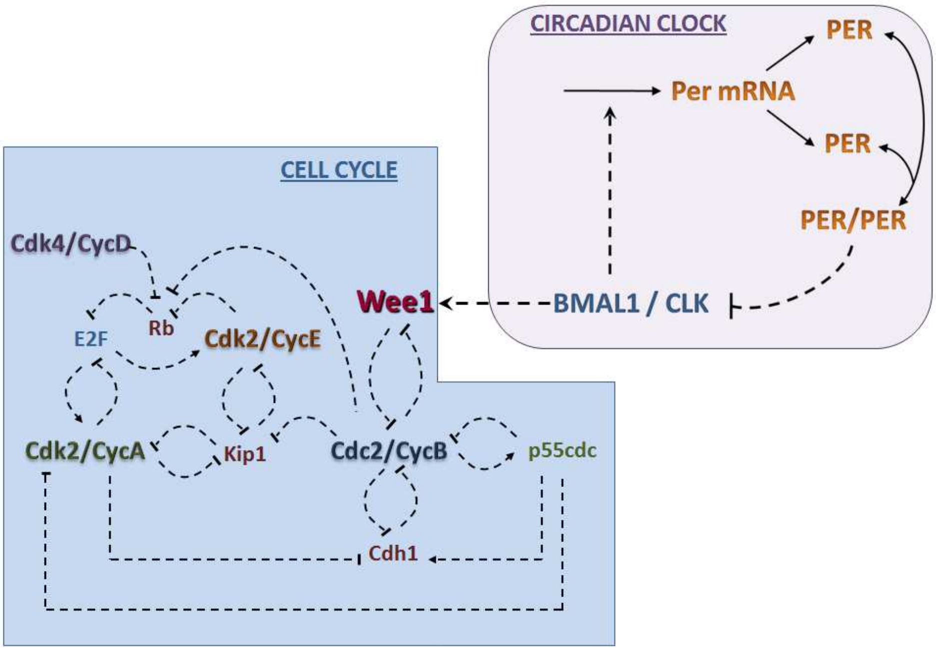


Figure 40: Simplified wiring diagram for the coupled cell cycle and circadian clock.

As we vary the mass doubling time (MDT) of the cell cycle, stochastic simulations reveal quantized cell cycles when the activity of Wee1 is influenced by clock components (Figure

41 and Figure 42). The quantized cell cycles disappear in the absence of coupling or when the strength of this link is reduced.

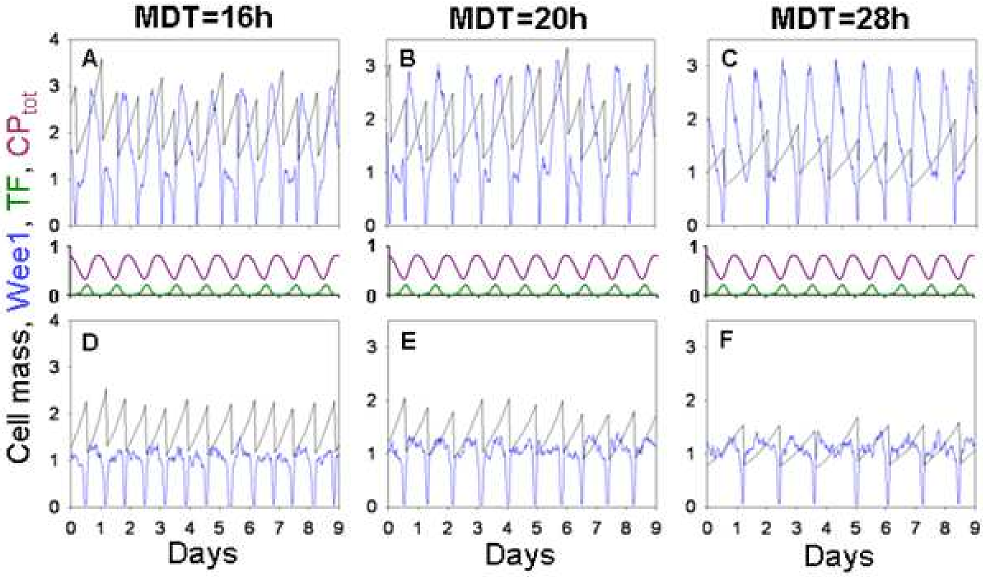


Figure 41: Clock-influenced cell cycle results in uneven distribution of cell cycle time. (A), (B), and (C) represent cell cycle simulations with strong coupling rate and the mass doubling time (MDT) at 16 h, 20 h, and 28 h, respectively. Clock-regulated Wee1 (blue) results in variations in sizes and cell cycle times at different MDTs. The black line represents cell mass, which grows exponentially and divides by a factor of 2. Such large deviations are not observed with weak coupling at MDTs 16 h, 20 h, and 28 h (D–F). The middle panels show a robust 24-h endogenous period of *CP*tot (purple) and transcription factor (*TF*; green) at various MDTs.

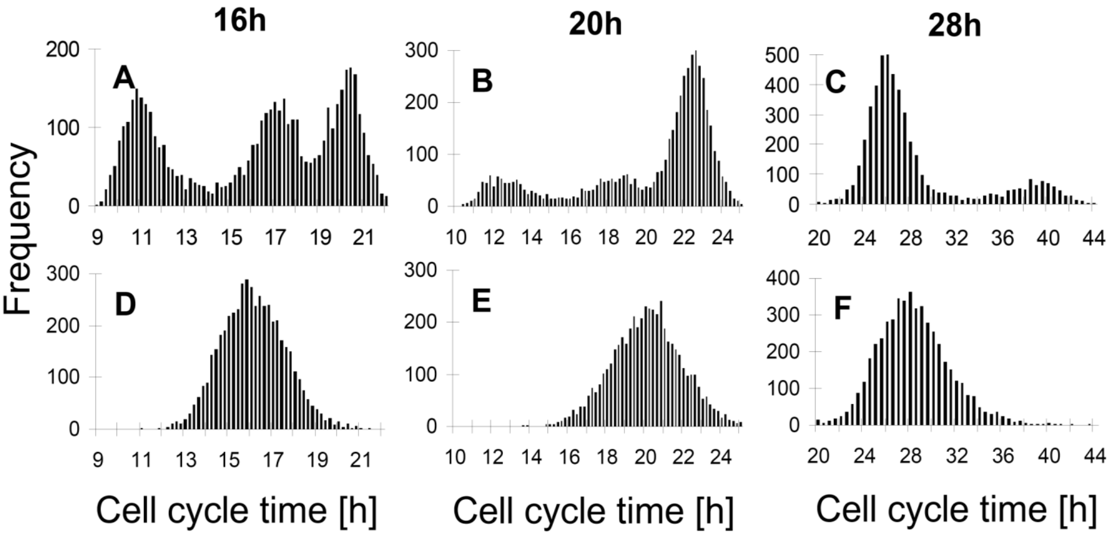


Figure 42: Histogram of cell cycle time distribution at 3 different mass doubling times (MDTs). The *y*-axis represents number of cells going through cell division with a particular cell cycle time. Strong coupling results in multimodal distribution of cell cycle times (A–C), while weak coupling results in normal distribution (D–F) at indicated MDTs (16 h, 20 h, and 28 h). Five thousand cell cycles are analyzed for each plot, which is calculated from 100 simulation runs with 50 consecutive cell cycles.

Quantized cell cycle distributions in mammalian cells have been measured experimentally by Robert Klevecz (Figure 43) and Nagoshi et al. [286]. Our simulation reveals the biological mechanism underlying this experimentally observed phenomenon [47].

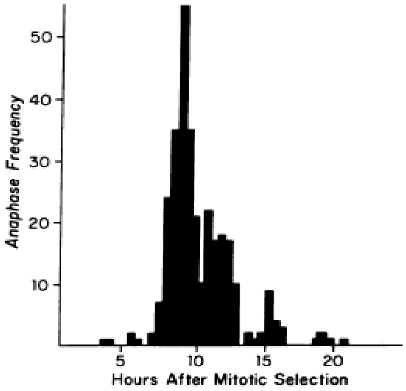


Figure 43: Quantized cell cycle distribution of mammalian cell culture measured experimentally by Klevecz [287].

More intriguingly, our simulations indicate that the circadian clock triggers critical size control in the mammalian cell cycle (Figure 44). Gating on the cell cycle progress via Wee1 enforces size control when the MDT (mass doubling time) is quite different from the circadian period which is 24h in mammalian cells. No size control is observed in the absence of coupling.

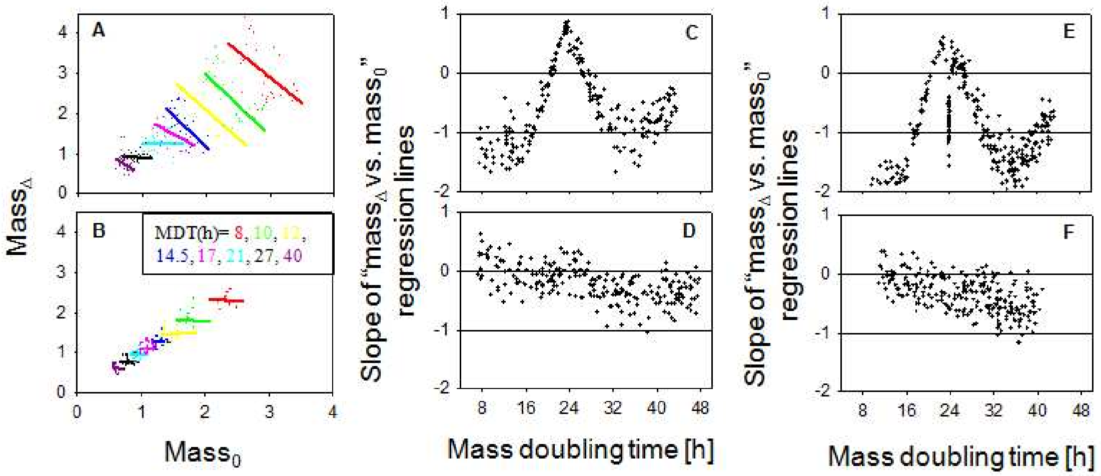


Figure 44: Analysis of critical mass control. (A, B) Growth from cell birth to division (massΔ) is plotted as a function of birth mass (mass0) for multiple simulations at different mass doubling times (MDTs). Data points are color coded and clustered according to particular MDTs. Cell size control is reflected when there is a negative correlation (slope of about –1) between massΔ and mass0. Strong coupling results in strict size control when cell masses are either large or small but no apparent correlation at intermediate cell masses (A). Weak coupling (B) shows no clear size control. About 250 simulation runs are calculated at different MDTs. For clear representation, not all data points are displayed on panels (A) and (B), and the legends for both panels are inserted on panel (B). (C, D) Slopes of linear regression lines from (A, B) are plotted as a function of the MDT. Strong coupling results in strict mass control (slope about –1) when the MDTs are either much shorter

or longer than 24 h, but size control is not observed when the MDT is close to 24 h (C). Weak coupling shows no apparent mass control (D). (E, F) Similar results are shown with linear growth rate. Unique slope of regression lines of massΔ vs. mass0 plots are observed with strong coupling as a function of MDT (E), as seen with exponential growth rate (C). The MDT is calculated from the average cell cycle time of 50 cycles.

The issue of size control in the mammalian system is debatable, whereas it is well established in yeast. It is possible that the size control is more readily observed in cell lines that contain circadian rhythms, since not all cell types have a circadian clock.

Following the ideas presented ahead [47] we translated the ODEs describing the coupled system of the cell cycle and the circadian rhythm into BlenX [288]. The main difference between the previous stochastic model and the BlenX one is that the current, translated model is fully stochastic and discrete. The ODE model presents noise only in the cell cycle part but not in the circadian clock. Furthermore, the stochasticity of the model originates from additional noise terms described by Langevin’s equations while in case of the BlenX model, Gillespie’s stochastic simulation algorithm is implemented into both the cell cycle and the circadian oscillator. The expression power of BlenX has been extended previously with the definition of rate functions through the process calculus events. Namely, BlenX let the user to specify a rate function that is used in place of the Gillespie method to compute the propensity function. We translated the biochemical reactions describing the system in ODEs into the BlenX framework with the help of these functions (see Chapter 2.10). Analysis of the simulations using a method based on the Fourier analysis confirmed our simulation results [288] that multimodial distribution occur when circadian clock gates the cell division cycle through Wee1 transcription.

As a further step, we extended the circadian rhythm model and fit it to experimental measurements. We added the transcription and degradation of BMAL1 mRNA and refined the model by adding different direct feedbacks (a positive, a negative and a combination of them) from the translated BMAL1 on the transcription of its mRNA.

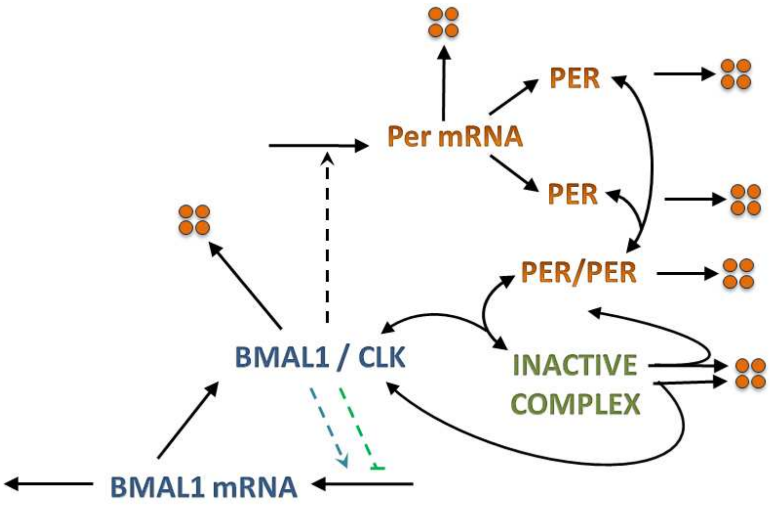


Figure 45: Modified circadian clock model based on the results of Sato et al. [191] and Preitner et al. [190]. Our simplified model has been extended with the synthesis and degradation of the messenger of the transcription factor BMAL1/CLOCK and the positive (blue dashed arrow) and negative (green dashed |-- sign) feedback regulation of the BMAL1/CLOCK protein complex.

We tried to infer reaction rates from the experimental data using KInfer [140], a tool for estimating rate constants of biochemical network models. We concluded that revision or extension of our model is necessary in order to fit experimental models quantitatively. Although, we see that such a small regulatory system describing circadian clock could propose possible answers for our questions raised about coupled oscillators.

We went further and compared our simple model to a detailed regulatory network proposed by Leloup and Goldbeter [211]. The author’s model contains additional regulatory loops to our simplified version of clock. Among other differences, they implemented the BMAL1/CLOCK driven inhibition of the Bmal1/Clock messenger. However, their system lacks a positive feedback loop that we proposed to present due to the dimer formation of the PER proteins resulting in a more stable form of the main clock negative element. By analyzing both models, we found this positive limb important in a different context.

In the following session our investigation on the effect of DNA-damaging agents on the circadian clock and how this simple model can match an experimentally observed and surprising phenomenon will be presented.

5.8.2 Analysis of DNA damage-induced phase advances in circadian rhythms

Since the early discoveries of circadian clock–gated cell cycles in lower eukaryotes [222], numerous molecular findings that connect the cell cycle and circadian clock are now being addressed [175,225]. In addition to the cell division cycles found to be gated by the circadian clock, ionizing radiation (IR) treatments cause cells to undergo a DNA damage response, which leads to phase shifts (mostly advances) also in circadian rhythms. DNA damage-induced activation of the cell cycle regulator, Chk2 kinase, results in phosphorylation and destruction of a circadian clock component (i.e., PER1 in *Mus musculus* or FRQ in *Neurospora crassa*) [228,229].

One of the identifying properties of circadian rhythms is the ability to phase shift upon a stimulus from external cues. This property allows organisms to adapt efficiently to the external environment. For example, a person traveling east to Europe from the U.S. will experience a jet-lag in the process to adapt advanced phase. It is intuitive to assume that a phase shifting agent will create both phase advances and delays depending on the timing and strength of the pulse by uniformly affecting molecular pathways in the circadian system [289]. It has been observed that 2h treatments of Rat-1 fibroblasts with the drug called dexamethasone (Dex) result in large advances and delays (Type 0 resetting of the phase), possibly by inducing transcription of both Per1 and Per2 genes [229,290]. This Dex-dependent phase response curve (PRC) is also observed in the NIH3T3-Bmal1-Luc-1 cells [286]. If the Dex-dependent induction of Per transcripts causes both phase advances and delays, we would also predict that DNA damage-dependent phosphorylation and degradation of PERs by Chk2 [228,291] would result in similar PRCs. Recent findings indicate that this intuition is wrong [228,229]. Upon experiencing DNA damage, the cell cycle machinery influences the circadian clock in such a way that creates predominantly phase advances in Rat-1 fibroblasts and mice, as well as in *Neurospora crassa*. In order to address the criteria of such behavior, we employed computational modeling to simulate different phase response curves (PRCs) resulting from Dex and IR treatments [48].

Phase response curves (PRCs) illustrate the relationship between the timing and the effect of a treatment designed to affect the circadian clock on a population level (Figure 46). A PRC is a graph showing, by convention, the time of the subject's endogenous day (along the x-axis) and the amount of the phase shift that occurs upon the stimulus (along the y-axis). Phase advances are plotted as positive values while phase delays take negative numbers.

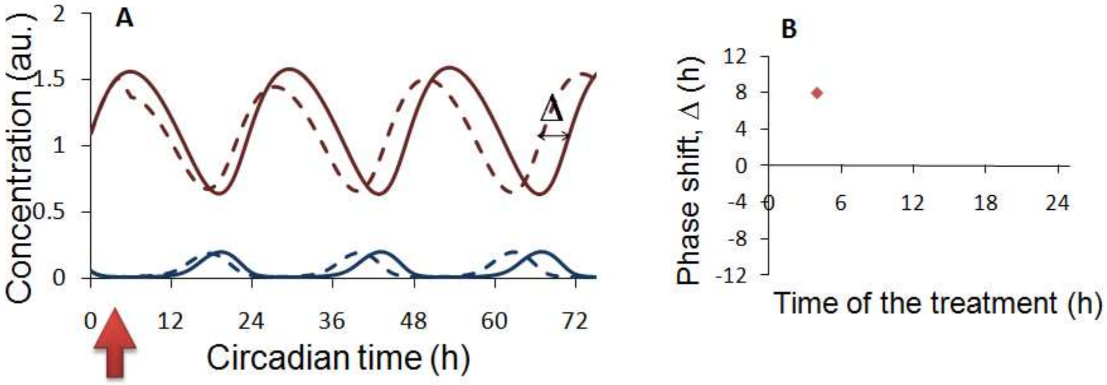


Figure 46: Measurement of the phase response curves. On (A) circadian cycles of clock components are shown with (dashed lines) and without (solid lines) phase-shifting agents given at a circadian time 4h in this example (red arrow). Value of the phase shift compared to the original phase (solid lines) in a later period is signed as **Δ**. (B) shows a phase response curve with the time of the treatment on the x-axis (4h in this example) and the hours of advances (positive values) or delays (negative values) on the y-axis.

Dexamethasone (Dex) is known to synchronize circadian rhythms in cell cultures and may generate both phase advances and delays (Figure 47 left handside). On the other hand, ionizing radiation treatment leads to unique phase responses with minimum delays of the circadian clock (Figure 47 right handside).

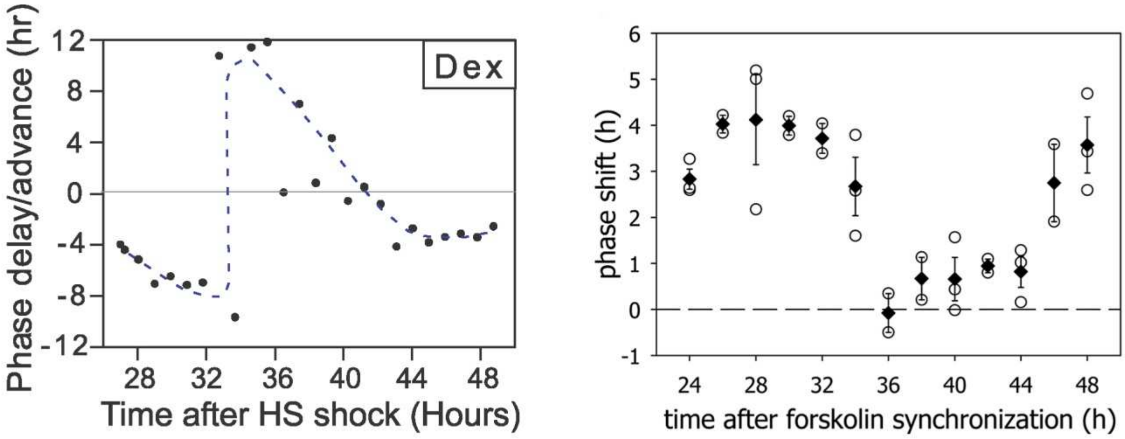


Figure 47: Phase response curves of mammalian circadian clocks upon dexamethasone (left side figure) and upon ionizing radiation treatment (on the right side). Note the different shape of the curves with large advances and delays in the left case and mostly advances on the right side figure. Both plots are taken from published articles of the experimental work done by Izumo et al. [290] and Oklejewicz et al. [229].

We explored a simple mammalian circadian clock model from our previous work [47] to investigate whether we can simulate different PRCs from the Dex and IR treatment experiments. Based on the experimental data, we added the following in our previous model: 1) Dex increases the transcripts of PER but not BMAL1, and 2) Chk2 phosphorylates PERs and facilitates their degradation upon DNA damage (Figure 48).

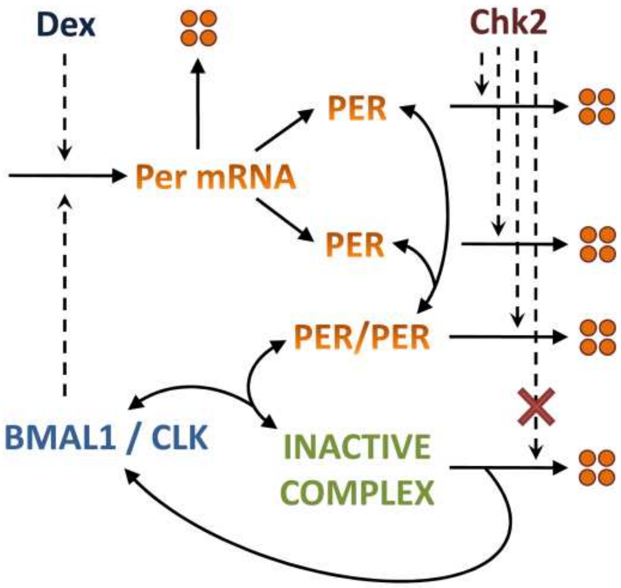


Figure 48: Graphical representation of the circadian model applied in this study. The effect of Dex and Chk2 has been investigated. Our simulations propose that PER proteins in the inactive complex are not affected by the Chk2 kinase (red cross).

Our simulations [48] show that the Dex-dependent increase of Per messages creates both Type 0 (as shown in the experiment, strong resetting of the phase) and Type 1 PRCs (weak resetting of the phase) depending on the strength (concentration) of the Dex treatments (Figure 49). It is, however, not trivial to simulate a PRC with mostly phase advances reproducing the phenotype from the IR treatment experiments. We can achieve this phenomenon if Chk2 prematurely degrades PERs that are not bound to BMAL1/CLK to advance the clock, while allowing continued repression of BMAL1/CLK by not degrading the PERs that are in a complex with BMAL1/CLK (Figure 48).

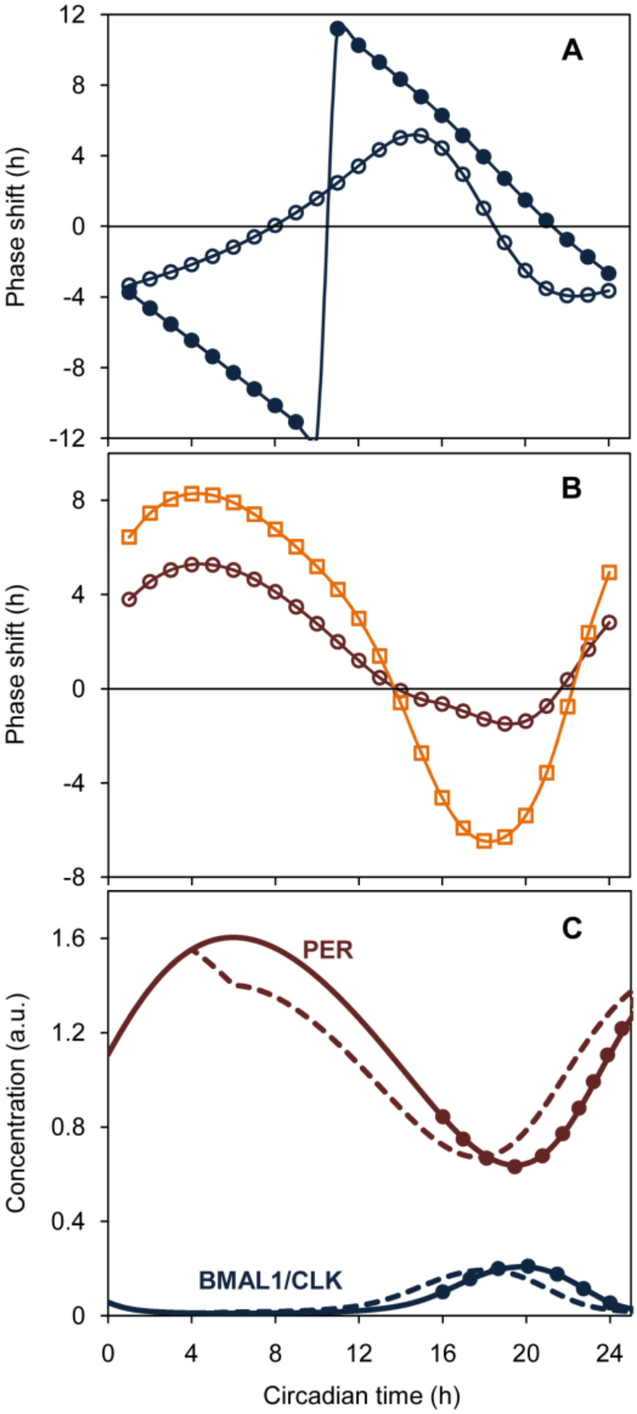


Figure 49: In silico Dex and IR treated experiments. (A) Strong pulses of Dex generate Type 0 PRC (filled circles; strong resetting of the circadian clock to the new phase which does not depend on the old phase) whereas weak pulses of Dex generates Type 1 PRC (blank circles; weak resetting of the phase where the new phase changes as a function of the old phase). (B) Large advances and delays are observed when Chk2 is assumed to affect all forms of PERs including the complex with BMAL1/CLK (orange squares). Chk2-dependent phase advances and minimum delays of the circadian clock are observed only if Chk2 does not affect the PERs that are in complex with BMAL1/CLK (red circles). (C) DNA damage-induced Chk2 activation causes phase advances of circadian clock. Solid lines represent endogenous profiles of PER and BMAL1/CLK. Dashed lines indicate PER (red - CPtotal) and BMAL1/CLK (blue - TF) in response to a 2 h IR treatment at simulation hour 4 and dots represent the results after the same 2 hr treatment at hour 16 (hour 0 corresponds to the peak of PER monomers (CP)).

We based our conclusions on the analysis of a simplified model and also on the investigation of a more comprehensive model proposed by Leloup and Goldbeter [211,215]. Comparison of their model and our simple circadian clock system is presented in the subsequent session.

5.8.3 An autocatalytic positive feedback mechanism as one of the criteria

Theoretically, a time-delayed negative feedback is sufficient to create robust oscillations. However, biological systems contain both negative and positive feedbacks in their wiring networks. Positive feedback mechanisms are essential for proper eukaryotic cell divisions [292] whereas their roles in circadian rhythms remain elusive. Recently, Tsai and colleagues indicated that a general function of positive feedbacks in different networks is to create tunable robustness in the system [293]. After proposing a molecular mechanism that accounts for Chk2-dependent PRC in circadian rhythms (in Chapter 0), we also tried to answer if the positive feedback mechanism is necessary for the observed PRC.

The autocatalytic positive feedback mechanism in the model arises from different stabilities between PER monomers vs. PER complexes. Based on molecular data from Drosophila system [294-296], we assume that PER monomers are more susceptible to degradation than PER in complexes (i.e. PER/PER, PER/CRY, etc.). This creates autocatalytic PER dynamics as PER stabilizes itself by forming complexes. To date, this is the only circadian rhythm model that employs an essential positive feedback mechanism that is necessary to maintain a robust oscillator [283]. Hence, we wondered whether the incorporated essential positive feedback is required (or disposable) in simulating the unique PRCs upon DNA damage.

In order to test our hypothesis, we removed the autocatalysis in the model by assuming no stability differences between PER monomers and complexes. Then, we re-parameterized the system to rescue oscillations. Note that we had to use a Hill-coefficient = 4 for highly cooperative negative feedback in order to rescue oscillations in our four-variable model in the absence of the autocatalytic positive feedback mechanism. To our surprise, we were not able to generate the unique PRC with predominantly phase advances upon DNA damage even by assuming differential phosphorylation and degradation of PER monomers vs. PER complexes with BMAL1/CLK.

We wondered whether above conclusions from our simple model can be generalized to a more comprehensive model with distinct wiring network. Hence, we tested Leloup and Goldbeter's mammalian model [211,215]. They used four sets of parameters in order to investigate possible functions of multiple feedback loops in the circadian system. For our purposes, we concentrated to parameter sets 1 and 3. In the parameter set 1, robust oscillations of their model can arise from two different time-delayed negative feedback loops: PER-driven and PER/CRY-independent BMAL1/CLK-driven negative feedback loops. For this parameter set, they can generate an oscillator based on BMAL1/CLK-driven negative feedback loop in the absence of the PER-driven negative feedback loop. In the parameter set 3, they disabled the BMAL1/CLK-driven negative feedback loop making the system a PER/CRY-dependent single negative feedback oscillator. We did not explore parameter sets 2 and 4 because PER is not required for oscillations in parameter sets 2 and 4. The wiring network of Leloup and Goldbeter's model is significantly different from our model which consists of an intertwined dynamics between an essential autocatalytic positive feedback and time-delayed negative feedback [186,283].

We incorporated Chk2-induced degradation of PER molecules that are not bound to BMAL1/CLK in the Leloup and Goldbeter's model. Then, we tested Chk-2-dependent differential degradation of PER as in our simple model. Our simulations indicate that we see both TYPE 1 and TYPE 0 PRC depending on the strength of Chk2, but we do not observe asymmetric PRCs with mostly advances (Table 14). These results show that the differential effect of Chk2-dependent degradation of PER complexes is not enough to create the observed DNA-damage induced PRCs with the innate wiring of the Leloup and Goldbeter's model.

Our next step was to introduce an autocatalytic positive feedback mechanism in the Leloup and Goldbeter's model and investigate its role in reproducing the asymmetric PRC upon DNA-damage. First, we added an autocatalytic positive feedback in the parameter set 1 of Leloup and Goldbeter's model in a similar way as in our simple model. PER complexes are assumed to be more stable than PER monomers. To our surprise, we were not able to generate the PRCs with predominantly phase advances with differential degradations of PER complexes by Chk2 even with an added autocatalytic positive feedback mechanism. We wondered whether this was due to the PER independent BMAL1/CLK-driven negative feedback loop which is built in the parameter set 1. Hence, we tested the parameter set3 which consists of the PER-driven single negative feedback. Interestingly, we were able to simulate the observed asymmetric PRC with predominantly phase advances as we have observed in our simple model only when both the autocatalytic positive feedback and the differential effect of Chk2 on PERs were implemented in the absence of BMAL1/CLK-driven negative feedback loop. This suggests that there exists an important dynamical relationship between negative feedback loops and an autocatalytic positive feedback mechanism.

MODEL	Circadian period	Changed parameters	Chk2 value	Maximum advance (h)	Maximum delay (h)	Ratio of maximum advance and maximum delay	Positive feedback
Simple model	24.0	-	0.2	5.27	-1.49	3.54	Yes
Simple model, positive feedback removed	26.8	many[1]	0.05	8.89	-11.56	0.77	No
Leloup and Goldbeter Set1	24.0	-	1	1.24	-2.18	0.57	No
Leloup and Goldbeter Set 3	23.9	-	1	3.94	-3.55	1.11	No
Leloup and Goldbeter Set1 with positive feedback	25.2	k_{dnp} = 0.3[2]	1	1.69	-2.39	0.71	Yes
Leloup and Goldbeter Set 3 with positive feedback	20.8	k_{dnp} = 0.3[2]	1	10.66	-4.32	2.47	Yes

[1]: Parameter set (without positive feedback):
Rate constants (h^{-1}): k_{ms} = 0.5, k_{md} = 0.045, k_{cps} = 10, k_{cpd} = 0.0001, k_a = 100, k_d = 0.001, k_{cp2d} = 0.0001, k_{icd} = 0.001, k_{ica} = 4, k_{p1} = 1.97, k_{p2} = 1.97. Dimensionless constants: TF_{tot} = 1, J_p = 0.05, J = 0.4, n = 4.

[2]: We introduced a new rate constant k_{dnp} as the nonspecific degradation rate constant of nonphosphorylated PER proteins in the cytosol (P_C in their model).

Table 14: Detailed results of the positive feedback necessity analysis

In the conditions that we have tested, we discovered that we can only simulate the Chk2-dependent PRC with predominantly phase advances when Chk2 only affects PERs that are not bound to BMAL1/CLK in the presence of an autocatalytic positive feedback mechanism. Both conditions are required for proper simulations. Our study is the only in silico experiment to indicate the necessity of an autocatalytic positive feedback mechanism in simulating specific phenotype in the circadian system.

The presented scheme is an example of a simplified model of biological rhythms where the assumptions applied in the complex nonlinearity terms have important roles in producing oscillatory cycles. Conversion of the model into stochastic simulations requires the decomposition of complex rate functions into elementary reactions. Goldbeter and Leloup proposed an analysis for a stochastic version of a *Drosophila* circadian model. Their model system is based on the assumption of Michaelis-Menten kinetics for most of the reactions. A highly nonlinear Hill term is also present with a Hill coefficient equal to 4. The authors decomposed the reactions into elementary steps by hand.

We decided to apply a systematic way to decompose our circadian model that accounts for several hypotheses corresponding to biological rhythms. During the compositional study, assumptions have to stay valid for the whole model if we would like to match the global properties of the system, provided by nonlinear functions in the original model. The proteins present in the clock have multiple roles [297], thus the possibility of choosing between complex rate functions or elementary reactions containing the required elements (e.g. an enzyme) of the reaction explicitly provides a more flexible use of the BlenX language for later extension of the model. Compositionality remains an important and helpful advantage of BlenX in a template based, modular environment.

5.8.4 A circadian clock model built up from BlenX templates

We have shown (see Chapter 5.8.3) that our simplified circadian clock model possesses a crucial role for a positive feedback presented in the system. In order to investigate a stochastic version of this model, we decided to transfer the system including nonlinear complex reaction terms with the use of our novel template library. The circadian clock presented above is a simplified picture where non-elementary steps create robust behavior. If we would like to have a realistic model where assumptions are not used to model complex reaction schemes, we rely on single step mechanisms. The BlenX templates presented ahead suit this need.

The system is divided into the following modules: (1) transcriptional regulation following Hill kinetics (2) translation mechanism (assumed to follow mass action kinetics in this study) (3) homodimerization of clock proteins (CP) (4) formation of an inactive complex providing a negative effect inside the loop (5) There are three degradation terms catalytically activated by enzymes (following the Michaelis-Menten assumptions) and the system also contains linear (so called background) degradation of the elements. This network composed of the reactions presented above (Figure 50) is built with complex rate functions and provides a 24h periodic oscillator.

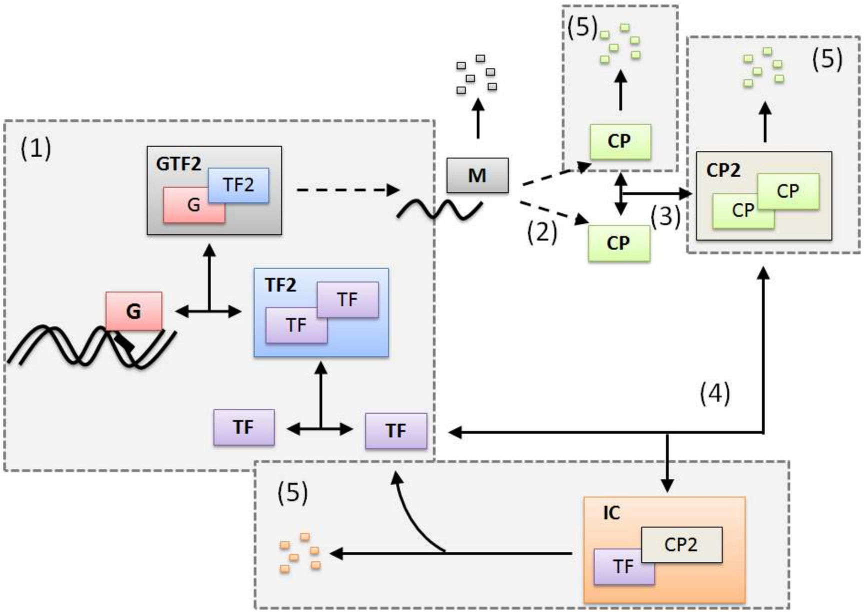


Figure 50: Circadian clock model as a composition of transcriptional (1), translational (2), reverse complex formation (3, 4) and the enzyme catalyzed degradation modules (5). The highlighted parts are described as multi-step reactions assumed to follow Hill function and Michaelis-Menten kinetics.

We chose the number of enzymes having role in the system to be less than the corresponding substrates, making the assumptions of Michaelis-Menten kinetics valid. The parameters originating from the complex functions are also scaled up to be fast enough. Thus the reactions assumed to be in equilibrium do not limit the system and provide accordance with the original assumptions. The products of the enzymatic reactions are degraded immediately (with an infinite rate) after their production in order to serve the catalyzed degradation scheme in the system.

Following the template-definition, we simply merge the modules and insert the boxes (enzymes and intermediates) of the novel entities. We also replace the events corresponding to the complex functions for the ones from the “unpacked” modules. This method can be easily automatized as it does not require the modification of the reactions that are independent of the substituted complex functions and the functions calculating the rate of complex reactions does not involve binders. The novel internal behavior of the boxes can be easily parallelized with the original ones. The composition of modules is shown in Figure 52. Additional boxes encoding the originally indefined enzymes (U1, U2, U3) acting in the Michaelis-Menten modules are listed as well as the temporary boxes waiting for degradation after the catalytic step (CPU1_D, CP2U2_D, ICU3_D). The gene and the dimers for the transcription module are also shown on Figure 53.b. Several steps are not modified (Figure 53.g). Internal processes are added parallel to the original boxes.

Simulation of the “unpacked” system (Figure 51 1b and 2b) shows larger noise than the original (Figure 51 1a and 2a), but still produces regular oscillations.

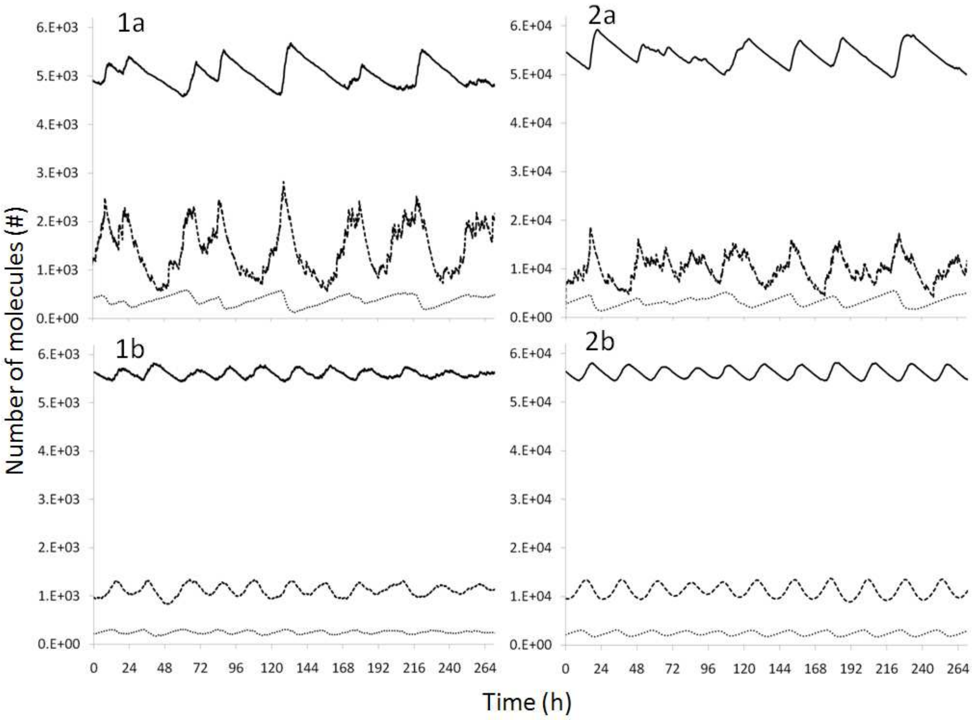
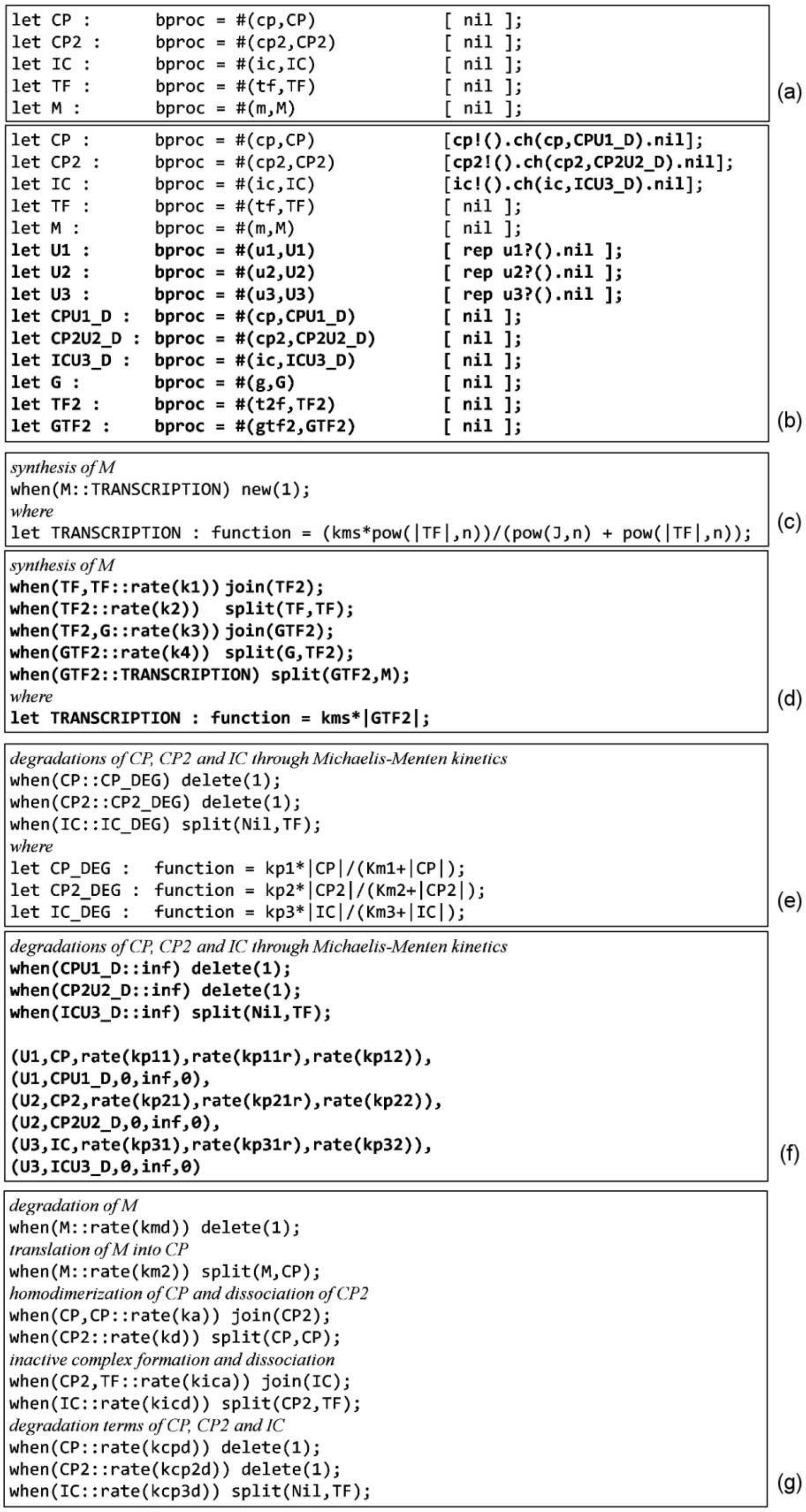


Figure 51: Simulation results for the stochastic model containing complex rate functions (1-2b) and for the “unpacked” versions (1-2a) in case of a conversion factor α=0.000167 (1a-b) and α=0.0000167 (2a-b). The total amount of CP in the system is plotted as solid curves; dashed curves represent the messenger (M) while dotted points demonstrate the free transcription factors (TF) in the model.



```
let CP :       bproc = #(cp,CP)          [ nil ];
let CP2 :      bproc = #(cp2,CP2)        [ nil ];
let IC :       bproc = #(ic,IC)          [ nil ];
let TF :       bproc = #(tf,TF)          [ nil ];
let M :        bproc = #(m,M)            [ nil ];
```

(a)

```
let CP :       bproc = #(cp,CP)          [cp!().ch(cp,CPU1_D).nil];
let CP2 :      bproc = #(cp2,CP2)        [cp2!().ch(cp2,CP2U2_D).nil];
let IC :       bproc = #(ic,IC)          [ic!().ch(ic,ICU3_D).nil];
let TF :       bproc = #(tf,TF)          [ nil ];
let M :        bproc = #(m,M)            [ nil ];
let U1 :       bproc = #(u1,U1)          [ rep u1?().nil ];
let U2 :       bproc = #(u2,U2)          [ rep u2?().nil ];
let U3 :       bproc = #(u3,U3)          [ rep u3?().nil ];
let CPU1_D :   bproc = #(cp,CPU1_D)      [ nil ];
let CP2U2_D :  bproc = #(cp2,CP2U2_D)    [ nil ];
let ICU3_D :   bproc = #(ic,ICU3_D)      [ nil ];
let G :        bproc = #(g,G)            [ nil ];
let TF2 :      bproc = #(t2f,TF2)        [ nil ];
let GTF2 :     bproc = #(gtf2,GTF2)      [ nil ];
```

(b)

```
synthesis of M
when(M::TRANSCRIPTION) new(1);
where
let TRANSCRIPTION : function = (kms*pow(|TF|,n))/(pow(J,n) + pow(|TF|,n));
```

(c)

```
synthesis of M
when(TF,TF::rate(k1)) join(TF2);
when(TF2::rate(k2))   split(TF,TF);
when(TF2,G::rate(k3)) join(GTF2);
when(GTF2::rate(k4))  split(G,TF2);
when(GTF2::TRANSCRIPTION) split(GTF2,M);
where
let TRANSCRIPTION : function = kms*|GTF2|;
```

(d)

```
degradations of CP, CP2 and IC through Michaelis-Menten kinetics
when(CP::CP_DEG) delete(1);
when(CP2::CP2_DEG) delete(1);
when(IC::IC_DEG) split(Nil,TF);
where
let CP_DEG :   function = kp1*|CP|/(Km1+|CP|);
let CP2_DEG :  function = kp2*|CP2|/(Km2+|CP2|);
let IC_DEG :   function = kp3*|IC|/(Km3+|IC|);
```

(e)

```
degradations of CP, CP2 and IC through Michaelis-Menten kinetics
when(CPU1_D::inf) delete(1);
when(CP2U2_D::inf) delete(1);
when(ICU3_D::inf) split(Nil,TF);

(U1,CP,rate(kp11),rate(kp11r),rate(kp12)),
(U1,CPU1_D,0,inf,0),
(U2,CP2,rate(kp21),rate(kp21r),rate(kp22)),
(U2,CP2U2_D,0,inf,0),
(U3,IC,rate(kp31),rate(kp31r),rate(kp32)),
(U3,ICU3_D,0,inf,0)
```

(f)

```
degradation of M
when(M::rate(kmd)) delete(1);
translation of M into CP
when(M::rate(km2)) split(M,CP);
homodimerization of CP and dissociation of CP2
when(CP,CP::rate(ka)) join(CP2);
when(CP2::rate(kd)) split(CP,CP);
inactive complex formation and dissociation
when(CP2,TF::rate(kica)) join(IC);
when(IC::rate(kicd)) split(CP2,TF);
degradation terms of CP, CP2 and IC
when(CP::rate(kcpd)) delete(1);
when(CP2::rate(kcp2d)) delete(1);
when(IC::rate(kcp3d)) split(Nil,TF);
```

(g)

Figure 52: BlenX source code. The boxes of the original model are shown in (a) while the “unpacked” version is in (b). Composition of (a) and (b) is a straightforward job by parallelization. Events of the original model are in (c), (e), (g), while (d), (f), (g) contains the unpacked version of the model. Note that there is no change in (g). The substituited modules are highlighted (bold font) in the text. The model and the parameters are in Appendix B.

We note that the stochastic simulation results of our unpacked circadian model containing only elementary reaction steps show larger noise than the one with the complex terms. However, this is not necessary the case in all systems. There are several scenarios when adding multiple reaction steps decrease the noise in the overall network [263]. In our model stochastic fluctuations may be reduced by increasing the number of molecules within the model. This can be inefficient and difficult to realize in case of certain oscillators where the amplitude of the cycles can bring the number of species down even if the peaks are large enough. Another solution could be to reduce noise (also in our circadian clock model) to extend the model with further regulatory modules, such as negative feedbacks. We know that the molecular network of daily rhythms include several additional negative feedback loops, thus we think that interconnected negative feedback loops also possess crucial roles in circadian clocks. In order to build a more realistic model based on our simple and predictive system, we can compose larger models with the help of predefined biological network motifs that ease the model building process in future.

It has been shown in several works [238] that with process calculus based languages dynamic models can be constructed and existing continuous models can be transferred into the stochastic framework providing additional predictions of the biological system results to the existing models [92]. Herein we have to remind the reader that generally distributed reaction times have been also implemented into the BlenX framework recently [137]. The use of this extension fits well the idea of a template based modeling framework as, depending on the question the user asked, biological models might be characterized through complex rate laws and handled by generalized distributions of time; while templates (including only elementary steps) offer a straightforward, flexible and more precise way of compositional modeling in BlenX making additional extension of the model easy.

5.9 Experimental perspectives

The joint effort of experimental work and modeling approaches has been already provided interesting findings for biology [3]. To verify our predictions about the interconnected circadian and cell cycles (presented in this thesis), during my PhD intern period I had the opportunity to visit Chris Hong's laboratory at the University of Cincinnati, Ohio (USA) and to realize experiments. We chose an experimental model system, the *Neurospora crassa* to investigate the regulation of daily rhythms.

Neurospora crassa is a type of filamentus fungi (bread mold). It has been a popular model organism for circadian clock studies from the beginning because it is easy to grow and under constant conditions it shows apparent conidiation banding pattern every 22h as an output of its endogenous clock. Additionally, the entire genome of *Neurospora crassa* has been sequenced in 2003 [298]. Genetical manipulation of the organisms is simple enough to study molecular genetics in a straightforward way. Individual mutants are stored in a Neurospora database [299]. With the thousands of mutant stocks that can be ordered and with the several convenient techniques that are available for studying this model organism, we could start to set up experiments. The levels of conservation observed among the eukaryotic circadian oscillators highlight the importance of using *Neurospora* as a model system for circadian clocks.

Analogous to the daily oscillator of *Drosophila melanogaster* and mammals, the frequency (frq) and white collar (wc) genes have been presented to encode componenents of the molecular feedback loop essential for the circadian rhythmicity in *Neurospora* (see detailed review by [300]). They show similarity to the period (per) and Bmal1 (Bmal1) genes' functions in mammals, respectively. Transcription of frq is activated by a WC complex (WCC) composed of the WC-1 [301] and WC-2 [302] proteins (two PER-ARNT-SIM (PAS) domain-containing transcription factors). After the FRQ protein is synthesized, it dimerizes with itself and forms the FFC complex with FRH (an FRQ-interacting RNA helicase) [303]. In the nucleus, FFC inhibits the activity of WCC, resulting in a decrease in frq mRNA levels. Post-translational modifications of FRQ play an important role in the circadian system. When FRQ is synthesized, it is progressively phosphorylated by several kinases (CKI and CKII (casein kinases), CAMK-1 (a calcium/calmodulin-dependent kinase)) and dephosphorylated by phosphatases (PP2A (protein phosphatase 2), PP1, PP4) [304]. When FRQ becomes hyperphosphorylated, FWD-1 (an F box/WD-40 repeat-containing protein) binds to it and a SCF-type ubiquitin ligase complex ubiquitinates FRQ resulting in its degradation [305].

We decided to follow the expression of the circadian clock and the cell cycle components in different strains. We designed a vector for achieve genetic modification of the *Neurospora* genome, making measurement with the customized strain available. The designed DNA fragments were combined in yeast (with natural yeast homologous recombination [306]) and a highly-efficient Cyclosporin A-resistance based gene insertion method [307] was used to implement the designed constructs into the *Neurospora* genome. The inserted sequences contain the promoter region of the gene we wish to follow. The coding sequence of the luciferase enzyme [308] has been fused to the promoter mentioned above (Figure 53). Luciferase possesses enzymatic activity and with a bioluminescence reaction occurring with luciferin it provides light that we can detect with a digital camera (Figure 54). The luciferase gene is transcribed when the upstream fused promoter of the gene we would like to follow is activated. With a sensitive camera equipped with CCD sensor we can detect the expression profile of different genes real time with the help of this method.

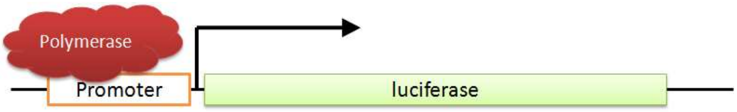


Figure 53: Schematic representation of a promoter-fused luciferase construct. When RNA polymerase is bound to the promoter region of a specific gene, luciferase is transcribed. In a luciferin-media, detection of the induced expression is possible that provides us information about the transcription pattern of the gene having the same promoter region.

Neurospora cells were grown in equipements called race tubes (hollow glass tubes about 40 cm long and 16 mm in diameter, bent up at both ends in order to hold an agar growth medium). Cultures were inoculated onto agar media containing luciferin (a substrate of the enzymatic reaction induced by luciferase) where they grow across the surface at constant rate (Figure 55). Following inoculation and growth for a day in constant light, the position of the growth front was marked and the culture was transferred to constant darkness (LD transfer). This sets the clock running from CT 12 and daily patterns of the banding could be captured with a camera.

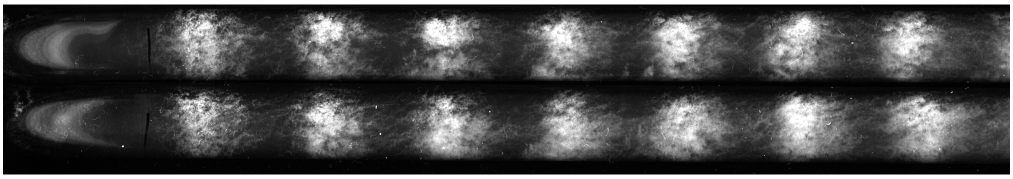


Figure 54: Luminescent Neurospora strains grown in race tubes. Banding pattern of conidiation is visible every 22h.

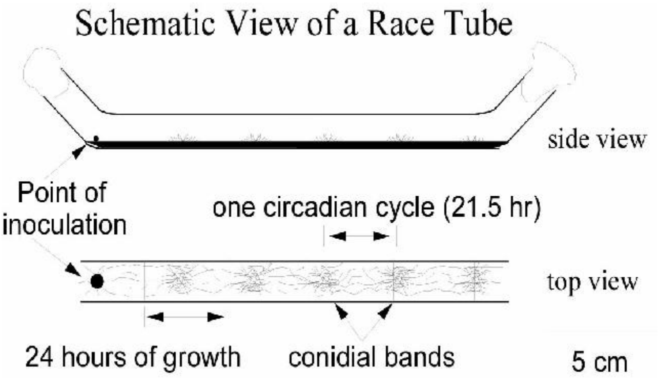


Figure 55: Schematic view of a race tube. The source of this figure is http://www.fgsc.net.

We know that PRD-4 (a checkpoint kinase in Neurospora) and FRQ physically interact, and that PRD-4 phosphorylates FRQ to reset the circadian clock. Furthermore, other links between the two systems have been shown in other organisms. However, in *Neurospora* we lack the detailed knowledge of the cell cycle that is well-characterized in yeasts and other model organisms. We followed the growth of the mycelia and the expression of several cell cycle and circadian clock mutants in different genetical backgrounds. We made this in real-time. Our preliminary results identify uncovered circadian clock regulated cell cycle elements in *Neurospora* (unpublished data) and in the future we plan to investigate our predictions carried out by theoretical work and we wish to understand the role of circadian clock in a DNA damage induced cell cycle pathway more. With the help of sophisticated experimental methods we will be able to extend our computational model with additional regulatory motifs and loops and investigate them under a stochastic framework, such as BlenX.

Conclusions

Biological systems are complex, thus novel systematic approaches try to handle them on a modular manner [78]. A molecular network is often represented as a composition of repetitive motifs that hold crucial behavior contributing to global properties. Several tools, e.g. [309,310] ensure a user-friendly interface for modeling such networks sometimes with the help of a graphical interface [311]. BlenX is one of the novel process calculus initiatives that support modularity by allowing biological systems to be composed from their components providing novel insights into systematic modeling. In this programming language designed for algorithmic systems biology, the affinities of molecules defining their present in reactions are coded in a formal way. However, the current modeling process with BlenX lacks biologically relevant modules abailable to reveal modularity on a larger scale. Compositionality supports that programs are built as an assembly of relatively independent computational units. In order to exploit the compositionality of BlenX, we defined re-usable and biologically relevant program templates that support modeling of complex reaction mechanisms often applied for kinetic description of biological networks. These modules enable the combination of several modeling methods and may also provide the programming bases for a user-friendly graphical approach. The main contribution of this thesis to systems biology is to have produced reusable and validated quantified modules and demonstrated their value in designing biological models within the BlenX language.

Model composition may begin at different stages. The modeler can build up the system from the bottom starting from the basic elements and primitives of the BlenX language. In order to fasten this process up, we provide the following flow of the use of pre-defined templates:

1) The desired modules can be chosen from the BlenX library.

2) Parameters of the whole model are asked from the user.

a) Users can define the whole set of parameters that they wish to use

b) Or users can define the known parameters

c) Or users can ask for estimation of the parameters from the software. This can be achieved by different algorithms or tools or can be calculated in some cases from deterministic simulations as we suggested in the Results session.

3) The program checks the parameters if they satisfy the assumptions of the particular templates.
4) If the assumptions are valid than the user is asked if he wants to insert the complex mathematical term based on the approximations of the theory. These formulas are also available in the template library and are easily composed within the process calculus framework of BlenX.
5) If the assumptions cannot be verified, the user can choose from the unpacked modules containing only elementary steps always satisfying Gillespie's stochastic algorithm.

The method presented above might include additional solutions for modeling nonlinear functions, for instance the usage of a generic distribution is a promising initiative proposed by Mura et al. [137].

There is another case when modeling has to deal with the problem of compositionality when we want to extend a network with novel links and biochemical reactions. The switch between the complex terms and the unpacked modules provided by the tool and stored in a template library bring a solution also to the problem of the model extension. Their structure and realization enable hidden enzymes and complexes to be present and explored within a novel biological system.

In a relevant case study, we investigated the effects of the circadian clock on cell cycle regulation in a deterministic and two different stochastic approaches (Langevin-type and SSA). Based on our computational analysis, we report in all cases quantized cell cycles when the transcription of a cell cycle regulator, Wee1, is strongly influenced by the circadian clock. This occurs from a "mode-lock" phenomenon that creates various periodic repetitions of cell division cycles with different mass doubling times. Strong circadian clock regulation on Wee1 transcription triggers cell size control at different mass doubling times (MDTs). Cell size control is observed during specific ranges of MDTs farther from the cycle of the circadian clock. When the circadian clock induces 24h periodic perturbations, it forces the cell cycle out of homeostasis from its division time. Our model shows qualitatively similar behaviors that have been observed in experiments by Robert R Klevecz [287].

Furthermore, our in silico experiments revealed two molecular criteria that accounts for another interesting clock phenomenon, that DNA damage shifts circadian clocks in a specific manner. The predominant present of phase advances of the daily rhythm upon DNA damage requires 1) the existence of an autocatalytic positive feedback mechanism in addition to the time-delayed negative feedback loop in the clock system, and 2) Chk2 dependent phosphorylation and degradation of PERs that are not bound to BMAL1/CLOCK. We confirmed our hypothesis with investigating another circadian clock model published by Leloup and Goldbeter [211].

Templates of the Michaelis-Menten and the Hill function have been applied for the G2/M transition and for the circadian clock study and compositionality has been carried out with the help of the process calculus origin of the BlenX language. Our current collection including addition modules which are our proposals for modeling is shown in Appendix A. Implementation of the template library into the CoSBi Lab framework - together with a parameter estimation toolkit or with the general distributions - results in a user-friendly and efficient tool for systematic modeling stochastic biological systems.

Comparison of different structures and different levels of abstraction may contribute to understand biological systems more. Noise has a large impact in the overall system and also it has been discussed that the effect of noise on regulatory circuits, feedback systems is important in the overall picture [312]. Analysis of the modules and motifs with different structures or different effect of noise (complex functions or elementary steps in case of low number of molecules) may lead to novel insights in the field. With BlenX as a proposed framework where analysis can be carried out easily with the help of compositionality may open the interest of biologists. Breaking a system down into a few reaction steps to be examined, while abstract the rest of the network in a different way, would lead to specific analysis of the biological system. Similar to the idea of synthetic biology [313] as an in vivo research line including artificial biological circuits, we approach biological systems with an in silico composition of regulatory motifs. The use of a template library in the model building process improves the current degree of compositionality and easens the systematic modeling with BlenX.

Bibliography

[1] B. Alberts, A. Johnson, J. Lewis, M. Ra, K. Roberts, and P. Walter, *Molecular Biology of the Cell*, Garland, 2002.

[2] H. Kitano, “Computational systems biology,” *Nature*, vol. 420, 2002.

[3] H. Kitano, “Systems biology: a brief overview.,” *Science (New York, N.Y.)*, vol. 295, Mar. 2002, pp. 1662-4.

[4] M.W. Kirschner, “The meaning of systems biology.,” *Cell*, vol. 121, May. 2005, pp. 503-4.

[5] H.V. Westerhoff and B.O. Palsson, “The evolution of molecular biology into systems biology.,” *Nature biotechnology*, vol. 22, Oct. 2004, pp. 1249-52.

[6] B. Palsson, “In silico biology through "omics",” *Nat Biotech*, vol. 20, Jul. 2002, pp. 649-650.

[7] D. Noble, “The rise of computational biology.,” *Nature reviews. Molecular cell biology*, vol. 3, Jun. 2002, pp. 459-63.

[8] G. Karlebach and R. Shamir, “Modelling and analysis of gene regulatory networks.,” *Nature reviews. Molecular cell biology*, vol. 9, Oct. 2008, pp. 770-80.

[9] S. Hoops, S. Sahle, R. Gauges, C. Lee, J. Pahle, N. Simus, M. Singhal, L. Xu, P. Mendes, and U. Kummer, “COPASI—a COmplex PAthway SImulator,” *Bioinformatics*, vol. 22, 2006, pp. 3067-3074.

[10] N.A. Allen, L. Calzone, K.C. Chen, A. Ciliberto, N. Ramakrishnan, C.A. Shaffer, J.C. Sible, J.J. Tyson, M.T. Vass, L.T. Watson, and J.W. Zwolak, “Modeling Regulatory Networks at Virginia Tech,” *OMICS: A Journal of Integrative Biology*, vol. 7, 2003, pp. 285-299.

[11] XPP, “XPPAUT,” *http://www.math.pitt.edu/~bard/xpp/xpp.html*.

[12] B.M. Slepchenko and L.M. Loew, “Use of Virtual Cell in Studies of Cellular Dynamics,” K. Jeon, ed., Academic Press, 2010, pp. 1-56.

[13] M. Hucka, A. Finney, B.J. Bornstein, S.M. Keating, B.E. Shapiro, J. Matthews, B.L. Kovitz, M.J. Schilstra, A. Funahashi, J.C. Doyle, and H. Kitano, “Evolving a lingua franca and associated software infrastructure for computational systems biology : the Systems Biology Markup Language (SBML) project,” *Systems Biology*, vol. 1, 2004, pp. 41-53.

[14] A. Regev, E.M. Panina, W. Silverman, L. Cardelli, and E. Shapiro, “BioAmbients: an abstraction for biological compartments,” *Theoretical Computer Science*, vol. 325, Sep. 2004, pp. 141-167.

[15] N. Chabrier-Rivier, F. Fages, and S. Soliman, “The Biochemical Abstract Machine BIOCHAM,” *Computational Methods in Systems Biology*, V. Danos and V. Schachter, eds., Springer Berlin / Heidelberg, 2005, pp. 172-191.

[16] Z. Darzynkiewicz and X. Huang, *Analysis of Cellular DNA Content by Flow Cytometry*, John Wiley & Sons, Inc., 2001.

[17] S. Di Talia, J.M. Skotheim, J.M. Bean, E.D. Siggia, and F.R. Cross, “The effects of molecular noise and size control on variability in the budding yeast cell cycle.,” *Nature*, vol. 448, Aug. 2007, pp. 947-51.

[18] A. Csikász-Nagy, “Computational systems biology of the cell cycle.,” *Briefings in bioinformatics*, vol. 10, Jul. 2009, pp. 424-34.

[19] A. Eldar and M.B. Elowitz, “Functional roles for noise in genetic circuits,” *Nature*, vol. 467, Sep. 2010, pp. 167-173.

[20] H.H. McAdams and a Arkin, “It’s a noisy business! Genetic regulation at the nanomolar scale.,” *Trends in genetics : TIG*, vol. 15, Feb. 1999, pp. 65-9.

[21] E.M. Ozbudak, M. Thattai, I. Kurtser, A.D. Grossman, and A. van Oudenaarden, “Regulation of noise in the expression of a single gene.,” *Nature genetics*, vol. 31, May. 2002, pp. 69-73.

[22] C.V. Rao, D.M. Wolf, and A.P. Arkin, “Control, exploitation and tolerance of intracellular noise.,” *Nature*, vol. 420, Nov. 2002, pp. 231-7.

[23] R. Milner, *A Calculus of Communicating Systems*, Springer Verlag, 1980.

[24] C. Priami, A. Regev, E. Shapiro, and W. Silverman, “Application of a stochastic name-passing calculus to representation and simulation of molecular processes,” *Information Processing Letters*, vol. 80, 2001, pp. 25-31.

[25] L. Dematté, R. Larcher, A. Palmisano, C. Priami, and A. Romanel, “Programming Biology in BlenX,” *Systems Biology for Signaling Networks*, S. Choi, ed., Springer New York, 2010, pp. 777-820.

[26] C. Priami and P. Quaglia, “Operational Patterns in Beta-Binders,” *Transactions on Computational Systems Biology I*, C. Priami, ed., Springer Berlin / Heidelberg, 2005, p. 133.

[27] L. Dematté, C. Priami, and A. Romanel, “The Beta Workbench: a computational tool to study the dynamics of biological systems.,” *Briefings in bioinformatics*, vol. 9, Sep. 2008, pp. 437-49.

[28] J. Fisher and T.A. Henzinger, “Executable cell biology,” *Nat Biotech*, vol. 25, Nov. 2007, pp. 1239-1249.

[29] C. Priami, “Algorithmic systems biology,” *COmmun. ACM*, vol. 52, 2009, pp. 80-88.

[30] A. Murray and T. Hunt, *The cell cycle: An introduction*, OXFORD UNIVERSITY PRESS, NEW YORK, NY (USA)., 1993.

[31] C.S. Pittendrigh, “Circadian Rhythms and the Circadian Organization of Living Systems,” *Cold Spring Harb Symp Quant Biol*, vol. 25, 1960, pp. 159-184.

[32] C.P. Fall, E.S. Marland, J.M. Wagner, and J.J. Tyson, *Computational Cell Biology*, Springer, 2002.

[33] J.J. Tyson and B. Novák, “Functional motifs in biochemical reaction networks.,” *Annual review of physical chemistry*, vol. 61, Mar. 2010, pp. 219-40.

[34] J. Weiss, “The Hill equation revisited: uses and misuses.,” *The FASEB Journal*, vol. 11, 1997, pp. 835-841.

[35] M. Hucka, A. Finney, H.M. Sauro, H. Bolouri, J.C. Doyle, H. Kitano, and the rest of the S. Forum:, A.P. Arkin, B.J. Bornstein, D. Bray, A. Cornish-Bowden, A.A. Cuellar, S. Dronov, E.D. Gilles, M. Ginkel, V. Gor, I.I. Goryanin, W.J. Hedley, T.C. Hodgman, J.-H. Hofmeyr, P.J. Hunter, N.S. Juty, J.L. Kasberger, A. Kremling, U. Kummer, N. Le Novère, L.M. Loew, D. Lucio, P. Mendes, E. Minch, E.D. Mjolsness, Y. Nakayama, M.R. Nelson, P.F. Nielsen, T. Sakurada, J.C. Schaff, B.E. Shapiro, T.S. Shimizu, H.D. Spence, J. Stelling, K. Takahashi, M. Tomita, J. Wagner, and J. Wang, “The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models ,” *Bioinformatics* , vol. 19 , Mar. 2003, pp. 524-531.

[36] F. Ciocchetta and J. Hillston, “Process Algebras in Systems Biology,” *Formal Methods for Computational Systems Biology*, M. Bernardo, P. Degano, and G. Zavattaro, eds., Springer Berlin / Heidelberg, 2008, pp. 265-312.

[37] D.T. Gillespie, “Exact Stochastic Simulation of Coupled Chemical Reactions,” *The Journal of Physical Chemistry*, vol. 81, 1977, pp. 2340-2361.

[38] B. Novák and J.J. Tyson, “Design principles of biochemical oscillators.,” *Nature reviews. Molecular cell biology*, vol. 9, Dec. 2008, pp. 981-91.

[39] L. Bortolussi and A. Policriti, “Modeling Biological Systems in Stochastic Concurrent Constraint Programming,” *WCB2006*, 2008, pp. 66-90.

[40] R. Blossey, L. Cardelli, and A. Phillips, “Compositionality, stochasticity, and cooperativity in dynamic models of gene regulation.,” *HFSP journal*, vol. 2, Feb. 2008, pp. 17-28.

[41] A. Bockmayr and A. Courtois, “Constraint Programming to Model Dynamic Biological Systems,” 2002, pp. 85-99.

[42] E. De Maria, F. Fages, and S. Soliman, “On Coupling Models Using Model-Checking : Effects of Irinotecan Injections on the Mammalian Cell Cycle,” *CMSB2009, LNBI 5688*, Springer-Verlag Berlin Heidelberg, 2009, pp. 142-157.

[43] F. Ciocchetta and J. Hillston, “Bio-PEPA: A framework for the modelling and analysis of biological systems,” *Theoretical Computer Science*, vol. 410, Aug. 2009, pp. 3065-3084.

[44] U. Alon, “Network motifs: theory and experimental approaches,” *Nat Rev Genet*, vol. 8, Jun. 2007, pp. 450-461.

[45] H.M. Sauro and B.N. Kholodenko, “Quantitative analysis of signaling networks.,” *Progress in biophysics and molecular biology*, vol. 86, Sep. 2004, pp. 5-43.

[46] R.V. Kondratov and M.P. Antoch, “Circadian proteins in the regulation of cell cycle and genotoxic stress responses.,” *Trends in cell biology*, vol. 17, Jul. 2007, pp. 311-7.

[47] J. Zámborszky, C.I. Hong, and A. Csikász Nagy, “Computational analysis of mammalian cell division gated by a circadian clock: quantized cell cycles and cell size control.,” *Journal of biological rhythms*, vol. 22, Dec. 2007, pp. 542-53.

[48] C.I. Hong, J. Zámborszky, and A. Csikász-Nagy, “Minimum criteria for DNA damage-induced phase advances in circadian rhythms.,” *PLoS computational biology*, vol. 5, May. 2009, p. e1000384.

[49] A.M. Turing, “The chemical basis of morphogenesis,” *Philos. T. Roy. Soc. B.*, vol. 237, 1951, pp. 37-72.

[50] B.C. Goodwin, *Temporal Organization in Cells*, Academic Press, New York., 1963.

[51] H. de Jong, “Modeling and simulation of genetic regulatory systems: a literature review.,” *Journal of computational biology : a journal of computational molecular cell biology*, vol. 9, Jan. 2002, pp. 67-103.

[52] J. Monod and F. Jacob, “General conclusions: Teleonomic mechanisms in cellular metabolism, growth, and differentiation,” *Cold Spring Harbor Symp. Quant. Biol.*, vol. 26, 1961, p. 389–401.

[53] R. Thomas, *On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations (Numerical Methods in the Study of Critical Phenomena)*, Springer-Verlag, Berlin, 1981.

[54] J.S. Griffith, “Mathematics of cellular control processes I. Negative feedback to one gene,” *Journal of Theoretical Biology*, vol. 20, Aug. 1968, pp. 202-208.

[55] J.S. Griffith, “Mathematics of cellular control processes II. Positive feedback to one gene,” *Journal of Theoretical Biology*, vol. 20, Aug. 1968, pp. 209-216.

[56] J. Tyson, K.C. Chen, and B. Novak, “Sniffers, buzzers, toggles and blinkers: dynamics of regulatory and signaling pathways in the cell,” *Current Opinion in Cell Biology*, vol. 15, Apr. 2003, pp. 221-231.

[57] J.E. Ferrell, “Self-perpetuating states in signal transduction: positive feedback, double-negative feedback and bistability,” *Current Opinion in Cell Biology*, vol. 14, Apr. 2002, pp. 140-148.

[58] D. Thieffry and L. Sanchez, “Dynamical modelling of pattern formation during embryonic development,” *Current Opinion in Genetics & Development*, vol. 13, 2003, pp. 326-330.

[59] D. Noble, “From the Hodgkin-Huxley axon to the virtual heart.,” *The Journal of physiology*, vol. 580, Apr. 2007, pp. 15-22.

[60] A.M. Turing, “The chemical basis of morphogenesis,” *Phil. Trans. R. Soc. Lond. B*, vol. 327, 1952, pp. 37-72.

[61] H.G. Othmer and L.E. Scriven, “Instability and Dynamic Pattern in Cellular Networks,” *J. theor. Biol.*, vol. 32, 1971, pp. 507-537.

[62] S. Strogatz, *Nonlinear Dynamics And Chaos: With Applications To Physics, Biology, Chemistry, And Engineering (Studies in Nonlinearity) SE - Studies in nonlinearity*, Perseus Books Group, 1994.

[63] A. Zhabotinsky, “A history of chemical oscillations and waves.,” *Chaos*, vol. 1, 1991, pp. 379-386.

[64] D. Noble, “Modelling the heart: insights, failures and progress,” *BioEssays*, vol. 24, 2002, pp. 1155-1163.

[65] a L. Koch and M. Schaechter, “A model for statistics of the cell division process.,” *Journal of general microbiology*, vol. 29, Nov. 1962, pp. 435-54.

[66] E. Klipp, R. Herwig, A. Kowald, C. Wierling, and H. Lehrach, *Systems Biology in Practice: Concepts, Implementation and Application*, Wiley-VCH, 2005.

[67] H. Matsuno, A. Doi, M. Nagasaki, and S. Miyano, “HYBRID PETRI NET REPRESENTATION OF GENE REGULATORY NETWORK,” *5th Pacific Symposium on Biocomputing*, 2000, pp. 338-349.

[68] P.J. Goss and J. Peccoud, “Quantitative modeling of stochastic systems in molecular biology by using stochastic Petri nets.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 95, Jun. 1998, pp. 6750-5.

[69] J. Collado-Vides, “A transformational-grammar approach to the study of the regulation of gene expression,” *Journal of Theoretical Biology*, vol. 136, Feb. 1989, pp. 403-425.

[70] A. Regev and E. Shapiro, “Cells as computation,” *Nature*, vol. 419, 2002, p. 343.

[71] R. Thomas, “Boolean formalization of genetic control circuits,” *Journal of Theoretical Biology*, vol. 42, Dec. 1973, pp. 563-585.

[72] D. Thieffry, “Dynamical roles of biological regulatory circuits.,” *Briefings in bioinformatics*, vol. 8, Jul. 2007, pp. 220-5.

[73] M.I. Davidich and S. Bornholdt, “Boolean network model predicts cell cycle sequence of fission yeast.,” *PloS one*, vol. 3, Jan. 2008, p. e1672.

[74] A. Fauré, A. Naldi, C. Chaouiya, and D. Thieffry, “Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle.,” *Bioinformatics (Oxford, England)*, vol. 22, Jul. 2006, pp. e124-31.

[75] F. Li, T. Long, Y. Lu, Q. Ouyang, and C. Tang, “The yeast cell-cycle network is robustly designed.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, Apr. 2004, pp. 4781-6.

[76] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn, “Probabilistic model checking of complex biological pathways,” *Theoretical Computer Science*, vol. 391, Feb. 2008, pp. 239-257.

[77] P.T. Monteiro, D. Ropers, R. Mateescu, A.T. Freitas, and H. de Jong, “Temporal logic patterns for querying dynamic models of cellular interaction networks ,” *Bioinformatics* , vol. 24 , Aug. 2008, p. i227-i233.

[78] L.H. Hartwell, J.J. Hopfield, S. Leibler, and a W. Murray, “From molecular to modular cell biology.,” *Nature*, vol. 402, Dec. 1999, pp. C47-52.

[79] U. Alon, N. Barkai, D. a Notterman, K. Gish, S. Ybarra, D. Mack, and a J. Levine, “Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, Jun. 1999, pp. 6745-50.

[80] N. Friedman, M. Linial, I. Nachman, and D. Pe’er, “Using Bayesian networks to analyze expression data.,” *Journal of computational biology : a journal of computational molecular cell biology*, vol. 7, Jan. 2000, pp. 601-20.

[81] K. Murphy and S. Mian, “Modelling gene expression data using dynamic bayesian networks,” 1999.

[82] S. Kauffman, “Metabolic stability and epigenesis in randomly constructed genetic nets.,” *J Theor Biol*, vol. 22, 1969, pp. 437-67.

[83] M. Mitchell, J.P. Crutchfield, and R. Das, “Evolving Cellular Automata with Genetic Algorithms : A Review of Recent Work,” *Proceedings of the First International Conference on Evolutionary Computation and Its Applications (EvCA’96). Russian Academy of Sciences*, 1996.

[84] E.H. Snoussi and R. Thomas, “Logical identification of all steady states: The concept of feedback loop characteristic states,” *Bulletin of Mathematical Biology*, vol. 55, Sep. 1993, pp. 973-991.

[85] R. Thomas, D. Thieffry, and M. Kaufman, “Dynamical behaviour of biological regulatory networks—I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state,” *Bulletin of Mathematical Biology*, vol. 57, pp. 247-276.

[86] R. Thomas and R. D’Ari, “Biological Feedback,” 1990.

[87] V.N. Reddy, M.N. Liebman, and M.L. Mavrovouniotis, “Qualitative analysis of biochemical reaction systems,” *Computers in Biology and Medicine*, vol. 26, Jan. 1996, pp. 9-24.

[88] V.N. Reddy, M.L. Mavrovouniotis, and M.N. Liebman, “Petri net representations in metabolic pathways.,” *Proceedings of the 1st International Conference on Intelligent Systems for Molecular Biology*, vol. 1, Jan. 1993, pp. 328-36.

[89] H. Genrich, R. Küffner, and K. Voss, “Executable Petri net models for the analysis of metabolic pathways,” *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 3, pp. 394-404.

[90] M. Heiner, D. Gilbert, and R. Donaldson, “Petri Nets for Systems and Synthetic Biology,” *Formal Methods for Computational Systems Biology*, M. Bernardo, P. Degano, and G. Zavattaro, eds., Springer Berlin / Heidelberg, 2008, pp. 215-264.

[91] L.J. Steggles, R. Banks, O. Shaw, and A. Wipat, “Qualitatively modelling and analysing genetic regulatory networks: a Petri net approach.,” *Bioinformatics (Oxford, England)*, vol. 23, Feb. 2007, pp. 336-43.

[92] I. Mura and A. Csikász-Nagy, “Stochastic Petri Net extension of a yeast cell cycle model.,” *Journal of theoretical biology*, vol. 254, Oct. 2008, pp. 850-60.

[93] C. Chaouiya, E. Remy, and D. Thieffry, “Petri net modelling of biological regulatory networks,” *Journal of Discrete Algorithms*, vol. 6, Jun. 2008, pp. 165-177.

[94] K. Jensen, “A brief introduction to coloured Petri Nets,” *Tools and Algorithms for the Construction and Analysis of Systems*, E. Brinksma, ed., Springer Berlin / Heidelberg, 1997, pp. 203-208.

[95] S. Christensen, “Modular Analysis of Petri Nets,” *The Computer Journal*, vol. 43, Mar. 2000, pp. 224-242.

[96] D.A. Zaitsev, “Decomposition of Petri nets,” *Cybernetics and Systems Analysis*, vol. 40, 2004, pp. 739-746.

[97] V. Sassone, “The Algebraic Structure of Petri Nets,” 2000.

[98] W. Hlavacek, J. Faeder, M. Blinov, R. Posner, M. Hucka, and W. Fontana, “Rules for modeling signal-transduction systems.,” *Sci STKE*, vol. 344, 2006.

[99] A. Mallavarapu, M. Thomson, B. Ullian, and J. Gunawardena, “Programming with models: modularity and abstraction provide powerful capabilities for systems biology.,” *Journal of the Royal Society, Interface / the Royal Society*, vol. 6, Mar. 2009, pp. 257-70.

[100] C.A.R. Hoare, “Communicating sequential processes,” *Comm. ACM*, vol. 21, 1978, p. 666–677.

[101] G.D. Plotkin, *A Structural Approach to Operational Semantics. Technical Report DAIMI FN-19*, 1981.

[102] J.R. Norris, *MArkov chains*, Cambridge University Press, 1997.

[103] R. Milner, *Communicating and Mobile Systems: the Pi-Calculus*, Cambridge University Press, 1999.

[104] M. Kwiatkowski and I. Stark, "The Continuous pi-Calculus: A Process Algebra for Biochemical Modelling," *Computational Methods in Systems Biology*, M. Heiner and A. Uhrmacher, eds., Springer Berlin / Heidelberg, 2008, pp. 103-122.

[105] C. Priami, "Stochastic pi-Calculus," *The Computer Journal*, vol. 38, 1995, p. 578–589.

[106] A. Phillips and L. Cardelli, "Efficient , Correct Simulation of Biological Processes in the Stochastic pi-calculus," *CMSB*, Springer, 2001, p. 184.

[107] J. Hillston, *A Compositional Approach to Performance Modelling*, Cambridge University Press: Cambridge University Press, 1996.

[108] M. Kwiatkowska, G. Norman, and D. Parker, "Using probabilistic model checking in systems biology," *ACM SIGMETRICS Performance Evaluation Review*, vol. 35, Mar. 2008, p. 14.

[109] F. Ciocchetta, S. Gilmore, M.L. Guerriero, and J. Hillston, "Integrated Simulation and Model-Checking for the Analysis of Biochemical Systems," *Electr. Notes Theor. Comput. Sci.*, vol. 232, 2009, pp. 17-38.

[110] V. Danos and J. Krivine, "Formal Molecular Biology Done in CCS-R," *Electr. Notes Theor. Comput. Sci.*, vol. 180, 2007, p. 31–49.

[111] A.D. Gordon and L. Cardelli, "Mobile ambients," *Theor. Comput. Sci.*, vol. 240, 2000, p. 177– 213.

[112] L. Cardelli, "Brane Calculi," *Computational Methods in Systems Biology (Lecture Notes in Computer Science)*, Springer, 2005, p. 257–280.

[113] V. Danos and S. Pradalier, "Projective Brane Calculus," *In CMSB, volume 3082 of Lecture Notes in Computer Science*, Springer, 2004, p. 134–148.

[114] C. Priami and P. Quaglia, "Beta Binders for Biological Interactions," *Computational Methods in Systems Biology*, V. Danos and V. Schachter, eds., Springer Berlin / Heidelberg, 2005, pp. 20-33.

[115] P. Degano, D. Prandi, C. Priami, and P. Quaglia, "Beta-binders for Biological Quantitative Experiments," *Electronic Notes in Theoretical Computer Science*, vol. 164, Oct. 2006, pp. 101-117.

[116] D. Prandi, C. Priami, and P. Quaglia, "Communicating by compatibility☆," *Journal of Logic and Algebraic Programming*, vol. 75, Apr. 2008, pp. 167-181.

[117] L. Dematte, C. Priami, and A. Romanel, *The BlenX Language : A Tutorial*, 2008.

[118] Z.G. Szabo, "Compositionality," *Stanford Encyclopedia of Philosophy*, 2007.

[119] P.P. Nayak and L. Joskowicz, “Efficient compositional modeling for generating causal explanations,” *Artificial Intelligence*, vol. 83, 1996, pp. 193-227.

[120] B. Falkenhainer and K.D. Forbus, “Cmpositional modeling: finding the right model for the job,” *Artificial Intelligence*, vol. 51, Oct. 1991, pp. 95-143.

[121] J. Saez-Rodriguez, a Kremling, H. Conzelmann, K. Bettenbrock, and E.D. Gilles, “Modular analysis of signal transduction networks - How engineering tools can be applied to the analysis of cellular machinery,” *IEEE Control Systems Magazine*, vol. 24, Aug. 2004, pp. 35-52.

[122] M. a Marchisio and J. Stelling, “Computational design of synthetic gene circuits with composable parts.,” *Bioinformatics (Oxford, England)*, vol. 24, Sep. 2008, pp. 1903-10.

[123] A. Dräger, N. Hassis, J. Supper, A. Schröder, and A. Zell, “SBMLsqueezer: a CellDesigner plug-in to generate kinetic rate equations for biochemical networks.,” *BMC systems biology*, vol. 2, Jan. 2008, p. 39.

[124] M. Pedersen and G.D. Plotkin, “A Language for Biochemical Systems : Design and Formal Specification,” *Computational Methods in Systems Biology*, 2008, pp. 63-82.

[125] L. Bortolussi and A. Policriti, “Stochastic Concurrent Constraint Programming and Differential Equations,” *Electronic Notes in Theoretical Computer Science*, vol. 190, Sep. 2007, pp. 27-42.

[126] J. Hillston, “Fluid flow approximation of PEPA models,” *Quantitative Evaluation of Systems, 2005. Second International Conference*, 2005, pp. 33-42.

[127] M. Calder, S. Gilmore, and J. Hillston, “Automatically deriving ODEs from process algebra models of signalling pathways,” 2005.

[128] S. Gilmore, J. Hillston, and M. Ribaudo, “An efficient algorithm for aggregating PEPA models,” *IEEE Transactions on Software Engineering*, vol. 27, May. 2001, pp. 449-464.

[129] A. Bockmayr and A. Courtois, “Using Hybrid Concurrent Constraint Programming to Model Dynamic Biological Systems,” *Logic Programming*, P. Stuckey, ed., Springer Berlin / Heidelberg, 2002, pp. 411-418.

[130] H. Hermanns, U. Herzog, V. Mertsiotakis, and M. Rettelbach, “Exploiting stochastic process algebra achievements for generalized stochastic Petri nets,” *Proceedings of the Seventh International Workshop on Petri Nets and Performance Models*, 1997, pp. 183-192.

[131] W.J. Bosl, “Systems biology by the rules: hybrid intelligent systems for pathway modeling and discovery.,” *BMC systems biology*, vol. 1, Jan. 2007, p. 13.

[132] A. Ciliberto, F. Capuani, and J.J. Tyson, “Modeling networks of coupled enzymatic reactions using the total quasi-steady state approximation.,” *PLoS computational biology*, vol. 3, Mar. 2007, p. e45.

[133] L. Jategaonkar Jagadeesan and R. Jagadeesan, “Causality and true concurrency: A data-flow analysis of the Pi-Calculus,” *Algebraic Methodology and Software Technology*, V. Alagar and M. Nivat, eds., Springer Berlin / Heidelberg, 1995, pp. 277-291.

[134] J.E. Ferrell Jr., “The Biochemical Basis of an All-or-None Cell Fate Switch in Xenopus Oocytes,” *Science*, vol. 280, May. 1998, pp. 895-898.

[135] A. Novick and M. Weiner, “Enzyme induction as an all-or-none phenomenon,” *Proc. Natl. Acad. Sci.*, vol. 43, 1957, pp. 553-566.

[136] I. Mura, *Exactness and Approximation of the Stochastic Simulation Algorithm Technical Report*, 2008.

[137] I. Mura, D. Prandi, C. Priami, and a Romanel, “Exploiting non-Markovian Bio-Processes,” *Electronic Notes in Theoretical Computer Science*, vol. 253, Nov. 2009, pp. 83-98.

[138] A.C. Shaffer, J.W. Zwolak, R. Randhawa, and J.J. Tyson, “Modeling Molecular Regulatory Networks with JigCell and PET,” *Methods Mol Biol*, Springer, 2009, pp. 1-30.

[139] Z. Zi and E. Klipp, “SBML-PET: a Systems Biology Markup Language-based parameter estimation tool.,” *Bioinformatics (Oxford, England)*, vol. 22, Nov. 2006, pp. 2704-5.

[140] P. Lecca, A. Palmisano, A. Ihekwaba, and C. Priami, “Calibration of dynamic models of biological systems with KInfer,” *European Biophysics Journal*, vol. 39, pp. 1019-1039.

[141] L. Michaelis and M.L. Menten, “{Die kinetik der invertinwirkung},” *Biochem. Z*, vol. 49, 1913, p. 352.

[142] A.V. Hill, “The possible effects of the aggregation of the molecules of haemoglobin on its dissociation curves,” *J Physiol*, vol. 40, 1910, pp. 389-403.

[143] U. Alon, *An Introduction to Systems Biology: Design Principles of Biological Circuits*, Chapman & Hall/CRC, 2006.

[144] N. Wilhelmová, “Cornish-Bowden, A.: Fundamentals of Enzyme Kinetics,” *Biologia Plantarum*, vol. 38, 1995, p. 430.

[145] B.N. Kholodenko, “Cell-signalling dynamics in time and space.,” *Nature reviews. Molecular cell biology*, vol. 7, Mar. 2006, pp. 165-76.

[146] L. Hartwell and T. Weinert, “Checkpoints: controls that ensure the order of cell cycle events,” *Science*, vol. 246, 1989, pp. 629-34.

[147] a Zetterberg, O. Larsson, and K.G. Wiman, “What is the restriction point?,” *Current opinion in cell biology*, vol. 7, Dec. 1995, pp. 835-42.

[148] K. Nasmyth, “At the heart of the budding yeast cell cycle,” *Trends in Genetics*, vol. 12, Oct. 1996, pp. 405-412.

[149] S. López-Avilés, O. Kapuy, B. Novák, and F. Uhlmann, “Irreversibility of mitotic exit is the consequence of systems-level feedback.,” *Nature*, vol. 459, May. 2009, pp. 592-5.

[150] K. Nasmyth, “Separating sister chromatids,” *Trends in Biochemical Sciences*, vol. 24, Mar. 1999, pp. 98-104.

[151] P. Nurse, “Universal control mechanism regulating onset of M-phase,” *Nature*, vol. 344, Apr. 1990, pp. 503-508.

[152] L.H. Hartwell, R.K. Mortimer, J. Culotti, and M. Culotti, “Genetic Control of the Cell Division Cycle in Yeast: V. Genetic Analysis of cdc Mutants.,” *Genetics*, vol. 74, Jun. 1973, pp. 267-86.

[153] P. Nurse, “Genetic control of cell size at cell division in yeast.,” *Nature*, vol. 256, 1975, pp. 547-51.

[154] T. Hunt, “Maturation promoting factor, cyclin and the control of M-phase,” *Curr Opin Cell Biol.*, vol. 1, 1989, pp. 268-74.

[155] J. Zámborszky and A. Csikász-Nagy, “Modeling the cell cycle,” *UNESCO Encyclopedia of Life Support Systems*, A. De Gaetano and P. Palumbo, eds.

[156] A.T. Winfree, “The prehistory of the Belousov-Zhabotinsky oscillator,” *Journal of Chemical Education*, vol. 61, Aug. 1984, p. 661.

[157] A. Goldbeter, “Computational approaches to cellular rhythms,” *Nature*, vol. 420, Nov. 2002, pp. 238-245.

[158] B.D. Aguda, “A quantitative analysis of the kinetics of the G2 DNA damage checkpoint system.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 96, Sep. 1999, pp. 11352-7.

[159] Z. Qu, W.R. MacLellan, and J.N. Weiss, “Dynamics of the cell cycle: checkpoints, sizers, and timers.,” *Biophysical journal*, vol. 85, Dec. 2003, pp. 3600-11.

[160] B. Novák and J.J. Tyson, “A model for restriction point control of the mammalian cell cycle.,” *Journal of theoretical biology*, vol. 230, Oct. 2004, pp. 563-79.

[161] A.B. Pardee, “A restriction point for control of normal animal cell proliferation,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 71, Apr. 1974, pp. 1286-90.

[162] K.C. Chen, a Csikasz-Nagy, B. Gyorffy, J. Val, B. Novak, and J.J. Tyson, “Kinetic analysis of a molecular model of the budding yeast cell cycle.,” *Molecular biology of the cell*, vol. 11, Jan. 2000, pp. 369-91.

[163] J.E. Toettcher, A. Loewer, G.J. Ostheimer, M.B. Yaffe, B. Tidor, and G. Lahav, “Distinct mechanisms act in concert to mediate cell cycle arrest.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, Jan. 2009, pp. 785-90.

[164] A. Csikász-Nagy, D. Battogtokh, K.C. Chen, B. Novák, and J.J. Tyson, “Analysis of a generic model of eukaryotic cell-cycle regulation.,” *Biophysical journal*, vol. 90, Jun. 2006, pp. 4361-79.

[165] B. Pfeuty, T. David-Pfeuty, and K. Kaneko, “Underlying principles of cell fate determination during G1 phase of the mammalian cell cycle,” *Cell Cycle*, vol. 7, 2008, pp. 3246-3257.

[166] R. Pepperkok and J. Ellenberg, “High-throughput fluorescence microscopy for systems biology,” *Nature Reviews Molecular Cell Biology*, vol. 7, 2006, pp. 690-696.

[167] R. Steuer, “Effects of stochasticity in models of the cell cycle: from quantized cycle times to noise-induced oscillations.,” *Journal of theoretical biology*, vol. 228, Jun. 2004, pp. 293-301.

[168] M. Sabouri-Ghomi, A. Ciliberto, S. Kar, B. Novak, and J.J. Tyson, “Antagonism and bistability in protein interaction networks.,” *Journal of theoretical biology*, vol. 250, Jan. 2008, pp. 209-18.

[169] Y. Zhang, M. Qian, Q. Ouyang, M. Deng, F. Li, and C. Tang, “Stochastic model of yeast cell-cycle network,” *Physica D: Nonlinear Phenomena*, vol. 219, Jul. 2006, pp. 35-39.

[170] L. Calzone and S. Soliman, “Coupling the Cell cycle and the Circadian Cycle,” *INRIA*, vol. 5835, 2006.

[171] J. Dunlap, “Molecular Bases for Circadian Clocks,” *Cell*, vol. 96, Jan. 1999, pp. 271-290.

[172] C.S. Pittendrigh, “Temporal organization: reflections of a Darwinian clock-watcher.,” *Annual review of physiology*, vol. 55, Jan. 1993, pp. 16-54.

[173] J.-C. Leloup and A. Goldbeter, “Modeling the circadian clock: from molecular mechanism to physiological disorders.,” *BioEssays : news and reviews in molecular, cellular and developmental biology*, vol. 30, Jun. 2008, pp. 590-600.

[174] S. Sahar and P. Sassone-corsi, “Circadian Clock and Breast Canceer,” *Cell Cycle*, vol. 6, 2007, pp. 1329-1331.

[175] L. Fu, H. Pelicano, J. Liu, P. Huang, and C. Lee, “The circadian gene Period2 plays an important role in tumor suppression and DNA damage response in vivo.,” *Cell*, vol. 111, Oct. 2002, pp. 41-50.

[176] A.J. Davidson, M.T. Sellix, D.S. Yamazaki, M. Menaker, and G.D. Block, “Chronic jet-lag increases mortality in aged mice,” *Current Biology*, vol. 16, 2006, p. R914-R916.

[177] E. Filipski, F. Delaunay, V.M. King, M.-W. Wu, B. Claustrat, A. Gréchez-Cassiau, C. Guettier, M.H. Hastings, and L. Francis, “Effects of chronic jet lag on tumor progression in mice.,” *Cancer research*, vol. 64, Nov. 2004, pp. 7879-85.

[178] T. Roenneberg and M. Merrow, “The network of time: understanding the molecular circadian system.,” *Current biology : CB*, vol. 13, Mar. 2003, pp. R198-207.

[179] A. Balsalobre, F. Damiola, and U. Schibler, “A serum shock induces circadian gene expression in mammalian tissue culture cells.,” *Cell*, vol. 93, Jun. 1998, pp. 929-37.

[180] C.S. Pittendrigh, *Circadian Clocks: What are They? (The Molecular Basis Of Circadian Rhythms)*, Berlin: Abakon Verlagsgesellschaft, 1975.

[181] T. Roenneberg and M. Merrow, “Circadian clocks - the fall and rise of physiology,” *Nature Review Molecular Cell Biology*, vol. 6, 2005, pp. 965-971.

[182] C.R. McClung, “Plant circadian rhythms.,” *The Plant cell*, vol. 18, Apr. 2006, pp. 792-803.

[183] T. Roenneberg and M. Merrow, “"What watch?...such much!" Complexity and evolution of circadian clocks.,” *Cell and tissue research*, vol. 309, Jul. 2002, pp. 3-9.

[184] D.B. Forger and C.S. Peskin, “Model based conjectures on mammalian clock controversies.,” *Journal of theoretical biology*, vol. 230, Oct. 2004, pp. 533-9.

[185] D. Gonze, J. Halloy, and A. Goldbeter, “Robustness of circadian rhythms with respect to molecular noise.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, Jan. 2002, pp. 673-8.

[186] C.I. Hong, E.D. Conrad, and J.J. Tyson, “A proposal for robust temperature compensation of circadian rhythms.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 104, Jan. 2007, pp. 1195-200.

[187] R.J. Konopka and S. Benzer, “Clock mutants of Drosophila melanogaster.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 68, Sep. 1971, pp. 2112-6.

[188] J.F. Feldman and M.N. Hoyle, “Isolation of circadian clock mutants of Neurospora crassa.,” *Genetics*, vol. 75, Dec. 1973, pp. 605-13.

[189] H.R. Ueda, S. Hayashi, W. Chen, M. Sano, M. Machida, Y. Shigeyoshi, M. Iino, and S. Hashimoto, “System-level identification of transcriptional circuits underlying mammalian circadian clocks.,” *Nature genetics*, vol. 37, Mar. 2005, pp. 187-92.

[190] N. Preitner, F. Damiola, J. Zakany, D. Duboule, U. Albrecht, and U. Schibler, “The Orphan Nuclear Receptor REV-ERB a Controls Circadian Transcription within the Positive Limb of the Mammalian Circadian Oscillator,” *Cell*, vol. 110, 2002, pp. 251-260.

[191] T.K. Sato, S. Panda, L.J. Miraglia, T.M. Reyes, R.D. Rudic, P. McNamara, K. a Naik, G. a FitzGerald, S. a Kay, and J.B. Hogenesch, “A functional genomics strategy reveals Rora as a component of the mammalian circadian clock.,” *Neuron*, vol. 43, Aug. 2004, pp. 527-37.

[192] C.H. Johnson, M. Egli, and P.L. Stewart, “Structural Insights into a Circadian Oscillator,” *Science*, vol. 322, Oct. 2008, pp. 697-701.

[193] A. Mehra, C.L. Baker, J.J. Loros, and J.C. Dunlap, “Post-translational modifications in circadian rhythms,” *Trends in Biochemical Sciences*, vol. 34, Oct. 2009, pp. 483-490.

[194] M. Doi, J. Hirayama, and P. Sassone-Corsi, “Circadian Regulator CLOCK Is a Histone Acetyltransferase,” *Cell*, vol. 125, May. 2006, pp. 497-508.

[195] J. Hirayama, S. Sahar, B. Grimaldi, T. Tamaru, K. Takamatsu, Y. Nakahata, and P. Sassone-Corsi, “CLOCK-mediated acetylation of BMAL1 controls circadian function,” *Nature*, vol. 450, Dec. 2007, pp. 1086-1090.

[196] Y. Nakahata, M. Kaluzova, B. Grimaldi, S. Sahar, J. Hirayama, D. Chen, L.P. Guarente, and P. Sassone-Corsi, “The NAD+-Dependent Deacetylase SIRT1 Modulates CLOCK-Mediated Chromatin Remodeling and Circadian Control,” *Cell*, vol. 134, Jul. 2008, pp. 329-340.

[197] L. Cardone, J. Hirayama, F. Giordano, T. Tamaru, J.J. Palvimo, and P. Sassone-Corsi, “Circadian Clock Control by SUMOylation of BMAL1,” *Science*, vol. 309, Aug. 2005, pp. 1390-1394.

[198] T. Roenneberg, E.J. Chua, R. Bernardo, and E. Mendoza, “Modelling biological rhythms.,” *Current biology*, vol. 18, Sep. 2008, p. R826-R835.

[199] G. a Carpenter and S. Grossberg, “A neural theory of circadian rhythms: Aschoff’s rule in diurnal and nocturnal mammals.,” *The American journal of physiology*, vol. 247, Dec. 1984, pp. R1067-82.

[200] R.E. Kronauer, D.B. Forger, and M.E. Jewett, “Quantifying human circadian pacemaker response to brief, extended, and repeated light stimuli over the phototopic range,” *Journal of Biological Rhythms*, vol. 14, 1999, pp. 500-515.

[201] C.S. Pittendrigh and S. Daan, “A functional analysis of circadian pacemakers in nocturnal rodents,” *Journal of Comparative Physiology A: Neuroethology, Sensory, Neural, and Behavioral Physiology*, vol. 106, 1976, pp. 291-331.

[202] S. Becker-Weimann, J. Wolf, H. Herzel, and A. Kramer, “Modeling feedback loops of the Mammalian circadian oscillator.,” *Biophysical journal*, vol. 87, Nov. 2004, pp. 3023-34.

[203] M. a Lema, D. a Golombek, and J. Echave, “Delay model of the circadian pacemaker.,” *Journal of theoretical biology*, vol. 204, Jun. 2000, pp. 565-73.

[204] J.M.G. Vilar, H.Y. Kueh, N. Barkai, and S. Leibler, “Mechanisms of noise-resistance in genetic oscillators.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 99, Apr. 2002, pp. 5988-92.

[205] B.C. Goodwin, “Oscillatory behavior in enzymatic control processes,” *Adv. Enzyme Regul.*, vol. 3, 1965, p. 425–438.

[206] P. Ruoff, M. Vinsjevik, C. Monnerjahn, and L. Rensing, “The Goodwin model: simulating the effect of light pulses on the circadian sporulation rhythm of Neurospora crassa.,” *Journal of theoretical biology*, vol. 209, Mar. 2001, pp. 29-42.

[207] A.T. Winfree, *The geometry of biological time*, Springer-Verlag, New York, 1980.

[208] M.E. Jewett, D.B. Forger, and R.E. Kronauer, “Revised Limit Cycle Oscillator Model of Human Circadian Pacemaker,” *Journal of Biological Rhythms*, vol. 14, Dec. 1999, pp. 493-500.

[209] A. Goldbeter, “A Model for Circadian Oscillations in the Drosophila Period Protein (PER),” *Proc. R. Soc. Lond. B*, vol. 261, 1995, pp. :319-324.

[210] S. Becker-Weimann, J. Wolf, H. Herzel, and A. Kramer, “Modeling Feedback Loops of the Mammalian Circadian Oscillator,” *Biophysical journal*, vol. 87, Nov. 2004, pp. 3023-3034.

[211] J.-C. Leloup and A. Goldbeter, “Toward a detailed computational model for the mammalian circadian clock,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, Jun. 2003, pp. 7051-6.

[212] D.B. Forger, D.A.D. Ii, K. Gurdziel, J.-christophe Leloup, C. Lee, C.V.O.N. Gall, J.-pierre Etchegaray, R.E. Kronauer, A. Goldbeter, C.S. Peskin, M.E. Jewett, and D.R. Weaver, “Mammalian Circadian Oscillators,” *OMICS: A Journal of Integrative Biology*, vol. 7, 2003, pp. 387-400.

[213] D.B. Forger and C.S. Peskin, “Stochastic simulation of the mammalian circadian clock.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, Jan. 2005, pp. 321-4.

[214] M.H. Hastings and E.D. Herzog, “Clock genes, oscillators, and cellular networks in the suprachiasmatic nuclei.,” *Journal of biological rhythms*, vol. 19, Oct. 2004, pp. 400-13.

[215] J.-C. Leloup and A. Goldbeter, “Modeling the mammalian circadian clock: sensitivity analysis and multiplicity of oscillatory mechanisms,” *Journal of theoretical biology*, vol. 230, Oct. 2004, pp. 541-62.

[216] M. Gallego, E.J. Eide, M.F. Woolf, D.M. Virshup, and D.B. Forger, “An opposite role for tau in circadian rhythms revealed by mathematical modeling.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 103, Jul. 2006, pp. 10618-23.

[217] D. Gonze, S. Bernard, C. Waltermann, A. Kramer, and H. Herzel, “Spontaneous synchronization of coupled circadian oscillators.,” *Biophysical journal*, vol. 89, Jul. 2005, pp. 120-9.

[218] Q. Yang, B.F. Pando, G. Dong, S.S. Golden, and A. van Oudenaarden, “Circadian gating of the cell cycle revealed in single cyanobacterial cells.,” *Science (New York, N.Y.)*, vol. 327, Mar. 2010, pp. 1522-6.

[219] J. Bradley, “A Ticking Clock: Performance Analysis of a Circadian Rhythm with Stochastic Process Algebra,” *Computer Performance Engineering*, N. Thomas and C. Juiz, eds., Springer Berlin / Heidelberg, 2008, pp. 79-94.

[220] O.E. Akman, F. Ciocchetta, A. Degasperi, and M.L. Guerriero, “Modelling Biological Clocks with Bio-PEPA: Stochasticity and Robustness for the Neurospora crassa Circadian Network,” *CMSB ’09: Proceedings of the 7th International Conference on Computational Methods in Systems Biology*, 2009, pp. 52-67.

[221] M.B. Kastan and J. Bartek, “Cell-cycle checkpoints and cancer.,” *Nature*, vol. 432, Nov. 2004, pp. 316-23.

[222] L.N. Edmunds, D.L. Laval-Martin, and K. Goto, “Cell division cycles and circadian clocks. Modeling a metabolic oscillator in the algal flagellate Euglena.,” *Annals of the New York Academy of Sciences*, vol. 503, Jan. 1987, pp. 459-75.

[223] K. Homma and J.W. Hastings, “Cell Cycle Synchronization of Gonyaulax polyedra by Filtration: Quantized Generation Times,” *J Biol Rhythms*, vol. 3, 1988, pp. 49-58.

[224] T. Mori, B. Binder, and C.H. Johnson, “Circadian gating of cell division in cyanobacteria growing with average doubling times of less than 24 hours.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 93, Sep. 1996, pp. 10183-8.

[225] T. Matsuo, S. Yamaguchi, S. Mitsui, A. Emi, F. Shimoda, and H. Okamura, “Control mechanism of the circadian clock for timing of cell division in vivo.,” *Science (New York, N.Y.)*, vol. 302, Oct. 2003, pp. 255-9.

[226] A. Altinok, F. Lévi, and A. Goldbeter, “Identifying mechanisms of chronotolerance and chronoefficacy for the anticancer drugs 5-fluorouracil and oxaliplatin by computational modeling.,” *European journal of pharmaceutical sciences : official journal of the European Federation for Pharmaceutical Sciences*, vol. 36, Jan. 2009, pp. 20-38.

[227] B. Kang, Y.-Y. Li, X. Chang, L. Liu, and Y.-X. Li, “Modeling the effects of cell cycle M-phase transcriptional inhibition on circadian oscillation.,” *PLoS computational biology*, vol. 4, Mar. 2008, p. e1000019.

[228] A.M. Pregueiro, Q. Liu, C.L. Baker, J.C. Dunlap, and J.J. Loros, “The Neurospora checkpoint kinase 2: a regulatory link between the circadian and cell cycles.,” *Science (New York, N.Y.)*, vol. 313, Aug. 2006, pp. 644-9.

[229] M. Oklejewicz, E. Destici, F. Tamanini, R. a Hut, R. Janssens, and G.T.J. van der Horst, “Phase resetting of the mammalian circadian clock by DNA damage.,” *Current biology : CB*, vol. 18, Feb. 2008, pp. 286-91.

[230] J. Saez-Rodriguez, S. Gayer, M. Ginkel, and E.D. Gilles, “Automatic decomposition of kinetic models of signaling networks minimizing the retroactivity among modules.,” *Bioinformatics (Oxford, England)*, vol. 24, Aug. 2008, pp. i213-9.

[231] M. a Marchisio and J. Stelling, “Computational design of synthetic gene circuits with composable parts.,” *Bioinformatics (Oxford, England)*, vol. 24, Sep. 2008, pp. 1903-10.

[232] C.V. Rao and A.P. Arkin, “Stochastic chemical kinetics and the quasi-steady-state assumption: Application to the Gillespie algorithm,” *The Journal of Chemical Physics*, vol. 118, 2003, p. 4999.

[233] F. Ciocchetta, “The BlenX Language with Biological Transactions,” *Transactions on Computational Systems Biology IX*, C. Priami, ed., Springer Berlin / Heidelberg, 2008, pp. 114-152.

[234] F. Ciocchetta and J. Hillston, “Bio-PEPA: An Extension of the Process Algebra PEPA for Biochemical Networks,” *Electronic Notes in Theoretical Computer Science*, vol. 194, Jan. 2008, pp. 103-117.

[235] F. Fages, S. Soliman, and N. Chabrier-Rivier, “Modelling and querying interaction networks in the biochemical abstract machine BIOCHAM,” *Journal of Biological Physics and Chemistry*, vol. 4, 2004, pp. 64-73.

[236] S. Mangan and U. Alon, “Structure and function of the feed-forward loop network motif.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, Oct. 2003, pp. 11980-5.

[237] A. Csikász-Nagy, O. Kapuy, A. Tóth, C. Pál, L.J. Jensen, F. Uhlmann, J.J. Tyson, and B. Novák, “Cell cycle regulation by feed-forward loops coupling transcription and phosphorylation.,” *Molecular systems biology*, vol. 5, Jan. 2009, p. 236.

[238] R. Blossey, L. Cardelli, and A. Phillips, “A Compositional Approach to the Stochastic Dynamics of Gene Networks,” *Transactions on Computational Systems Biology IV*, C. Priami, L. Cardelli, and S. Emmott, eds., Springer Berlin / Heidelberg, 2006, pp. 99-122.

[239] A. Phillips and L. Cardelli, “A programming language for composable DNA circuits.,” *Journal of the Royal Society, Interface / the Royal Society*, vol. 6 Suppl 4, Aug. 2009, pp. S419-36.

[240] L. Bortolussi and A. Policriti, “Hybrid Systems and Biology,” *Formal Methods for Computational Systems Biology*, M. Bernardo, P. Degano, and G. Zavattaro, eds., Springer Berlin / Heidelberg, 2008, pp. 424-448.

[241] A. Goldbeter and D.E. Koshland, “An amplified sensitivity arising from covalent modification in biological systems,” *Proc. Natl. Acad. Sci. USA*, vol. 78, 1981, pp. 6840-6844.

[242] A. Goldbeter, “A minimal cascade model for the mitotic oscillator involving cyclin and cdc2 kinase.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 88, Oct. 1991, pp. 9107-11.

[243] B.N. Kholodenko, “Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades.,” *European journal of biochemistry / FEBS*, vol. 267, Mar. 2000, pp. 1583-8.

[244] N. Blüthgen, F.J. Bruggeman, S. Legewie, H. Herzel, H.V. Westerhoff, and B.N. Kholodenko, “Effects of sequestration on signal transduction cascades.,” *The FEBS journal*, vol. 273, Mar. 2006, pp. 895-906.

[245] O.G. Berg, J. Paulsson, and M. Ehrenberg, “Fluctuations and quality of control in biological cells: zero-order ultrasensitivity reinvestigated.,” *Biophysical journal*, vol. 79, Sep. 2000, pp. 1228-36.

[246] a Palmisano, I. Mura, and C. Priami, “From ODES to language-based, executable models of biological systems.,” *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, Jan. 2009, pp. 239-50.

[247] F. Ciocchetta, “Bio-PEPA: An Extension of the Process Algebra PEPA for Biochemical Networks,” *Electronic Notes in Theoretical Computer Science*, vol. 194, Jan. 2008, pp. 103-117.

[248] R. Bundschuh, F. Hayot, and C. Jayaprakash, “Fluctuations and slow variables in genetic networks.,” *Biophysical journal*, vol. 84, Mar. 2003, pp. 1606-15.

[249] J. Paulsson, O.G. Berg, and M. Ehrenberg, “Stochastic focusing: fluctuation-enhanced sensitivity of intracellular regulation.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 97, Jun. 2000, pp. 7148-53.

[250] L. Dematté, C. Priami, and A. Romanel, “The Beta Workbench: a computational tool to study the dynamics of biological systems.,” *Briefings in bioinformatics*, vol. 9, Sep. 2008, pp. 437-49.

[251] L. Cardelli, “Artificial Biochemistry,” *Algorithmic Bioprocesses*, A. Condon, D. Harel, J.N. Kok, A. Salomaa, and E. Winfree, eds., Springer Berlin Heidelberg, 2009, pp. 429-462.

[252] C. Alexander, “The origins of pattern theory: the future of the theory, and the generation of a living world,” *IEEE Software*, vol. 16, 1999, pp. 71-82.

[253] B. Sevella, *Biomérnöki Műveletek és Folyamatok*, Műegyetemi kiadó, 1998.

[254] L. Matyska and J. Kovár, “Comparison of several non-linear-regression methods for fitting the Michaelis-Menten equation.,” *The Biochemical journal*, vol. 231, Oct. 1985, pp. 171-7.

[255] “CoSBi Lab,” *http://www.cosbi.eu/index.php/research/prototypes/overview*, 2009.

[256] C. Priami, P. Ballarini, and P. Quaglia, “BlenX4Bio - BlenX for Biologists,” *Computational Methods in Systems Biology*, P. Degano and R. Gorrieri, eds., Springer Berlin / Heidelberg, 2009, pp. 26-51.

[257] D.E. Koshland, G. Némethy, and D. Filmer, “Comparison of Experimental Binding Data and Theoretical Models in Proteins Containing Subunits,” *Biochemistry*, vol. 5, Jan. 1966, pp. 365-385.

[258] J. Monod, J. Wyman, and J. Changeux, “ON THE NATURE OF ALLOSTERIC TRANSITIONS: A PLAUSIBLE MODEL.,” *J Mol Biol.*, vol. 12, 1965, pp. 88-118.

[259] J.M. Pedraza and J. Paulsson, “Effects of molecular memory and bursting on fluctuations in gene expression.,” *Science (New York, N.Y.)*, vol. 319, Jan. 2008, pp. 339-43.

[260] F. Toledo and G.M. Wahl, “Regulating the p53 pathway: in vitro hypotheses, in vivo veritas.,” *Nature reviews. Cancer*, vol. 6, Dec. 2006, pp. 909-23.

[261] T.K. Kerppola and T. Curran, “Fos-Jun heterodimers and jun homodimers bend DNA in opposite orientations: Implications for transcription factor cooperativity,” *Cell*, vol. 66, Jul. 1991, pp. 317-326.

[262] L. Cai, C.K. Dalal, and M.B. Elowitz, “Frequency-modulated nuclear localization bursts coordinate gene regulation.,” *Nature*, vol. 455, Sep. 2008, pp. 485-90.

[263] A. Csikász-Nagy and I. Mura, “Role of mRNA Gestation and Senescence in Noise Reduction during the Cell Cycle,” *In Silico Biology*, vol. 10, Jan. 2010, pp. 81-88.

[264] P. Cohen, “The regulation of protein function by multisite phosphorylation - a 25 year update,” *Trends in Biochemical Sciences*, vol. 25, Dec. 2000, pp. 596-601.

[265] O.S. Soyer, H. Kuwahara, and A. Csikász-Nagy, “Regulating the total level of a signaling protein can vary its dynamics in a range from switch like ultrasensitivity to adaptive responses,” *FEBS Journal*, vol. 276, 2009, pp. 3290-3298.

[266] E.R. Stadtman and P.B. Chock, “Superiority of interconvertible enzyme cascades in metabolic regulation: analysis of monocyclic systems.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 74, Jul. 1977, pp. 2761-5.

[267] J. Xing and J. Chen, “The Goldbeter-Koshland switch in the first-order region and its response to dynamic disorder.,” *PloS one*, vol. 3, Jan. 2008, p. e2140.

[268] R. Thomas, “Laws for the dynamics of regulatory networks.,” *The International journal of developmental biology*, vol. 42, Jan. 1998, pp. 479-85.

[269] E.M. Ozbudak, M. Thattai, H.N. Lim, B.I. Shraiman, and A. van Oudenaarden, “Multistability in the lactose utilization network of Escherichia coli,” *Nature*, vol. 427, Feb. 2004, pp. 737-740.

[270] T.S. Gardner, C.R. Cantor, and J.J. Collins, “Construction of a genetic toggle switch in Escherichia coli,” *Nature*, vol. 403, Jan. 2000, pp. 339-342.

[271] H. Bolouri and E.H. Davidson, “Modeling transcriptional regulatory networks.,” *BioEssays : news and reviews in molecular, cellular and developmental biology*, vol. 24, Dec. 2002, pp. 1118-29.

[272] T. Zhang, P. Brazhnik, and J.J. Tyson, “Computational analysis of dynamical responses to the intrinsic pathway of programmed cell death.,” *Biophysical journal*, vol. 97, Jul. 2009, pp. 415-34.

[273] B. Novak and J.J. Tyson, “Numerical analysis of a comprehensive model of M-phase control in Xenopus oocyte extracts and intact embryos.,” *Journal of cell science*, vol. 106 (Pt 4, Dec. 1993, pp. 1153-68.

[274] J.J. Tyson and B. Novak, “Regulation of the eukaryotic cell cycle: molecular antagonism, hysteresis, and irreversible transitions.,” *Journal of theoretical biology*, vol. 210, May. 2001, pp. 249-63.

[275] W. Sha, J. Moore, K. Chen, A.D. Lassaletta, C.-S. Yi, J.J. Tyson, and J.C. Sible, “Hysteresis drives cell-cycle transitions in Xenopus laevis egg extracts.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, Feb. 2003, pp. 975-80.

[276] J.R. Pomerening, E.D. Sontag, and J.E. Ferrell, “Building a cell cycle oscillator: hysteresis and bistability in the activation of Cdc2.,” *Nature cell biology*, vol. 5, Apr. 2003, pp. 346-51.

[277] M. Laurent and N. Kellershohn, “Multistability: a major means of differentiation and evolution in biological systems,” *Trends in Biochemical Sciences*, vol. 24, Nov. 1999, pp. 418-422.

[278] A. Ciliberto, B. Novak, and J.J. Tyson, “Steady states and oscillations in the p53/Mdm2 network,” *Cell Cycle*, vol. 4, 2005, pp. 488-493.

[279] D.O. Morgan, “Principles of CDK regulation,” *Nature*, vol. 374, Mar. 1995, pp. 131-134.

[280] O. Shoval and U. Alon, “SnapShot: Network Motifs,” *Cell*, vol. 143, Oct. 2010, pp. 326.e1-326.e2.

[281] A. Csikász-Nagy and O.S. Soyer, “Adaptive dynamics with a single two-state protein.,” *Journal of the Royal Society, Interface / the Royal Society*, vol. 5 Suppl 1, Aug. 2008, pp. S41-7.

[282] J. Zámborszky and C. Priami, “BlenX-based compositional modeling of complex reaction mechanisms,” *Electronic Proceedings in Theoretical Computer Science*, vol. 19, Feb. 2010, pp. 85-102.

[283] J. Tyson, C.I. Hong, D. Thron, and B. Novak, “A Simple Model of Circadian Rhythms Based on Dimerization and Proteolysis of PER and TIM,” *Biophysical Journal*, vol. 77, Nov. 1999, pp. 2411-2417.

[284] N.G. van Kampen, *Stochastic processes in physics and chemistry*, North-Holland Publishing Co., Amsterdam, 1981.

[285] J.J. Tyson, “The coordination of cell growth and division — intentional or Incidental?,” *BioEssays*, vol. 2, 1985, pp. 72-77.

[286] E. Nagoshi, C. Saini, C. Bauer, T. Laroche, F. Naef, and U. Schibler, “Circadian Gene Expression in Individual Fibroblasts: Cell-Autonomous and Self-Sustained Oscillators Pass Time to Daughter Cells,” *Cell*, vol. 119, Nov. 2004, pp. 693-705.

[287] R.R. Klevecz, “Quantized generation time in mammalian cells as an expression of the cellular clock.,” *Proceedings of the National Academy of Sciences of the United States of America*, vol. 73, Nov. 1976, pp. 4012-6.

[288] A. Csikász-Nagy, A. Faure, R. Larcher, P. Lecca, I. Mura, F. Jordan, A. Palmisano, A. Romanel, S. Sedwards, H. Siebert, S. Soliman, D. Thieffry, J. Zámborszky, T. Mazza, and

P. Ballarini, “Analyzing various models of Circadian Clock and Cell Cycle coupling,” 2009.

[289] M.P. Myers, K. Wager-Smith, A. Rothenfluh-Hilfiker, and M.W. Young, “Light-Induced Degradation of TIMELESS and Entrainment of the Drosophila Circadian Clock,” *Science*, vol. 271, Mar. 1996, pp. 1736-1740.

[290] M. Izumo, T.R. Sato, M. Straume, and C.H. Johnson, “Quantitative Analyses of Circadian Gene Expression in Mammalian Cell Cultures,” *PLoS Comput Biol*, vol. 2, Oct. 2006, p. e136.

[291] S. Gery, N. Komatsu, L. Baldjyan, A. Yu, D. Koo, and H.P. Koeffler, “The circadian gene per1 plays an important role in cell growth and DNA damage control in human cancer cells.,” *Molecular cell*, vol. 22, May. 2006, pp. 375-82.

[292] B. Novak, J.J. Tyson, B. Gyorffy, and A. Csikasz-Nagy, “Irreversible cell-cycle transitions are due to systems-level feedback.,” *Nature cell biology*, vol. 9, Jul. 2007, pp. 724-8.

[293] T.Y.-C. Tsai, Y.S. Choi, W. Ma, J.R. Pomerening, C. Tang, and J.E. Ferrell, “Robust, tunable biological oscillations from interlinked positive and negative feedback loops.,” *Science (New York, N.Y.)*, vol. 321, Jul. 2008, pp. 126-9.

[294] H.W. Ko, J. Jiang, and I. Edery, “Role for Slimb in the degradation of Drosophila Period protein phosphorylated by Doubletime,” *Nature*, vol. 420, Dec. 2002, pp. 673-678.

[295] B. Kloss, J.L. Price, L. Saez, J. Blau, A. Rothenfluh, C.S. Wesley, and M.W. Young, “The Drosophila Clock Gene double-time Encodes a Protein Closely Related to Human Casein Kinase I epsilon,” *Cell*, vol. 94, Jul. 1998, pp. 97-107.

[296] Z.J. Huang, K.D. Curtin, and M. Rosbash, “PER protein interactions and temperature compensation of a circadian clock in Drosophila,” *Science*, vol. 267, Feb. 1995, pp. 1169-1172.

[297] U. Knippschild, A. Gocht, S. Wolff, N. Huber, J. Lohler, and M. Stoter, “The casein kinase 1 family: participation in multiple cellular processes in eukaryotes,” *Cellular Signalling*, vol. 17, Jun. 2005, pp. 675-689.

[298] J.E. Galagan, S.E. Calvo, K.A. Borkovich, E.U. Selker, N.D. Read, D. Jaffe, W. FitzHugh, L.-J. Ma, S. Smirnov, S. Purcell, B. Rehman, T. Elkins, R. Engels, S. Wang, C.B. Nielsen, J. Butler, M. Endrizzi, D. Qui, P. Ianakiev, D. Bell-Pedersen, M.A. Nelson, M. Werner-Washburne, C.P. Selitrennikoff, J.A. Kinsey, E.L. Braun, A. Zelter, U. Schulte, G.O. Kothe, G. Jedd, W. Mewes, C. Staben, E. Marcotte, D. Greenberg, A. Roy, K. Foley, J. Naylor, N. Stange-Thomann, R. Barrett, S. Gnerre, M. Kamal, M. Kamvysselis, E. Mauceli, C. Bielke, S. Rudd, D. Frishman, S. Krystofova, C. Rasmussen, R.L. Metzenberg, D.D. Perkins, S. Kroken, C. Cogoni, G. Macino, D. Catcheside, W. Li, R.J. Pratt, S.A. Osmani, C.P.C. DeSouza, L. Glass, M.J. Orbach, J.A. Berglund, R. Voelker, O. Yarden, M. Plamann, S. Seiler, J. Dunlap, A. Radford, R. Aramayo, D.O. Natvig, L.A. Alex, G. Mannhaupt, D.J. Ebbole, M. Freitag, I. Paulsen, M.S. Sachs, E.S. Lander, C. Nusbaum, and B. Birren, “The genome sequence of the filamentous fungus Neurospora crassa,” *Nature*, vol. 422, Apr. 2003, pp. 859-868.

[299] "BROAD Institute," *http://www.broadinstitute.org/annotation/genome/neurospora/MultiHome.html*.

[300] D. Bell-Pedersen, S.K. Crosthwaite, P.L. Lakin-Thomas, M. Merrow, and M. Økland, "The Neurospora circadian clock: simple or complex?," *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, vol. 356, Nov. 2001, pp. 1697-709.

[301] P. Ballario, P. Vittorioso, a Magrelli, C. Talora, a Cabibbo, and G. Macino, "White collar-1, a central regulator of blue light responses in Neurospora, is a zinc finger protein.," *The EMBO journal*, vol. 15, Apr. 1996, pp. 1650-7.

[302] H. Linden and G. Macino, "White collar 2, a partner in blue-light signal transduction, controlling expression of light-regulated genes in Neurospora crassa," *EMBO J*, vol. 16, Jan. 1997, pp. 98-109.

[303] P. Cheng, Q. He, Q. He, L. Wang, and Y. Liu, "Regulation of the Neurospora circadian clock by an RNA helicase," *Genes & Development*, vol. 19, Jan. 2005, pp. 234-241.

[304] C. Heintzen and Y. Liu, "The Neurospora crassa Circadian Clock," J.C.H.B.T.-A. in Genetics, ed., Academic Press, 2007, pp. 25-66.

[305] Q. He, P. Cheng, Y. Yang, Q. He, H. Yu, and Y. Liu, "FWD1-mediated degradation of FREQUENCY in Neurospora establishes a conserved mechanism for circadian clock regulation.," *The EMBO journal*, vol. 22, Sep. 2003, pp. 4421-30.

[306] H. Ma, S. Kunes, P.J. Schatz, and D. Botstein, "Plasmid construction by homologous recombination in yeast," *Gene*, vol. 58, 1987, pp. 201-216.

[307] N. Bardiya and P.K.T. Shiu, "Cyclosporin A-resistance based gene placement system for Neurospora crassa.," *Fungal genetics and biology : FG & B*, vol. 44, May. 2007, pp. 307-14.

[308] V.D. Gooch, A. Mehra, L.F. Larrondo, J. Fox, M. Touroutoutoudis, J.J. Loros, and J.C. Dunlap, "Fully codon-optimized luciferase uncovers novel temperature characteristics of the Neurospora clock.," *Eukaryotic cell*, vol. 7, Jan. 2008, pp. 28-37.

[309] L.M. Loew and J.C. Schaff, "The Virtual Cell: a software environment for computational cell biology," *Trends in Biotechnology*, vol. 19, Oct. 2001, pp. 401-406.

[310] H. Kitano, A. Funahashi, Y. Matsuoka, and K. Oda, "Using process diagrams for the graphical representation of biological networks," *Nat Biotech*, vol. 23, Aug. 2005, pp. 961-966.

[311] N.L. Novere, M. Hucka, H. Mi, S. Moodie, F. Schreiber, A. Sorokin, E. Demir, K. Wegner, M.I. Aladjem, S.M. Wimalaratne, F.T. Bergman, R. Gauges, P. Ghazal, H. Kawaji, L. Li, Y. Matsuoka, A. Villeger, S.E. Boyd, L. Calzone, M. Courtot, U. Dogrusoz, T.C. Freeman, A. Funahashi, S. Ghosh, A. Jouraku, S. Kim, F. Kolpakov, A. Luna, S. Sahle, E. Schmidt, S. Watterson, G. Wu, I. Goryanin, D.B. Kell, C. Sander, H. Sauro, J.L. Snoep,

K. Kohn, and H. Kitano, “The Systems Biology Graphical Notation,” *Nat Biotech*, vol. 27, Aug. 2009, pp. 735-741.

[312] L. Sun and A. Becskei, “Systems biology: The cost of feedback control,” *Nature*, vol. 467, Sep. 2010, pp. 163-164.

[313] E. Andrianantoandro, S. Basu, D.K. Karig, and R. Weiss, “Synthetic biology: new engineering rules for an emerging discipline,” *Mol Syst Biol*, vol. 2, May. 2006.

Acknowledgement

First of all, I wish to thank my supervisor, Corrado Priami, his support. I am heartily grateful to my co-advisor, Attila Csikász-Nagy, whose expertise and encouragement have been guided me from the very beginning of my scientific studies. I would like to thank my collaborator, Chris Hong (University of Cincinnati), all the exciting and fruitful discussions on the research on circadian rhythms. I am grateful to all my friends who have always stood by me and my colleagues in The Microsoft Research - University of Trento Centre for Computational and Systems Biology who have helped to make everyday life as a PhD candidate more fun. This thesis would not have been possible unless the support of my family and their faith in me. Finally, I am particularly grateful to my husband for his imperishable patience and his love.

Appendix A: Template library for BlenX

a) Enzyme catalytic reaction (Michaelis-Menten kinetics)

program file:

```
[ steps = …, delta = … ]
<<   BASERATE:inf,   CHANGE:inf >>

///*************** species ******************
/// PRODUCT:
let P : bproc = #(s:0,P) [ nil ];

/// SUBSTRATE:
let S : bproc = #(s:0,S) [ s!().ch(s,P).nil ];

/// ENZYME:
let E1 : bproc = #(e1:0,E1) [ rep e1?().nil ];

///*************** initial ******************
run … E1 || … S || … P
```

type file:

```
{ S,P,E1 }
%%{  (S,E1,rate(k1),rate(k1r),rate(k2)),
  (P,E1,0,inf,0)}
```

b) transcription module (Hill function)

program file:

```
[ steps = …, delta = … ]
<< BASERATE:inf, CHANGE:inf >>

////////.............. TRANSCRIPTION ...................
///*************** species ******************
            /// gene:
let G       : bproc = #(g:0,G)  [ nil ];
            /// transcription factor:
let TF           : bproc = #(tf,TF)  [ nil ];
            /// TF dimer:
let TF2     : bproc = #(t2f,TF2)     [ nil ];
            /// G-TF-TF trimer:
let GTF2    : bproc = #(gtf2,GTF2)   [ nil ];
            /// transcript (messenger RNA):
let M       : bproc = #(m:0,M)  [ nil ];

///*************** reactions ****************
       ///dimerization:
when(TF,TF::rate(k1)) join(TF2);
       ///decomplexation:
when(TF2::rate(k2)) split(TF,TF);
       ///'trimerization':
when(TF2,G::rate(k3)) join(GTF2);
       ///decomplexation:
when(GTF2::rate(k4)) split(G,TF2);
       ///transcriptional activation:
when(GTF2::rate(kms)) split(GTF2,M);
       /// degradation of messenger:
when(M::rate(kmd)) delete(1);
```

```
///*************** initial ******************
run … M || … TF || 1 G ||  0 GTF2 || 0 TF2
```

type file:

```
{ G, TF, TF2, GTF2, M } %% {}
```

c) Goldbeter-Koshland ultrasensitive switch

program file:

```
[ steps = …, delta = … ]
<< BASERATE:inf, HIDE:inf, UNHIDE:inf, CHANGE:inf >>

///*************** species ******************
       ///two forms of the protein:
let P :              bproc = #(p:0,P)         [ nil ];
let Pmod :       bproc = #(p:0,Pmod)     [ nil ];
       ///enzymes:
let E1 :             bproc = #(e1:0,E1)       [ nil ];
let E2 :             bproc = #(e2:0,E2)       [ nil ];
       ///intermediate complexes:
let E1_P :       bproc = #(e1p:0,E1P)    [ nil ];
let E2_Pmod :        bproc = #(e2pm:0,E2Pm)   [ nil ];

///*************** reactions ****************
when   (E1,P::rate(e1_1))      join(E1_P);
when   (E1_P::rate(e1_2))      split(E1,P);
when   (E1_P::rate(e1_3))      split(E1,Pmod);
when   (Pmod,E2::rate(e2_1))   join(E2_Pmod);
when   (E2_Pmod::rate(e2_2))   split(E2,Pmod);
when   (E2_Pmod::rate(e2_3))   split(E2,P);
```

```
///*************** initial *****************
run … Pmod || … E1 || … E2 || … P
```

type file:

```
{ P, E1, E2, Pmod, E1P, E2Pm }
%%{}
```

d) Mutual antagonism

program file:

```
[ steps = …, delta = … ]
<<   BASERATE:inf,  HIDE:inf,  UNHIDE:inf,  CHANGE:inf >>

///*************** processes ****************
let inactivate_ENZ :
pproc = ( if (substrate, bound) then hide(enzyme).nil endif
);

let inactivate_SUB :
pproc = ( if (enzyme, bound) then hide(substrate).nil endif
);

let activate_ENZ :
pproc = (if (not(substrate, bound)) then un-
hide(enzyme).rec_enzyme!().nil endif );

let activate_SUB : pproc = (if (not(enzyme, bound)) then un-
hide(substrate).rec_substrate!().nil endif );

let activate_ENZ_X :
pproc = (if (not(substrate, bound) and not(substrate_YPX,
bound)) then unhide(enzyme).rec_ENZ!().nil endif);
```

```
let activate_SUB_X :
pproc = (if (not(enzyme, bound) and not(substrate_YPX,
bound)) then unhide(substrate).rec_SUB!().nil endif);
let activate_ENZ_X_YP :
pproc = (if (not(enzyme, bound) and not(substrate, bound))
then unhide(substrate_YPX).rec_ENZ_X!().nil endif);

let inactivate_ENZ_X :
pproc = ( if ((substrate, bound) or (substrate_YPX, bound))
then hide(enzyme).nil endif );

let inactivate_SUB_X :
pproc = ( if ((enzyme, bound) or (substrate_YPX, bound)) then
hide(substrate).nil endif );

let inactivate_ENZ_X_YP :
pproc = ( if ((enzyme, bound) or (substrate, bound)) then
hide(substrate_YPX).nil endif );

///*************** species ******************
/// SIGNAL
let S : bproc = #(s:0,S)  [ rep s?().nil ];
/// enzyme
let A : bproc = #(a:0,A)  [ rep a?().nil ];

/// X
let X : bproc =  #(substrate:0,X_sub),
                 #(enzyme:0,X_enz),
                 #(substrate_YPX:0,X)
  [    inactivate_ENZ_X | activate_ENZ_X |
       rep rec_ENZ?().(inactivate_ENZ_X | activate_ENZ_X ) |
       inactivate_SUB_X | activate_SUB_X |
       rep rec_SUB?().(inactivate_SUB_X | activate_SUB_X ) |
          inactivate_ENZ_X_YP | activate_ENZ_X_YP |
          rep rec_ENZ_X?().(inactivate_ENZ_X_YP |
          activate_ENZ_X_YP ) |
          rep enzyme?().nil |
          substrate!().ch(substrate, YX).nil |
          substrate_YPX!().ch(substrate_YPX, YPX).nil  ];

/// Y
let Y : bproc =  #(substrate:0,Y_sub),
                 #(enzyme:0,Y_enz)
  [     inactivate_ENZ | activate_ENZ |
        rep rec_enzyme?().(inactivate_ENZ | activate_ENZ ) |
        inactivate_SUB | activate_SUB |
        rep rec_substrate?().(inactivate_SUB | activate_SUB ) |
        rep enzyme?().nil |
        substrate!().ch(substrate, XY).nil ];

/// XP
let XP : bproc =     #(substrate:0,XP_sub),
                 #(enzyme:0,XP_enz)
  [     inactivate_ENZ | activate_ENZ |
        rep rec_enzyme?().(inactivate_ENZ | activate_ENZ ) |
        inactivate_SUB | activate_SUB |
        rep rec_substrate?().(inactivate_SUB | activate_SUB ) |
        rep enzyme?().nil |
        substrate!().ch(substrate, XP).nil ];

/// YP
let YP : bproc =     #(substrate:0,YP_sub),
                 #(enzyme:0,YP_enz)
  [     inactivate_ENZ | activate_ENZ |
        rep rec_enzyme?().(inactivate_ENZ | activate_ENZ ) |
  inactivate_SUB | activate_SUB |
  rep rec_substrate?().(inactivate_SUB | activate_SUB ) |
             rep enzyme?().nil |
        substrate!().ch(substrate, AYP).nil ];

/// temporary species
let X_YX : bproc =    #(substrate:0,YX),
               #(enzyme:0,X_enz),
               #(substrate_YPX:0,X)
  [    inactivate_ENZ_X | activate_ENZ_X |
       rep rec_ENZ?().(inactivate_ENZ_X | activate_ENZ_X ) |
       inactivate_SUB_X | activate_SUB_X |
       rep rec_SUB?().(inactivate_SUB_X | activate_SUB_X ) |
            inactivate_ENZ_X_YP | activate_ENZ_X_YP |
       rep rec_ENZ_X?().(inactivate_ENZ_X_YP |
       activate_ENZ_X_YP ) |
       rep enzyme?().nil |
       substrate_YPX!().ch(substrate_YPX, YPX).nil  ];

let Y_XY : bproc =    #(substrate:0,XY),
               #(enzyme:0,Y_enz)
  [    inactivate_ENZ | activate_ENZ |
       rep rec_enzyme?().(inactivate_ENZ | activate_ENZ ) |
       inactivate_SUB | activate_SUB |
       rep rec_substrate?().(inactivate_SUB | activate_SUB ) |
       rep enzyme?().nil ];

let YP_AYP : bproc =  #(substrate:0,AYP),
                      #(enzyme:0,YP_enz)
  [    inactivate_ENZ | activate_ENZ |
       rep rec_enzyme?().(inactivate_ENZ | activate_ENZ ) |
       inactivate_SUB | activate_SUB |
       rep rec_substrate?().(inactivate_SUB | activate_SUB ) |
       rep enzyme?().nil ];

let X_YPX : bproc =   #(substrate:0,X_sub),
                     #(enzyme:0,X_enz),
                     #(substrate_YPX:0,YPX)
  [    inactivate_ENZ_X | activate_ENZ_X |
       rep rec_ENZ?().(inactivate_ENZ_X | activate_ENZ_X ) |
       inactivate_SUB_X | activate_SUB_X |
       rep rec_SUB?().(inactivate_SUB_X | activate_SUB_X ) |
            inactivate_ENZ_X_YP | activate_ENZ_X_YP |
       rep rec_ENZ_X?().(inactivate_ENZ_X_YP |
       activate_ENZ_X_YP ) |
       rep enzyme?().nil |
       substrate!().ch(substrate, YX).nil ];

///*************** reactions ****************
when(Y_XY::inf) split(YP,Nil);
when(X_YX::inf) delete(1);
when(YP_AYP::inf) split(Y,Nil);
when(X_YPX::inf) delete(1);
when(S::rate(k1)) split(S,X);
when(X::rate(k2p)) delete(1);

///*************** initial *****************
///run … S || … X || … Y || … YP || … A || … YP_AYP || … Y_XY
|| … X_YX || … X_YPX
```

type file:

```
{S, A, X_sub, X_enz, X, Y_sub, Y_enz, XP_enz, XP_sub, YP_sub,
YP_enz, YX, XY, AYP, YPX, XP }
%%
{
  (X_sub, Y_enz, rate(k2f), rate(k2r), rate(k2)),
  (YX, Y_enz, 0, inf, 0),
  (YP_sub, A, rate(k3f), rate(k3r), rate(k3)),
  (AYP, A, 0, inf, 0),
  (X_enz, Y_sub, rate(k4f), rate(k4r), rate(k4)),
  (X_enz, XY, 0, inf, 0),
  (YP_enz, X, rate(k5f), rate(k5r), rate(k5)),
  (YP_enz, YPX, 0, inf, 0)
}
```

e) Mutual activation

```
[ steps=…, delta=… ]
<<   BASERATE:inf,  HIDE:inf,  UNHIDE:inf,  CHANGE:inf >>

let activate_ENZ :
pproc = (if (not(substrate, bound)) then un-
hide(enzyme).rec_ENZ!().nil endif );

let activate_SUB :
pproc = (if (not(enzyme, bound)) then un-
hide(substrate).rec_SUB!().nil endif );

let inactivate_ENZ :
pproc = ( if (substrate, bound) then hide(enzyme).nil endif
);

let inactivate_SUB :
pproc = ( if (enzyme, bound) then hide(substrate).nil endif
);

let CYCB : bproc =    #(substrate:0,CYCB_sub),
                      #(enzyme:0,CYCB_enz_CDC25)
  [    inactivate_ENZ | activate_ENZ |
       rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
       inactivate_SUB | activate_SUB |
       rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
       rep enzyme?().nil |
       substrate!().ch(substrate, CYCB_sub_mod).nil     ];


let CYCB_TEMP : bproc =   #(substrate:0,CYCB_sub_mod),
                          #(enzyme:0,CYCB_enz_CDC25)
[      inactivate_ENZ | activate_ENZ |
       rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
       inactivate_SUB | activate_SUB |
       rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
       rep enzyme?().nil                 ];

let CYCBP : bproc = #(substrate:0,CYCBP_sub)
       [ substrate!().ch(substrate, CYCBP_sub_mod).nil ];

let CYCBP_TEMP : bproc = #(substrate:0,CYCBP_sub_mod)
       [ nil ];

let WEE1 : bproc = #(enzyme:0,WEE1_enz)
       [ rep enzyme?().nil ];

let CDC25 :     bproc = #(substrate:0,CDC25_sub)
       [ substrate!().ch(substrate, CDC25_sub_mod).nil  ];

let CDC25_TEMP : bproc = #(substrate:0,CDC25_sub_mod)
       [ nil ];

let CDC25P : bproc = #(enzyme:0,CDC25P_enz),
                     #(substrate:0,CDC25P_sub)
       [    inactivate_ENZ | activate_ENZ |
            rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
            inactivate_SUB | activate_SUB |
            rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
            rep enzyme?().nil |
            substrate!().ch(substrate, CDC25P_sub_mod).nil
        ];

let CDC25P_TEMP : bproc =      #(enzyme:0,CDC25P_enz),
                               #(substrate:0,CDC25P_sub_mod)
[       inactivate_ENZ | activate_ENZ |
        rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
        inactivate_SUB | activate_SUB |
        rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
        rep enzyme?().nil                               ];

let ENZYME1 : bproc = #(enzyme:0,ENZYME1_enz)
  [ rep enzyme?().nil ];

let SIGNAL :     bproc = #(s:0,S)    [ nil ];

when(CYCB_TEMP::inf) split(CYCBP, Nil);
when(CDC25P_TEMP::inf) split(CDC25,Nil);
when(CYCBP_TEMP::inf) split(CYCB,Nil);
when(CDC25_TEMP::inf) split(CDC25P,Nil);
when(SIGNAL::rate(s)) split(SIGNAL,CYCB);
when(CYCB::rate(deg)) delete(1);
```

type file:

```
{ CYCB_sub, CYCB_sub_mod, CYCB_enz_CDC25,
  CDC25P_enz, CDC25P_sub, CDC25P_sub_mod,
  ENZYME1_enz,
  CDC25_sub, CDC25_sub_mod,
  CYCBP_sub, CYCBP_sub_mod      }
%%
{ (CYCB_enz_CDC25, CDC25_sub, rate(k1), rate(k1r), rate(k2)),
  (CYCB_enz_CDC25, CDC25_sub_mod, 0, inf , 0 ),
  (ENZYME1_enz, CDC25P_sub, rate(l1), rate(l1r), rate(l2)),
  (ENZYME1_enz, CDC25P_sub_mod, 0, inf , 0),
  (CDC25P_enz, CDC25P_sub,rate(m1), rate(m1r), rate(m2) ),
  (CDC25P_enz, CYCBP_sub_mod, 0, inf , 0 ),
  (WEE1_enz, CYCB_sub, rate(n1), rate(n1r),rate(n2)),
  (WEE1_enz, CYCB_sub_mod, 0, inf , 0)    }
```

f) G2/M transition

```
let activate_ENZ :
     pproc = (if (not(substrate, bound)) then un-
hide(enzyme).rec_ENZ!().nil endif );

let activate_SUB :
     pproc = (if (not(enzyme, bound)) then un-
hide(substrate).rec_SUB!().nil endif );

let inactivate_ENZ :
     pproc = ( if (substrate, bound) then hide(enzyme).nil
endif );

let inactivate_SUB :
     pproc = ( if (enzyme, bound) then hide(substrate).nil
endif );

let activate_ENZ_CYCB :
     pproc = (if (not(substrate, bound) and
     not(enzyme_CYCBCDC25, bound))
     then unhide(enzyme).rec_ENZ!().nil endif);

let activate_SUB_CYCB :
     pproc = (if (not(enzyme, bound) and
     not(enzyme_ CYCBCDC25, bound))
     then unhide(substrate).rec_SUB!().nil endif);
```

```
let activate_ENZ_CYCB_CDC25 :
     pproc = (if (not(enzyme, bound) and
     not(substrate, bound))
     then unhide(enzyme_CYCBCDC25).rec_ENZ_CDC25!().nil endif);

let inactivate_ENZ_CYCB :
     pproc = ( if ((substrate, bound) or (enzyme_CYCBCDC25,
bound))
      then hide(enzyme).nil endif );

let inactivate_SUB_CYCB :
     pproc = ( if ((enzyme, bound) or (enzyme_ CYCBCDC25,
bound))
     then hide(substrate).nil endif );

let inactivate_ENZ_CYCB_CDC25 :
     pproc = ( if ((enzyme, bound) or (substrate, bound))
     then hide(enzyme_ CYCBCDC25).nil endif );

let CYCB : bproc =    #(enzyme:0,CYCB_enz),
  #(substrate:0,CYCB_sub),
    #(enzyme_ CYCBCDC25:0,CYCB_enz_CDC25)
  [    inactivate_ENZ_CYCB | activate_ENZ_CYCB |
 rep rec_ENZ?().(inactivate_ENZ_CYCB |
 activate_ENZ_CYCB ) |
       inactivate_SUB_CYCB | activate_SUB_CYCB |
     rep rec_SUB?().(inactivate_SUB_CYCB |
     activate_SUB_CYCB ) |
       inactivate_ENZ_CYCB_CDC25 | activate_ENZ_CYCB_CDC25 |
     rep rec_ENZ_CDC25?().(inactivate_ENZ_CYCB_CDC25 |
     activate_ENZ_CYCB_CDC25 ) |
       rep enzyme?().nil |
       substrate!().ch(substrate, CYCB_sub_mod).nil |
       rep enzyme_ CYCBCDC25?().nil                 ];

let CYCB_TEMP : bproc =    #(enzyme:0,CYCB_enz),
     #(substrate:0,CYCB_sub_mod),
     #(enzyme_ CYCBCDC25:0,CYCB_enz_CDC25)
     [      inactivate_ENZ_CYCB | activate_ENZ_CYCB |
     rep rec_ENZ?().(inactivate_ENZ_CYCB |
     activate_ENZ_CYCB ) |
            inactivate_SUB_CYCB | activate_SUB_CYCB |
     rep rec_SUB?().(inactivate_SUB_CYCB |
     activate_SUB_CYCB ) |
            inactivate_ENZ_CYCB_CDC25 |
 activate_ENZ_CYCB_CDC25 |
     rep rec_ENZ_CDC25?().(inactivate_ENZ_CYCB_CDC25 |
     activate_ENZ_CYCB_CDC25 ) |
            rep enzyme?().nil |
            rep enzyme_ CYCBCDC25?().nil                 ];

let CYCBP : bproc = #(substrate:0,CYCBP_sub)
     [ substrate!().ch(substrate, CYCBP_sub_mod).nil ];

let CYCBP_TEMP : bproc = #(substrate:0,CYCBP_sub_mod)
 [ nil ];

let WEE1 : bproc = #(enzyme:0,WEE1_enz), #(sub-
strate:0,WEE1_sub)
       [    inactivate_ENZ | activate_ENZ |
     rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
            inactivate_SUB | activate_SUB |
     rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
            rep enzyme?().nil |
            substrate!().ch(substrate, WEE1_sub_mod).nil  ];

let WEE1_TEMP : bproc =    #(enzyme:0,WEE1_enz),
 #(substrate:0,WEE1_sub_mod)
      [    inactivate_ENZ | activate_ENZ |
     rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
            inactivate_SUB | activate_SUB |
     rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
            rep enzyme?().nil                            ];

let WEE1P :  bproc = #(substrate:0,WEE1P_sub)
     [ substrate!().ch(substrate, WEE1P_sub_mod).nil      ];

let WEE1P_TEMP : bproc = #(substrate:0,WEE1P_sub_mod)
[ nil  ];

let CDC25 : bproc = #(substrate:0,CDC25_sub)
     [  substrate!().ch(substrate, CDC25_sub_mod).nil    ];

let CDC25_TEMP : bproc = #(substrate:0,CDC25_sub_mod)
     [ nil ];

let CDC25P : bproc = #(enzyme:0,CDC25P_enz),
 #(substrate:0,CDC25P_sub)
      [    inactivate_ENZ | activate_ENZ |
     rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
            inactivate_SUB | activate_SUB |
     rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
            rep enzyme?().nil |
            substrate!().ch(substrate, CDC25P_sub_mod).nil
            ];

let CDC25P_TEMP : bproc =  #(enzyme:0,CDC25P_enz),
     #(substrate:0,CDC25P_sub_mod)
     [      inactivate_ENZ | activate_ENZ |
     rep rec_ENZ?().(inactivate_ENZ | activate_ENZ ) |
            inactivate_SUB | activate_SUB |
     rep rec_SUB?().(inactivate_SUB | activate_SUB ) |
            rep enzyme?().nil                            ];

let ENZYME1 : bproc = #(enzyme:0,ENZYME1_enz)
 [ rep enzyme?().nil ];

let ENZYME2 : bproc = #(enzyme:0,ENZYME2_enz)
 [ rep enzyme?().nil ];

when(WEE1_TEMP::inf) split(WEE1P, Nil);
when(CYCB_TEMP::inf) split(CYCBP, Nil);
when(CDC25P_TEMP::inf) split(CDC25,Nil);
when(CYCBP_TEMP::inf) split(CYCB,Nil);
when(WEE1P_TEMP::inf) split(WEE1,Nil);
when(CDC25_TEMP::inf) split(CDC25P,Nil);

///*************** initial *****************
run … CYCB || … CYCBP || … CDC25P || … CDC25 || … WEE1 || …
WEE1P || … ENZYME1 || … ENZYME2
```

type file:

```
{ WEE1_enz, WEE1_sub, WEE1_sub_mod,
  CYCB_enz, CYCB_sub, CYCB_sub_mod, CYCB_enz_CDC25,
 CDC25P_enz, CDC25P_sub, CDC25P_sub_mod,
  ENZYME1_enz,   ENZYME2_enz,
  CDC25_sub, CDC25_sub_mod,
  WEE1P_sub, WEE1P_sub_mod,
  CYCBP_sub, CYCBP_sub_mod      } %%
{ (CYCB_enz_CDC25, CDC25_sub, rate(a1), rate(a1r), rate(a2)),
  (CYCB_enz_CDC25, CDC25_sub_mod, 0, inf , 0 ),
  (CYCB_enz, WEE1_sub, rate(b1), rate(b1r), rate(b2)),
  (CYCB_enz, WEE1_sub_mod, 0, inf , 0),
  (ENZYME1_enz, CDC25P_sub, rate(c1), rate(c1r), rate(c2)),
  (ENZYME1_enz, CDC25P_sub_mod, 0, inf , 0),
  (CDC25P_enz, CDC25P_sub,rate(d1), rate(d1r), rate(d2) ),
  (CDC25P_enz, CYCBP_sub_mod, 0, inf , 0 ),
  (WEE1_enz, CYCB_sub, rate(e1), rate(e1r),rate(e2)),
  (WEE1_enz, CYCB_sub_mod, 0, inf , 0),
  (ENZYME2_enz, WEE1P_sub, rate(f1), rate(f1r) ,rate(f2) ),
  (ENZYME2_enz, WEE1P_sub_mod, 0, inf ,0)           }
```

Appendix B: The circadian clock model

program file:

```
[ steps = 7200, delta = 0.1 ]

<< BASERATE:inf, CHANGE:inf >>

     /// SUBSTRATE (CP)
let CP :    bproc = #(cp:0,CP)
     [ cp!().ch(cp,CPU1_DEG).nil ];

     /// SUBSTRATE (CP2)
let CP2 :   bproc = #(cp2:0,CP2)
     [ cp2!().ch(cp2,CP2U2_DEG).nil ];

  /// SUBSTRATE (IC)
let IC :    bproc = #(ic:0,IC)
     [ ic!().ch(ic,ICU3_DEG).nil ];

       /// ENZYME
let U1 :    bproc = #(u1:0,U1)
     [ rep u1?().nil ];

       /// ENZYME
let U2 :    bproc = #(u2:0,U2)
     [ rep u2?().nil ];

       /// ENZYME
let U3 :    bproc = #(u3:0,U3)
     [ rep u3?().nil ];
```

```
                /// PRODUCT
let CPU1_DEG :   bproc = #(cp:0,CPU1_DEG)
     [ nil ];

       /// PRODUCT
let CP2U2_DEG : bproc = #(cp2:0,CP2U2_DEG)
     [ nil ];

       /// PRODUCT
let ICU3_DEG :   bproc = #(ic:0,ICU3_DEG)
     [ nil ];

////////.............. TRANSCRIPTION MODULE
...................
///*************** species ******************

       /// gene
let G       : bproc = #(g:0,G)  [ nil ];

       /// transcription factor /// BMAL1/CLK
let TF           : bproc = #(tf,TF)  [ nil ];

       /// G-TF dimer
let TF2     : bproc = #(t2f,TF2)     [ nil ];

       /// G-TF-TF trimer
let GTF2    : bproc = #(gtf2,GTF2)   [ nil ];

       /// messenger RNA
let M       : bproc = #(m:0,M)  [ nil ];

///*************** reactions ****************
       ///dimerization
when(TF,TF::rate(k1)) join(TF2);
```

```
       ///decomplexation
when(TF2::rate(k2)) split(TF,TF);
       ///'trimerization'
when(TF2,G::rate(k3)) join(GTF2);
       ///decomplexation
when(GTF2::rate(k4)) split(G,TF2);
     /// transcriptional activation
when(GTF2::TRANSCRIPTION) split(GTF2,M);
       /// degradation of messenger
when(M::rate(kmd)) delete(1);


/////////.............. TRANSLATION ...................
when(M::rate(km2)) split(M,CP);

/////////.............. HOMODIMERIZATION ...................
when(CP,CP::rate(ka)) join(CP2);
when(CP2::rate(kd)) split(CP,CP);

/////////.............. COMPLEX FORMATION ...................
when(CP2,TF::rate(kica)) join(IC);
when(IC::rate(kicd)) split(CP2,TF);

/////////.............. DEGRADATIONS ...................
when(CP::rate(kcpd)) delete(1);
when(CP2::rate(kcp2d)) delete(1);
when(IC::rate(kcp3d)) split(Nil,TF);

//////// Michaelis Menten kinetics for degradation terms:
///*************** species *****************
when(CPU1_DEG::inf) delete(1);
when(CP2U2_DEG::inf) delete(1);
when(ICU3_DEG::inf) split(Nil,TF);
```

```
//////* Initial
run 96300 M || 600 CP ||  3400 CP2 || 21700 TF || 277700 IC
||
1 G || 0 TF2 || 0 GTF2 ||
60 U1 || 240 U2 || 26830 U3
```

function file:

```
let alfa : const = 0.00000167;

let kms2 : const = 1;
let k1 : const = 10*alfa;
let k2 : const = 100;
let k3 : const = 10000*alfa;
let k4 : const = 90;

let kms : const = 1/alfa;
let kmd : const = 0.1;
let km2 : const = 0.5;
let ka_c : const = 100;
let ka : const = ka_c*alfa;
let kd : const = 0.01;
let kica_c : const = 20;
let kica : const = kica_c*alfa;
let kicd : const = 0.01;

let kcpd : const = 0.0525;
let kcp2d : const = 0.000525;
let kcp3d : const = 0.000525;

let kp11 : const = 10000000*alfa;
let kp11r : const = 1530;
let kp12 : const = 1000;

let kp21 : const = 100000*alfa;
let kp21r : const = 47.5;
let kp22 : const = 2.5;

let kp31 : const = 100*alfa;
let kp31r : const = 0.027678571;
let kp32 : const = 0.022321429;

let TRANSCRIPTION : function = kms*kms2*|GTF2|;

///**** complex reaction terms for Michaelis-Menten kinetics:
///let U1 : const = 0.0001/alfa;
///let U2 : const = 0.0004/alfa;
///let U3 : const = 0.0448/alfa;
///let CP_DEG : function =
kp12*U1*|CP|/(((kp11r+kp12)/kp11)+|CP|);
///let CP2_DEG : function =
kp22*U2*|CP2|/(((kp21r+kp22)/kp21)+|CP2|);
///let IC_DEG : function =
kp32*U3*|IC|/(((kp31r+kp32)/kp31)+|IC|);

///**** parameters and complex reaction terms for Hill function:
///let n : const = 2;
///let J : const = 0.3/alfa;
///let TRANSCRIPTION : function = (kms*pow(|TF|,n))/(pow(J,n)
+ pow(|TF|,n));
```

type file:

```
{ CPU1_DEG, CP2U2_DEG, ICU3_DEG, U1, CP, U2, CP2, U3, IC, G,
TF, TF2, GTF2, M }
%%
{
 (U1,CP,rate(kp11),rate(kp11r),rate(kp12)),
 (U1,CPU1_DEG,0,inf,0),
 (U2,CP2,rate(kp21),rate(kp21r),rate(kp22)),
 (U2,CP2U2_DEG,0,inf,0),
 (U3,IC,rate(kp31),rate(kp31r),rate(kp32)),
 (U3,ICU3_DEG,0,inf,0)
}
```