A Review of Modelling and Verification Approaches for Computational Biology

Savas Konur*

* *Department of Computer Science, University of Bradford, UK*
s.konur@bradford.ac.uk

Abstract

This paper reviews most frequently used computational modelling approaches and formal verification techniques in computational biology. The paper also compares a number of model checking tools and software suits used in analysing biological systems and biochemical networks and verifiying a wide range of biological properties.

Keywords: modelling, verification, model checking, biology, computational biology, biological systems, biochemical networks, gene regulatory networks

1 Introduction

The formalisation of biological systems using *computational modelling* approach as an alternative to mathematical equations has recently received much interest, as this approach provides a deeper mechanistic understanding of biological systems. Formalisms where molecular populations and interactions are modelled as discrete entities and events have come to be known collectively as *executable biology* [1], or *algorithmic systems biology* [2]. There are various computational formalisms studied in this context, including *state transition systems*, *rule-based systems*, *Petri nets* and *process algebra*.

Formal methods have been considerably used to construct and analyse computational (or executable) biological systems so as to obtain important information about system behaviour, which can be considered as a complementary approach to standard computational techniques. *Formal verification* is a method which *exhaustively* analyses *all* possible system behaviour to evaluate the correctness of systems. Formal verification provides more insight into a natural system than standard methods, e.g. simulation and testing, allowing us to infer more novel information about system properties.

Model checking [3], an *algorithmic* approach for verification, is a computational technique verifying whether a finite structure satisfies a property. Model checking requires a formal modelling of the system and a formal specification of the property, expressed in a logical notation [4, 5, 6, 7, 8, 9]. It then evaluates the formal specification against all possible behaviours of the system, which are computed by enumerating all possible sequence of traces.

Model checking has been extensively used in computer science and engineering for the last two decades in the verification of various systems, e.g. safety-critical systems [10], concurrent systems [11], distributed systems [12], network protocols [13], stochastic systems [14], multi-agent systems [15, 16], pervasive systems [17, 18, 19] and swarm robotics [20, 21] as well as some engineering applications [22, 23, 24]. Due to its novel approach to extract information about system behaviour, there is a growing interest to apply this technique in the analysis of biological systems and biochemical networks [25, 26]. Recently, it has been applied to analysis of various biological systems, e.g. ERK/MAPK pathway [27], FGF signalling pathway [28, 29], cell cycle in eukaryotes [30, 31], EGFR pathway [32], T-cell receptor signaling pathway [33, 34, 35, 36], cell cycle control [37, 38, 39, 40], mammalian cell cycle regulation [41, 42], apoptosis network [43], bladder tumorigenesis [44], quorum sensing [45, 46, 47], DNA

Computing [48, 49], genetic oscillator [50], genetic Boolean gates [51, 45, 52, 53] and switches [54, 55].

In this paper, we review some of the most utilised modelling approaches in computational biology, and compare several model checking tools and software suits used in analysing biological systems and biochemical networks.

The paper is organised as follows: Section 2 presents computational modelling approaches. Section 3 compares tools used in verifying biological and biochemical systems. Section 4 presents some biological systems analysed using model checking. Section 5 concludes the paper and discusses our future research.

2 Modelling approaches for biological and biochemical systems

In this section, we overview some of the most utilised mathematical and computational modelling approaches (see Table 1) in analysing biological systems and biochemical networks.

Table 1: Modelling approaches most commonly used in analysing biological systems and biochemical networks.

Modelling approach	**Formalism**
ODEs	XS-systems, Piecewise-linear, Piecewise-multiaffine
State transition systems	Kripke structures, Boolean networks, Multivalued logical formalism, Reactive modules, Statecharts, Live sequence charts
Rule-based systems	P Systems, Rewriting systems, Pathway logic, Biocham, BioNetGen
Petri nets	Stochastic, Coloured, Discrete, Continuous Petri nets
Process algebra	Stochastic $\pi-$calculus, Bio-PEPA, BioAmbients

2.1 Mathematical modelling approach: ODEs

Mathematical modelling approach has a long tradition inherited from various disciplines including genetics, biochemistry, evolutionary biology and systems biology to understand the system dynamics and characteristics [56]. Until recently, coupled *ordinary differential equations (ODEs)* were commonly used as a modelling approach for complex biological and biochemical systems.

In a typical ODE model, a variable represents the rate of the concentration change of a molecular species in the system in question. A key assumption in an ODE model is that variables are *continuous*, i.e. concentrations change *continuously* over time, and the system model is deterministic, i.e. resulting in precise solutions. This assumption becomes valid when concentrations are sufficiently high, with an approximate lower bound of 10^3 molecules, and reactions are fast enough [57]. When lower concentrations are considered, this assumption cannot be considered any more.

Biological systems have been also modelled by some approximation methods, e.g., *piecewise-linear* and *piecewise-multiaffine* differential equations. ODEs can be rewritten in some canonical forms: *S-system* is a canonical form where each differential equation is described in such a way that the concentration change of a product is expressed in terms of the concentration changes of its reactants [58]. An *XS-system* is a form containing a list of expressions describing the concentration change of molecular species and a set of equations which represents some constraints regarding model parameters [58].

2.2 Computational modelling approach

A different formalism based on computational models, as an alternative to mathematical approaches, has been investigated more intensively in the last years. The mathematical models rely on equations to describe quantities and their relationships over time, whereas the computational ones present the system

in a rather operational way, as a sequence of steps, or an algorithm, executed by an abstract or virtual machine [1]. Two terms have been coined in connection with these models, *executable/computational biology* [1] or *algorithmic systems biology* [2, 59]. This new formalism comes with a set of domain specific languages in modelling and methods and tools largely utilised in programming analysis and verification. Below we review a selection of these alternative representations and their capabilities.

2.2.1 State transitions systems

A (discrete) *state system* is a simple computational model that describes the dynamic behaviour of a system. A state system is composed of set of finite set of states and behaviours and it defines how certain changes on input events cause output events to take place. *State transition systems* comprise a finite set of *states*, representing the states that the system can be in, and a finite set of *transitions*, representing the conditions to traverse between these states. The machine moves from one state to another when an event or condition in the corresponding transitions holds. In a state transition system, transitions can also be labeled. Depending on the context, a label can be used for different purposes, e.g. a condition to trigger the transition, an action to perform when the transition is taken and a probability denoting the likelihood to take the transition.

Kripke structures [60] are simple state transition systems, describing the dynamic behaviour of a system. In a Kripke structure, a state represents a snapshot of system; a transition represents how the system state evolves; and a path represents a computation of the system.

Boolean networks [61] are Boolean state transition systems representing gene interactions as a directed graph, where each node represents a gene that is either active or inactive; edges represent positive or negative regulation; and Boolean functions represent gene expressions. Boolean networks are deterministic and have a fixed topology. So, the starting configuration does not change with time. Boolean networks are especially useful when the data about the system is incomplete. In such cases, we can easily construct a model using only topological information and necessary binary relationships. They are therefore often selected as a modelling approach for their amenability to analysis rather than realism [62].

Thomas' multivalued logical formalism [63, 64] is a multi-valued state-transition system, providing a *discrete* modelling framework for regulatory networks. It can also be considered as "an abstraction of a special case of piecewise-linear differential equations or as a generalization of a restriction of Boolean networks" [65]. Thomas' regulatory network models are represented by a labeled directed graph, where a vertex denotes variables representing biological entities (e.g. genes and proteins) and a directed edge denotes abstract interaction between variables. Each edge is labeled with a discrete *threshold* representing the maximum concentration level and a sign "+" representing activation and "-" representing inhibition. A regulatory network is associated with a(n) (a)synchronous state graph denoting the dynamics of the network [63].

Reactive modules [66] define a language, providing a compact representation for state transition systems by dividing them into a set of *modules*. A module comprises a collection of local variables and a set of transitions representing the behaviour of the variables over time. The behaviour of a reactive module is represented by *guarded* transitions, which specify that if the guard is true, then the transition is carried out by performing a certain action and updating the local variables.

Statecharts [67] are qualitative but fine-grained state-based transition diagrams used to model the mechanism underneath the system behaviour. Statecharts model the dynamic behaviour using states and events triggering transitions between states. Statecharts permit modelling at multiple levels by allowing states to be composed of substates, which can be "zoomed in" and "zoomed out". States can also be divided into "orthogonal states", thus concurrency can be described [68].

Live sequence charts (LSCs) [69], an extension of message sequence charts, are a scenario-based approach to analyse system behaviour. Live sequence charts formalise various scenarios of behaviours between different system elements, e.g., "required", "possible", and "forbidden" scenarios.

2.2.2 Rule-based systems

Rule-based systems model the state of the system as molecular species and state change as molecular interactions specified by rules.

P Systems are the key models of the *membrane computing* theory [70], which is a branch of natural computing. These models emphasise the compartmentalised nature of biological systems. The central element of any P system is a membrane structure, consisting of regions (or compartments) containing multisets of objects interacting through locally specified rewriting and communication rules [71]. P systems *evolve* by repeatedly applying these rules, mimicking chemical reactions and transportation across membranes. The close similarity between the model and key elements of the cellular biology makes it suitable not only for modelling purposes, but it also enhances the communication between modellers and biologists, or any other wet laboratory experimentalists.

Rewriting systems are methods containing sequences of discrete steps where a subterm of a formula is replaced with another [72]. Rewriting systems comprise a set of objects and rewrite rules representing the transformation of these objects. In systems biology context, a rewriting rule defines "a step in a biological process such as metabolism or intra/inter-cellular signaling" and describes "the behavior of proteins and other components depending on modification state and biological context" [73].

Pathway logic [32] is a rewriting systems based "algebraic structure" enabling the building and analysing network models of biological processes, where system states and behaviour are represented by algebraic structures and rewriting rules, respectively. Pathway logic consists of "data types representing cellular components such as proteins, small molecules, complexes, compartments/locations protein state, and post-translational modifications" [73]. Pathway logic permits qualitative analysis, e.g. "static and dynamic structure of reaction networks", but does not permit specifying kinetic constants for reactions; stochastic simulations, therefore, are not supported.

BioNetGen [74] is ruled-based system based on structured building blocks which represent proteins and protein complexes. BioNetGen captures "site-specific details of protein-protein interactions" and provides visualisations of these interactions. The language also allows molecules to be combined into complexes through "binding sites". Rules specify the biochemical reactions occurring in the system and can be used to construct bimolecular networks.

Biocham [75, 76] is another rule based system comprising three languages, modelling biochemical processes at different levels of abstraction: Boolean (i.e. "asynchronous Boolean transition systems"), concentration (i.e. ODEs) and stochastic (i.e. continuous time Markov chains).

Other rule-based systems, including **NFsim** [77], **Kappa** [78] and **little b** [79], have been introduced in order address the issue of combinatorial explosion in the number of interactions, by explicitly providing mechanisms to model coincidental modifications or conformations that need to be represented.

2.2.3 Petri nets

Petri nets are well-established formalisms describing distributed and concurrent systems behaviour. A Petri net is a graph with two types of nodes, places containing tokens and representing various resources available, and transitions describing events that will fire under certain circumstances. Petri nets can be considered as a more general form of Boolean networks, because several tokens might fire concurrently

at the same time. We also note that while Boolean networks can be executed deterministically, Petri nets can be executed nondeterministically.

There are several variants of Petri nets. **Stochastic Petri nets** [80, 81] make use of a stochastic simulation algorithm where transitions with associated rate are fired and a period of time is calculated and added to the global clock. **Coloured Petri nets** [82] provide a means to deal with multiple possible values associated with each place and produce a more compact representation of a system [83]. Petri nets can also be defined in *discrete* and *continuous* semantics.

2.2.4 Process algebras

Process algebras (process calculi) are a diverse family of related formalisms that describe distributed concurrent processes, such as tasks inside a computer program or a collection of programs, interacting in accordance with certain communication protocols. π-calculus [84] is a model for concurrent mobile processes where the communication topologies evolve dynamically. In the biological models based on π-calculus molecules with binding sites are represented as processes with communication channels. The simulation is similar to a standard Petri net and provides a qualitative view on the system's behaviour.

Several variants of process algebras have been introduced: **Stochastic π-calculus** [85] enables quantitative simulations by associating a rate constant with each channel and providing means to compute a probability based on this rate. **PEPA** (Performance Evaluation Process Algebra) is a different stochastic process algebra used for modelling reagent-centric and pathway-centric approaches [86] and for different signalling pathways [86, 87, 88, 89] and synthetic biology designs [90]. **Bio-PEPA** [91] is a modification of PEPA, incorporating stoichiometry and the use of kinetic laws in rate functions, adequate for modelling biological systems. **BioAmbients** [92], an extension of the Ambient calculus with stochastic features, as in stochastic π-calculus, have been developed to model dynamics of biological compartments. **BlenX** [93] is a high level stochastic process algebra based textual language, explicitly designed to model biological entities and their interactions.

2.2.5 Other computational systems

As well as the computational models discussed above, there are other executable modelling formalisms for biological and biochemical systems. In the following we discuss them very briefly:

The Systems Biology Markup Language (SBML) [94] is an XML dialect used to store models of biological systems and facilitate communication between different tools. SBML files store information about model compartments, species and reactions, as well as events, units, etc. that are relevant to some models and approaches but not others.

Hybrid systems [95, 96, 97] combine both continuous and discrete aspects in one single model. In general, the continuous behaviour is represented by (piecewise linear) differential equations and the discrete behaviour represented by computational models. Hybrid systems provide a means to close "the gap between mathematical models and computational models by combining the two" [1]. Some hybrid model examples are hybrid discrete-continuous systems, hybrid automata [98] and hybrid Petri nets [99].

Cellular automaton is a formalism frequently used for modelling biological systems [100], amongst them pattern formation (morphogenesys) [101], ecology and population biology, immunology, oscillations, diffusion processes, fibroblast aggregation, ant trails and others - [102] is an overview of different cellular automata models.

Other two notable approaches are **agent-based systems** [103] and **knowledge-based systems** [104].

2.3 Example: Gene expression

In this section, we illustrate some of the modelling approaches on a running example, taken from [105]. The biomolecular system comprises *positive*, *negative* and *constitutive* expressions of a gene. The model consists of a gene with its transcribed RNA and the corresponding translated protein and activator and repressor molecules which bind to the gene. Figure 1 shows models of this system using different formalisms.

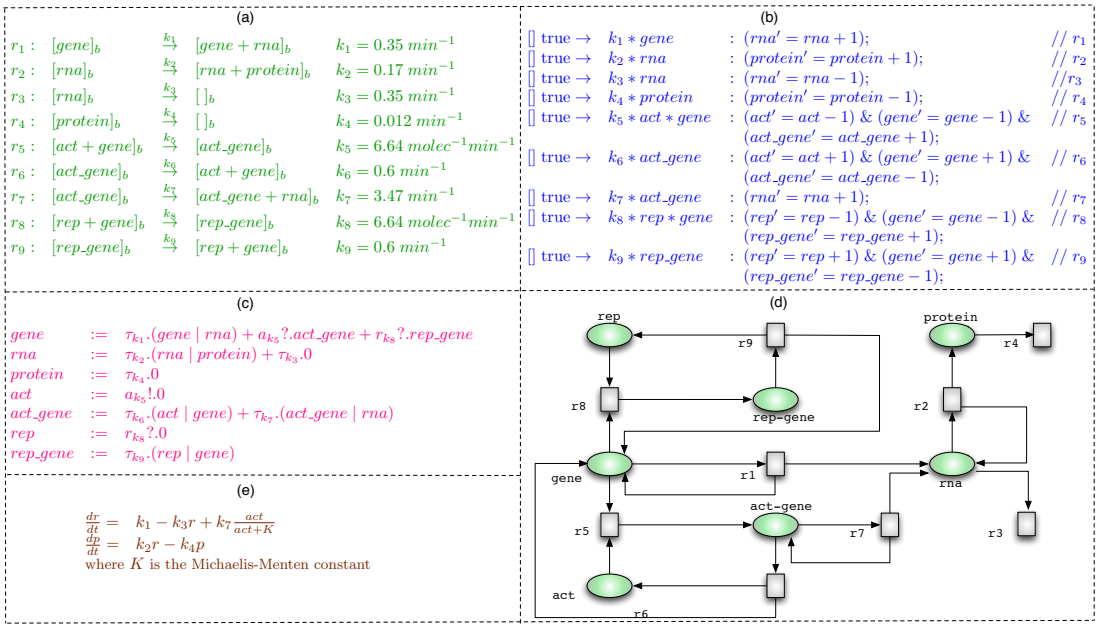

Figure 1: (redrawn from [105]) A system comprising positive, negative and constitutive expression of a gene. (a) **Rule-based model (P systems).** The system consists of a bacterium represented using a membrane, called b. The transcription and translation are represented as multiset rewriting rules with kinetic constants associated with. (b) **State-machine model (reactive modules of PRISM).** The model consists of modules and each module is described by a set of commands. A command has the form: $[act]\ g \ \rightarrow \ \gamma_1 : u_1 + ... + \gamma_n$, where g is a predicate over all the variables of the model and each u_i describes a tradition of the module, where new values of the variables are calculated. The expressions γ_i are used to express rates associated to transitions. The label act is used to synchronise commands occurring in different modules. (c) **Process algebra model.** The process called *gene* defines all possible interactions consisting of a constitutive reaction, a positive regulation, or negative regulation. Other processes describe subsequent interactions. (d) **Petri net model.** This describes all the interactions presented above. If a token appears in the place *gene*, then we only describe constitutive expression; if a token is in *gene* and n in *act* with $n \geq 1$, then we describe a positive regulation; and if we start with one token in *gene* and m in *rep* with $m \geq 1$, then we have negative regulation. (e) **ODE model.** The same interactions described as a set of differential equations.

3 Tools used in verifying biological and biochemical systems

We have carried out a survey that contains a comparative summary of important features of the tools and methods used/developed in this context.

3.1 Standalone model checkers

Here, we present the state of the art model checkers used in systems biology and biochemical networks. An overview of the standalone model checking tools is given in Table 2, which provides a summary of the tools and the important features.

A comparison of these tools is presented in Table 3, where we compare the tools according to *modelling formalism* (i.e. which modelling approach the tool supports), *specification language* (i.e. logical languages used to express properties), *type* (i.e. model checking method), *usage* (i.e. for which purpose(s) the tool is used, e.g. *property checking* to verify required properties, *parameter estimation* to find missing parameters in models, and *parameter optimisation* to find optimal set of values), the last *release* date of the tool and the corresponding *reference*.

3.2 Software suits employing model checkers

Model checkers have also been integrated and used in software platforms developed to analyse biological systems. In Table 4, we present an overview of the integrated software suits employing model checking tools. In Table 5, we compare the tools similar to Table 3. In Table 5, we also mention which model checking tools are employed.

In addition to the most popular software suits presented in Table 4, there are some other tools developed such as U-CHECK BMS15, MULE [116], PYBIONETFIT [39], Bio-ModelChecker [35].

4 Biological systems analysed using model checking

In this section, we present biological systems analysed using model checking techniques. Table 6 presents a selected range of systems. The extended list with references is presented below.

4.1 Kripke structures

Kripke structures are formal models for standard model checkers. They are mostly described in a high-level language, e.g. PROMELA [106] and NUSMV language [107]. Such high-level models correspond to Kripke structures. **SPIN** [106] and **NUSMV** [107] are two popular tools to model and analyse Kripke structures. Several biological systems have been analysed using this approach:

SPIN: mucus production in *Pseudomonas aeruginosa* [135], biological regulatory network for *Indoleamine 2,3-dioxygenase* [135], genetic network of *Arabidopsis thaliana* [136], quorum sensing [46], genetic gates [137, 45], pulse generator [124, 138].

NUSMV: EGFR network [139], molecular interaction network of a macrophage [139], quorum sensing [46], genetic gates [45], pulse generator [124], T-helper cell plasticity [34].

4.2 Boolean Networks

Boolean networks are the earliest computational modelling frameworks for gene regulatory networks. It has been applied to model various GRNs [140, 141, 142, 143, 144, 145] (e.g. yeast cell-cycle regulation). Apart from **MATLAB**, several tools have been developed that enable modelling and analysis on Boolean networks such as **BOOLEANNET** [146], **BOOLNET** [147], **ANTELOPE** [108] and **PYBOOLEANNET** [148]. In [149], Boolean networks are extended to *Qualitative networks*, which allow variables representing "the level of protein activity or gene expression to range over larger domains". Some biological systems formally analysed are presented below:

Table 2: Model checking tools used in analysing biological systems.

SPIN [106] is a popular model checking tool to verify qualitative temporal properties of various systems. System models to be verified are specified in the PROMELA language, corresponding to a Kripke structure, and temporal properties are expressed as LTL formulas (or 'assertions'). SPIN also provides an interactive and guided simulator displaying model execution traces. The tool employs a number of methods to improve model checking process and reduce the storage space.
NUSMV [107] is one of the most widely used model checking tools to verify the correctness of finite state systems. NUSMV is a symbolic model checker, because it uses a symbolic compact representation of states to reduce the state space. The NUSMV high level language allows writing modular hierarchical descriptions and reusable components. A NUSMV model corresponds to a Kripke structure. Temporal properties can be expressed in both LTL and CTL.
ANTELOPE [108] is a model checking tool for modelling and analysing (branching-time) Boolean networks. The branching time can represent some important phenomena, e.g."asynchrony", incomplete system information and "interaction with the environment" [108]. In order to express more Boolean network specific properties, ANTELOPE employs a "*hybrid* extension of CTL, enabling to specify the set of stable and unstable steady states" [108]. Rather than just checking if the given property is true at the initial state, ANTELOPE returns all states satisfying the given property. This adoption is more preferable in the context of Boolean networks [108]. ANTELOPE is a symbolic model checker, building the state space based on compact representations.
PRISM [109] is the most widely used probabilistic model checking tool for formal modelling and analysis of probabilistic systems. Models are written in a "state-based" high-level language based on "reactive modules". PRISM allows building and analysing various probabilistic models, e.g. Discrete-time Markov Chains (DTMCs), Continous-time Markov Chains (CTMCs) and Markov Decision Processes (MDPs). It uses compact and structured data structures based on (multi-terminal) binary decision diagrams (BDDs) to reduce the size of probabilistic models. As property language, various probabilistic logics are supported, including LTL, PCTL and CSL. PRISM also provides a "discrete-event simulation engine", supporting statistical model checking, and employs various analysis techniques.
PLASMA LAB [50] is a statistical model checking tool, deployed for probabilistic systems, e.g. DTMCs and CTMCs. PLASMA LAB supports several modelling languages, including reactive modules and a rule based biological language, and some property specification languages, including BLTL. The BLTL language allows specifying temporal properties with bounds expressed in step or time units. PLASMA LAB extends BLTL with a probability operator, which permits specifying probabilistic properties.
MOCHA [110] is the first model checker developed to formally analyse reactive modules in a modular and hierarchical way. The description languages allows modelling systems with synchronous and asynchronous components. In MOCHA, properties are expressed in an extension of CTL, called ATL. The logic ATL, unlike CTL and LTL, provides a selective quantification, which allows one to specify a strategy to reach a desired state. MOCHA is a symbolic model checker, employing a BDD engine. The tool also provides a tool support for simulating the reactive systems and displaying the simulation traces in a message sequence chart.
MAUDE [111] is a "high-performance reflective language" based on rewriting systems and an integrated software suit using a range of tools for the modelling and analyses of rewriting systems. MAUDE models can be translated into various formalisms to make various analyses such as model checking, theorem proving, statistical model checking, debugging and searching. The MAUDE model checker is the part of the MAUDE system, which checks if a rewriting system satisfied a property, specified in LTL. The MAUDE model checker does not use any compact BDD structures to store states. MAUDE also provides support for Pathway logic.
BIOLAB [33] is a statistical model checking tool, enabling formal verification for BIONETGEN models. BIOLAB simulates models using a BIONETGEN simulator, and then verifies properties, expressed as probabilistic bounded linear temporal logic (PBLTL), against generated stochastic traces. PBLTL is an extension of the bounded linear temporal logic (BLTL) with a probability operator to bound the likelihood of a BLTL formula to hold. BIOLAB generates as many simulations as needed to verify a property. If more samples are needed to infer whether the property is verified or not, then BIOLAB continues to generate more simulation traces until a decision about the property has been made.
MC2 [112] is a statistical model checker based on "Monte Carlo approximation". The tool accepts a set of simulation traces as input rather than a system model specified in a high-level language. Simulation traces can be obtained from any simulation output, e.g. ODEs, CTMCs and Gillespie. As property language, MC2 employs a probabilistic temporal logic with numerical constraints, i.e. PLTL with numerical constraints. MC2 can cope with state spaces beyond the current limits of the model checkers that perform exhaustive analyses. Also, the model checking process is in general much quicker because the entire search space is not explored.
BIODIVINE [113, 114] is a tool developed for model checking biological interaction networks specified as ODEs. It provides a user interface, called GENESIM, to build models of biochemical systems as piecesewise-linear or piecewise-multiaffine ODEs. Properties to be verified are expressed as temporal logic formulas specified in LTL. BIODIVINE is related to several other tools, e.g. DIVINE tool [114] (a general purpose parallel LTL model checker), PARASIM (a tool to analyse robustness of biochemical systems) and PARSYBONE (a tool for synthesis of discrete kinetic parameters in gene regulatory networks – specified in Thomas' formalism).
SIMPATHICA [58] is a tool set deployed for modelling and analysing biological networks. A network system, e.g. regulatory and signalling, is constructed using a graphical model editor, which allows to describe ODEs (based on XS-systems). The system can be analysed using different tools and techniques. The OCTAVE tool enables simulating the model to analyse its behaviour. SIMPATHICA also allows analysing temporal behaviour of a biological network using its backend model checker XSSYS. The XSSYS tool can verify CTL queries, repressing questions about the behavior of a biological network. These queries are checked against a set of simulation traces. XSSYS can therefore be considered as an approximate model checker. XSSYS also provides a natural language interface that can be used to construct temporal logic formulas using plain English.
BIOMODELANALYZER [115] is a web-based tool set developed for the modelling and analysis of gene and protein interaction networks. Based upon the Qualitative Networks formalism, users can graphically draw biological models manually or using a built-in library. Users have access to three analysis engines to test their models, which are standard simulation tools, an interface to the stability?testing algorithm and to a graphical Linear Temporal Logic (LTL) editor and analysis tool. The tool also allows users to construct LTL properties using a graphical language supported by drag-and-drop feature to visually construct queries as well as evaluate results. The tool hosts a knowledge base that consists of definitions as well as example usages of LTL operators and developmental end states.

Table 3: Comparison of model checking tools.

Tool	Modelling formalism	Specification language	Type	Usage
SPIN	Kripke structures	LTL	temporal	property checking
NUSMV	Kripke structures	LTL, CTL	temporal	property checking
ANTELOPE	Boolean networks	CTL (hybrid)	temporal	property checking
PRISM	Reactive modules	LTL, PCTL, CSL	temporal, probabilistic	property checking
PLASMA LAB	Reactive modules	BLTL + probability	temporal, probabilistic	property checking
MOCHA	Reactive modules	ATL	temporal	property checking
MAUDE	Rewriting systems, Pathway logic	LTL	temporal	property checking
BIOLAB	BioNetGen + Simulation traces	PBLTL	probabilistic	property checking, parameter estimation
MC2	Simulation traces	PLTL + constraints	probabilistic	property checking, parameter optimisa.
BIODIVINE	ODEs	LTL	temporal	property checking
SIMPATHICA & XSSYS	ODEs	CTL	temporal	property checking
BIOMODEL ANALYZER	Qualitative networks (GUI)	LTL	temporal	property checking

BOOLEANNET: abscisic acid [146], mammalian immune response [146], T-cell large granular *lymphocyte leukemia* [146], cell cycle gene identification [150]

ANTELOPE: *Arabidopsis thaliana* root stem cell niche [108], flower organ specification [108], root stem niche [108].

BOOLENET: mammalian cell cycle [147], yeast cell cycle [147].

NUSMV: *D. melanogaster* embryo development [151], budding yeast cell cycle [145], bladder tumorigenesis [44].

4.3 Thomas' multivalued logical formalism

Thomas' formalism is supported by the **SMBIONET** [120] software platform, which selects models of a regulatory graph and generates all corresponding asynchronous state graphs based on some behavioural properties. Several biological models have been analysed using the Thomas' formalism and the SMBIONET tool, e.g. mucus production in *Pseudomonas aeruginosa* [64], immunity control in bacteriophage lambda [152], tail resorption in tadpole metamorphosis [120], biosurfactants production in *Pseudomonas* fluorescens [153], biological regulatory network in breast cancer [154], FGF signalling in *drosophila melanogaster* [29]. **GINSIM** [155] is another software tool built upon Thomas' logical approach, enabling the qualitative modelling, simulation and analysis of the dynamics of regulatory networks. The tool has been used in many systems, including budding yeast cell cycle, (various) *Drosophila* signalling pathways [156], MAPK pathway [156], TCR signalisation [156] and mammalian network [42].

Thomas' work has been extended in some papers: A hybrid modelling approach to generalize Thomas' regulatory networks is applied to analyse various possible dynamics of the mucus production in the *Pseudomonas aeruginosa* bacterium [157] and the reaction of *E.coli* bacterium to carbon starvation [158] using the **HYTEC** tool [159]. In [160], Thomas' approach is extended with constraint programming; namely, the method generates all asynchronous state graphs of a regulatory network graph based on some behavioural properties and constraints. The approach has been illustrated in the mucus production system of *Pseudomonas aeruginosa*. A similar work has been published in [161], where a constraint-based extension of the Thomas' method has been proposed and applied to the mucus production of *Pseudomonas*

Table 4: Software suits integrating model checking tools.

PLAY-ENGINE [117] is a tool developed to build, execute, analyse and verify scenario-based requirements. Behavioural requirements are specified in a visual formalism called *live sequence charts (LSCs)*. The required behaviour is captured by playing and constructing scenarios in a GUI. This process is called *play-in*. The output of the play-in process is then automatically translated to a formal specification visualised as LSCs [118]. In the *play-out* process, these requirements are validated by executing resulting LSCs. The play-out process is extended with *smart play-out*, which makes use of model checking while executing the behavioral requirements. PLAY-ENGINE relies on some third party model checkers, including SMV [119].
SMBIONET [120] is a tool for modelling and analysing biological regulatory networks. The description language of the tool is the BioNetGen language, based on Thomas' multivalued logical formalism, and the property specification language is CTL. SMBIONET allows selecting models of a biological regulatory network according to some properties. Namely, temporal properties of the regulatory network are expressed as CTL properties. The tool then generates all corresponding asynchronous state graphs satisfying these properties. SMBIONET employs NUSMV to perform model checking. It therefore uses symbolic model checking approach.
The **INFOBIOTICS WORKBENCH (IBW)** [57] is an integrated software toolkit to perform various analyses for a stochastic extension of P systems. The software platform enables modelling, simulation, model checking and parameter optimisation using different tools and methods: (i) models are simulated either using stochastic simulation or deterministic numerical method using MCSS (a simulator for multi-compartment stochastic P system models) (ii) model structure and parameters can be optimised with evolutionary algorithms using POPTIMIZER; (iii) properties of a model's temporo-spatial behaviour can be verified using PMODELCHECKER; and (iv) all experiments can be visualised using the INFOBIOTICS DASHBOARD [57]. PMODELCHECKER employs the PRISM and MC2 tools to perform probabilistic and statistical model checking, respectively. PMODELCHECKER supports all the specification languages that these tools support, i.e. LTL, PCTL, CSL and PLTL with numerical constraints. An important feature of PMODELCHECKER is that it provides a natural language query tool to assist in constructing properties using natural language statements without using any formal syntax.
The **kPWORKBENCH** [121] is an integrated software toolkit that allows modelling and analysing a unified membrane P systems, called kP systems [122, 123]. The platform permits simulation and formal verification of membrane using several simulation and verification tools and methods. The framework features a native simulator, allowing the simulation of kP system models [124]. In addition, it also integrates an agent based high-performance simulation environment [125, 51]. KPWORKBENCH's model checking environment permits the formal verification of kernel P system models [126, 127]. The framework supports both LTL and CTL properties by making use of the SPIN and NUSMV model checkers. In order to facilitate the formal specification, KPWORKBENCH features a property language comprising a list of natural language statements representing formal property patterns, from which the formal syntax of the SPIN and NUSMV formulas are automatically generated.
The **BIOCHAM** system [128] is a modelling and analysis platform for rule-based systems. The tool allows modelling biochemical systems, simulating Boolean, differential and stochastic models and verifying biological properties as well as "developing, correcting, completing and coupling models". BIOCHAM can estimate missing model parameters from temporal logic properties. The tool can also check that temporal logic properties are not violated during the model building process. In addition, BIOCHAM can "automatically search for parameter values that reproduce the specified behavior of the system in different conditions" [76]. BIOCHAM employs NUSMV to carry out the model checking task. As property specification language, the tool supports two types of formulas: qualitative properties are expressed in CTL and quantitative properties "about concentrations and their derivatives" are expressed in LTL with numerical constraints.
The **MODEL-CHECKING KIT (MCKIT)** [129] is a software suit for a collection of tools including model checkers, providing (various variants of) Petri net models with formal analysis and verification as well as deadlock and reachability checking. The kit can be considered as a common interface for various verification tools. This allows the kit to run different state construction techniques, e.g. explicit-state and symbolic methods. Among the model checking tools are SPIN, SMV and LOLA. MCKIT accepts both LTL and CTL as the property specification language.
BIO-PEPA ECLIPSE PLUG-IN & WORKBENCH [91] are two software tools developed to support biochemical networks, based on the Bio-PEPA language. The BIO-PEPA ECLIPSE PLUG-IN is a modelling environment, incorporating various features, e.g. a simulator based on the stochastic Gillespie algorithm, static and model coverage analyses and a GUI to visualise results. The **BIO-PEPA WORKBENCH** tool enables modelling and analysing biochemical networks using different techniques, such as stochastic simulation and model checking. The tool maps Bio-PEPA models to different targets (e.g. ODEs, CTMCs and SBML) to use different analysis methods. The BIO-PEPA WORKBENCH does not integrate a model checker directly, but it complies Bio-PEPA models to CTMCs in the reactive modules format that PRISM accepts. These modes can then be verified against some CSL properties.
GENETIC NETWORK ANALYZER (GNA) [130] is a tool developed for qualitative modelling and analysis of genetic regulatory networks given in the form of piecewise-linear ODEs. GNA provides a GUI to build, edit and visualise GRN models. These networks can be analysed using simulation and model checking. The network dynamics can be observed through a qualitative simulation, "resulting in predictions adapted to available gene expression data". Qualitative properties of GRNs are specified in a temporal logic using natural language query templates. These properties can be expressed in CTL and CTRL. CTRL, subsuming both CTL and LTL, is an extension of CTL with regular expressions (and fairness operators). GNA employs NUSMV and CADP [131] for model checking. GNA can export to and import from SBML. The tool is also compatible with the Systems Biology Graphical Notation (SBGN) format.
The **TAVERNA SERVICES FOR SYSTEMS BIOLOGY (TAV4SB)** [132] is a web-based software platform to design and analyse kinetics model of biological systems, described as a set of ODEs. TAV4SB accepts ODE models in the SBML format. The services provided by the tool are simulation of the kinetic model using the SBML ODE Solver library, probabilistic model checking of CSL formulas using PRISM, "visualization of data series, such as ODEs trajectories or values of parametrized CSL properties, and probabilistic distribution sampling, using MATHEMATICA", and "high-level analysis, such as multi-parameter sensitivity analysis" [132].
ROVERGENE [133] is a tool for analysing biological regulatory networks, described as (piecewise multiaffine) ODEs. ROVERGENE can be used to carry out robustness analysis, meaning that "a dynamical property is satisfied by every parameter in a given set and for every initial state in a given region", and parameter synthesis, meaning that "searching for valid subsets of a given parameter set". ROVERGENE employs the NUSMV model checker, and temporal properties are expressed in LTL.
ANIMO [134] is a toolset for analysing biological pathways, represented by networks of Stochastic Hybrid Automata. The tool combines UPPAAL Stochastic Model Checking, a plugin of the tool used by biologists, as well as with SimBiology, a plugin of Matlab to simulate reactions. By integrating translators from SBML (and XGMML) used by Cytoscape and SimBiology to stochastic and hybrid automata, it allows stochastic model checking analysis techniques for stochastic and hybrid systems using UPPAAL SMC and the specification formalism of weighted metric temporal logic .

Table 5: Comparison of software suits.

Tool	Modelling formalism	Specification language	3rd party tools	Type	Usage
PLAY-ENGINE	Live sequence charts	Play-in, Play-out	SMV	temporal	capturing & executing requirements
SMBIONET	Multivalued logical form.	CTL	NUSMV	temporal	model selection
IBW	P systems	LTL, PCTL, CSL, PLTL + constraints	PRISM, MC2	temporal, probabilistic	property checking
BIOCHAM	Biocham (ODEs, stochastic, Boolean)	CTL, LTL + constraints	NUSMV	temporal	property checking, parameter estimation
MCKIT	Petri nets	LTL, CTL	SMV, SPIN, LOLA	temporal	property checking
BIO-PEPA	Bio-PEPA	CSL	PRISM	probabilistic	property checking
GNA	ODEs	CTL, CTRL	NUSMV, CADP	temporal	property checking
TAV4SB	ODEs	CSL	PRISM	probabilistic	property checking, param. sensitivity analysis
ROVERGENE	ODEs	LTL	NUSMV	temporal	parameter estimation, robustness analysis
ANIMO	SBML	WMTL	UPPAAL-SMC	temporal	property checking

aeruginosa using the **AGATHA** tool [162]. Another related work has been carried out in [163], presenting a computational approach used in finding all attractors in multi-valued regulatory networks. The method has been applied to the network models of *Arabidopsis thaliana*, budding yeast, *Drosophila* melanogaster, fission yeast, mammalian cell, T-cell receptor and T-helper cell.

4.4 Reactive modules

Reactive modules can be used as a description language for various models including *Discrete Markov Chains, Continuous Markov Chains* and *Markov Decision Processes*. These models can be analysed using some tools, e.g., **PRISM** and **PLASMA**. Reactive modules can also be analysed formally using the **MOCHA** tool [110]. Many biological systems have been analysed so far:

PRISM: enzymatic activity [164], ERK/MAPK (Mitogen-Activated Protein Kinase) signaling pathway [165, 166, 27, 167], FGF signalling pathway [28, 168], codon bias [169], phase switching in *E. coli* film system [170], protein folding kinetics [171], ribosome kinetics [172], codon misreading errors [173], transient oscillator [174], influenza virus fusion [175], bone pathologies [176], sodium-potassium exchange pump [177], genetic network with a negative feedback [178], DNA strand displacement [179], PDGF pathway [180], TGF-β, WNT and MAPK pathways [181], crosstalk between the cyclic AMP and the Raf-1/MEK/ERK signalling pathways [182], cell energy-related reactions of sodium-potassium exchange pump [183], genetic gates [184, 52, 53], genetic toggle switch [55].

PLASMA: genetic oscillator [50], simple biochemical system [50].

MOCHA: signaling crosstalk during *C. elegans* vulval development [185], cell fate specification during *C. elegans* vulval development [186].

Table 6: Some biological systems and biochemical networks analysed using model checking.

Modelling formalism	Tool	Biological systems modelled
Kripke structures	SPIN	mucus production in *P. aeruginosa*, genetic network of *Arabidopsis thaliana*, quorum sensing, genetic gates, pulse generator
	NUSMV	EGFR network, molecular interaction network of a macrophage, quorum sensing, genetic gates, pulse generator, T-helper cell plasticity
Boolean networks	BOOLEANNET	abscisic acid, mammalian immune response, T-cell large granular *lymphocyte leukemia*, cell cycle gene identification
	ANTELOPE	*A. thaliana* root stem cell niche, flower organ specification, root stem niche
	BOOLENET	mammalian cell cycle, yeast cell cycle
	NUSMV	*D. melanogaster* embryo development, budding yeast cell cycle, bladder tumorigenesis
Multivalued logical for	SMBIONET	mucus production in *P. aeruginosa*, biosurfactants production in *Pseudomonas* fluorescents, breast cancer, FGF signalling in *drosophila melanogaster*
	GINSIM	east cell cycle, *Drosophila* signalling pathways, MAPK pathway, TCR signalisation, mammalian cell cycle
Reactive modules	PRISM	ERK/MAPK & FGF signaling pathways, codon bias, ribosome kinetics, transient oscillator, bone pathologies, genetic gates, genetic toggle switch
	PLASMA	genetic oscillator, simple biochemical system
	MOCHA	signaling crosstalk during *C. elegans* vulval development, cell fate specification during *C. elegans* vulval development
P Systems	IBW	cell cycle in eukaryotes, gene expression, liposome logic, quorum sensing, repressilator, pulse generator, genetic gates
	kPWORKBENCH	quorum sensing, genetic logic gates, pulse generator
Rewriting systems	MAUDE	ERK/MAPK & EGFR signaling networks, molecular interaction network of a macrophage, Rho GTP-binding cycle, signal transduction
Pathway logic	PATHWAY LOGIC ASSISTANT	MAPK & EGFR signaling networks, HGF/HGFR & IL6/IL6R signaling pathways, response of melanoma cancer cells to drugs, crosstalk in breast cancer
BioNetGen	BIONETGEN, BIOLAB	T-cell receptor signaling pathway, activation of Jak-family protein tyrosine kinases, yeast pheromone response pathway, phosphorylation and scaffolding in MAPK pathways, B-cell antigen receptor signaling, p53-induced apoptosis
Biocham	BIOCHAM	gene expression regulation, cell cycle control, ERK/MAPK, synthetic transcriptional cascade, translation initiation in sea urchin, cell cycle on the mitosis phase, phosphorylation cycles
Petri nets	PRISM, MC2, MCKIT, BIOCHAM	ERK signal transduction pathway, receptor signalling, kinase cascades, cell-cycle regulation, wound healing, neuronal cell fate decision model in *Caenorhabditis elegans*, angiogenetic process, Wnt/β-catenin signalling pathway, DNA walker, quorum sensing
Process algebra	BIO-PEPA, BIOSPI, etc.	epidemiological models *avian influenza*, yeast pheromone pathway, circadian clock in *Ostreococcus tauri*, genetic network with a negative feedback, cell growth and damage from cancer treatment, hypertorus communication grid, mumps virus
ODEs	NUSMV	nutritional stress response in *E. coli*, mammalian cell cycle regulation
	BIODIVINE	transcription in *Bacillus subtilis*, ammonium transport in *E. coli*, genetic regulatory networks, FGFR3 signalling pathway
	SIMPATHICA/XSSYS	repressilator, purine metabolism, yeast cell
	GNA	gene regulatory networks, nutritional stress and carbon starvation response in *E. coli*
	TAV4SB	enzymatic reaction model
	ROVERGENE	synthetic transcriptional cascade, toggle switch, two-genes network with stimulus

4.5 Statecarts / Live sequence charts

Statecharts can be designed and executed using the **RHAPSODY** tool [187]. Statecharts have been applied to modelling and analysis of T-cell maturation [188, 68] and the *C.elegans* vulval development [189].

Both statecharts and live sequence charts work in a complementary fashion to model different aspects of a biological system. The **PLAY-ENGINE** [190] tool supports modelling and execution with LSCs. Statecharts and live sequence charts have been applied together to model the cell fate decision of the development of the *C.elegans* somatic gonad [117].

4.6 Membrane (P) systems

There are many variants of P systems. Some of the most well-studied P systems with relevance for modelling biological systems are: *stochastic P systems* [191] (supported by the **INFOBIOTICS WORKBENCH (IBW)** software suit [57]), *kernel P systems* [192, 193] (supported by **kPWORKBENCH** [121]), *metabolic P systems* [194] (supported by the **METAPLAB** software system); *non-deterministic P systems* [195], *dynamical probabilistic P systems* [196] (supported by the **BIOSIMWARE** software platform [197]), *probabilistic dynamics population P systems* [198] (supported by the **MECOSIM** software environment [199]), *probabilistic P systems with peripheral proteins* [200] (supported by **CYTOSIM** simulation environment [201]) and *P systems with string objects* [202] (supported by the **SRSIM** software environment [202]). P systems have been used in modelling and analysis of various biological systems

INFOBIOTICS WORKBENCH: cell cycle in eukaryotes [30], gene expression [105], liposome logic [203], auxin transport [204], repressilator [205], quorum sensing [46], pulse generator [57, 124, 206] and genetic Boolean gates [51, 45].

kPWORKBENCH: quorum sensing[46], genetic logic gates [45], pulse generator [124]

4.7 Rewriting systems

Rewriting systems are supported by the **MAUDE** system [111], which enables executing, searching and model-checking. Some biological systems modelled using rewriting systems are growth and heterocyst differentiation in *Anabaena* [207], cascades of protein interactions in signalling pathways [208], EGFR signaling network [32, 139], ERK/MAPK pathway [209], molecular interaction network of a macrophage [139], Rho GTP-binding cycle [210], signal transduction [211, 212].

4.8 Pathway logic

Pathway logic is supported by **PATHWAY LOGIC ASSISTANT (PLA)** [211]. PLA provides a user interface mapping the Pathway logic models to Petri net models for analysis and visualisiation as well as querying to find pathways. This feature also permits using tools developed for analysing the Petri net models e.g. **PATHALYZER** [213] for carrying out computational analyses and **LOLA** [214] for model-checking. Pathway logic is also supported by the **MAUDE** system. Pathway logic has been used in the analysis of various systems, including Rac1 activation [209, 73], EGFR pathway [32, 215], MAPK pathway [211], HGF/HGFR and IL6/IL6R signaling pathways [216], response of melanoma cancer cells to drugs [217], pathway crosstalk between TNF, TGFB1 and EGF in basal-like breast cancer [218].

4.9 BioNetGen

The BioNetGen language is supported by the **BIONETGEN** software tool [74], providing modelling, simulation and analysis of biochemical systems. The BIONETGEN system has been used in various

systems, including T-cell receptor signaling pathway [33, 36], activation of Jak-family protein tyrosine kinases [219], yeast pheromone response pathway [220], phosphorylation and scaffolding in MAPK pathways [221], B-cell antigen receptor signaling [222], signaling pathways exhibiting bistability [223].

The only model checker that supports BioNetGen is **BIOLAB**, which has been used to formally analyse T-Cell receptor signaling pathway [33], p53-induced apoptosis [224].

4.10 Biocham

Biocham is supported by the **BIOCHAM** [128] software system enabling modelling biochemical systems, simulating Boolean, kinetic and stochastic models and verifying some biological properties. Some systems modelled and analysed using the BIOCHAM system are gene expression regulation [75], cell cycle control [75, 37, 225], ERK/MAPK [166, 27, 128], synthetic transcriptional cascade [226], cap-dependent translation initiation in sea urchin [227], cell cycle on the mitosis phase [38], phosphorylation cycles [228].

4.11 Petri Nets

Petri nets were used to model initially biological pathways [229, 230] and more recently ERK signal transduction pathway [166, 27, 231], receptor signalling [232], kinase cascades [232], cell-cycle regulation [232], wound healing [232], neuronal cell fate decision model in *Caenorhabditis elegans* [233], angiogenetic process [231], Wnt/β-catenin signalling pathway [234], DNA walker [48], quorum sensing [47]. An overview of using Petri nets for modelling, simulation and analysis of biological systems can be found in [235, 236, 237]. There are numerous tools deployed to create and analyse Petri nets, e.g. **SNOOPY** [238], **PATHALYZER** [213], **CELL ILLUSTRATOR** [239] and **LOLA** [214]. We refer the reader to [240, 241, 237] for more information on the available tools.

4.12 Process Algebra

Different variants of process algebra have been supported by different tools, e.g. **BIO-PEPA** [91], **BIOSPI** [242], **SPIM** [243], **COSBI LAB** [59], **BETA WORKBENCH** [244] and **KINFER** [245]. Some biological systems formally analysed are epidemiological models of *avian influenza* [246], yeast pheromone pathway [247], circadian clock in *Ostreococcus tauri* [248], genetic network with a negative feedback [91, 178], circadian oscillators [249], cell growth and damage from cancer treatment [250], hypertorus communication grid [251], mumps virus [252].

4.13 ODEs

ODEs have been predominantly solved using mathematical tools e.g. **MATLAB** and **CVODE**. Many biological systems have been modelled using this approach. For a detailed survey on mathematical modelling in biological and biochemical networks, we refer the reader to [253, 254].

In order to infer more information about the system dynamics, various tools have been deployed to analyse ODEs using formal techniques. Several model checking tools available allowing formal analysis of certain ODEs. Some examples are given below:

NUSMV: nutritional stress response in *E.coli* [255], mammalian cell cycle regulation [41].

BIODIVINE: transcription in *Bacillus subtilis* [256], ammonium transport in *E.coli* [113], genetic regulatory networks [257], FGFR3 signalling pathway [258].

SIMPATHICA/ XSSYS: repressilator [58], purine metabolism [58], yeast cell cycle [259], Wnt signaling pathway [260], purine metabolism pathway [261].

GNA: gene regulatory networks [262], nutritional stress response in *E. coli* [255], carbon starvation response of *E. coli* [263, 264, 130].

TAV4SB: enzymatic reaction model [132]

ROVERGENE: synthetic transcriptional cascade [133], toggle switch [54], two-genes network with stimulus [54].

4.14 SBML

Tools for the visual specification of models in SBML, e.g. **CELLDESIGNER** [265], **VCELL** [266] and **COPASI** [267], enable the visual creation of models from a collection of symbols for various types of molecular and interactions. Recently, a statistical model checking tool, **MIRACH** [268], has been devised to formally analyse SBML models. The tool has been applied to ERK/MAPK signalling pathway and cell fate in *C. elegans*. The **BIOCHAM** [128] tool has also support for SBML. Another tool in this context is **IBIOSIM** [269], which supports both SBML and SBOL (Synthetic Biology Open Language).

4.15 Hybrid systems

Among the tools for modelling and analysis of hybrid systems are **HYTECH** [159] and **CELL ILLUSTRATOR** [239]. Various hybrid system models have been studied using hybrid system, e.g. Delta-notch signalling network [270], insulin control diabetic patients [271], sporulation in *Bacillus subtilis* [271], lactose metabolism in *E.Coli* [272] and biochemical systems theory [273]. A recent survey on applications of hybrid systems in biology is presented in [274].

5 Discussions and conclusion

In this paper, we have reviewed mostly used modelling approaches in computational biology, and also compared some model checking tools used in analysing biological systems and biochemical networks. There is a growing interest in using model checking systems biology, as this approach provides a deeper mechanistic understanding of biological systems. Numerous biological systems have been analysed, from biological pathways to genetic bio-devices.

Our survey has showed that most of the related work focuses on systems biology. Although there are a few case studies, the application in synthetic biology is quite limited. This is mainly due to the fact that synthetic biology introduces new challenges, difficult for existing model checking approaches to cope with.

As future research, we are currently working on an integrative perspective combining different model checking approaches based on different property types and the use of some natural language patterns to express various properties. This approach will make the use of the model checking methods very effective and easy to use, even for non-experts in formal methods.

Acknowledgements

The work of SK is supported by EPSRC (EP/I031812/1 & EP/R043787/1).

References

[1] J. Fisher and T. A. Henzinger. Executable cell biology. *Nature Biotechnology*, 25(11):1239–1249, 2007.

[2] C. Priami. Algorithmic systems biology. *Commun. ACM*, 52:80–88, 2009.

[3] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model checking*. MIT Press, 1999.

[4] S. Konur. A decidable temporal logic for events and states. In *Thirteenth International Symposium on Temporal Representation and Reasoning (TIME'06)*, pages 36–41, 2006.

[5] S. Konur. An interval logic for natural language semantics. In *Proceedings of the seventh conference on Advances in Modal Logic, Nancy, France, 9-12 September 2008*, pages 177–191, 2008.

[6] S. Konur. A survey on temporal logics. *CoRR*, abs/1005.3199, 2010.

[7] S. Konur. Real-time and probabilistic temporal logics: An overview. *CoRR*, abs/1005.3200, 2010.

[8] S. Konur. An event-based fragment of first-order logic over intervals. *Journal of Logic, Language and Information*, 20:49 – 68, 2011.

[9] S. Konur. A survey on temporal logics for specifying and verifying real-time systems. *Front. Comput. Sci.*, 7(3):370–403, June 2013.

[10] S. Konur. Specifying safety-critical systems with a decidable duration logic. *Science of Computer Programming*, 80(PB):264–287, 2014.

[11] R. Alur, K. McMillan, and D. Peled. Model-checking of correctness conditions for concurrent objects. *Information and Computation*, 160(1-2):167 – 188, 2000.

[12] M. Yabandeh. *Model checking of distributed algorithm implementations*. PhD thesis, IC, 2011.

[13] S. Konur and M. Fisher. Formal analysis of a VANET congestion control protocol through probabilistic verification. In *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*, pages 1–5, 2011.

[14] S. Konur. Towards light-weight probabilistic model checking. *Journal of Applied Mathematics*, 2014:15, 2014.

[15] H. Abbink, R. van Dijk, T. Dobos, M. Hoogendoorn, C. Jonker, S. Konur, P.-P. van Maanen, V. Popova, A. Sharpanskykh, P. van Tooren, J. Treur, J. Valk, L. Xu, and P. Yolum. Automated support for adaptive incident management. In *Proc. of the 1st Int. Workshop on Information Systems for Crisis Response and Management, ISCRAM'04. Brussels*, pages 153–170, 2004.

[16] S. Konur, M. Fisher, and S. Schewe. Combined model checking for temporal, probabilistic, and real-time logics. *Theoretical Computer Science*, 503:61 – 88, 2013.

[17] M. Arapinis, M. Calder, L. Denis, M. Fisher, P. Gray, S. Konur, A. Miller, E. Ritter, M. Ryan, S. Schewe, C. Unsworth, and R. Yasmin. Towards the verification of pervasive systems. *Electronic Communications of the EASST*, 22, 2009.

[18] S. Konur, M. Fisher, S. Dobson, and S. Knox. Formal verification of a pervasive messaging system. *Formal Aspects of Computing*, 26(4):677–694, 2014.

[19] S. Konur and M. Fisher. A roadmap to pervasive systems verification. *The Knowledge Engineering Review*, 30(3):324341, 2015.

[20] S. Konur, C. Dixon, and M. Fisher. Formal verification of probabilistic swarm behaviours. In M. Dorigo, M. Birattari, G. A. Di Caro, R. Doursat, A. P. Engelbrecht, D. Floreano, L. M. Gambardella, R. Groß, E. Şahin, H. Sayama, and T. Stützle, editors, *Swarm Intelligence*, volume 6234 of *Lecture Notes in Computer Science*, pages 440–447, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[21] S. Konur, C. Dixon, and M. Fisher. Analysing robot swarm behaviour via probabilistic model checking. *Robotics and Autonomous Systems*, 60(2):199 – 213, 2012.

[22] F. Camci, O. F. Eker, S. Baskan, and S. Konur. Comparison of sensors and methodologies for effective prognostics on railway turnout systems. *Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit*, 230(1):24–42, 2016.

[23] R. Lefticaru, S. Konur, Ü. Yildirim, A. Uddin, F. Campean, and M. Gheorghe. Towards an integrated approach to verification and model-based testing in system engineering. In *The International Workshop on Engineering Data- & Model-driven Applications (EDMA-2017)*, pages 131–138, 2017.

[24] R. Lefticaru, M. E. Bakir, S. Konur, M. Stannett, and F. Ipate. Modelling and validating an engineering application in kernel P systems. In M. Gheorghe, G. Rozenberg, A. Salomaa, and C. Zandron, editors, *Membrane Computing*, pages 183–195, Cham, 2018. Springer International Publishing.

[25] M. E. Bakir, M. Gheorghe, S. Konur, and M. Stannett. Comparative analysis of statistical model checking tools. In A. Leporati, G. Rozenberg, A. Salomaa, and C. Zandron, editors, *Membrane Computing*, pages 119–135, Cham, 2017. Springer International Publishing.

[26] M. E. Bakir, S. Konur, M. Gheorghe, N. Krasnogor, and M. Stannett. Automatic selection of verification tools for efficient analysis of biochemical models. *Bioinformatics*, 34(18):3187–3195, 2018.

[27] M. Heiner, D. Gilbert, and R. Donaldson. Petri nets for systems and synthetic biology. In *Formal Methods for Computational Systems Biology*, volume 5016 of *Lecture Notes in Computer Science*, pages 215–264. Springer Berlin Heidelberg, 2008.

[28] J. Heath, M. Kwiatkowska, G. Norman, D. Parker, and O. Tymchyshyn. Probabilistic model checking of complex biological pathways. *Theoretical Computer Science*, 319(3):239–257, 2008.

[29] M. T. Saeed, J. Ahmad, J. Baumbach, J. Pauling, A. Shafi, R. Z. Paracha, A. Hayat, and A. Ali. Parameter estimation of qualitative biological regulatory networks on high performance computing hardware. *BMC Systems Biology*, 12(1):146:1–146:15, 2018.

[30] F. J. Romero-Campero, M. Gheorghe, L. Bianco, D. Pescini, M. J. Perez-Jimenez, and R. Ceterchi. Towards probabilistic model checking on P systems using Prism. In *Membrane Computing*, volume 4361 of *Lecture Notes in Computer Science*, pages 477–495. Springer Berlin Heidelberg, 2006.

[31] A. Shukla, A. Bhattacharyya, L. Kuppusamy, M. Srivas, and M. Thattai. Discovering vesicle traffic network constraints by model checking. *PLOS ONE*, 12(7):1–15, 2017.

[32] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, J. Meseguer, and K. Sonmez. Pathway logic: symbolic analysis of biological signaling. In *Proceedings of the Pacific Symposium on Biocomputing*, pages 400–412, 2002.

[33] E. M. Clarke, J. R. Faeder, C. J. Langmead, L. A. Harris, S. K. Jha, and A. Legay. Statistical model checking in Biolab: applications to the automated analysis of T-cell receptor signaling pathway. In *Proceedings of the 6th International Conference on Computational Methods in Systems Biology*, CMSB'08, pages 231–250. Springer-Verlag, 2008.

[34] W. Abou-Jaoud, P. T. Monteiro, A. Naldi, M. Grandclaudon, V. Soumelis, C. Chaouiya, and D. Thieffry. Model checking to assess t-helper cell plasticity. *Frontiers in Bioengineering and Biotechnology*, 2:86, 2015.

[35] H. Sedghamiz, M. Morris, T. J. A. Craddock, D. Whitley, and G. Broderick. Bio-modelchecker: Using bounded constraint satisfaction to seamlessly integrate observed behavior with prior knowledge of biological networks. *Frontiers in Bioengineering and Biotechnology*, 7:48, 2019.

[36] A. Khalid. Biometa: A multiple specification parameter estimation system for stochastic biochemical models, 2020.

[37] L. Calzone, N. Chabrier-Rivier, F. Fages, and S. Soliman. Machine learning biochemical networks from temporal logic properties. In *Transactions on Computational Systems Biology VI*, volume 4220 of *Lecture Notes in Computer Science*, pages 68–94. Springer Berlin Heidelberg, 2006.

[38] P. Traynard, F. Fages, and S. Soliman. Model-based investigation of the effect of the cell cycle on the circadian clock through transcription inhibition during mitosis. In O. Roux and J. Bourdon, editors, *Computational Methods in Systems Biology*, pages 208–221, Cham, 2015. Springer International Publishing.

[39] E. D. Mitra, R. Suderman, J. Colvin, A. Ionkov, A. Hu, H. M. Sauro, R. G. Posner, and W. S. Hlavacek. Pybionetfit and the biological property specification language. *iScience*, 19:1012 – 1036, 2019.

[40] B. A. Hall and J. Fisher. Constructing and analyzing computational models of cell signaling with biomodelanalyzer. *Current Protocols in Bioinformatics*, 69:e95, 2020.

[41] P. Traynard, A. Faur, F. Fages, and D. Thieffry. Logical model specification aided by model-checking techniques: application to the mammalian cell cycle regulation. *Bioinformatics*, 32(17):i772–i780, 2016.

[42] A. Naldi, C. Hernandez, W. Abou-Jaoud, P. T. Monteiro, C. Chaouiya, and D. Thieffry. Logical modeling and analysis of cellular regulatory networks with GINsim 3.0. *Frontiers in Physiology*, 9:646, 2018.

[43] O. Wei, Z. Guo, Y. Niu, and W. Liao. Model checking optimal finite-horizon control for probabilistic gene regulatory networks. *BMC Systems Biology*, 11(6):75–88, 2017.

[44] H. Klarner, H. Siebert, S. Nee, and F. Heinitz. Basins of attraction, commitment sets and phenotypes of boolean networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 1–1, 2018.

[45] S. Konur, M. Gheorghe, C. Dragomir, F. Ipate, and N. Krasnogor. Conventional verification for unconventional computing: a genetic XOR gate example. *Fundamenta Informaticae*, 134:97–110, 2014.

[46] S. Konur, M. Gheorghe, C. Dragomir, L. Mierlă, F. Ipate, and N. Krasnogor. Qualitative and quantitative analysis of systems and synthetic biology constructs using P systems. *ACS Synthetic Biology*, 4(1):83–92, 2015.

[47] D. Gilbert, M. Heiner, L. Ghanbar, and J. Chodak. Spatial quorum sensing modelling using coloured hybrid Petri nets and simulative model checking. *BMC Bioinformatics*, 20, 04 2019.

[48] M. A. Boemo, A. E. Lucas, A. J. Turberfield, and L. Cardelli. The formal language and design principles of autonomous dna walker circuits. *ACS Synthetic Biology*, 5(8):878–884, 2016.

[49] W. Zhu, Y. Han, and Q. Zhou. Performing CTL model checking via DNA computing. *Soft Comput.*, 23(12):3945 – 3963, 2019.

[50] A. Legay, S. Sedwards, and L.-M. Traonouez. Plasma lab: A modular statistical model checking platform. In T. Margaria and B. Steffen, editors, *Leveraging Applications of Formal Methods, Verification and Validation: Foundational Techniques*, pages 77–93, Cham, 2016. Springer International Publishing.

[51] D. Sanassy, H. Fellermann, N. Krasnogor, S. Konur, L. Mierlă, M. Gheorghe, C. Ladroue, and S. Kalvala. Modelling and stochastic simulation of synthetic biological Boolean gates. In *16th IEEE International Conference on High Performance Computing and Communications*, pages 404–408, 2014.

[52] M. Gheorghe, S. Konur, and F. Ipate. *Kernel P Systems and Stochastic P Systems for Modelling and Formal Verification of Genetic Logic Gates*, pages 661–675. Springer International Publishing, Cham, 2017.

[53] S. Konur, H. Fellermann, L. Marian Mierla, D. Sanassy, C. Ladroue, S. Kalvala, M. Gheorghe, and N. Krasnogor. *An Integrated In Silico Simulation and Biomatter Compilation Approach to Cellular Computation*, pages 655–676. Springer International Publishing, Cham, 2017.

[54] S. Bogomolov, C. Schilling, E. Bartocci, G. Batt, H. Kong, and R. Grosu. Abstraction-based parameter synthesis for multiaffine systems. In N. Piterman, editor, *Hardware and Software: Verification and Testing*, pages 19–35. Springer International Publishing, 2015.

[55] T. Neupane, Z. Zhang, C. Madsen, H. Zheng, and C. J. Myers. *Approximation Techniques for Stochastic Analysis of Biological Systems*, pages 327–348. Springer International Publishing, Cham, 2019.

[56] J. Gunawardena. Models in systems biology: the parameter problem and the meanings of robustness. In *Elements of Computational Systems Biology*, pages 19–47. John Wiley & Sons, Inc., 2010.

[57] J. Blakes, J. Twycross, S. Konur, F. Romero-Campero, N. Krasnogor, and M. Gheorghe. Infobiotics workbench: A P systems based tool for systems and synthetic biology. In *Applications of Membrane Computing in Systems and Synthetic Biology*, volume 7 of *Emergence, Complexity and Computation*, pages 1–41. Springer, 2014.

[58] M. Antoniotti, A. Policriti, N. Ugel, and B. Mishra. Model building and model checking for biochemical processes. *Cell Biochemistry and Biophysics*, 38(3):271–286, 2003.

[59] C. Priami. Algorithmic systems biology – computer science propels systems biology. In *Handbook of Natural Computing*, pages 1835–1862. Springer Berlin Heidelberg, 2012.

[60] S. A. Kripke. Semantical analysis of modal logic i normal modal propositional calculi. *Mathematical Logic Quarterly*, 9(56):67–96, 1963.

[61] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *Journal of theoretical biology*, 22(3):437–467, 1969.

[62] A. Feiglin, A. Hacohen, A. Sarusi, J. Fisher, R. Unger, and Y. Ofran. Static network structure can be used to model the phenotypic effects of perturbations in regulatory networks. *Bioinformatics*, 28(21):2811–2818, 2012.

[63] R. Thomas and M. Kaufman. Multistationarity, the basis of cell differentiation and memory. i. structural conditions of multistationarity and other nontrivial behavior. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 11(1):170–179, 2001.

[64] G. Bernot, J.-P. Comet, A. Richard, and J. Guespin. Application of formal methods to biological regulatory networks: extending Thomas' asynchronous logical approach with temporal logic. *Journal of Theoretical Biology*, 229(3):339 – 347, 2004.

[65] M. Carrillo, P. A. Gongora, and D. A. Rosenblueth. An overview of existing modeling tools making use of model checking in the analysis of biochemical networks. *Frontiers in Plant Science*, 3(155), 2012.

[66] R. Alur and T. Henzinger. Reactive modules. *Formal Methods in System Design*, 15(1):7–48, 1999.

[67] D. Harel. Statecharts: A visual formalism for complex systems. *Sci. Comput. Program.*, 8(3):231–274, 1987.

[68] S. Efroni, D. Harel, and I. R. Cohen. Toward rigorous comprehension of biological complexity: modeling, execution, and visualization of thymic T-cell maturation. *Genome Research*, 13:2485–2497, 2003.

[69] J. Fisher, D. Harel, E. Hubbard, N. Piterman, M. Stern, and N. Swerdlin. Combining state-based and scenario-based approaches in modeling biological systems. In *Computational Methods in Systems Biology*, volume 3082 of *Lecture Notes in Computer Science*, pages 236–241. Springer Berlin Heidelberg, 2005.

[70] G. Păun. Computing with membranes. *Journal of Computer and System Sciences*, 61 (2000)(1):108–143, 2000.

[71] F. J. Romero-Campero, M. Gheorghe, G. Ciobanu, J. M. Auld, and M. J. Pérez-Jiménez. Cellular modelling using P systems and process algebra. *Progress in Natural Science*, 17:375–383, 2007.

[72] J. W. Klop. Term rewriting systems. In *Handbook of Logic in Computer Science (Vol. 2)*, pages 1–116. Oxford University Press, Inc., 1992.

[73] C. Talcott. Symbolic modeling of signal transduction in pathway logic. In *Proceedings of the 38th Conference on Winter Simulation*, WSC '06, pages 1656–1665. Winter Simulation Conference, 2006.

[74] J. R. Faeder, M. L. Blinov, and W. S. Hlavacek. Rule-based modeling of biochemical systems with Bionetgen. In *Methods in Molecular Biology, Systems Biology*, volume 500 of *Methods in Molecular Biology*. Humana Press, 2009.

[75] N. Chabrier-Rivier, M. Chiaverini, V. Danos, F. Fages, and V. Schächter. Modeling and querying biomolecular interaction networks. *Theoretical Computer Science*, 325(1):25 – 44, 2004.

[76] L. Calzone, F. Fages, and S. Soliman. BIOCHAM: an environment for modeling biological systems and formalizing experimental knowledge. *Bioinformatics*, 22(14):1805–1807, 2006.

[77] M. W. Sneddon, J. R. Faeder, and T. Emonet. Efficient modeling, simulation and coarse-grain of biological complexity with NFsim. *Nature Methods*, 8:177–183, 2011.

[78] V. Danos, J. Feret, W. Fontana, and J. Krivine. Scalable modelling of biological pathways. In *Asian Symposium on Programming Systems, LNCS 4807*, LNCS, pages 139–157, 2007.

[79] 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*, 6:257–270, 2009.

[80] S. Natkin. *Les reseaux de Petri stochastiques et leur application a levaluation des systemes informatiques*. PhD thesis, CNAM, Paris, France, 1980.

[81] F. J. W. Symons. Introduction to numerical Petri nets, a general graphical model of concurrent processing systems. *Australian Telecommun. Res.*, 14(1), 1980.

[82] K. Jensen and L. M. Kristensen. *Coloured Petri Nets: Modelling and Validation of Concurrent Systems*. Springer, 2009.

[83] Q. Gao, F. Liu, D. Tree, and D. Gilbert. Multi-cell modelling using coloured petri nets applied to planar cell polarity. In *Proceedings of the 2nd International Workshop on Biological Processes & Petri Nets (BioPPN2011)*, pages 135–150, 2011.

[84] R. Milner. *Communicating and Mobile Systems: π-calculus*. Cambridge University Press, Cambridge, 1999.

[85] C. Priami. Stochastic π-calculus. *Computer Journal*, 38(7):578–589, 1995.

[86] M. Calder, S. Gilmore, and J. Hillston. Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. In *Transactions on Computational Systems Biology VII*, volume 4230 of *Lecture Notes in Computer Science*, pages 1–23. Springer Berlin Heidelberg, 2006.

[87] M. Calder, S. Gilmore, and J. Hillston. Automatically deriving odes from process algebra models of signalling pathways. In *Proceedings of CMSB 2005*, pages 204–215, 2005.

[88] M. Calder, S. Gilmore, J. Hillston, and V. Vyshemirsky. Formal methods for biochemical signalling pathways. In *Formal Methods: State of the Art and New Directions*, pages 185–215. Springer, 2006.

[89] M. Calder, A. Duguid, S. Gilmore, and J. Hillston. Stronger computational modelling of signalling pathways using both continuous and discrete-state methods. In *Proceedings of CMSB 2006*, volume 4210 of *LNCS*, pages 63–77, 2006.

[90] L. Gerosa. Stochastic process algebras as design and analysis framework for synthetic biology modelling. Master's thesis, University of Trento, 2007.

[91] F. Ciocchetta and J. Hillston. Bio-pepa: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33-34):3065–3084, 2009.

[92] A. Regev, E. M. Panina, W. Silverman, L. Cardelli, and E. Shapiro. BioAmbients: an abstraction for biological compartments. *Theoretical Computer Science*, 325(1):141 – 167, 2004.

[93] L. Dematté, C. Priami, and A. Romanel. The Blenx language: A tutorial. In *Formal Methods for Computational Systems Biology, SFM 2008*, number 5054 in LNCS, pages 123–138, 2008.

[94] M. Hucka, A. Finney, J. Bornstein, M. Keating, E. Shapiro, J. Matthews, L. Kovitz, J. Schilstra, A. Funahashi, 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*, 1:41–53, 2004.

[95] K.-H. Cho, K. H. Johansson, and O. Wolkenhauer. A hybrid systems framework for cellular processes. *Biosystems*, 80(3):273 – 282, 2005.

[96] P. V. Coveney and P. W. Fowler. Modelling biological complexity: a physical scientist's perspective. *Journal of The Royal Society Interface*, 2(4):267–280, 2005.

[97] A. Tiwari. Abstractions for hybrid systems. *Formal Methods in Systems Design*, 32:57–83, 2008.

[98] T. Henzinger. The theory of hybrid automata. In *Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science*, pages 278–292, 1996.

[99] H. Matsuno, M. Nagasaki, and S. Miyano. Hybrid Petri net based modeling for biological pathway simulation. *Natural Computing*, 10(3):1099–1120, 2011.

[100] S. Wolfram. *A new kind of science*. Champaign, IL: Wolfram Media, 2002.

[101] A. Deutsch and S. Dormann. *Cellular automata modeling of biological pattern formation*. Springer, 2009.

[102] G. B. Ermentrout and L. Edelstein-Keshet. Cellular automata approaches to biology. *Journal of Theoretical Biology*, 160:97–133, 1993.

[103] M. Soheilypour and M. Mofrad. Agent-based modeling in molecular systems biology. *BioEssays*, 40, 06 2018.

[104] M. R. Aniba and J. D. Thompson. Knowledge based expert systems in Bioinformatics. In P. Vizureanu, editor, *Expert Systems*, chapter 10. IntechOpen, Rijeka, 2010.

[105] F. Bernardini, M. Gheorghe, F. Romero-Campero, and N. Walkinshaw. A hybrid approach to modelling biological systems. In *Membrane Computing*, volume 4860 of *Lecture Notes in Computer Science*, pages 138–159. Springer, 2007.

[106] G. J. Holzmann. The model checker Spin. *IEEE Transactions on Software Engineering*, 23(5):275–295, 1997.

[107] A. Cimatti, E. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. NuSMV version 2: an open source tool for symbolic model checking. In *Proc. International Conference on Computer-Aided Verification (CAV 2002)*, volume 2404 of *LNCS*. Springer, 2002.

[108] G. Arellano, J. Argil, E. Azpeitia, M. Benitez, M. Carrillo, P. Gongora, D. Rosenblueth, and E. R. Alvarez-Buylla. Antelope: A hybrid-logic model checker for branching-time Boolean GRN analysis. *BMC Bioinformatics*, 12(1):1–15, 2011.

[109] A. Hinton, M. Kwiatkowska, G. Norman, and D. Parker. Prism: A tool for automatic verification of probabilistic systems. In *Proc. TACAS*, volume 3920 of *LNCS*, pages 441–444. Springer, 2006.

[110] R. Alur, T. Henzinger, F. Mang, S. Qadeer, S. Rajamani, and S. Tasiran. MOCHA: Modularity in model checking. In *Computer Aided Verification*, volume 1427 of *Lecture Notes in Computer Science*, pages 521–525. Springer Berlin Heidelberg, 1998.

[111] K. Bae, S. Escobar, and J. Meseguer. Abstract Logical Model Checking of Infinite-State Systems Using Narrowing. In F. van Raamsdonk, editor, *24th International Conference on Rewriting Techniques and Applications (RTA 2013)*, volume 21 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 81–96. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013.

[112] R. Donaldson and D. Gilbert. A Monte Carlo model checker for probabilistic LTL with numerical constraints. Research Report TR-2008-282, Dept. of Computing Science, University of Glasgow, 2008.

[113] J. Barnat, L. Brim, I. Cerna, S. Drazan, J. Fabrikova, J. Lanik, D. Safranek, and H. Ma. BioDiVinE: A framework for parallel analysis of biological models. In *Proceedings of 2nd International Workshop on Computational Models for Cell Processes*, volume 6 of *EPTCS*, 2009.

[114] J. Barnat, L. Brim, V. Havel, J. Havlícek, J. Kriho, M. Lenco, P. Rockai, V. Still, and J. Weiser. DiVinE 3.0 - an explicit-state model checker for multithreaded C & C++ programs. In *CAV*, pages 863–868, 2013.

[115] Y. Z. Paterson, D. Shorthouse, M. W. Pleijzier, N. Piterman, C. Bendtsen, B. A. Hall, and J. Fisher. A toolbox for discrete modelling of cell signalling dynamics. *Integrative Biology*, 10:370–382, 2018.

[116] O. Prvu and D. Gilbert. A novel method to verify multilevel computational models of biological systems using multiscale spatio-temporal meta model checking. *PLOS ONE*, 11(5):1–43, 2016.

[117] A. Sadot, J. Fisher, D. Barak, Y. Admanit, M. J. Stern, E. J. A. Hubbard, and D. Harel. Toward verified biological models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(2):223–234, 2008.

[118] D. Harel, H. Kugler, R. Marelly, and A. Pnueli. Smart Play-out of behavioral requirements. In *Formal Methods in Computer-Aided Design*, volume 2517 of *Lecture Notes in Computer Science*, pages 378–398. Springer Berlin Heidelberg, 2002.

[119] J. Burch, E. Clarke, K. McMillan, D. Dill, and L. Hwang. Symbolic model checking: 10E20 states and beyond. *Information and Computation*, 98(2):142 – 170, 1992.

[120] Z. Khalis, J. P. Comet, A. Richard, and G. Bernot. The SMBioNet method for discovering models of gene regulatory networks. *Genes, Genomes and Genomics*, 3:15–22, 2009.

[121] S. Konur, L. Mierl, F. Ipate, and M. Gheorghe. kpworkbench: A software suit for membrane systems. *SoftwareX*, 11:100407, 2020.

[122] M. Gheorghe, R. Ceterchi, F. Ipate, and S. Konur. Kernel P systems modelling, testing and verification - sorting case study. In A. Leporati, G. Rozenberg, A. Salomaa, and C. Zandron, editors, *Membrane Computing*, pages 233–250, Cham, 2017. Springer International Publishing.

[123] A. Ţurlea, M. Gheorghe, F. Ipate, and S. Konur. Search-based testing in membrane computing. *Journal of Membrane Computing*, 1(4):241–250, 2019.

[124] M. E. Bakir, F. Ipate, S. Konur, L. Mierlă, and I. Niculescu. Extended simulation and verification platform for kernel P systems. In M. Gheorghe, G. Rozenberg, A. Salomaa, P. Sosík, and C. Zandron, editors, *Membrane Computing*, pages 158–178, Cham, 2014. Springer International Publishing.

[125] M. E. Bakir, S. Konur, M. Gheorghe, I. Niculescu, and F. Ipate. High performance simulations of kernel P systems. In *16th IEEE Int. Conference on High Performance Computing and Communications*, pages 409–412, 2014.

[126] C. Dragomir, F. Ipate, S. Konur, R. Lefticaru, and L. Mierlă. Model checking kernel P systems. In *14th Int. Conference on Membrane Computing*, volume 8340 of *LNCS*, pages 151–172. Springer, 2013.

[127] M. Gheorghe, S. Konur, F. Ipate, L. Mierla, M. E. Bakir, and M. Stannett. An integrated model checking toolset for kernel P systems. In *Membrane Computing*, pages 153–170. Springer, 2015.

[128] F. Fages and S. Soliman. On robustness computation and optimization in Biocham-4. In M. Češka and D. Šafránek, editors, *Computational Methods in Systems Biology*, pages 292–299, Cham, 2018. Springer International Publishing.

[129] C. Schroter, S. Schwoon, and J. Esparza. The model-checking kit. In *Applications and Theory of Petri Nets 2003*, volume 2679 of *Lecture Notes in Computer Science*, pages 463–472. Springer Berlin Heidelberg, 2003.

[130] G. Batt, B. Besson, P.-E. Ciron, H. Jong, E. Dumas, J. Geiselmann, R. Monte, P. Monteiro, M. Page, F. Rechenmann, and D. Ropers. Genetic network analyzer: a tool for the qualitative modeling and simulation of bacterial regulatory networks. In *Bacterial Molecular Networks*, volume 804 of *Methods in Molecular Biology*, pages 439–462. Springer New York, 2012.

[131] H. Garavel, F. Lang, R. Mateescu, and W. Serwe. CADP 2010: A toolbox for the construction and analysis of distributed processes. In *Tools and Algorithms for the Construction and Analysis of Systems*, volume 6605 of *Lecture Notes in Computer Science*, pages 372–387. Springer Berlin Heidelberg, 2011.

[132] M. Rybinski, M. Lula, P. Banasik, S. Lasota, and A. Gambin. Tav4SB: integrating tools for analysis of kinetic models of biological systems. *BMC Systems Biology*, 6(1):25, 2012.

[133] G. Batt, B. Yordanov, R. Weiss, and C. Belta. Robustness analysis and tuning of synthetic gene networks. *Bioinformatics*, 23(18):2415–2422, 2007.

[134] A. David, K. G. Larsen, A. Legay, M. Mikucionis, D. B. Poulsen, and S. Sedwards. Statistical model checking for biological systems. *Int. J. Softw. Tools Technol. Transf.*, 17(3):351–367, 2015.

[135] U. Rauf, U. Siddique, J. Ahmad, and U. Niazi. Formal modeling and analysis of biological regulatory networks using Spin. In *IEEE International Conference on Bioinformatics and Biomedicine*, pages 304–308, 2011.

[136] D. Bošnački and R. S. Mans. *Modeling and analysis of biological networks with model checking*, pages 915–940. John Wiley & Sons, Inc., 2011.

[137] S. Konur and M. Gheorghe. Design and analysis of genetically constructed logic gates, June 2014.

[138] S. Konur, M. Kiran, M. Gheorghe, M. Burkitt, and F. Ipate. Agent-based high-performance simulation of biological systems on the GPU. In *2015 IEEE 17th International Conference on High Performance Computing and Communications*, pages 84–89, 2015.

[139] Romulus apolzan, 2005. Rapid Prototyping Applications of Formal Reasoning Tools to Biological Cellular Signalling Networks, University of New England.

[140] I. Shmulevich, H. Lähdesmäki, E. R. Dougherty, J. Astola, and W. Zhang. The role of certain post classes in boolean network models of genetic networks. *Proceedings of the National Academy of Sciences*, 100(19):10734–10739, 2003.

[141] R. Albert and H. G. Othmer. The topology of the regulatory interactions predicts the expression pattern of the segment polarity genes in Drosophila melanogaster. *Journal of Theoretical Biology*, 223(1):1 – 18, 2003.

[142] 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*, 101(14):4781–4786, 2004.

[143] S. Martin, Z. Zhang, A. Martino, and J.-L. Faulon. Boolean dynamics of genetic regulatory networks inferred from microarray time series data. *Bioinformatics*, 23(7):866–874, 2007.

[144] S. Bornholdt. Boolean network models of cellular regulation: prospects and limitations. *Journal of The Royal Society Interface*, 5(Suppl 1):S85–S94, 2008.

[145] C. Hong, M. Lee, D. Kim, D. Kim, K.-H. Cho, and I. Shin. A checkpoints capturing timing-robust Boolean model of the budding yeast cell cycle regulatory network. *BMC Systems Biology*, 6(1):129, 2012.

[146] I. Albert, J. Thakar, S. Li, R. Zhang, and R. Albert. Boolean network simulations for life scientists. *Source Code for Biology and Medicine*, 3(1):16, 2008.

[147] C. Müssel, M. Hopfensitz, and H. A. Kestler. BoolNet – an R package for generation, reconstruction and analysis of Boolean networks. *Bioinformatics*, 26(10):1378–1380, 2010.

[148] H. Klarner, A. Streck, and H. Siebert. PyBoolNet: a Python package for the generation, analysis and visualization of boolean networks. *Bioinformatics*, 33(5):770–772, 2016.

[149] M. Schaub, T. Henzinger, and J. Fisher. Qualitative networks: a symbolic approach to analyze biological signaling networks. *BMC Systems Biology*, 1(1):4, 2007.

[150] S. A. Dabydeen, A. Desai, and D. Sahoo. Unbiased boolean analysis of public gene expression data for cell cycle gene identification. *Molecular Biology of the Cell*, 30(14):1770–1779, 2019.

[151] C. J. Langmead and S. K. Jha. Symbolic approaches for finding control strategies in boolean networks. *J. Bioinformatics and Computational Biology*, 7(2):323–338, 2009.

[152] A. Richard, J.-P. Comet, and G. Bernot. Formal methods for modeling biological regulatory networks. In *Modern Formal Methods and Applications*, pages 83–122. Springer Netherlands, 2006.

[153] A. Richard, G. Rossignol, J.-P. Comet, G. Bernot, J. Guespin-Michel, and A. Merieau. Boolean models of biosurfactants production in *pseudomonas fluorescens*. *PLoS ONE*, 7(1):e24651, 01 2012.

[154] S. Khalid, R. Hanif, S. H. Tareen, A. Siddiqa, Z. Bibi, and J. Ahmad. Formal modeling and analysis of ER-α associated biological regulatory network in breast cancer. *PeerJ*, 4:e2542, 2016.

[155] C. Chaouiya, A. Naldi, and D. Thieffry. *Logical Modelling of Gene Regulatory Networks with GINsim*, volume 804, pages 463–79. Springer, New York, NY, 2012.

[156] A. Naldi, D. Berenguier, A. Faure, F. Lopez, D. Thieffry, and C. Chaouiya. Logical modelling of regulatory networks with {GINsim} 2.3. *Biosystems*, 97(2):134 – 139, 2009.

[157] J. Ahmad, G. Bernot, J.-P. Comet, D. Lime, and O. Roux. Hybrid modelling and dynamical analysis of gene regulatory networks with delays. *Complexus*, 3(4):231–251, 2006.

[158] J. Ahmad, J. Bourdon, D. Eveillard, J. Fromentin, O. Roux, and C. Sinoquet. Temporal constraints of a gene regulatory network: refining a qualitative simulation. *Biosystems*, 98(3):149 – 159, 2009.

[159] T. A. Henzinger, P.-H. Ho, and H. Wong-Toi. HyTech: A model checker for hybrid systems. In *Computer Aided Verification*, volume 1254 of *Lecture Notes in Computer Science*, pages 460–463. Springer Berlin Heidelberg, 1997.

[160] J. Fromentin, J.-P. Comet, P. Le Gall, and O. Roux. Analysing gene regulatory networks by both constraint programming and model-checking. In *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, pages 4595–4598, 2007.

[161] D. Mateus, J.-P. Gallois, J.-P. Comet, and P. L. Gall. Symbolic modeling of genetic regulatory networks. *Journal of Bioinformatics and Computational Biology*, 05(02b):627–640, 2007.

[162] C. Bigot, A. Faivre, J.-P. Gallois, A. Lapitre, D. Lugato, J.-Y. Pierron, and N. Rapin. Automatic test generation with Agatha. In *Proceedings of the 9th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, TACAS'03, pages 591–596. Springer-Verlag, 2003.

[163] E. Dubrova, M. Teslenko, and L. Ming. Finding attractors in synchronous multiple-valued networks using SAT-based bounded model checking. In *40th IEEE International Symposium on Multiple-Valued Logic (ISMVL)*, pages 144–149, 2010.

[164] R. Barbuti, S. Cataudella, A. Maggiolo-Schettini, P. Milazzo, and A. Troina. A probabilistic model for molecular systems. *Fundamenta Informaticae*, 67(1-3):13–27, 2005.

[165] M. Calder, V. Vyshemirsky, D. Gilbert, and R. Orton. Analysis of signalling pathways using continuous time Markov chains. In *Transactions on Computational Systems Biology VI*, volume 4220 of *Lecture Notes in Computer Science*, pages 44–67. Springer Berlin Heidelberg, 2006.

[166] D. Gilbert, M. Heiner, and S. Lehrack. A unifying framework for modelling and analysing biochemical pathways using Petri nets. In *Proceedings of the 2007 International Conference on Computational Methods in Systems Biology*, CMSB'07, pages 200–216. Springer-Verlag, 2007.

[167] M. Kwiatkowska, G. Norman, and D. Parker. Using probabilistic model checking in systems biology. *ACM SIGMETRICS Performance Evaluation Review*, 35(4):14–21, 2008.

[168] M. Kwiatkowska, G. Norman, and D. Parker. *Symbolic Systems Biology*, chapter Probabilistic Model Checking for Systems Biology, pages 31–59. Jones and Bartlett, 2010.

[169] T. Pronk, E. de Vink, D. Bosnacki, and T. Breit. Stochastic modeling of codon bias with Prism. In *Proc. MTCoord 2007, Paphos*. Computer Science Department, University of Cyprus, Nicosia, 2007.

[170] D. Chu and I. C. Blomfield. Orientational control is an efficient control mechanism for phase switching in the *E. coli* film system. *Journal of Theoretical Biology*, 244(3):541 – 551, 2007.

[171] C. J. Langmead and S. K. Jha. Predicting protein folding kinetics via temporal logic model checking. In *Algorithms in Bioinformatics*, volume 4645 of *Lecture Notes in Computer Science*, pages 252–264. Springer Berlin Heidelberg, 2007.

[172] D. Bosnacki, T. E. Pronk, and E. P. de Vink. In silico modelling and analysis of ribosome kinetics and aa-trna competition. In *Transactions on Computational Systems Biology XI*, volume 5750 of *Lecture Notes in Computer Science*, pages 69–89. Springer, 2009.

[173] H. M. M. T. Eikelder, M. N. Steijaert, and E. P. D. Vink. Stochastic analysis of amino acid substitution in protein synthesis. In *Proc. CMSB 2008*, pages 367–386, 2008.

[174] P. Ballarini, R. Mardare, and I. Mura. Analysing biochemical oscillation through probabilistic model checking. *Electronic Notes in Theoretical Computer Science*, 229(1):3 – 19, 2009.

[175] M. P. Dobay, A. Dobay, J. Bantang, and E. Mendoza. How many trimers? modeling influenza virus fusion yields a minimum aggregate size of six trimers, three of which are fusogenic. *Mol. BioSyst.*, 7:2741–2749, 2011.

[176] P. Liò, E. Merelli, and N. Paoletti. Multiple verification in computational modeling of bone pathologies. In *Proc. 3rd International Workshop on Computational Models for Cell Processes (COMPMOD'11)*, volume 68 of *EPTCS*, pages 82–96, 2011.

[177] M. A. Crepalde, A. C. Faria-Campos, and S. V. A. Campos. Modeling and analysis of cell membrane systems with probabilistic model checking. *BMC Genomics*, 12(4), 2011.

[178] F. Ciocchetta, S. Gilmore, M. L. Guerriero, and J. Hillston. Integrated simulation and model-checking for the analysis of biochemical systems. *Electron. Notes Theor. Comput. Sci.*, 232:17–38, 2009.

[179] M. Lakin, D. Parker, L. Cardelli, M. Kwiatkowska, and A. Phillips. Design and analysis of DNA strand displacement devices using probabilistic model checking. *Journal of the Royal Society Interface*, 9(72):1470–1485, 2012.

[180] Q. Yuan, P. Trairatphisan, J. Pang, S. Mauw, M. Wiesinger, and T. Sauter. Probabilistic model checking of the PDGF signalling pathway. In *Transactions on Computational Systems Biology XIV*, volume 7625 of *Lecture Notes in Computer Science*, pages 151–180. Springer, 2012.

[181] R. Donaldson and M. Calder. Modular modelling of signalling pathways and their cross-talk. *Theor. Comput. Sci.*, 456:30–50, 2012.

[182] O. Andrei and M. Calder. Trend-based analysis of a population model of the AKAP scaffold protein. In *Transactions on Computational Systems Biology XIV*, volume 7625 of *Lecture Notes in Computer Science*, pages 1–25. Springer, 2012.

[183] F. A. F. Braz, J. S. Cruz, A. C. Faria-Campos, and S. V. A. Campos. Probabilistic model checking analysis of Palytoxin effects on cell energy reactions of the Na+/K+-atpase. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 10(6):1530–1541, 2013.

[184] S. Konur, C. Ladroue, H. Fellermann, D. Sanassy, L. Mierla, F. Ipate, S. Kalvala, M. Gheorghe, and N. Krasnogor. Modeling and analysis of genetic boolean gates using infobiotics workbench. In *Verification of Engineered Molecular Devices and Programs, Proceedings of VEMDP14, Vienna, Austria*, pages 26–37, 2014.

[185] J. Fisher, N. Piterman, A. Hajnal, and T. A. Henzinger. Predictive modeling of signaling crosstalk during *c. elegans* vulval development. *PLoS Comput Biol*, 3(5):e92, 2007.

[186] C. Li, M. Nagasaki, K. Ueno, and S. Miyano. Simulation-based model checking approach to cell fate specification during *caenorhabditis elegans* vulval development by hybrid functional petri net with extension. *BMC Systems Biology*, 3(1):42, 2009.

[187] D. Harel and E. Gery. Executable object modeling with Statecharts. In *Proceedings of the 18th International Conference on Software Engineering*, pages 246–257, 1996.

[188] N. Kam, I. Cohen, and D. Harel. The immune system as a reactive system: modeling T cell activation with Statecharts. In *Proceedings IEEE Symposia on Human-Centric Computing Languages and Environments, 2001*, pages 15–22, 2001.

[189] J. Fisher, N. Piterman, E. J. A. Hubbard, M. J. Stern, and D. Harel. Computational insights into *caenorhabditis elegans* vulval development. *PNAS*, 102(6):1951–1956, 2005.

[190] D. Harel and R. Marelly. *Come, Let's Play: Scenario-based Programming Using LSC's and the Play-engine*. Springer-Verlag New York, Inc., 2003.

[191] F. J. Romero-Campero, J. Twycross, H. Cao, J. Blakes, and N. Krasnogor. A multiscale modeling framework based on P systems. In *WMC9 2008*, pages 63–77. Springer-Verlag Berlin, 2009.

[192] M. Gheorghe, R. Ceterchi, F. Ipate, S. Konur, and R. Lefticaru. Kernel P systems: from modelling to verification and testing. *Theoretical Computer Science*, 724:45–60, 2018.

[193] M. Gheorghe, F. Ipate, and S. Konur. Testing based on identifiable P systems using cover automata and X-machines. *Information Sciences*, 372:565 – 578, 2016.

[194] V. Manca. The metabolic algorithm for P systems: principles and applications. *Theor. Comput. Sci.*, 404(1-2):142–155, 2008.

[195] J. Jack and A. Păun. Discrete modeling of biochemical signaling with memory enhancement. In *Transactions on Computational Systems Biology XI*, volume 5750 of *Lecture Notes in Computer Science*, pages 200–215. Springer Berlin Heidelberg, 2009.

[196] D. Pescini, D. Besozzi, G. Mauri, and C. Zandron. Dynamic probabilistic P systems. *International Journal of Foundations of Computer Science*, 1(17):183–204, 2006.

[197] D. Besozzi, P. Cazzaniga, G. Mauri, and D. Pescini. BioSimWare : A software for the modeling, simulation and analysis of biological systems. In *CMC 2010, LNCS 6501*, pages 119–143. Springer, 2010.

[198] M. A. M. del Amor, I. Pérez-Hurtado, M. J. Pérez-Jiménez, A. Riscos-Núñez, and F. Sancho-Caparrini. A formal verification algorithm for multienvironment probabilistic P systems. *International Journal of Foundations of Computer Science*, 22(1):107–118, 2011.

[199] I. Pérez-Hurtado, L. Valencia, M. Pérez-Jiménez, M. Colomer, and Riscos-Núńez. A general purpose software tool for simulating biological phenomena by means of P systems. In *Proceedings 2010 IEEE Fifth International Conference BIC-Ta 2010, Changsha, China*, pages 637–643, 2010.

[200] M. Cavaliere, T. Mazza, and S. Sedwards. Statistical model checking of membrane systems with peripheral proteins: quantifying the role of estrogen in cellular mitosis and DNA damage. In P. Frisco, M. Gheorghe, and M. Pérez-Jiménez, editors, *Applications of Membrane Systems to Biology*, Emergence, Complexity and Computation. Springer, 2013.

[201] F. Nedelec and D. Foethke. Collective langevin dynamics of flexible cytoskeletal fibers. *New Journal of Physics*, 9(11):427, 2007.

[202] T. Hinze, T. Lenser, G. Escuela, I. Heiland, and S. Schuster. Modelling signalling networks with incomplete information about protein activation states: A P system framework for KaiABC oscillator. In *Workshop on Membrane Computing, LNCS 5957*, LNCS, pages 316–334, 2010.

[203] J. Smaldon, F. J. Romero-Campero, F. Fernandez Trillo, M. Gheorghe, C. Alexander, and N. Krasnogor. A computational study of liposome logic: towards cellular computing from the bottom up. *Systems and Synthetic Biology*, 4(3):157–179, 2010.

[204] J. Twycross, L. R. Band, M. J. Bennett, J. R. King, and N. Krasnogor. Stochastic and deterministic multiscale models for systems biology: an auxin-transport case study. *BMC Systems Biology*, 4(1), 2010.

[205] M. Gheorghe, V. Manca, and F. J. Romero-Campero. Deterministic and stochastic P systems for modelling cellular processes. *Natural Computing*, 9(2):457–473, 2010.

[206] S. Konur and M. Gheorghe. A property-driven methodology for formal analysis of synthetic biology systems. In *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, volume 12, pages 360–371, 2015.

[207] P. Prusinkiewicz. Modeling of spatial structure and development of plants: a review. *Scientia Horticulturae*, 74(1-2):113 – 149, 1998.

[208] M. J. Fisher, G. Malcolm, and R. C. Paton. Spatio-logical processes in intracellular signalling. *Biosystems*, 55(1-3):83 – 92, 2000.

[209] C. Talcott, S. Eker, M. Knapp, P. Lincoln, and K. Laderoute. Pathway logic modeling of protein functional domains in signal transduction. In *Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE*, pages 618–619, 2003.

[210] W. Cools, S. Rijkers, S. S. Mobarakeh, and M. van den Heuvel. Report on formal models of two biological processes. Technical report, University of Twente, 2008.

[211] C. Talcott. Pathway logic. In *Proceedings of the Formal Methods for the Design of Computer, Communication, and Software Systems 8th International Conference on Formal Methods for Computational Systems Biology*, SFM'08, pages 21–53. Springer-Verlag, 2008.

[212] M. Baggi, D. Ballis, and M. Falaschi. Quantitative pathway logic for computational biology. In *Computational Methods in Systems Biology*, volume 5688 of *Lecture Notes in Computer Science*, pages 68–82. Springer Berlin Heidelberg, 2009.

[213] D. L. Dill, M. A. Knapp, P. Gage, C. Talcott, K. Laderoute, and P. Lincoln. The Pathalyzer: A tool for analysis of signal transduction pathways. In *Proceedings of the 2005 Joint Annual Satellite Conference on Systems Biology and Regulatory Genomics*, RECOMB'05, pages 11–22. Springer-Verlag, 2007.

[214] K. Schmidt. LoLA: A low level analyser. In *Application and Theory of Petri Nets 2000*, volume 1825 of *Lecture Notes in Computer Science*, pages 465–474. Springer Berlin Heidelberg, 2000.

[215] S. Eker, M. Knapp, K. Laderoute, P. Lincoln, and C. Talcott. Pathway logic: executable models of biological networks. *Electronic Notes in Theoretical Computer Science*, 71(0):144 – 161, 2004.

[216] G. Santos-Garca, C. Talcott, and J. De Las Rivas. Analysis of cellular proliferation and survival signaling by using two ligand/receptor systems modeled by pathway logic. In *LNCS*, volume 9271, pages 226–245, 09 2015.

[217] C. Talcott and M. Knapp. Explaining response to drugs using pathway logic. In J. Feret and H. Koeppl, editors, *Computational Methods in Systems Biology*, pages 249–264, Cham, 2017. Springer International Publishing.

[218] K. Jo, B. S. Buitrago, M. Kim, S. Rhee, C. Talcott, and S. Kim. Logic-based analysis of gene expression data predicts pathway crosstalk between TNF, TGFB1 and EGF in basal-like breast cancer. *bioRxiv*, 2019.

[219] D. Barua, J. R. Faeder, and J. M. Haugh. A bipolar clamp mechanism for activation of Jak-family protein tyrosine kinases. *PLoS Comput Biol*, 5(4):e1000364, 2009.

[220] T. M. Thomson, K. R. Benjamin, A. Bush, T. Love, D. Pincus, O. Resnekov, R. C. Yu, A. Gordon, A. Colman-Lerner, D. Endy, and R. Brent. Scaffold number in yeast signaling system sets tradeoff between system output and dynamic range. *Proceedings of the National Academy of Sciences*, 2011.

[221] P. Kocieniewski, J. R. Faeder, and T. Lipniacki. The interplay of double phosphorylation and scaffolding in MAPK pathways. *Journal of Theoretical Biology*, 295(0):116 – 124, 2012.

[222] D. Barua, W. S. Hlavacek, and T. Lipniacki. A computational model for early events in B cell antigen receptor signaling: analysis of the roles of Lyn and Fyn. *The Journal of Immunology*, 2012.

[223] S. M. Abel, J. P. Roose, J. T. Groves, A. Weiss, and A. K. Chakraborty. The membrane environment can promote or suppress bistability in cell signaling networks. *The Journal of Physical Chemistry B*, 116(11):3630–3640, 2012.

[224] B. Liu and J. R. Faeder. Parameter estimation of rule-based models using statistical model checking. In *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1453–1459, 2016.

[225] F. Fages and A. Rizk. On temporal logic constraint solving for analyzing numerical data time series. *Theor. Comput. Sci.*, 408(1):55–65, 2008.

[226] A. Rizk, G. Batt, F. Fages, and S. Soliman. A general computational method for robustness analysis with applications to synthetic gene networks. *Bioinformatics*, 25(12):i169–i178, 2009.

[227] R. Belle, S. Prigent, A. Siegel, and P. Cormier. Model of cap-dependent translation initiation in sea urchin: A step towards the eukaryotic translation regulation network. *Molecular Reproduction and Development*, 77(3):257–264, 2010.

[228] A. Baudier, F. Fages, and S. Soliman. Graphical requirements for multistationarity in reaction networks and their verification in biomodels. *Journal of Theoretical Biology*, 459:79 – 89, 2018.

[229] V. N. Reddy, M. L. Mavrovouniotis, and M. N. Liebman. Petri net representations in metabolic pathways. In *Proc Int Conf Intell Syst Mol Biol*, pages 328–336, 1993.

[230] V. N. Reddy, M. N. Liebman, and M. L. Mavrovouniotis. Qualitative analysis of biochemical reaction systems. *Comput. Biol. Med.*, 26:9–24, 1996.

[231] C. Rohr. *Simulative Model Checking of Steady State and Time-Unbounded Temporal Operators*, pages 142–158. Springer Berlin Heidelberg, 2013.

[232] D. Gilbert, H. Fuss, X. Gu, R. Orton, S. Robinson, V. Vyshemirsky, M. J. Kurth, C. S. Downes, and W. Dubitzky. Computational methodologies for modelling, analysis and simulation of signalling networks. *Brief. Bioinformatics*, 7:339–353, 2006.

[233] C. Li, M. Nagasaki, C. H. Koh, and S. Miyano. Online model checking approach based parameter estimation to a neuronal fate decision simulation model in Caenorhabditis elegans with hybrid functional petri net with extension. *Mol. BioSyst.*, 7:1576–1592, 2011.

[234] P. Ballarini, E. Gallet, P. Le Gall, and M. Manceny. Formal analysis of the Wnt/β-catenin pathway through statistical model checking. In *International Symposium On Leveraging Applications of Formal Methods, Verification and Validation*, volume 8803. Springer International Publishing, 2014.

[235] J. Will and M. Heiner. Petri nets in biology, chemistry, and medicine. bibliography. Technical report, Brandenbury University of Technology at Cottbus, 2002.

[236] C. Chaouiya. Petri net modelling of biological networks. *Briefings in Bioinformatics*, 8(4):210–219, 2007.

[237] R. V. Carvalho, F. J. Verbeek, and C. J. Coelho. *Bio-modeling Using Petri Nets: A Computational Approach*, pages 3–26. Springer International Publishing, Cham, 2018.

[238] M. Heiner, M. Herajy, F. Liu, C. Rohr, and M. Schwarick. Snoopy - a unifying Petri net tool. In *Application and Theory of Petri Nets*, volume 7347 of *Lecture Notes in Computer Science*, pages 398–407. Springer Berlin Heidelberg, 2012.

[239] M. Nagasaki, A. Saito, E. Jeong, C. Li, K. Kojima, E. Ikeda, and S. Miyano. Cell illustrator 4.0: A computational platform for systems biology. In *Biological Petri Nets*, volume 162 of *Studies in Health Technology and Informatics*, pages 160–181. Springer Berlin Heidelberg, 2008.

[240] Petri nets tool database. url: http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/db.html.

[241] F. Kordon, A. Linard, D. Buchs, M. Colange, S. Evangelista, K. Lampka, N. Lohmann, E. Paviot-Adet, Y. Thierry-Mieg, and H. Wimmel. Report on the model checking contest at Petri nets 2011. In *Transactions on Petri Nets and Other Models of Concurrency VI*, volume 7400 of *Lecture Notes in Computer Science*, pages 169–196. Springer Berlin Heidelberg, 2012.

[242] 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*, 80(1):25 – 31, 2001.

[243] A. Phillips, L. Cardelli, and G. Castagna. A graphical representation for biological processes in the stochastic π-calculus. *Transactions in Computational Systems Biology*, 4230:123–152, 2006.

[244] L. Dematté, C. Priami, and A. Romanel. The Beta workbench: a computational tool to study the dynamics of biological systems. *Briefings in Bioinformatics*, 9(5):437–49, 2008.

[245] P. Lecca, A. Palmisano, A. Ihekwaba, and C. Priami. Calibration of dynamic models of biological systems with KInfer. *European Biophysics Journal*, 39(6):1019–1039, 2010.

[246] F. Ciocchetta and J. Hillston. Bio-pepa for epidemiological models. *Electronic Notes in Theoretical Computer Science*, 261(0):43 – 69, 2010.

[247] F. Ciocchetta, M. L. Guerriero, and J. Hillston. Investigating modularity in the analysis of process algebra models of biochemical systems. In *FBTC*, pages 55–69, 2010.

[248] O. E. Akman, M. L. Guerriero, L. Loewe, and C. Troein. Complementary approaches to understanding the plant circadian clock. In *FBTC*, pages 1–19, 2010.

[249] C. Banks, D. Seaton, and I. Stark. Analysis of a post-translational oscillator using process algebra and spatio-temporal logic. In *Computational Methods in Systems Biology*, Lecture Notes in Computer Science, pages 222–238. Springer International Publishing, 2015.

[250] J. Coulter. *Process Algebra with Layers: Multi-scale Integration Modelling Applied to Cancer Therapy*, pages 118–133. Springer, 2017.

[251] D. A. Zaitsev, T. R. Shmeleva, and J. F. Groote. Verification of hypertorus communication grids by infinite petri nets and process algebra. *IEEE/CAA Journal of Automatica Sinica*, 6(3):733–742, 2019.

[252] D. Hamami, B. Atmani, R. Cameron, K. G. Pollock, and C. Shankland. Improving process algebra model structure and parameters in infectious disease epidemiology through data mining. *J. Intell. Inf. Syst.*, 52(3):477–499, 2019.

[253] N. Tenazinha and S. Vinga. A survey on methods for modeling and analyzing integrated biological networks. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 8(4):943–958, July 2011.

[254] D. D. Nguyen, Z. Cang, and G.-W. Wei. A review of mathematical representations of biomolecular data. *Phys. Chem. Chem. Phys.*, 22:4343–4367, 2020.

[255] G. Batt, D. Ropers, H. De Jong, J. Geiselmann, R. Mateescu, M. Page, and D. Schneider. Validation of qualitative models of genetic regulatory networks by model checking: analysis of the nutritional stress response in *escherichia coli*. *Bioinformatics*, 21(1):19–28, 2005.

[256] J. Barnat, L. Brim, I. Cerna, S. Drazan, and D. Safranek. From simple regulatory motifs to parallel model checking of complex transcriptional networks. In *Proceedings of PDMC 2008 - Parallel and Distributed Methods Ins Verification*, pages 83–96, 2008.

[257] J. Barnat, L. Brim, I. Cerna, S. Drazan, J. Fabrikova, and D. Safranek. On algorithmic analysis of transcriptional regulation by LTL model checking. *Theoretical Computer Science*, 410(33–34):3128–3148, 2009.

[258] M. Hajnal. Formal methods for model selection in systems biology, 2018.

[259] M. Antoniotti, I. T. Lau, and B. Mishra. Naturally speaking: A system biology tool with natural language based interfaces. In *Biological Language Conference*, 2004.

[260] B. Mishra, M. Antoniotti, S. Paxia, and N. Ugel. Chapter 5 - Simpathica: A computational systems biology tool within the VALIS bioinformatics environment. In *Computational Systems Biology*, pages 79 – 102. Academic Press, 2006.

[261] B. Mishra. Intelligently deciphering unintelligible designs: Algorithmic algebraic model checking in systems biology. *Journal of the Royal Society, Interface / the Royal Society*, 6:575–97, 05 2009.

[262] G. Batt, D. Bergamini, H. de Jong, H. Garavel, and R. Mateescu. Model checking genetic regulatory networks using GNA and CADP. In *Model Checking Software*, volume 2989 of *Lecture Notes in Computer Science*, pages 158–163. Springer Berlin Heidelberg, 2004.

[263] P. Monteiro, E. Dumas, B. Besson, R. Mateescu, M. Page, A. Freitas, and H. de Jong. A service-oriented architecture for integrating the modeling and formal verification of genetic regulatory networks. *BMC Bioinformatics*, 10(1):1–12, 2009.

[264] R. Mateescu, P. T. Monteiro, E. Dumas, and H. de Jong. CTRL: Extension of CTL with regular expressions and fairness operators to verify genetic regulatory networks. *Theor. Comput. Sci.*, 412(26):2854–2883, 2011.

[265] A. Funahashi, Y. Matsuoka, A. Jouraku, M. Morohashi, N. Kikuchi, and H. Kitano. CellDesigner 3.5: A versatile modeling tool for biochemical networks. *Proceedings of the IEEE*, 96(8):1254–1265, 2008.

[266] I. I. Moraru, J. C. Schaff, B. M. Slepchenko, and L. L. M. The virtual cell: an integrated modeling environment for experimental and computational cell biology. *Annals of the New York Academy of Sciences*, 971:595–596, 2002.

[267] 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*, 22(24):3067–3074, 2006.

[268] C. H. Koh, M. Nagasaki, A. Saito, C. Li, L. Wong, and S. Miyano. MIRACH: efficient model checker for quantitative biological pathway models. *Bioinformatics*, 27(5):734–735, 2011.

[269] L. Watanabe, T. NGUYEN, M. Zhang, Z. Zundel, Z. Zhang, C. Madsen, N. Roehner, and C. Myers. iBioSim 3: A tool for model-based genetic circuit design. *ACS Synthetic Biology*, 8, 06 2018.

[270] R. Ghosh, A. Tiwari, and C. Tomlin. Automated symbolic reachability analysis: with application to Delta-notch signaling automata. In *Proceedings of the 6th International Conference on Hybrid Systems: Computation and Control*, HSCC'03, pages 233–248. Springer-Verlag, 2003.

[271] P. Lincoln and A. Tiwari. Symbolic systems biology: hybrid modeling and analysis of biological networks. In *Hybrid Systems: Computation and Control HSCC*, volume 2993 of *LNCS*, pages 660–672. Springer, 2004.

[272] A. Halasz, V. Kumar, M. Imielinski, C. Belta, O. Sokolsky, S. Pathak, and H. Rubin. Analysis of lactose metabolism in E.coli using reachability analysis of hybrid systems. *Systems Biology, IET*, 1(2):130–148, 2007.

[273] J. Wu and E. Voit. Hybrid modeling in biochemical systems theory by means of functional Petri nets. *Journal of Bioinformatics and Computational Biology*, 07(01):107–134, 2009.

[274] X. Li, O. Omotere, L. Qian, and E. Dougherty. Review of stochastic hybrid systems with applications in biological systems modeling and analysis. *EURASIP Journal on Bioinformatics and Systems Biology*, 2017, 12 2017.