On Verifying Bio-PEPA Models

Allan Clark
School of Informatics
University of Edinburgh, UK
a.d.clark@ed.ac.uk

Stephen Gilmore*
LFCS and Centre for Systems Biology at Edinburgh
University of Edinburgh, UK
stg@inf.ed.ac.uk

Maria Luisa Guerriero
Centre for Systems Biology at Edinburgh
University of Edinburgh, UK
mguerrie@inf.ed.ac.uk

Peter Kemper†
Department of Computer Science
College of William and Mary
Williamsburg, VA 23187, USA
kemper@cs.wm.edu

ABSTRACT

Verifying that a computational model implements the conceptual model of some dynamic biological phenomena is an important yet non-trivial task. In this paper, we discuss a variety of steps that contribute to this verification process, using the Bio-PEPA process algebra as a modelling language and describing the verification steps that are supported by the Bio-PEPA tool. In particular, we elaborate on both static analysis based on the structure of models and dynamic analysis of generated stochastic simulation traces performed using the Traviando trace analyser. We illustrate the approach with a model of a JAK/STAT signalling pathway.

Categories and Subject Descriptors

G.3 [**Mathematics of Computing**]: Probability and Statistics—*Stochastic processes*; I.6.4 [**Computing Methodologies**]: Simulation and Modelling—*Model Validation and Analysis*; I.6.6 [**Computing Methodologies**]: Simulation and Modelling—*Simulation Output Analysis*

General Terms

Dynamic modelling, stochastic simulation, trace analysis, static analysis.

*The Centre for Systems Biology at Edinburgh is a Centre for Integrative Systems Biology (CISB) funded by BBSRC and EPSRC, reference BB/D019621/1.

†This work is partially funded by a Distinguished Visitor fellowship from the Scottish Informatics and Computer Science Alliance (SICSA).

CMSB '10, September 29 – October 1, Trento, Italy

Keywords

Bio-PEPA, Traviando, Gillespie algorithm, verification.

1. INTRODUCTION AND MOTIVATIONS

Verification and validation of a model is the process of checking that the model represents correctly the system as described in its specification in order to substantiate that it "possesses sufficient representational and behavioural accuracy" [1]. This process, hence, is a crucial phase of model analysis which must be performed before scientific conclusions are based on the results which are obtained from this model.

In biochemical modelling, specifications are often given in terms of informal diagrams representing known interactions between the various chemical species which populate the system. The translation of this kind of specification into computational models is an error-prone process. Moreover, validation of models by comparison with the expected behaviour of the system is hard, since the behaviour of the system globally is often unknown. However, there is generally some existing knowledge about the behaviour of individual components or about relative ratios in their amounts; though such information cannot be explicitly encoded in models or directly used for comparison with simulation results, it still can be exploited in model verification and validation.

The use of a formal language where models must have a precise structure (e.g. process algebra or Petri nets) can greatly help in reducing the number of errors introduced in model definition. Indeed, static analysis on the syntax and the structure of a model provides a quick method to identify common errors such as trivial typos in variable names or rates, to check for the presence of deadlocks or for expected causal and ordering relations between events.

However, this kind of structural analysis is qualitative and hence it does not guarantee the absence of errors arising from quantitative problems, such as typos in parameters or incorrectly specified dynamic interactions. Therefore, dynamic analysis is necessary to complete the verification and validation of a model.

Moreover, if the focus is on stochastic processes – which is increasingly the case in biochemical modelling, especially in

the investigation of small-scale phenomena – the variability of the system must also be taken into account, and thus the behaviour of the system is generally obtained by computing ensemble measures. This can be done in two different ways:

- verification via model-checking: exact, but often not practical due to the exponential blow-up in the size of the reachable state-space of the system; infeasible if there are unknown parameters or if experimentation with different conditions is desired;
- testing via stochastic simulation: involves the identification of a certain number of cases to analyse: these can be either a big enough sample of random cases or a selection of "interesting" cases.

In the stochastic setting dynamic analysis is, hence, even more important, since by considering quantitative aspects and dynamical interactions between the system components we can identify rare events and use this information to select the "interesting" cases.

In this paper we identify some common modelling errors and illustrate a procedure for verification and validation which reports such errors; we present the software tool support for verification and validation of Bio-PEPA models and demonstrate the procedure on a model of a signalling pathway.

The methodology which we propose is to perform *static analysis* and *trace analysis* before embarking on a lengthy simulation study. The use of static analysis and trace analysis brings at least two benefits: i) it can identify errors in a model in advance of simulation, and thus save many hours of compute time by avoiding the execution of a pointless simulation study on a cluster of workstations; and ii) it can identify different time-scales and influence the choice of reactions for which the stochastic partial equilibrium approximation can be applied, facilitating the use of faster simulation algorithms such as the Slow-Scale SSA.

2. FORMAL MODELLING OF BIOLOGY

Dynamic modelling of biological processes proceeds by conjecturing a mechanism which could explain a biological phenomenon and then generating predictions from the model which can be compared to experimental data. If the model predictions show good agreement with the experimental data then the mechanism is a plausible explanation of the phenomenon. If not, then the mechanism or the model parameters need to be adjusted in the hope of finding a more credible explanation.

2.1 Static analysis

Before the dynamic modelling process is initiated, the model can be checked for well-formedness using static analysis methods which investigate without recourse to dynamic execution of the model. Some methods applied at this point, such as *choke point analysis* or *flux balance analysis* [17] do not require knowledge of species populations or concentrations, or knowledge of kinetic parameters but can produce effective insights into model behaviour despite being offered only a partial view of the model.

An important structural property of chemical reaction systems is the identification of the maximal conserved moieties from the stoichiometry of these systems. These identify invariant building blocks of the system and both the presence and the absence of these invariants can be informative to the modeller. Where others use their own custom algorithms to determine conserved moieties [19] we have drawn on the rich theory of Petri nets and used the Fourier-Motzkin algorithm (also known as the Farkas algorithm, see [14] for an introduction). Although this algorithm is known to work well on average and in practice, its worst case complexity is exponential due to a possible exponential growth in the number of matrix rows and invariants generated in the solution process. Consequently we use a robust implementation which uses a greedy strategy to pick columns; imposes a threshold on the number of rows to produce; and delivers an intermediate output of valid invariants such that one harvests at least those invariants that have been computed if the calculation explodes. The generated set of vectors is a superset of a set of base vectors, which is then reduced to a minimal set using standard procedures of matrix algebra.

2.2 Trace, path and time-series analysis

We distinguish between three kinds of formal representations of stochastic simulations: traces, paths and time-series. A *trace* is a sequence $s_0 e_1 s_1 \dots e_n s_n$ of states $s_0, \dots, s_n \in S$ and events $e_1, \dots, e_n \in E$ over some (finite or infinite) sets S, E for an arbitrary but fixed $n \in \mathbb{N}$. A *path* (or *trajectory*) is a sequence $s_0 \dots s_n$ of states. A *time-series* (or *run*) is a sequence $t_0 \dots t_n$ of time-series observations, over some (finite or infinite) set T.

The most informative of these is a trace, because it includes information about the event which caused the state change, allowing the calculation of statistics on individual events (such as tally, mean and variance) and between pairs of events (such as joint moments). A path can be obtained from a trace by erasing the information about events. A time-series can be obtained from a path by discretising the output onto a regular grid of time points. Ensembles of time-series can be averaged in order to give a collective view of the molecular dynamics as a time-series. For this reason, objects in a time-series may include real-valued averages of molecular population samples whereas states only have integer-valued counts of molecular populations.

Even when an exact simulation algorithm such as Gillespie's Direct Method is used, some simulators used in computational biology generate a time-series where individual events have been omitted and the output has been discretised in order to reduce the volume of output produced and to facilitate averaging. While this is entirely appropriate for convenient investigation of the system dynamics the abstraction which is applied in a time-series is problematic for verification and validation purposes. For the present work we extended the Bio-PEPA Eclipse Plug-in modelling tool with fully exact generators for paths and traces which include every event and state transition without omission or adjustment. The Traviando analyser [12] operates on traces. BIOCHAM [7] and the MC2 component of the BioNessie software suite [16, 13] operate on paths.

2.3 Bio-PEPA: language and tool support

Bio-PEPA [5] is a stochastic process algebra, recently developed for the modelling and analysis of biological systems. We give here a brief overview of the main features of the language. For a detailed presentation of its syntax and semantics, see [5].

The main components of a Bio-PEPA system are the *spe-*

cies components, describing the behaviour of each species, and the *model component*, specifying all interactions and initial amounts of species. The syntax of Bio-PEPA components is given by:

$$S ::= (\alpha, \kappa)\ \mathtt{op}\ S \mid S + S \mid C \quad \text{with } \mathtt{op} = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot$$

$$P ::= P \underset{\mathcal{I}}{\bowtie} P \mid S(x)$$

where S is the *species component* and P is the *model component*. In the prefix term $(\alpha, \kappa)\ \mathtt{op}\ S$, κ is the *stoichiometry coefficient* of species S in reaction α, and the *prefix combinator* "op" represents the role of S in the reaction. Specifically, $\downarrow$ indicates a *reactant*, $\uparrow$ a *product*, $\oplus$ an *activator*, $\ominus$ an *inhibitor* and $\odot$ a generic *modifier*. The notation $\alpha\ \mathtt{op}$ is a shorthand for $(\alpha, \kappa)\ \mathtt{op}\ S$ when $\kappa = 1$. The operator "+" expresses a choice between possible actions, and the constant C is defined by an equation $C \stackrel{def}{=} S$. The process $P \underset{\mathcal{I}}{\bowtie} Q$ denotes synchronisation between components P and Q; the set $\mathcal{I}$ determines the activities on which the operands are forced to synchronise, with $\underset{*}{\bowtie}$ denoting a synchronisation on all common action types. In the model component $S(x)$, the parameter $x \in \mathbb{N}$ represents the initial number of molecules of S present. In addition to species and model components, a Bio-PEPA system consists of kinetic rates, parameters and, if needed, locations, events and other auxiliary information for the species. Complexes are sometimes denoted with colons, as in *E:S*, but the colon is just a letter in the name, not an operator.

The formal representation offered by Bio-PEPA allows for different kinds of analysis through the defined mapping into continuous-deterministic and discrete-stochastic analysis methods (see [5] for details). More on Bio-PEPA can be found at [2], including the Bio-PEPA Eclipse Plug-in [4, 6] which include a number of stochastic simulators and ODE solvers and export formats (e.g. SBML for interaction with other tools, PRISM for model-checking, Traviando for simulation trace analysis).

2.4 The Traviando trace analyser

Traviando [20] is a trace visualiser and analyser for simulation traces of discrete event system models. It is not dedicated to a particular modelling formalism. Its input is an XML-formatted trace with a prefix that defines finite sets of state variables that constitute a state $s \in S$ and actions that occur as events E and a suffix that contains data as we formally defined it for a trace.

Traviando provides a command-line interface to generate a report, an HTML formatted document, of various properties seen in a given trace. A selection of example reports can be accessed at [20]. The idea is to provide a modeller with human readable feedback with graphics, data and explanations for a simulation trace in a format that requires no further expertise to explore what a simulator really does with a given model. In addition to the generated web report, Traviando comes with an interactive graphical user interface that visualises any location in a trace in full detail with a variant of a message sequence chart and that provides various features intended to help a modeller locate areas of interest in a lengthy simulation trace. For example, Traviando provides an event browser that allows a modeller to investigate sequences of events and a model checker for linear temporal logic formulae where areas in a trace that fulfill given formulas are highlighted with particular colours. Traviando also provides a reduction mechanism to remove cycles in a trace which helps to generate plots for a measure of progress that show particular patterns to reveal certain errors in simulation models [12].

Traviando has been used to analyse traces generated from various sources, most recently Bio-PEPA integrated Traviando to support dynamic analysis.

3. VALIDATING BIO-PEPA MODELS

Bio-PEPA models representing biochemical systems can be very large and complex: they can be composed of many species and reactions, can involve multiple compartments, can include time-dependent rates and events, and can consist of a compositional definition of multiple modules. When dealing with this complexity, it is evident that the process of modelling is potentially subject to a range of errors, ranging from typos in variable names or in numerical values, to the omission of interactions between components.

The model verification and validation process is supported in Bio-PEPA thanks to the following design and implementation choices.

First, Bio-PEPA was explicitly designed to model biochemical systems and, hence, it only includes the minimal set of operators needed to model this kind of system. Bio-PEPA models conform to a structure closely resembling the definition of biochemical reaction networks. The simple and rigorous structure of models makes the automatic detection of several modelling errors easier[1].

Second, like all the other high-level modelling languages, but unlike classical mathematical modelling tools such as MATLAB [15], the Bio-PEPA language allows modellers to define mnemonic names for system components, reactions and parameters. The possibility to work with name definitions rather than numerical vectors for variables and parameters reduces the chances of introducing trivial modelling errors. Making models more readable makes it easier to identify potential sources of errors.

Finally, the tool support for Bio-PEPA enables the user to observe the system from a variety of points of views and to perform a variety of analyses: each different view and analysis highlights a specific aspect of the model, thus making the detection of specific errors or the verification of desired properties straightforward. Some of the different views and analyses supported by the Bio-PEPA tools are:

- biochemical reaction view (reaction-centric representation of a Bio-PEPA model, automatically derived in the Bio-PEPA Eclipse Plug-in);
- species and reaction invariants (computed by the Bio-PEPA Eclipse Plug-in using the Fourier-Motzkin procedure);
- interactive visualisation of simulation traces (trace exported by the Bio-PEPA Eclipse Plug-in and analysed in Traviando);
- detailed report on simulation traces (summary of various measures and statistics computed by Traviando and exported as browsable HTML documents).

[1]For example, sub-expressions which occur multiple times in an ODE model are lifted out into named functions, eliminating the problem of repeated copies of the sub-expression being inconsistent.

In the rest of this section we list the most common potential sources of errors in Bio-PEPA models; in the next sections we will describe how these are identified and signalled to modellers via the views which we have described.

The following errors can be due to typos in names or to forgotten definitions:

- species used in the model but not defined as species components, or not included in the system component;
- reaction used but with no functional rate associated;
- variable or function defined but not used;
- reaction appearing twice in the same species definition;
- reaction rate not depending on involved reactant(s), or depending also on species not involved in the reaction.

The following errors, instead, can be due to typos in numerical values or in the operators specifying the role of reactants in reactions:

- wrong functional rate, constant definition;
- wrong initial amount;
- wrong stoichiometric coefficient;
- predefined kinetic law used for a reaction not involving the expected number of components.

Some of these errors can be discovered with a simple syntactic check, others require some more advanced techniques. We consider three categories of errors:

- syntax errors: detected in Bio-PEPA Eclipse Plug-in built-in editor;
- errors that influence qualitative behaviour: detected by static analysis in the Bio-PEPA Eclipse Plug-in or analysis of traces in Traviando; and
- errors that influence quantitative behaviour: detected by analysis of traces in Traviando or analysis of time-series in the Bio-PEPA Eclipse Plug-in.

4. SYNTAX ANALYSIS

In this section, we discuss non-trivial errors that occur at the level of syntax. This includes also errors an editor can immediately recognise and highlight to a user with appropriate feedback. We include in the class of syntactical errors all those errors that prevent a modelling framework from producing an executable simulation model that it can run.

This includes the type of error when an entity is used but not declared, e.g., a constant is used but not declared with an initial value, a species is used in a reaction rate but not declared in an equation that describes the reactions that modify its value, or finally a reaction that appears in the declaration of a species but does not have a rate defined for it.

These errors can sometimes be subtle, and not easy for humans to detect. One check which is automatically applied (and which has proven to be useful in practice) is to check that the kinetic law for each reaction depends on the molecule count of all of the reactants which participate in the reaction. A warning message is produced if this is not the case. The warning indicates that the missing dependency could allow the reaction to drive the population of one chemical species negative—an event which has no physical explanation. When using a modelling formalism which does not require this static check such a modelling error could go undetected, and would only be caught if some simulation run follows a trajectory which does drive the molecule count negative.

A syntactically correct model allows us to derive an orthogonal representation that focuses on reactions. The Bio-PEPA tool automatically derives a model representation as a list of common chemical reactions. The use of chemical reaction notation is familiar to many biologists and thus provides a convenient view for a check of "face validity" (i.e. does the model in this view "look" like it should?).

As an example, a Bio-PEPA model of Michaelis-Menten kinetics is shown below.

$$\begin{aligned} E &\stackrel{def}{=} (r_1,1)\downarrow E + (r_{-1},1)\uparrow E + (r_2,1)\downarrow E \\ S &\stackrel{def}{=} (r_1,1)\downarrow S + (r_{-1},1)\uparrow S \\ E{:}S &\stackrel{def}{=} (r_1,1)\uparrow E{:}S + (r_{-1},1)\downarrow E{:}S + (r_2,1)\downarrow E{:}S \\ P &\stackrel{def}{=} (r_2,1)\uparrow P \end{aligned}$$

When seen in the Outline view in the Bio-PEPA tool this is displayed as follows.

$$\begin{aligned} r_1, &\quad E + S \to E{:}S \\ r_{-1}, &\quad E{:}S \to E + S \\ r_2, &\quad E{:}S \to E + P \end{aligned}$$

This summary view is very convenient for a face validity check.

A syntactically correct model can be executed but it does not necessarily meet common and plausible conditions for its behaviour. Those are the topic of the next section on static analysis.

5. STATIC ANALYSIS

In this section, we describe constraints that apply to a complete model and corresponding automated methods that check those constraints and give guidance to a modeller. The techniques considered here all analyse a given model description without exercising its dynamics, i.e., without performing any kind of simulation.

For this kind of analysis, the Bio-PEPA model is seen as a graph on which it is possible to perform structural checks adapting techniques classically used in the analysis of Petri nets.

A Bio-PEPA model can be mapped into a stochastic Petri net $(P, T, I^-, I^+, \lambda, M_0)$ with state-dependent transition rates in a straightforward manner. The set of species is the set of places P, the set of reactions is the set of transitions T, the stoichiometry coefficients κ provide entries in the $(P \times T)$ incidence matrices I^- and I^+, the initial amounts of species give values for the initial marking M_0. More precisely, a reaction α that consumes a species S, i.e., $(\alpha, \kappa)\downarrow S$ results in an entry $I^-(i,j) = \kappa$, where i is the place that corresponds to S and j is the transition that corresponds to α. If a reaction produces a species, its stoichiometry coefficient κ goes into I^+. Finally, the exponentially distributed firing times of transitions have a state (marking) dependent firing rate with a function $\lambda : T \times (P \to \mathbb{N}_0) \to \mathbb{R}_0$. For a static analysis, we will focus on the untimed fragment of the stochastic Petri net, which is a place/transition net (P, T, I^-, I^+, M_0). With the help of this mapping, we can readily apply Petri net techniques like that of invariant analysis. A place (species)

invariant is a non-negative column vector $x \in \mathbf{N}_0^P$ that solves $x^T \cdot I = 0$ for incidence matrix $I = I^+ - I^-$. A transition (reaction) invariant is a vector $y \in \mathbf{N}_0^T$ that solves $I \cdot y = 0$.

5.1 Invariant analysis

Is the model covered by place (species) and transition (reaction) invariants? Are the expected invariants present?

Species that are covered by an invariant can assume values in a bounded range. Species invariants are expected to be present in biochemical systems which do not include synthesis and degradation reactions; if an expected species invariant is missing, it means the model violates the law of conservation of mass. Reaction invariants describe combinations of reactions with a total effect of zero changes to species. For each reversible reaction, we expect a corresponding reaction invariant.

5.2 Source/sink species and source/sink actions

Are there source/sink species in the model? Are there source/sink actions?

Source species are those species that can be consumed but never produced, while sink species can be produced but never consumed. Though these kinds of species can be perfectly valid in a model, their presence might also represent a potential cause of unwanted behaviour: sources may cause the model to deadlock, while sinks may accumulate arbitrarily high amounts due to an unconstrained influx.

Source actions are reactions that have products but no reactants (i.e. synthesis reactions or actions representing an influx of molecules in the system), while sink actions are reactions that have reactants but no products (i.e. degradations or actions representing an efflux out of the system).

Note that a model covered by reaction invariants cannot have any source and sink species, whereas a model covered by species invariants cannot have any source and sink actions.

Certain classes of models, e.g., pathway models, naturally contain source/sink species or reactions. For these, it is often illustrative to understand the sequence of reactions that link sources with sinks and to learn the overall combined effect of those reactions. For any pair of sources and sinks, we compute such a sequence of reactions (a minimal path in the reaction graph) as well as the overall effect on all species if each reaction in this sequence is performed once. We found this helpful to see which sources have influence on which sink (due to the existence of a sequence between them) and what the input/output behaviour of the overall model is. These sequences provide some guidance to understand causal dependencies in a complex model.

5.3 Connectedness

Is the reaction graph strongly/weakly connected? Are the molecular populations bounded?

The reaction graph underlying a model should be connected. If it is not then the model consists of two independent submodels which could more conveniently be analysed in isolation. It might be difficult to see how any reasonable model could fail to be connected, but for models with locations or compartments omitting the transport reactions would be enough to make the model disconnected. So, we expect a hint for omission errors to be the prime result of this analysis. If static analysis determines that the model is not connected then this suggests that the modeller should analyse the submodels individually or look for a missing reaction or a missing species.

For models with neither source/sink species nor source/sink reactions, we may often expect that the model operates on a finite amount of species and its dynamics to continue. A useful theorem in Petri net theory states that a live and bounded Petri net that is connected must be strongly connected. We can apply this to the Petri net we associate with a Bio-PEPA model. Liveness means that there exists an initial number of molecules $S(x)$ such that the model can only reach states where for every reaction α there is a way to proceed further such that α can perform sometimes in the future. Boundedness means that for an arbitrary but fixed initial number of molecules the model can only reach amounts of molecules below some upper bound. While boundedness naturally matches with limited supplies and conservation of mass, liveness is desirable for models where there is interest in a long term behaviour. Note that this concept basically considers subgraphs being sources or sinks instead of individual reactions or species.

As a result of static analysis, models can be classified as being open (having source and sink reactions such that total amount of mass varies), closed (having no influx or efflux of matter) or mixed (some species observe conservation of mass while some others are created or degraded).

There is subtle difference between the possible behaviour of an associated, untimed Petri net used for static analysis and the behaviour of the stochastic model itself. Common state-dependent reaction rates, e.g., mass action kinetics, impose a natural upper bound for the average amount of reactants, because rates are proportional to reagent concentrations. This implies that even if the structure of a model does not ensure that a species is bounded, reaction rates may counterbalance the growth of a species such that the probability to reach arbitrarily high amounts goes down to zero.

6. DYNAMIC ANALYSIS

In this section, we consider aspects of the behaviour of the model that can only be observed during a simulation run, when the dynamics of a model are exercised. The additional information obtained by analysing the dynamic behaviour of the model can (i) provide additional knowledge on the system, (ii) be used as a quantitative counterpart of the qualitative static analysis results to investigate specific aspects, and (iii) provide insights in the way a simulator executes the model. Dynamic analysis should be performed on traces obtained from two model configurations. One configuration has artificial value settings for constants and initial values $S(x)$ such that all reactions frequently occur. The other configuration has value settings that are suitable for validation, i.e., to reproduce known results. Traviando provides us with a report document that is generated from a trace and an interactive graphical user interface for a detailed investigation at the level of states and events. We list here a series of checks based on the report document that can be performed using the measures automatically computed and

reported by Traviando. A drill down into the details of a trace is only advisable if the report stimulates suspicion of faulty behaviour but does not give sufficient details to track the root cause in the model itself.

6.1 Invariant analysis

Do observed "quantitative" invariants match the "qualitative" ones computed statically?

Invariants are computed based on the occurrences of the reactions in that particular trace. They can be seen as a dynamic/quantitative counterpart of the static invariants. If all reactions are present in the trace, the computed set of species and reaction invariants are equivalent. The first model configuration described above provides us with a trace such that we expect to obtain equivalent sets of invariants. If some reactions are missing, due to absence of rare events in a trace and the presence of creation/degradation reactions, the computed invariants will differ and are harder to interpret.

6.2 Reaction occurrences and variable ranges

Are variables always within the expected ranges? Are relative numbers of occurrences of events the same as expected?

The number of occurrences of reactions, number of changes to variables and observed ranges of species are automatically computed for each species/reaction. For each species, we see a distribution of values and common statistical measures (mean, variance, skewness) as well as the sequence of values observed throughout the trace. The latter often helps us to recognise if values for some species freeze at a particular level – in particular zero if it dies out – or if it reaches a fixed upper bound due to conservation of mass. This is often an indication of some internal, partial deadlock. This complements static analysis, because we can take timing effects into account where reaction rates may differ to an extent that certain reactions are practically infeasible. For reactions, Traviando reports the first and last occurrence, the total number of occurrences as well as estimates for the first five moments of the time before each reaction and after each reaction. In general, this information can be used as additional validity checks to verify whether the expected behaviour is observed in a trace.

Though generally there is no preexisting knowledge on the precise amount of the involved species, it is often the case that some rough estimate on relative values can be made (e.g. highly abundant vs. low copy number species). Sometimes, for instance, the ratio between different forms of the same molecule (e.g. phosphorylated or not, located in different compartments, etc.) are roughly known. Also, we expect some species to have a certain behaviour (e.g. constant, present in small amount, bounded, etc.) because otherwise they would not satisfy some model constraints (e.g. Michaelis-Menten assumption of constant enzyme, or impossibility of unconstrained growth). This kind of information can be used to validate the model by comparing the known facts with the values computed by Traviando.

A general limitation of trace analysis is that it draws conclusions from a finite set of observations, which is incomplete and not necessarily representative of the overall and long term model behaviour.

7. APPLICATION TO A CASE STUDY

The JAK/STAT signalling pathway is a well-studied biological system which plays a key role in several biological processes in humans. The signalling cascade is triggered by cytokines LIF and OSM binding to membrane receptors gp130, LIFR and OSMR forming different receptor complexes. One of the main targets of the signal transduction is the transcription factor STAT3, and a key feature of the pathway is the nuclear/cytoplasmic shuttling of STAT3: upon activation, STAT3 can translocate into the nucleus and activate the transcription of downstream gene targets or be transported back to the cytoplasm in its inactive form. Figure 1 shows a diagrammatic representation of the main components and reactions involved in the pathway, including the inhibitors SOCS3 and PIAS3.

The gp130/JAK/STAT pathway has been modelled in Bio-PEPA previously (see [9] and [2] for the full model): [9] reports a number of results obtained from dynamic analysis of the model via stochastic simulation and model-checking. Our aim here is to supplement those results with static analysis results and additional dynamic results that are obtained thanks to the recently added support for analysis of Bio-PEPA simulation traces via the Traviando trace analyser.

7.1 Syntax and static analysis

Syntax analysis of the model confirms that it has no syntax errors.

The "Outline view" provides a compact view of the model listing the species and reactions composing the model, and its *source/sink* species and actions, if there are any.

The model is composed of 63 species and 118 reactions. There is no sink species, and there are five sources species: the two ligands (LIF and OSM) and the three receptors (gp130, LIFR and OSMR). Indeed, the ligand/receptor binding are modelled as irreversible processes, so the amount of ligands and receptors can only decrease. Moreover, there is one source action (SOCS3 synthesis mediated by nuclear STAT3) and one sink action (SOCS3 degradation).

The steps needed to activate nuclear STAT3, which can be intuitively inferred looking at Figure 1, are formally represented by the minimal path between the nodes representing ligands and the node representing STAT3_PD_n in the reaction graph which is associated to the Bio-PEPA model. A minimal path computed by Traviando from LIF to STAT3-PD_n is

$$\text{LIF} \xrightarrow{\text{bind}_1} \text{gp130:LIF:LIFR} \xrightarrow{\text{phospho}_8} \text{gp130:LIF:LIFR-P} \xrightarrow{\text{phospho}_{10}}$$
$$\text{rcpt-DP} \xrightarrow{\text{bind}_{28}} \text{rcpt-DP:STAT3} \xrightarrow{\text{bind}_{30}} \text{STAT3:rcpt-DP:STAT3} \xrightarrow{\text{phospho}_{44}}$$
$$\text{STAT3:rcpt-DP:STAT3-P} \xrightarrow{\text{phospho}_{46}} \text{STAT3-P:rcpt-DP:STAT3-P} \xrightarrow{\text{unbind}_{55}}$$
$$\text{STAT3-PD\_c} \xrightarrow{\text{reloc\_cn}_{58}} \text{STAT3-PD\_n}$$

where rcpt-DP is the double phosphorylated receptor complex gp130-P:LIF:LIFR-P. The net effect of this path is to remove one gp130, one LIF, one LIFR and two STAT3_c molecules, and to produce one gp130-P:LIF:LIFR-P and one STAT3-PD_n molecules.

The species and reaction invariants of the model can be automatically computed from the Bio-PEPA model in the Eclipse Plug-in (reported in the "Invariants view"), and they can be used to verify whether there is any violation of the law of conservation of mass and whether any expected reversible reaction is missing.

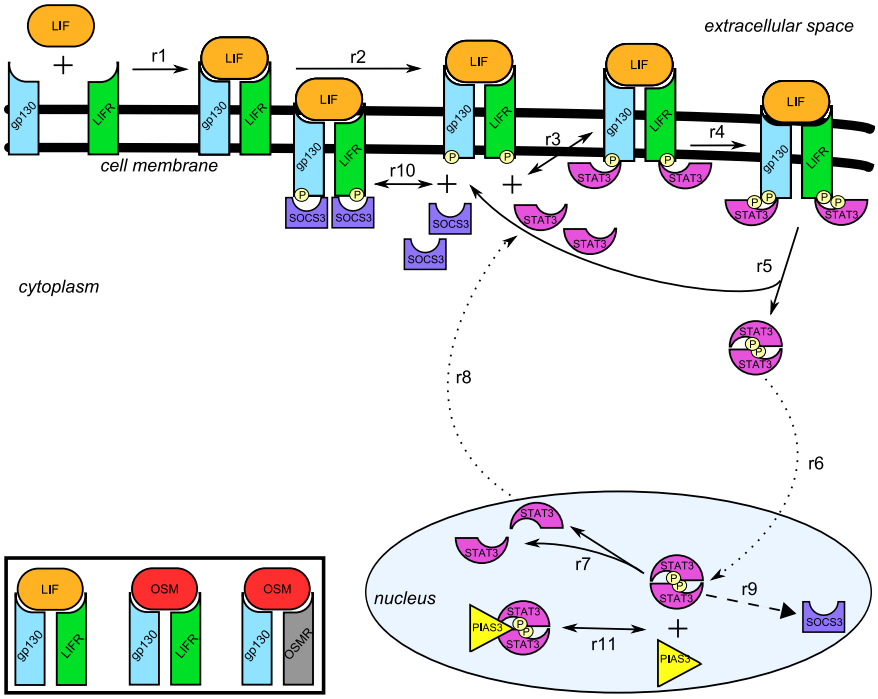

Figure 1: Gp130/JAK/STAT pathway map.

In our case, the model is only partially covered by *species invariants*, since SOCS3 is not covered by any invariant: the amount of SOCS3, in fact, is not constant since its synthesis and degradation are represented. On the other hand, given the absence of synthesis and degradation reactions for ligands, receptors and STAT3, for them, the sum of their amounts in their different forms is constant. For instance, the total amount of LIFR receptor present in its various possible forms is constant. (Free, as gp130:LIF:LIFR complex and as gp130:LIF:LIFER complex and as gp130:OSM:LIFR complex, with one or both of its subunits phosphorylated, and with STAT3 or SOCS3 bound.)

Many *reaction invariants* are reported (we can use the "Outline view" to find the reactions identified by the names listed in the "Invariants view"). Many invariants refer to binding/unbinding reactions for the reversible binding of STAT3 to the dimerised receptors, and there is one invariant related to SOCS3 synthesis/degradation. Moreover, there are six invariants which represent the shuttling of STAT3 between cytoplasm and nucleus (i.e. the cycle given by reactions r_3 to r_8 in Figure 1). There are two invariants for each of the three types of dimeric receptors (see insert in Figure 1), where the two invariants refer to a different ordering in phosphorylation of the two involved STAT3 molecules. In these invariants the reaction representing the relocation of STAT3 from nucleus to cytoplasm (r_8) is present twice (it has to be performed twice as many times as the other reactions since it transports STAT3 in its monomeric rather than dimeric form).

In Figure 2 we report a screenshot of the Eclipse Plug-in, showing the Bio-PEPA model (upper right), its biochemical reactions equivalent representation together with the list of source species (upper left), a time-series resulting from a single stochastic simulation run (bottom left) and the "Invariants view" showing some of the model invariants (bottom right).

7.2 Dynamic analysis

In [9], time-series results obtained via stochastic simulation are reported. Here we also use stochastic simulation, but we focus on the analysis of individual simulation traces and on the additional information which can be extracted from them in order to verify if the dynamic behaviour of the system is the same as expected.

As explained in Section 6, for trace analysis, we consider two versions of the model: the original one, and a "test" version obtained by changing constants and initial values to increase the probability that all reactions occur within the chosen simulation time.

As expected, the species and transition invariants observed on a trace of the test model match the structural invariants computed statically in the previous section[2]. Conversely, some of the structural invariants are not observed in traces of the original model: for instance, two of the possible complexes involving SOCS3 (namely gp130:OSM:LIFR-P:SOCS3 and SOCS3:gp130-P:OSM:LIFR) are not part of the species invariant for LIFR computed on one of the generated traces. To understand this difference, it is helpful to inspect the detailed information provided by Traviando about the behaviour of these species, which reports that their amount is constantly equal to zero (in fact, it also reports a warning

[2]It would be possible for a too-short trace to generate a number of false species invariants because the reactions which modify their molecule counts did not occur. A mismatch between the invariants computed statically and those observed on a trace could also indicate a mismatch between the model semantics intended by the modeller (or by the model development tool) and the one assumed by the simulator. Our results show that none of these problems occur here.

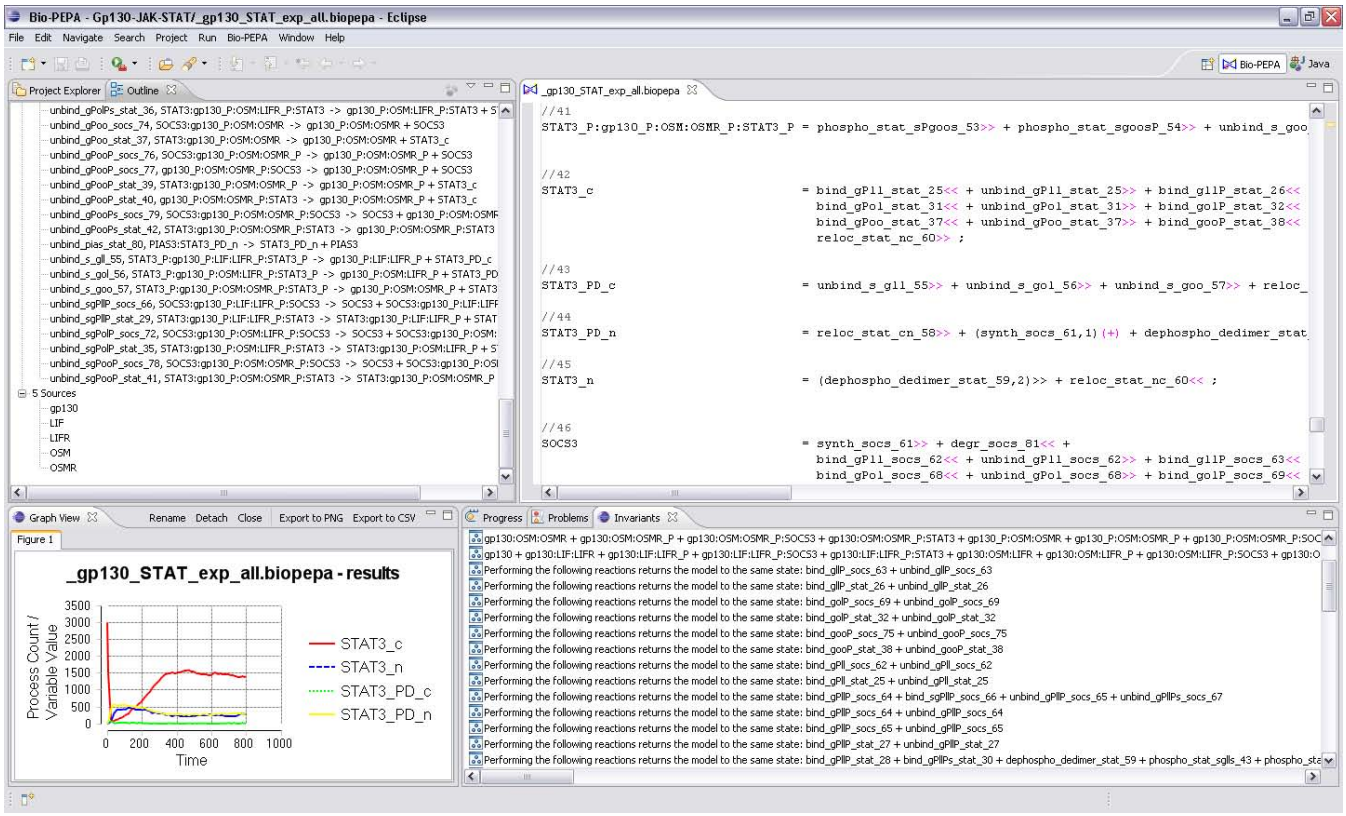

Figure 2: A screenshot of the Eclipse Plug-in.

stating that these variables are not modified and so their value is constant over the whole length of the trace). This kind of warning is meant to signal the possible presence of *dead* reactions (i.e. reactions that can never occur); in this case, we note that the rate of the binding of SOCS3 to the receptor dimers is much smaller than the one of the binding of STAT3 and, hence, since these reactions are competitive, we conclude that simply in this particular trace this binding did not occur. For the same reason, the reaction invariants for SOCS3 binding/unbinding reactions are sometimes missing in simulation traces compared to the structural invariants.

In addition to signalling warnings about reactions that are never occurring in a trace, Traviando also records and plots the number of occurrences for each reaction. In traces of the original model, for instance, we note that the number of occurrences of all SOCS3 bindings is much smaller than the number of occurrences of STAT3 bindings. This might either mean that one of the two sets of rates are wrong, or that the effect of SOCS3 inhibition is minor and, hence, the model could be simplified by removing SOCS3 without introducing a significant approximation. Also, we note that the number of occurrences of binding/unbinding reactions between STAT3 and PIAS3 is more than 10 times higher than any other reactions. Figure 3 is a visual representation of the number of occurrences for each reaction for a particular trace. Table 1 shows the collected statistics about reaction occurrences (we only report the results for a selection of reactions).

Another kind of information is focusing on the state variables of the model (i.e. the amount of species), such as the range of values each species attains in a trace, its mean value, and the number of changes to its value. For instance, in one of the traces of the original model, STAT3-PD_n is always in the range [0,590], with arithmetic mean = 354.42205 and standard deviation = 118.82829, and the total number of assignments of its value are 236587. The variable is modified for the first time at event 2553, for the last time at event 329612, and it has a leading non-decreasing sequence till event 2791. For instance, Figure 4 reports the distribution of values for STAT3-PD_n.

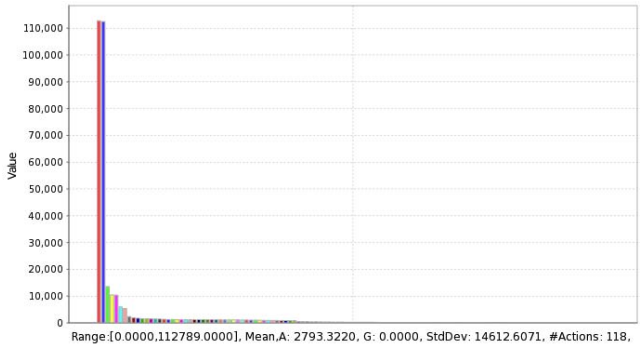

Figure 3: Number of occurrences observed in a trace as reported by Traviando.

Some final noteworthy observations are the following. The structural invariant analysis states that SOCS3 is not covered by species invariants, which implies that the amount

Action	Occurrences	First seen	Last seen	Remarks
bind_gl_l_1	161	1	1378	
bind_go_l_3	160	3	1269	
bind_go_o_6	181	11	1455	
bind_golP_socs_69	0			Action does not occur, may be dead code.
bind_golP_stat_32	161	929	16379	
bind_gPll_socs_62	1	12022	12022	Fitting of state transformation is not unique and based on too little data.
bind_gPll_stat_25	142	949	151407	
bind_gPllP_socs_64	280	11570	327965	
bind_gPllP_stat_28	1221	1540	329590	
bind_pias_stat_80	112789	2791	329612	
phospho_gll_10	44	1503	27467	
reloc_stat_nc_60	10481	8986	329565	
synth_socs_61	13562	4928	329611	
unbind_golP_socs_69	0			Action does not occur, may be dead code.
unbind_golP_stat_32	34	1529	18418	
unbind_pias_stat_80	112480	3914	329610	

Table 1: Measures reported by Traviando on reaction occurrence: for each reaction the table shows the number of occurrences, the first and last occurrences, and warnings signalling unusual behaviour. Warnings can refer to reactions that do not occur (potentially dead code) or that could cause problems in the analysis (e.g. analysis can be inaccurate if based on a single occurrence of an action, or a trace can be too short to observe all actions).

of SOCS3 may be unbounded. This is intuitive because of the presence of a source action representing SOCS3 synthesis. The quantitative behaviour observed in dynamic simulation, however, is not an unbounded increase of SOCS3. The presence of an opposed sink action representing SOCS3 degradation brings the system to an equilibrium. The state-

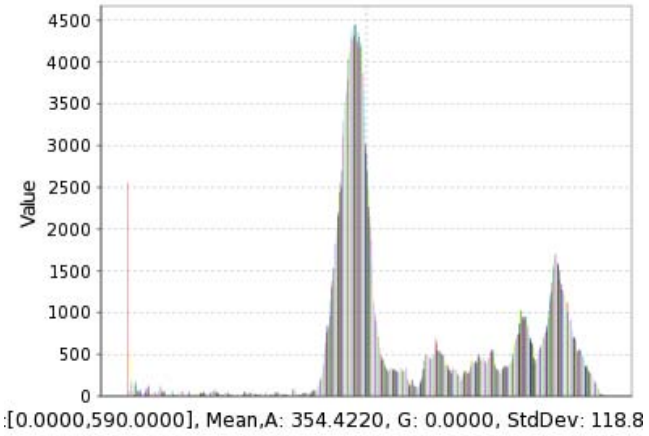

Figure 4: Distribution of values observed over a trace as reported by Traviando.

dependent rates of mass-action kinetics, in fact, generally counterbalance the growth of species since any draining reaction of a given species will become faster than a constant input reaction at a certain level of that species.

From this simple example it is evident that the behaviour inferred from static analysis is not necessarily observed in simulation, and that the dynamic behaviour is highly dependent on the given quantitative parameters and kinetic laws.

Additionally, we also point out that an observation of a system for a finite time, such as the one done via traces, is not guaranteed to be representative of the behaviour of the system; therefore, before drawing conclusions on the long-term behaviour from trace analysis we must consider carefully timing issues. In this case, existing knowledge, both experimental and from previous modelling, suggested a suitable time simulation interval such that the system reached an equilibrium.

8. RELATED WORK

A very accessible introduction to the application of the structural theory of Petri nets (including place and transition invariants) can be found in [10]. In that paper transition invariants are classified in order to provide a structured representation which aids the interpretation of a large collection of invariants.

Several works exist in the literature on the application of Petri nets based techniques to the analysis of biochemical pathways. Additional information can be found, for instance, in [3, 11]. In particular, structural theory and invariant analysis have been used previously for different purposes. In [8] Petri net t-invariants are used to assist in the modularisation of biochemical networks. In [18] the authors begin with minimal t-invariants and work towards feasible t-invariants, checking the biological significance of these.

An alternative approach to the verification of biochemical pathways is the use of formal verification techniques such as model-checking. In [9] this approach is used for the analysis of the gp130/JAK/STAT pathway.

9. CONCLUSIONS AND FURTHER WORK

In this paper we have put forward the opinion that static analysis and trace analysis have a valuable role to play in the investigation of biological phenomena via discrete stochastic modelling of their dynamics.

Static analysis of the gp130/JAK/STAT pathway which we took as our example here determined that the model was connected but was not covered by species and reaction invariants.

Trace analysis of our model identified two reactions which dominated the model dynamics, suggesting that it could be profitable to apply the stochastic partial equilibrium approximation and more computationally effective to simulate the model using the slow-scale variant of Gillespie's SSA. This was a fact which we had not previously known about this reaction pathway.

Other benefits come from our provision of both model-centric and trace-centric implementations of invariant calculation. The implementation of invariant calculation in the Bio-PEPA Eclipse Plug-in makes use of the stoichiometry matrix associated to the reaction network underlying the model. In contrast, the implementation of invariant inference in the Traviando trace analyser is given only a single trace as input, and does not have access to the model and the stoichiometry matrix. This allows us to make a qualitative judgement about the effectiveness of this particular trace, in the following way. If the invariants inferred from the trace by Traviando agree with the invariants computed by the Bio-PEPA Eclipse Plug-in then this suggests that the trace has been long enough to produce an informative observation of the model dynamics. If the two sets of invariants do not agree then this may suggest that the trace was too short to give a proper account of everything in the model, and that generating longer simulation runs should be considered; alternatively, this may be due to a big difference in rates of model reactions which causes the existence of rare events that are not necessarily observed in a single simulation, in which case generating multiple simulation runs is important.

As a final comment, we point out that traces represent examples of behaviour of a system, so they are good to demonstrate the occurrence of a phenomenon and they can also give an indication on the general behaviour. However, we cannot draw conclusions on the general behaviour just observing one example, and the analysis of multiple traces is often desirable. Currently, the generation and analysis of multiple traces is possible, but the aggregation of analysis results is not automated. The computation of aggregated measures over multiple traces is one of the planned future developments which could exploit the recent addition of trace clustering in Traviando.

Further work building on the static invariant analysis could include exploiting the invariant calculation to reduce the size of the model, or the size of traces generated from the model.

Finally, the computation of the minimal paths within the reaction graph associated to a Bio-PEPA model deserves further investigation as it is related to the identification of causal chains between events and could potentially be useful for model composition, abstraction and simplification.

10. ACKNOWLEDGEMENTS

Allan Clark and Stephen Gilmore are supported by the Engineering and Physical Sciences Research Council (EPSRC) grant EP/E031439/1 “Stochastic Process Algebra for Biochemical Signalling Pathway Analysis”.

Stephen Gilmore and Maria Luisa Guerriero are supported by the Centre for Systems Biology at Edinburgh. The Centre for Systems Biology at Edinburgh is a Centre for Integrative Systems Biology (CISB) funded by BBSRC and EPSRC, reference BB/D019621/1.

11. REFERENCES

[1] O. Balci. Quality assessment, verification, and validation of modeling and simulation applications. In *Winter Simulation Conference*, pages 122–129, 2004.

[2] Bio-PEPA Home Page. http://www.biopepa.org/.

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

[4] F. Ciocchetta, A. Duguid, S. Gilmore, M. L. Guerriero, and J. Hillston. The Bio-PEPA tool suite. In *Proceedings of QEST'09*, pages 309–310, 2009.

[5] 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.

[6] A. Duguid, S. Gilmore, M. L. Guerriero, J. Hillston, and L. Loewe. Design and Development of Software Tools for Bio-PEPA. In *Proc. of WSC'09*, pages 956–967. IEEE Press, 2009.

[7] F. Fages and A. Rizk. On the analysis of numerical data time series in temporal logic. In *Computational Methods in Systems Biology '07*, pages 48–63. Springer, 2007. LNCS 4695.

[8] E. Grafahrend-Belau, F. Schreiber, M. Heiner, A. Sackmann, B. H. Junker, S. Grunwald, A. Speer, K. Winder, and I. Koch. Modularization of biochemical networks based on classification of Petri net t-invariants. *BMC Bioinformatics*, 9(90), 2008.

[9] M. L. Guerriero. Qualitative and Quantitative Analysis of a Bio-PEPA Model of the Gp130/JAK/STAT Signalling Pathway. *Trans. on Computational and Systems Biology XI*, pages 90–115, 2009.

[10] M. Heiner. Understanding network behavior by structured representations of transition invariants. In *Algorithmic Bioprocesses*. Springer-Verlag, 2009.

[11] M. Heiner, D. Gilbert, and R. Donaldson. Petri Nets for Systems and Synthetic Biology. In *SFM'08*, volume 5016 of *LNCS*, pages 215–264, 2008.

[12] P. Kemper and C. Tepper. Automated trace analysis of discrete-event system models. *IEEE Transactions on Software Engineering*, 35(2):195–208, 2009.

[13] X. Liu and D. Gilbert. BioNessie: A biochemical networks simulation environment. *Journal of Cell Research*, 2010. To appear.

[14] J. Martinez and M. Silva. A simple and fast algorithm to obtain all invariants of a generalized Petri net. In *Selected Papers from the First and the Second European Workshop on Application and Theory of Petri Nets*, pages 301–310. Springer-Verlag, 1982.

[15] MATLAB. http://www.mathworks.com/.

[16] MC2(PLTLc) Home Page. http://www.brc.dcs.gla.ac.uk/software/mc2/.

[17] B. O. Palsson. *Systems Biology: Properties of Reconstructed Networks*. Cambridge University Press, 2006.

[18] A. Sackmann, D. Formanowicz, P. Formanowicz, I. Koch, and J. Blazewicz. An analysis of the Petri net based model of the human body iron homeostasis process. *Computational Biology and Chemistry*, 31(1):1–10, 2007.

[19] S. Schuster and C. Hilgetag. What information about the conserved-moiety structure of chemical reaction systems can be derived from their stoichiometry? *Journal of Physical Chemistry*, 99(20):8017–8023, 1995.

[20] Traviando Home Page. www.cs.wm.edu/~kemper/traviando.html.