Conservation of Mass Analysis for Bio-PEPA

Allan Clark[1]

SynthSys — Edinburgh
CH Waddington Building, Kings Buildings, Edinburgh, Scotland

Stephen Gilmore[2]

School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh, Scotland

Maria Luisa Guerriero[3]

Systems Biology Ireland, Conway Institute
University College Dublin, Belfield, Dublin 4, Ireland.

Jane Hillston[4]

School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh, Scotland

Abstract

This paper describes a static analysis for Bio-PEPA models based on the notion of conservation of mass. Failure to obey the law of mass conservation can be an indication that there is an error in the model description. Here we focus on the use of invariant analysis to identify such potential flaws in models. We extend the basic technique to consider *open* models, in which it is possible to automatically ignore some causes of mass production or consumption that are unlikely to be errors. Our approach is an improvement on direct application of invariant analysis because it does not depend on a deep understanding of the model and prior expectations of the sets of components which should have conserved mass. We demonstrate the use of our technique on a published model from the literature and explain how our analysis can be used to uncover potential problems in the model description. Of course, not all models which fail to conserve mass are flawed. Nevertheless, this represents an important method of model verification which can be applied before the model itself is evaluated — since the analysis does not depend on accurate dynamics it can be undertaken early in the model development process, before the model has been fully parameterised.

[1] Email: A.D.Clark@ed.ac.uk
[2] Email: Stephen.Gilmore@ed.ac.uk
[3] Email: Maria.Guerriero@ucd.ie
[4] Email: Jane.Hillston@ed.ac.uk

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

1 Introduction

As modelling approaches for systems biology grow in sophistication, and experimental techniques gather ever more data about biological function, constructed models grow ever more complex. One of the advantages of using a high-level modelling language, such as a process algebra, rather than the low-level mathematics, such as ordinary differential equations, is that the language facilitates automated techniques to support the modeller in constructing models that are faithful to her intentions, i.e. her current understanding of the biological mechanisms at play.

This paper is concerned with a particular kind of static analysis of dynamic biological models aimed at uncovering problems and errors in the model description as early as possible in the model development process and in particular before the model is simulated to produce analysis results. Our analysis is concerned with the conservation of mass in a model and the use of invariant generation to identify definitions of species or reactions in a model which may violate the principle of conservation of mass.

We are exclusively interested in automatic analysis procedures which can be applied without human intervention and automated with acceptable efficiency in practice. Such procedures can then be applied to every version of a biological model produced during the model development process, leading to a supervision process which we believe lessens the possibility of undetected errors in models. We have previously [9] considered the verification of Bio-PEPA models using a combination of static and dynamic analysis. In the present paper we extend the repertoire of static analyses.

A significant component of the analysis is the ability to generate invariants over the chemical species in the model which identify sets of species whose total quantity remains unchanged throughout simulation. Invariants such as these can be calculated from the stoichiometric information in a model using the Fourier-Motzkin elimination procedure [12]. This is a procedure of numerical linear algebra which operates on the stoichiometric matrix in order to calculate invariants over species and invariants over reactions. The Fourier-Motzkin procedure is known to have doubly exponential running time in the worst case but for small or medium-sized models we have always found the running time of the algorithm to be acceptable in practice.

1.1 Positioning with respect to the state-of-the-art

Many researchers working on static analysis of biological models have been generating invariants and inspecting these by eye. Thus, although the computation of invariants is automatic the modeller has been until now left with the problem of deciding whether the invariants which have been computed are the expected set of invariants for this model. If these match then the modeller is reassured that the model has been constructed as intended. If these do not match then this suggests

that an error has been made in constructing the model.

The difficulty with this current common practice is that knowing which invariants are expected often requires a combination of both strong mathematical reasoning skills and deep biological domain knowledge, making it a non-trivial task to decide whether the invariants computed suggest an error in the model. Added to this, standard invariant analysis is not so easily interpreted for models of *open* biological systems in which matter flows into the model via sources and flows out again via sinks. However, such models are entirely legitimate and thus we would like our analysis to apply to these also.

The novel contributions of our paper are

(i) transforming the invariant evaluation problem which relies on the application of human expertise and skill into a decision problem which can be entirely automated to give either a yes or no answer; and

(ii) devising a procedure to apply invariant analysis to models of open biological systems which bounds the problem to form a closed model which can be entirely covered by species invariants.

The discussion in this paper is focused on models written in the process algebra Bio-PEPA [7] which is designed to be particularly applicable to modelling biological systems such as signalling pathways. However much of what is said applies to other kinds of modelling paradigms and in particular to models which can be converted to an SBML [16] model. In fact since this is a qualitative analysis we require only knowledge of the stoichiometry matrix.

2 Related work

Invariant generation and analysis is regularly applied when modelling with Petri nets [19], also in the context of models of biological processes [15,14]. Invariants can be used for a range of analysis purposes (see [2] for a survey), including guiding the modular decomposition of large models [13].

Most implementations of invariant generation require the complete stoichiometric description of a model [5] and work forwards from this using the Fourier-Motzkin procedure to generate a set of invariants. In contrast, the Traviando trace analysis tool [17] contains a novel on-the-fly algorithm to infer a set of invariants from a trace generated by a discrete stochastic simulation. An implementation of the Fourier-Motzkin elimination procedure due to Peter Kemper is included in the Bio-PEPA tool suite in the Bio-PEPA Eclipse Plug-in [6].

We are working here with a high-level language which gives us the considerable advantage of being able to switch between different views of a model and between different regimes for dynamic analysis. The latter has already proven to be

[5] In Petri nets terms, the incidence matrix.

valuable in detecting previously unknown problems with the analysis of biological models [4]. Papers working directly with ODE models in order to find errors appear to be relatively rare, and when this is done – as in [20,18] – the authors require two independent implementations of their mathematical model, then need to generate residuals with fixed geometric properties, and subsequently isolate errors using feature matrices which describe the subspace imposed by such errors. Most of the methods required here would be relatively unfamiliar to most biologists and are not automated. In contrast we have automated methods which relate to the well-known notion of conserved moieties, familiar to biological modellers.

3 The Bio-PEPA language

Bio-PEPA [7] is a stochastic process algebra for modelling and analysis of biochemical systems. We give here a brief overview of the main features of the language. For a detailed presentation of its syntax and semantics, see [7].

In a Bio-PEPA model of a biochemical system, each molecular species (i.e. proteins, genes, mRNAs) is represented by a process. The state of the system at a given time is given by the current amount of the molecular species, and the result of the occurrence of a biochemical reaction is a change in the available amount of the involved species.

Processes interact by means of shared action names representing reactions and specifying their role in the reaction (reactant, product, catalyser, inhibitor, etc.) and their stoichiometric coefficient for that reaction. The effect of a reaction occurrence is to decrease the amount of reactants and increase the amount of products according to the stoichiometry [6] .

Species amounts in Bio-PEPA can either be concentrations (continuous semantics) or molecule counts (discrete semantics), hence allowing both numerical methods based on differential equations and also stochastic analysis either via stochastic simulation using the Gillespie algorithm or by numerical evaluation of the underlying continuous-time Markov chain. For each biochemical species, the modeller specifies the set of reactions in which the species is involved and the role of the species in each reaction. Each reaction is associated with a kinetic law which specifies the rate of occurrence of that reaction.

Formally, the main components of a Bio-PEPA system are the *species components*, describing the behaviour of each species, and the *model component*, specifying all interactions and initial amounts of species. The syntax of Bio-PEPA

[6] The stoichiometry of a species with respect to a reaction indicates how many molecules of this species are produced or consumed by this reaction. In Bio-PEPA the semantics automatically adjusts the quantitative variable for a species to reflect the stoichiometry whenever a reaction occurs.

components is given by:

$$S ::= (\alpha, \kappa) \odot S \mid S + S \mid C \quad \text{with} \quad \odot = \downarrow \mid \uparrow \mid \oplus \mid \ominus \mid \odot$$
$$P ::= P \underset{\mathscr{I}}{\bowtie} P \mid S(x)$$

where S is a *species component* and P is a *model component*. In the prefix term $(\alpha, \kappa) \odot S$, κ is the *stoichiometry coefficient* of species S in reaction α, and the *prefix combinator* "$\odot$" 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 $\alpha \odot$ in the definition of species S is a shorthand for $(\alpha, \kappa) \odot S$ 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 $P \underset{\mathscr{I}}{\bowtie} Q$ denotes synchronisation between components P and Q; the set $\mathscr{I}$ 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.

Here we illustrate the basic concepts using the following simple example. A reaction $S \xrightarrow{E} P$ which converts a substrate molecule S into a product molecule P catalysed by an enzyme E is modelled in Bio-PEPA as

$$S \stackrel{def}{=} r_1 \downarrow$$
$$P \stackrel{def}{=} r_1 \uparrow$$
$$E \stackrel{def}{=} r_1 \oplus$$

where r_1 is a name associated with the reaction. The kinetic law of r_1 is defined by the Michaelis-Menten kinetics

$$r_1 = \frac{k_{\text{cat}} \cdot E \cdot S}{K_{\text{M}} + S}$$

and k_{cat} and K_{M} are the reaction kinetic constants.

This represents the fact that S, P and E are all involved in the occurrence of reaction r_1, and that the result of the occurrence of r_1 is to decrease the total amount of S molecules ($\downarrow$) and increase the total amount of P molecules ($\uparrow$); the role of the enzyme E is to speed up the reaction, but its amount is unaffected ($\oplus$).

The Bio-PEPA Eclipse Plug-in [11] is a software framework for Bio-PEPA model development and analysis. In addition to dynamic time-series analysis via

stochastic simulation and the solution of differential equations, the Bio-PEPA Eclipse Plug-in enables modellers to perform static analysis — such as the identification of invariants, sources and sinks described in the following. More information on the Bio-PEPA language and on the features of the tool and its import/export formats can be found in [3,6,11]. The Bio-PEPA software tools are available from http://www.biopepa.org/.

4 Invariants

In this section we review the definition of an invariant with respect to a Bio-PEPA model and the process of computing the set of invariants for an entire model. The set of invariants for a model is usually computed during model construction in order to assist in model validation. There are two kinds of computed invariants; state invariants and reaction invariants. A *state invariant* involves a set of components or species in the model. At any time during a simulation one may sum together the populations of the components of a state invariant and the result will always be the same. Consider the following simple model presented in reaction syntax on the left and Bio-PEPA syntax on the right:

$$\begin{array}{l|l} E + S \xrightarrow{r} ES & E \stackrel{def}{=} r\downarrow + rm\uparrow + p\uparrow \\ ES \xrightarrow{rm} E + S & S \stackrel{def}{=} r\downarrow + rm\uparrow \\ ES \xrightarrow{p} E + P & ES \stackrel{def}{=} r\uparrow + rm\downarrow + p\downarrow \\ & P \stackrel{def}{=} p\uparrow \end{array}$$

In this model there are exactly two state invariants: $E + ES$ and $S + ES + P$. To see this, consider symbolically combining the Bio-PEPA definitions of the species E and the species ES. Each occurrence of a $\uparrow$ reaction is then matched with its corresponding $\downarrow$ reaction. The same is true when the definitions of S, ES and P are combined. In contrast, the quantity $E + S$ is *not* an invariant of this model and we can see this when the definitions of the species E and S are combined because the $p\uparrow$ term in the Bio-PEPA definition of the species E is not matched by a corresponding $p\downarrow$ term in the definition of S.

More generally, a state invariant may have a set of coefficients such that we may for example say that $(1 \times O_2) + (2 \times O)$ is a state invariant. A coefficient may also be negative. We may have that $A - B$ is invariant in a model in which the species A and the species B are only ever produced or consumed together.

A *reaction invariant* is a set of reactions such that from any state X that the model may reach, if one of each of the reactions in the reaction invariant is fired in sequence then the model is returned to X. In the example model above there is only

one reaction invariant: $r + rm$, as illustrated below,

$$E + S \xrightarrow{r} ES \xrightarrow{rm} E + S.$$

As in the case of state invariants, a reaction invariant may include a set of coefficients, such that some of the reactions may be required to fire more than once to return the model to the original state. Note that it does not matter in which order the reactions are fired. Reactions not included in the invariant may be interspersed with the reactions of an invariant, in which case the effect on the state of the model will be the same as if only the interspersed reactions had occurred, and therefore we will not return to the original state unless the interspersed reactions form a reaction invariant themselves.

5 Invariants, sources and sinks

In this section we detail the relationship of both component and reaction invariants to sources and sinks within the model. A model can have both sources and sinks. In the interests of readability we refer to sources and sinks collectively as *taps*. Taps may be component-based or reaction-based. A component source is consumed by at least one reaction but is never produced by any reactions. Conversely a component sink is a component which is produced by at least one reaction but never consumed by any reaction. The syntax of Bio-PEPA makes the identification of component taps trivial; a component is a source if its definition contains at least one $\downarrow$ operator and no $\uparrow$ operators, although it may contain $\oplus$, $\ominus$ or $\odot$. Conversely the definition of a sink component contains at least one $\uparrow$ operator and no $\downarrow$ operators. The following snippet of Bio-PEPA highlights this.

$$Source \stackrel{def}{=} a\downarrow + b\downarrow + c\oplus$$
$$Sink \stackrel{def}{=} a\uparrow + b\ominus + c\uparrow$$

Reaction taps are defined analogously. That is, a reaction source is a reaction which has no reactants but at least one product and a reaction sink is a reaction which has no products and at least one reactant. Simply put, a reaction source produces something but does not consume anything whereas a reaction sink consumes something but does not produce anything. In the Bio-PEPA syntax, while it is trivial to observe component source and sinks, reaction source and sinks can only be identified by viewing the entire model. The Bio-PEPA software however provides an outline view of your model showing reactions and in this view reactions which are source or sink reactions are trivial to identify. In any case the outline view lists all component and reaction sources and sinks. The following snippet shows two

reactions one of which is a source and the other of which is a sink.

$$source\_reaction : \longrightarrow P_1 + P_2$$
$$sink\_reaction : R_1 + R_2 \longrightarrow$$

5.1 Taps mark boundaries

When constructing a model the modeller must choose which features of a physical system to include. Components of the real system which are included are called *model components*. Those which are excluded are called *external components* or collectively referred to as the *external environment*. By their nature, models are finite in extent and scope. The external components are essentially everything not mentioned in the model.

For a model to be useful we hope that either in the physical system the influence of the external components on the model components is negligible or that this influence can be ignored for the purposes of the current analysis. This leads to boundaries between the model components and the external components and here we wish to argue that taps in the model represent such boundaries. (Note that we are not claiming that such boundaries are *only* represented by taps in the model.)

A reaction sink forms a natural boundary between the model components and the external environment. This is because from the point of view of the model a reaction sink is removing mass from the system. In reality, mass does not reduce to nothing, but the species that it does reduce to is an external component and hence not mentioned in the model. The same is of course true for a reaction source. Mass is not produced from nothing, but instead from an external component not mentioned in the model.

5.2 Invariants affected by taps

None of the reactions which modify a component tap can be involved in any reaction invariant. The reason for this is straightforward: if a component P is a source then any reaction r which modifies P must decrease the population of P, since there are no reactions which increase the population of a component source. The original population of component P before a firing of reaction r can never be restored by any combination of other reactions in the model. Thus, reaction r cannot be involved in any reaction invariant. The same reasoning applies to component sinks. A component tap may however be involved in a component invariant.

A reaction tap may be part of a reaction invariant but a component listed as a reactant or product of a reaction tap cannot form part of a component invariant. The reasoning is fairly straightforward when we consider a reaction source r. If r modifies the population of P it must *increase* the population P, because r is a reaction source. Being a source, r cannot reduce the population of any other component.

The consequence is that any putative invariant in which P is involved would also have its value increased by a firing of r and therefore would not be an invariant at all. Analogous reasoning applies in the case that r is a sink reaction. However just as a component tap may be involved in a component invariant, a reaction tap may be involved in a reaction invariant/loop, as in:

$$\begin{aligned} source\_reaction &: \longrightarrow P \\ sink\_reaction &: P \longrightarrow \end{aligned}$$

5.3 Removing reactions and components

When modelling it is sometimes desirable to remove a component or a reaction from the description of the model. This may be done to see the effect it has on the evaluation of the model. Removal may be done by hand, or can be automated in modelling software. In this section we briefly describe the removal of components and reactions from a Bio-PEPA model and then how this affects invariants.

The removal of a reaction is straightforward: simply delete the kinetic law and update any component description by removing the corresponding reaction behaviour from the component definition. The following shows the removal of a reaction named *am*. The model before removal is on the left; the model after removal is on the right.

$$\begin{aligned} E &\stackrel{def}{=} a\downarrow + am\uparrow + b\uparrow \\ S &\stackrel{def}{=} a\downarrow + am\uparrow \\ ES &\stackrel{def}{=} a\uparrow + am\downarrow + b\downarrow \\ P &\stackrel{def}{=} b\uparrow \end{aligned} \qquad \Bigg| \qquad \begin{aligned} E &\stackrel{def}{=} a\downarrow + b\uparrow \\ S &\stackrel{def}{=} a\downarrow \\ ES &\stackrel{def}{=} a\uparrow + b\downarrow \\ P &\stackrel{def}{=} b\uparrow \end{aligned}$$

If we remove the only reaction which a species is involved in then that species simply becomes a constant equal to its initial population.

Removal of a component involves simply the removal of that component's definition. In Bio-PEPA this automatically involves updating the reaction descriptions of any reactions in which the deleted component was involved. This then raises the question of what should be done to kinetic laws which involve the deleted component's population. In most cases it can be safely deleted from the kinetic law. Further treatment of this issue is outside the scope of this paper, since our main analysis is a rateless analysis.

What happens to the computed set of invariants when we remove a reaction from the model? The first observation is that by removing a reaction from the model we cannot invalidate a component invariant: any group of components which previously formed a component invariant will still do so. (If we removed all the reactions from a model then all of the components would have constant populations.) This

is illustrated by the example above where we can see that the component invariants are the same — $E + ES$ and $S + ES + P$ — in both the left hand and the right hand model.

Our second observation is that by removing a reaction we may create more component invariants, if we remove a reaction which violates an invariant property and there are no other reactions left which violate the candidate invariant property. A reaction invariant is disrupted by the removal of any reaction within the reaction invariant, but undisturbed by the removal of any reaction not within the reaction invariant.

What happens to the computed set of invariants when a component is removed from the model? If the removed component is part of a component invariant then naturally that invariant may be invalidated. However the removal of a component will not disrupt any component invariants in which it is not involved nor will it disrupt any reaction invariants. It may however cause the invariant analysis to report what was previously a single (reaction or component) invariant as two or more invariants. This is because the analysis procedure reports the set of minimal invariants and although an invariant will not be invalidated by the removal of a component it may cease to be a minimal invariant.

6 Conservation of Mass

We have discussed the invariants of a model and the taps which are the sources and sinks of a model. We have also discussed the effect on invariant analysis of the removal of components or reactions from the model. In this section we combine these concepts in order to check the consistency of a model.

Having performed invariant analysis over the model we can check if mass is conserved because every component in the model should be covered by at least one invariant. We can then obtain a single invariant which covers the entire model by summing together all of the component invariants in the model. This provides us with a convenient static analysis consistency check on all kinds of biological models. This check is always applicable: invariants can always be summed because the sum of two constants is a constant.

However, due to their finite extent, it is often the case in a biological model that mass is not conserved. This occurs because mass is lost to or gained from the external components which are not within the scope of the model. A simple way in which this occurs is to have a production or degradation reaction which produces or removes the mass of one or more model components. These reactions show up as source and sink reactions and are the reason that mass appears not to be conserved within the model.

As we have observed before, removing reactions from the model does not invalidate any component invariants. We utilise this by removing from the model all tap reactions, that is all source and sink reactions. Once we have done this we have

removed from the model any obvious means by which mass may be produced or consumed. Performing invariant analysis on the model with tap reactions removed should give us total invariant cover. That is, all components in the resulting model should be covered by at least one invariant. If that is not the case then it means that mass is produced or consumed internally within the model.

6.1 Summary of Analysis Steps

This section summarizes the steps to detecting, finding and fixing flaws in a model using conservation of mass analysis.

- Run the invariant check. The software automatically ignores all tap reactions and calculates the set of invariants. A warning is issued if not all components in the model are covered by some invariant indicating that mass is not conserved by the model. Additionally the modeller is shown the set of model components which are not covered by any invariant.
- Re-run the invariant check once for each reaction in the model. Each analysis ignores all of the tap reactions plus one of the non-tap reactions in the model. Each analysis gives a set of components which are not covered by any component invariant. All of these sets are equal to (or a subset of) the species which are not covered by any invariant when no non-tap reactions are ignored. This requires an invariant analysis run for every reaction in the model. This is tolerable in practice; the analysis for the model in our case study required less than one second on a conventional desktop computer.
- This provides a suspect list of reactions which may be contributing to the loss or production of mass in the model. Examine this list and in particular look for pairs of similar reactions. These should be similar both in their component participants and in the set of uncovered components produced by ignoring either of the two reactions. The invariant check can be performed while ignoring all the tap reactions plus both of the reactions in such a pair. Closely examine any reaction (or reaction pair) whose removal causes the analysis to determine that mass is conserved.
- If a reaction is found to have been defined in error then update the model description to amend the reaction and re-run the conservation of mass analysis. In Bio-PEPA amending the reaction(s) consists of modifying the component definitions to either add or remove components as reaction reactants or products. If the model is advanced enough such that rate laws have been written for reactions, it is a good time now to update the rate law(s) associated with the modified reactions. The Bio-PEPA software can again help with this, as there are warnings produced when a rate law expression does not contain a reference to a reactant, or does contain a reference to a non-reactant. This is more fully discussed in [10].

M1	TNFα	M17	NF-κB
M2	TNFR1	M18	RIP1/Caspase-8
M3	TNFα/TNFR1	M19	RIP1n
M4	TRADD	M20	RIP1c
M5	TNFα/TNFR1/TRADD	M21	FADD
M6	TRAF2	M22	Caspase-8
M7	TNFα/TNFR1/TRADD/TRAF2	M23	TNFα/TNFR1/TRADD/FADD
M8	IKK	M24	TNFα/TNFR1/TRADD/FADD/Caspase-8
M9	TNFα/TNFR1/TRADD/TRAF2/IKK	M25	Caspase-8*
M10	RIP1	M26	Caspase-8*/Effector
M11	TNFα/TNFR1/TRADD/TRAF2/RIP1	M27	Effector*
M12	TNFα/TNFR1/TRADD/TRAF2/RIP1/IKK	M28	DNA fragmentation
M13	IKK*	M29	Effector
M14	IκB/NF-κB	M30	Effector/c-IAP
M15	IκB/NF-κB/IKK*	M31	c-IAP
M16	IκB-P		

Fig. 1. ODE variable names and biological variable names in the model

7 Case Study

In this section we provide a case study to illustrate our techniques. The example model has already been analysed using our previous invariant analysis techniques [9,10]. We update the case study here to include our methods of finding the particular flaws in the model revealed by conservation of mass analysis and in particular how this analysis guides us to the erroneous parts of our model.

Our method works directly from the formal text of the differential equations and does not require additional graphical representations or other supplementary non-formal designs. Of course, we recommend working with high-level languages such as process algebras or Petri nets and generating differential equations from these, but many practitioners begin with ODE models. The model which we consider in our case study was presented in [5] as a series of ordinary differential equations which we first had to hand-translate into a Bio-PEPA model. This step represents another potential source of flaws in the model which we are keen to detect before any quantitative analysis is performed. Schematic variable names are preserved from the ODEs. These are related to (biologically) meaningful names in Fig 1.

The model outline computed by the Bio-PEPA software is provided in Fig. 2. This includes two additional reactions $r20alt$ and $r29alt$ which represent possible improvements from an earlier pass of model validation reported in [10]. The purpose here is to represent the methodology used in finding errors in models prior to evaluation of model results.

31 Species

$M1$
- with initial #molecules = 30
- $r1, M1 + M2 \rightarrow M3$
- $r2, M3 \rightarrow M1 + M2$

$M2$
- with initial #molecules = 15
- $r1, M1 + M2 \rightarrow M3$
- $r13, M12 \rightarrow M2 + M4 + M6 + M10 + M13$
- $r2, M3 \rightarrow M1 + M2$
- $r24, M24 \rightarrow M2 + M4 + M6 + M10 + M21 + M25$

$M28$
- with initial #molecules = 0 (*is-sink*)
- $r28, M27 \rightarrow M28$

$M29$
- with initial #molecules = 10
- $r25, M25 + M29 \rightarrow M26$
- $r26, M26 \rightarrow M25 + M29$
- $r29, M29 \rightarrow M30$
- $r30, M30 \rightarrow M29 + M31$

$M5$
- with initial #molecules = 0
- $r20, M5 \rightarrow$
- $r21, M23 \rightarrow M5 + M21$
- $r3, M3 + M4 \rightarrow M5$
- $r4, M5 \rightarrow M3 + M4$
- $r5, M5 + M10 \rightarrow M11$
- $r6, M11 \rightarrow M5 + M10$
- $r7, M5 + M6 \rightarrow M7$
- $r8, M7 \rightarrow M5 + M6$

...

26 species omitted

33 Reactions

- $r19, M17 \rightarrow M31$
- $r2, M3 \rightarrow M1 + M2$
- $r20, M5 \rightarrow$
- $r20alt, M21 \rightarrow M23$
- $r21, M23 \rightarrow M5 + M21$
- $r22, M22 + M23 \rightarrow M24$
- $r23, M24 \rightarrow M22 + M23$
- $r24, M24 \rightarrow M2 + M4 + M6 + M10 + M21 + M25$
- $r25, M25 + M29 \rightarrow M26$
- $r26, M26 \rightarrow M25 + M29$
- $r27, M26 \rightarrow M22 + M27$
- $r28, M27 \rightarrow M28$
- $r29, M29 \rightarrow M30$
- $r29alt, M31 \rightarrow$
- ...
- 19 reactions omitted

---

4 Sinks

- $M16$
- $M19$
- $M20$
- $M28$

---

2 Sink actions

- $r20, M5 \rightarrow$
- $r29alt, M31 \rightarrow$

Fig. 2. A condensed-for-space version of the outline view inferred from the Bio-PEPA model of the TNFα-mediated NF-κB signal transduction pathway. The $r20alt$ and $r29alt$ reactions have been added to the Bio-PEPA model during an earlier pass of model validation reported in [10].

7.1 Initial Invariant Analysis

Our initial invariant analysis may provide some clue as to the veracity of this model for anyone with a deep understanding of the intended semantics of the model and in particular which state invariants should hold within the system. The results of invariant analysis give six component invariants and eleven components not covered by any invariants, as reported in Fig. 3.

The fact that there are uncovered species might give us cause for concern, but we note that some of these are trivially expected, for example the component $M31$ is involved in the sink reaction: $r29alt, M31 \longrightarrow$, and the component $M5$ is involved

State Invariants:

(i) $M21 + M23 + M24$
(ii) $M12 + M13 + M15 + M8 + M9$
(iii) $M14 + M15 + M16$
(iv) $M18 + M22 + M24 + M25 + M26$
(v) $M26 + M27 + M28 + M29 + M30$
(vi) $M10 + M11 + M12 + M18 + M22 + M24 + M26 + M29 + M30$

Uncovered Species:

- $\{M1, M17, M19, M2, M20, M3, M31, M4, M5, M6, M7\}$

Fig. 3. The component invariants and reaction invariants computed for the TNFα-mediated NF-κB signal transduction pathway by the Bio-PEPA Eclipse Plug-in

in the sink reaction: $r20, M5 \longrightarrow$. We cannot expect either of these components to be included in any state invariant.

However when we repeat this analysis, choosing to ignore all tap reactions, we find the same set of invariants and the same set of components uncovered by any invariant. This removes the doubt that the uncovered components were caused by mass being produced or consumed at the boundary between the model components and the external environment. In other words our model is either producing or consuming mass within the model components. Although we would require a rather deep understanding of the model and of invariant laws to anticipate the expected set of invariants in this model we require only a relatively shallow understanding to know that we expect mass to be conserved by the model components.

7.2 Compositional Reduction Analysis

In this section we narrow down the causes of the non-conservation of mass detected in our system by iteratively re-analysing the model for invariant coverage whilst successively removing each reaction. We re-run the analysis once for every non-tap reaction. In each run all tap reactions plus one other non-tap reaction are ignored. We call this 'reduction analysis' since the model is smaller than the entire original model. It is compositional because we do this one reaction at a time.

We will use $\mathscr{L}$ to denote the original set of uncovered species. Table 1 shows the reductions relative to $\mathscr{L}$ when each reaction is, in turn, ignored for the purposes of the invariants check. The results in the first three lines of the table tell us that those reactions are not independently responsible for causing the non-conservation of mass. This has narrowed our search from thirty-one initial non-tap reactions to

Ignored Reaction	Uncovered Component Set
$r1 \ldots r17$	$\mathscr{L}$
$r25 \ldots r28,$	$\mathscr{L}$
$r20, r22, r23, r29alt, r31$	$\mathscr{L}$
$r18, r24$	$\mathscr{L} - \{M19, M20, M6, M7\}$
$r19$	$\mathscr{L} - \{M17\}$
$r20alt$	$\{M17, M31\}$
$r21$	$\mathscr{L} - \{M1, M3, M5, M7\}$
$r29, r30$	$\mathscr{L} - \{M17, M31\}$

Table 1
Table showing the effect which ignoring each reaction has on the set of components that are uncovered by any invariant. Only the removal of reactions $r18, r24, r19, r20alt, r21, r29$ and $r30$ have any effect.

seven reactions. These reactions are:

$$\begin{aligned}
r18 &: M18 \longrightarrow M19 + M20 + M22 \\
r24 &: M24 \longrightarrow M2 + M4 + M6 + M10 + M21 + M25 \\
r19 &: M17 \longrightarrow M31 \\
r20alt &: M21 \longrightarrow M23 \\
r21 &: M23 \longrightarrow M5 + M21 \\
r29 &: M29 \longrightarrow M30 \\
r30 &: M30 \longrightarrow M29 + M31
\end{aligned}$$

One point of interest in Table 1 is that there are two rows consisting of a pair of reactions, namely: $\{r18, r24\}$ and $\{r29, r30\}$. Removing either of the two reactions in these pairs 'fixes' the same set of components. The first pair seems unrelated, however examining the second pair we see a simple loop which creates mass. Reaction $r29$ consumes $M29$ producing $M30$, which is consumed by reaction $r30$ in producing both the original $M29$ and an additional $M31$. This seems likely to be a flaw in the model, either reaction $r29$ should also consume $M31$ or reaction $r30$ should not produce $M31$. Of course in order to actually fix this flaw one must fully understand the intention of the model to begin with. However in this case we can re-write the two reactions giving the involved components their descriptive biological names.

$$\begin{aligned}
r29 &: \text{Effector} \longrightarrow \text{Effector-IAP} \\
r30 &: \text{Effector-IAP} \longrightarrow \text{Effector} + \text{c-IAP}
\end{aligned}$$

From these two reactions we cautiously made the choice to amend reaction $r29$ to

Ignored Reaction	Uncovered Component Set
$r1 \ldots r17$	$\mathscr{M}$
$r19 \ldots r20$	$\mathscr{M}$
$r22, r23$	$\mathscr{M}$
$r26, r31$	$\mathscr{M}$
$r18, r24$	$\mathscr{M} - \{M6, M7, M19, M20\}$
$r21$	$\mathscr{M} - \{M1, M3, M5, M7\}$
$r20alt$	$\{\}$

Table 2
Table showing the effect which ignoring each reaction has on the set of components that are uncovered by any invariant for the model with a corrected version of reaction r29. Only the removal of reactions $r18, r24, r21$ and $r20alt$ have any effect.

consume c-IAP.

Having made this correction we re-analyse the model and observe that mass is still not conserved. This matches our expectations since the flaw that we have corrected allowed the production of the component $M31$. However, in the original model $M31$ is only involved in its own production. Hence the fact that there are other uncovered components could not have been caused by the production of this single component. We therefore go through the steps again and find a similar table to Table 1. Table 2 shows an updated version of the original table, this time analysing the model with a corrected version of reaction r29. Additionally $\mathscr{M} = \mathscr{L} - \{M17, M31\}$.

The most significant point of interest in Table 2 is the result for reaction $r20alt$. Removing $r20alt$ means that the invariant analysis calculates that mass is conserved within the model. This strongly implicates $r20alt$ or a related reaction. However a single reaction on its own cannot be at fault, there are only three other reactions which modify the set of uncovered components. We look at $r21$ first because it seems to be the most closely related to $r20alt$ and because $r18$ and $r24$ appear to be a pair. The two reactions $r20alt$ and $r21$ are:

$$r20alt : M21 \longrightarrow M23$$
$$r21 : M23 \longrightarrow M5 + M21$$

As before we have identified a simple loop: if $r20alt$ and $r21$ both fire once, the model is returned to the same original state but with an increased count of $M5$. Again the correction is that either $r20alt$ should have $M5$ as a reactant or $r21$ should not have $M5$ as a product. Just as before, we cannot authoritatively say which should be the optimal correction for this model but re-writing the reactions with

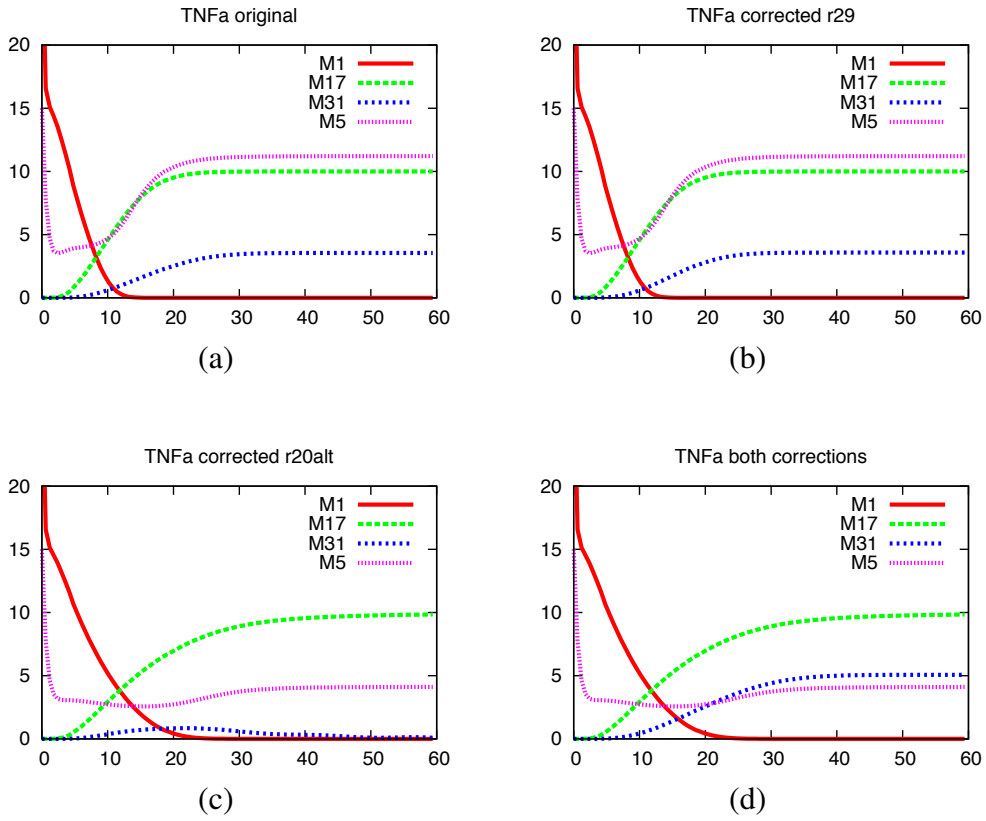

Fig. 4. Timeseries analysis of the four models: (a) the original model; (b) the model with $r29$ corrected; (c) the model with $r20$ corrected; and (d) the model with both flaws corrected. Clearly the biological insights which could be derived from (d) are quite different from the insights which could be derived from (a) – the results are qualitatively different.

the original component names gives us something of an indication:

$$r20alt : \text{FADD} \longrightarrow \text{TNF}\alpha\text{/TNFR1/TRADD/FADD}$$
$$r21 : \text{TNF}\alpha\text{/TNFR1/TRADD/FADD} \longrightarrow \text{TNF}\alpha\text{/TNFR1/TRADD} + \text{FADD}$$

It appears likely that $M5$ (TNFα/TNFR1/TRADD) is missing as a reactant of the reaction $r20alt$. Having fixed this in our model we re-analyse the model and as expected the analysis returns that mass is conserved in our new model.

7.3 Results

Figure 4 depicts the results from timeseries analysis of four versions of the TNFα-mediated NF-κB signal transduction pathway model. The first version is the original model before we began any conservation of mass analysis. The second version has an updated version of reaction $r29$ introduced to fix the first flaw we have detected. The third version corrects the original model with respect to our second

found flaw, as a result this updates the reaction $r20alt$. The fourth and final version updates both reactions $r29$ and $r20alt$ to represent a model which internally conserves mass.

Not all species in the model have been plotted for the sake of clarity. The interesting result is the progression of the population of the species $M5$. In both the first two versions in which $M5$ has been erroneously left out of the reactant list for reaction $r20alt$ after an initial decline in population size this is recovered and settles at an equilibrium value of over 10. However when the reaction is fixed to include $M5$ as a reactant, the population of $M5$ has only a brief attempt at a recovery before settling into an equilibrium value of less than 5. The correction we made to reaction $r29$ has a less obvious effect.

8 Limitations

In this section we discuss the limitations of our approach. No static analysis can detect all flaws in models, and we are not hoping to do this here. Additionally, any static analysis which warns about possible errors in a model becomes almost useless if the analysis produces too many false positives. When this occurs the modeller is likely to begin to ignore the analysis results. In this section we discuss some reasons for which there may be false positives and we argue that these do not detract from the general usefulness of our approach.

The most obvious scenario in which our conservation of mass analysis has difficulty is if the model itself is not expected to conserve mass. This is likely when the scale of the model is higher than a molecular level, for example cellular or even organism level.

A legitimate block occurs when using a biological model to measure population levels with birth and death rates, for example epidemiological models in Bio-PEPA are considered in [8]. In these cases, "mass" is clearly not expected to be conserved and hence we would expect our analysis to highlight problems with reactions which are in fact perfectly correct. Another example would be at the celluar level in which the growth and splitting of a single cell into two daughter cells (cytokinesis) is modelled.

Our analysis depends on the process of partitioning the set of all reactions into two sets; of tap and non-tap reactions. This allows us to ignore reactions which introduce mass into the model components from outside of the scope of the model, or discard mass from the model out to the external environment. In turn this allows us to focus on the reactions entirely within the scope of the model to determine whether mass is conserved there.

If reactions are written down correctly with respect to the model components, then our partitioning is conservative in the sense that a non-tap reaction is not erroneously ascribed as a tap reaction (although a tap reaction may be identified as a non-tap reaction). This is the correct relationship since the tap reactions are ignored

for the analysis so we are better to be conservative in ignoring too few rather than too many reactions. Additionally a reaction may be explicitly ignored as is done in the compositional reduction analysis step, see Section 7.2.

9 Conclusions

In this paper we have discussed a methodology for the static analysis of biological models with particular respect to determining whether the model in question conforms to the law of conservation of mass. We are careful to define the boundaries between the portions of the real system which lie within the scope of the model and those which do not. We expect that mass may be lost from the model components to the external environment or gained by the model components from the external environment. We ignore reactions which cause such losses or gains in order to determine whether reactions which describe behaviour entirely captured within the model conserve mass. If we determine that the model fails to conserve mass then we search for the reactions which are the cause. This involves compositional application of the conservation of mass analysis, for each reaction within the model scope. For each reaction we compare the results of conservation of mass analysis, with and without that reaction. This can dramatically narrow the search for erroneous reactions within the model.

We believe that our analysis has several advantages. It is a qualitative analysis which means that it can be performed on the model at all stages of development and in particular before parameters are finalised or even estimated.

In addition our analysis is quite inexpensive, although as we have previously noted [10], the invariant analysis used can be exponential in the worst case, practice has shown that this rarely occurs. The invariants check can be performed efficiently enough on a typical desktop computer to be perceived as instantaneous for models involving more than forty reactions.

The conservation analysis check can be performed automatically with little or no understanding of the model. Once a fault has been determined to exist, this can even be sought without detailed knowledge of the nature of the model. To actually modify the model in order to repair an error will of course require knowledge of the intentions of the modeller. However the point we wish to make here is that invariant analysis and the general expectation of conservation of mass combine well to form an automatic evaluation of the model. Previously we had relied on the modeller having a pre-existing expectation of the set of invariants which should be computed.

Although all of our analysis has taken place within the setting of the process algebra Bio-PEPA, the analysis is not only applicable there. Any formalism which can be written as a reaction matrix may adopt this analysis. We have performed this analysis over a Bio-PEPA model which included a use of compartments. This involved no extra work in order to perform our mass conservation analysis.

Combining these two final points we have begun work on the automatic classification of a large database of SBML models. We can mechanically categorise all such models into those which conserve mass and those which appear not to. The latter set can be later analysed more closely by a human who will be presented with a good idea of the reactions which may be erroneous.

We are convinced of the utility of performing static analysis on any kind of model at all stages in development of the model. In this paper we have described an inexpensive, qualitative and useful analysis together with a methodology which greatly reduces the time spent tracking down the source of an error once one has been detected to exist. We are confident that our approach is scalable as we have tested it against the models in the Biomodels database [1]. Models which ranged in size from 4 reactions to 4139 reactions could all be analysed in under 15 seconds.

Acknowledgements:

Clark, Gilmore and Hillston are supported by SynthSys — Edinburgh, a Centre for Integrative Systems Biology (CISB) funded by BBSRC and EPSRC, reference BB/D019621/1. The authors benefited from an introduction to invariant generation by Peter Kemper during his time as a SICSA Distinguished Visiting Fellow.

References

[1] http://www.ebi.ac.uk/biomodels-main/ (2012).

[2] Baldan, P., N. Cocco, A. Marin and M. Simeoni, *Petri nets for modeling metabolic pathways: A survey*, Natural Computing **9** (2010), pp. 955–989.

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

[4] Calder, M., A. Duguid, S. Gilmore and J. Hillston, *Stronger computational modelling of signalling pathways using both continuous and discrete-state methods*, in: *Proc. of CMSB'06*, LNCS **4210**, 2006, pp. 63–77.

[5] Cho, K., S.-Y. Shin, W. Kolch and O. Wolkenhauer, *Experimental design in systems biology, based on parameter sensitivity analysis using a Monte Carlo method: A case study for the TNFα-mediated NF-κB signal transduction pathway*, Simulation **79** (2003), pp. 726–739.

[6] Ciocchetta, F., A. Duguid, S. Gilmore, M. L. Guerriero and J. Hillston, *The Bio-PEPA Tool Suite*, in: *Proc. of QEST'09* (2009), pp. 309–310.
URL http://www.computer.org/portal/web/csdl/doi/10.1109/QEST.2009.27

[7] Ciocchetta, F. and J. Hillston, *Bio-PEPA: A framework for the modelling and analysis of biological systems*, Theoretical Computer Science **Concurrent Systems Biology: To Nadia Busi (1968–2007), 410** (2009), pp. 3065–3084.

[8] Ciocchetta, F. and J. Hillston, *Bio-pepa for epidemiological models*, Electron. Notes Theor. Comput. Sci. **261** (2010), pp. 43–69.
URL http://dx.doi.org/10.1016/j.entcs.2010.01.005

[9] Clark, A., S. Gilmore, M. L. Guerriero and P. Kemper, *On Verifying Bio-PEPA Models*, in: *Proc. of CMSB'10* (2010), pp. 23–32.
URL http://doi.acm.org/10.1145/1839764.1839769

[10] Clark, A., J. Hillston, S. Gilmore and P. Kemper, *Verification and testing of biological models*, in: *Winter Simulation Conference* (2010), pp. 620–630.

[11] Duguid, A., S. Gilmore, M. L. Guerriero, J. Hillston and L. Loewe, *Design and development of software tools for Bio-PEPA*, in: *Proc. of WSC'09* (2009), pp. 956–967.
URL http://www.informs-sim.org/wsc09papers/091.pdf

[12] Farkas, J., *Theorie der einfachen ungleichungen*, Journal für die Reine und Angewandte Mathematik **124** (1902), pp. 1–27.

[13] Grafahrend-Belau, E., 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** (2008).

[14] Heiner, M., *Understanding Network Behavior by Structured Representations of Transition Invariants*, in: *Algorithmic Bioprocesses*, Natural Computing Series (2009), pp. 367–389.

[15] Heiner, M., D. Gilbert and R. Donaldson, *Petri Nets for Systems and Synthetic Biology*, in: *SFM'08*, LNCS **5016**, Springer, 2008 pp. 215–264.

[16] Hucka, M., S. Hoops, S. Keating, N. Le Novère, S. Sahle and D. Wilkinson, *Systems Biology Markup Language (SBML) Level 2: Structures and facilities for model definitions*, Available from Nature Precedings (2008), ⟨http://dx.doi.org/10.1038/npre.2008.2715.1⟩.

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

[18] Lennox, J. A. and Z. Yuan, *An approach to verifying and debugging simulation models governed by ordinary differential equations: Part 2. residuals analysis and a case study*, International Journal for Numerical Methods in Engineering **57** (2003), pp. 707–722.

[19] Martinez, J. 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* (1982), pp. 301–310.

[20] Yuan, Z., M. L. Graham and J. A. Lennox, *An approach to verifying and debugging simulation models governed by ordinary differential equations: Part 1. methodology for residual generation*, International Journal for Numerical Methods in Engineering **57** (2003), pp. 685–706.