OXFORD

Briefings in Bioinformatics, 2017, 1–14

doi: 10.1093/bib/bbx096
Paper

Towards dynamic genome-scale models

David Gilbert, Monika Heiner, Yasoda Jayaweera and Christian Rohr

Corresponding author. David Gilbert, Department of Computer Science, Brunel University London, Uxbridge, UB8 3PH, United Kingdom.
E-mail: david.gilbert@brunel.ac.uk

Abstract

The analysis of the dynamic behaviour of genome-scale models of metabolism (GEMs) currently presents considerable challenges because of the difficulties of simulating such large and complex networks. Bacterial GEMs can comprise about 5000 reactions and metabolites, and encode a huge variety of growth conditions; such models cannot be used without sophisticated tool support. This article is intended to aid modellers, both specialist and non-specialist in computerized methods, to identify and apply a suitable combination of tools for the dynamic behaviour analysis of large-scale metabolic designs. We describe a methodology and related workflow based on publicly available tools to profile and analyse whole-genome-scale biochemical models. We use an efficient approximative stochastic simulation method to overcome problems associated with the dynamic simulation of GEMs. In addition, we apply simulative model checking using temporal logic property libraries, clustering and data analysis, over time series of reaction rates and metabolite concentrations. We extend this to consider the evolution of reaction-oriented properties of subnets over time, including dead subnets and functional subsystems. This enables the generation of abstract views of the behaviour of these models, which can be large—up to whole genome in size—and therefore impractical to analyse informally by eye. We demonstrate our methodology by applying it to a reduced model of the whole-genome metabolism of *Escherichia coli* K-12 under different growth conditions. The overall context of our work is in the area of model-based design methods for metabolic engineering and synthetic biology.

Key words: whole-genome-scale metabolic models; formal analysis; scalability; approximative stochastic simulation; model checking; reaction profiling; clustering; data analytics; delta leaping; subsystems behaviour; model-based design

Introduction

Currently, models of biochemical networks which can be simulated dynamically are limited in size; for instance, dynamic models for bacterial whole-genome metabolism are the exception [1, 2], with the lack of kinetic data usually given as the main reason. In addition, the dynamic simulation of such large and complex models is currently a bottleneck, presenting considerable difficulties both for stochastic and deterministic methods. These two difficulties together impede progress in the development of dynamic models of these large systems. Even when these models can be simulated to generate dynamic behaviour, there is a further challenge to analyse the large amount of data produced. Bacterial genome-scale models of metabolism (GEMs) can comprise about 5000 reactions and metabolites, and encode a huge variety of growth conditions; such models are far too large and complex in terms of their structure and number of observables (metabolite concentrations and reaction rates) to be checked without sophisticated tool support. Our approach is

David Gilbert is a Professor of Computing in the Department of Computer Science, Brunel University London, UK. His research interests include systems biology: modelling and analysis of biological systems; synthetic biology: computational design of novel biological systems.
Monika Heiner is a Professor of Computer Science in the Department of Computer Science, Brandenburg Technical University, Cottbus, Germany. Her research interests include modelling and analysis of technical as well as biological systems using qualitative and quantitative Petri nets, model checking and simulation techniques.
Yasoda Jayaweera is a PhD student in the Department of Computer Science, Brunel University London, UK. Her research interests are the application of data analytic techniques in systems and synthetic biology.
Christian Rohr has a PhD degree in simulative stochastic methods and is a Research Associate in the Department of Computer Science, Brandenburg Technical University, Cottbus, Germany. His research interests include the simulative analysis of stochastic Petri nets.
Submitted: 18 January 2017; **Received (in revised form):** 10 July 2017

intended to facilitate in the long term the development of dynamic whole-genome-scale models, by providing means to simulate and analyse these models. It can already be applied to explore properties, which are not dependent on specific kinetic parameters such as dead subnets, and furthermore be used for fine-tuning of kinetic parameters via optimization.

In this article, we describe a methodology and related workflow comprising methods and associated software tools to explore the dynamic behaviour of genome-scale metabolic models, and providing a guidance framework for modellers, either specialist or non-specialist in computerized methods. Our focus here is on analysis and exploration of already constructed models, rather than on the construction process itself. Typically, currently available large genome-scale models are designed to be analysed by constraint-based methods, such as flux balance analysis (FBA), flux variability analysis, etc., which explore steady-state behaviour [3]. We wish to complement these approaches by considering dynamic behaviour, which is required, for example, if the transient behaviour of microorganisms has to be described under changing external conditions, or when cell growth is crucial for metabolic engineering. Our methodology integrates the static analysis of graph properties and dynamic stochastic simulation, within a Petri net setting, exploiting a rich set of associated tools. This requires some model preparation before simulation followed by analysis of the generated time series traces. In this article, we use stochastic simulation based on δ-leaping [4], a recently introduced approach that permits the efficient simulation of these GEMs, enabling the observation of new and complex behaviours. Our approach is general, and not bound to Petri net representations.

We demonstrate the use of our techniques to explore model behaviours under different growth conditions, with a focus on the reaction perspective related to functionally meaningful subsystems (pathways). Other application scenarios include the comparison of model versions arrived at by model development, manual curation, variant exploration (knockout, etc.), production target-based optimization (e.g. within the framework of synthetic biology or metabolic engineering) across populations and over generations for evolutionary approaches. The intention is that our techniques will aid the exploration and understanding of large models, and comparison between model versions and configurations. These methods may also support the modification of such models as part of the design process in synthetic biology.

Novel techniques that we use include simulative model checking using libraries of properties and derived network variables (observers) for reaction behaviour, as well as analysis of the enlarging/shrinking of property-induced subnets over time with respect to functional subsystem or network location, possibly connected with structural network properties. Our methodology involves behaviour-based network decomposition using clustering and model checking, and abstraction over the data based on functional grouping. We illustrate our methods by considering metabolic models based on the whole genome of *Escherichia coli* K-12, and use a reduced version as running example.

Outline

In the following section, we introduce the kind of models, which we consider and our running example. Next, we give an overview of our workflow and a detailed presentation of our core methods. We conclude with a brief summary and outlook on future work.

Materials

Models. The models that we consider in this article comprise networks of biochemical reactions, which are often exchanged via the Systems Biology Markup Language (SBML) [5]. This article builds on experience gained while working with a set of 55 public domain models of whole-genome metabolism [6] of various *E. coli* and *Shigella* strains available in SBML [7]. We use a subset of the information contained in these models comprising:

- compartments, e.g. cytosol, periplasm and extracellular space;
- metabolites (species), given by their names, initialization values (set to zero), compartment membership and if they act as inputs or outputs called boundary conditions; and
- reactions, given by their names, substrates, products and related stoichiometry, subsystem membership, reversibility information and flux bounds. Kinetic information is not included in the original GEMs.

These public domain models incorporate the ability to represent different growth conditions by use of the boundary conditions. Individual boundary conditions are switched on or off in a Boolean manner by allowing/disabling inflow of particular metabolites. This is achieved by modifying the exchange reactions, which model the transport between the external environment and internal compartments by setting flux values to maximum or zero, corresponding closely to the laboratory experimental protocols used. We replicate this in our approach, although of course we could modulate rates in a continuous manner.

The growth conditions include a minimal growth medium based on M9 [8] comprising 25 metabolites, plus a set of additional possible nutritional sources whose exchange reactions are deactivated by default.

These public domain models also provide subsystem information. A subsystem is uniquely defined by a set of functionally related reactions, which belong uniquely to one subsystem only, whereas metabolites may be shared between subsystems, acting as communication channels between them, or they may be uniquely members of only one subsystem. A biochemical pathway is a subsystem with a recognized biological function, for example glycolysis or the citric acid cycle. There is no SBML tag to indicate the membership of reactions to subsystems, but this information has been added by the original modellers using a 'notes' annotation, which we assume to be correct.

Biochemical reaction networks can be considered as classical bipartite graphs, with the two distinct types of nodes representing reactions and metabolites. Hence, they can be immediately encoded as Petri nets, and thus their analysis can benefit from the rich set of Petri net techniques and tools, not considered further in this article, covering both qualitative and (stochastic and deterministic) quantitative aspects. We deploy standard terminology of Petri net theory; see [9] for a general introduction, or the Supplementary Data for this article. Systems biology and Petri nets use related terms, e.g. reactions/transitions and metabolites/places, which we use interchangeably; see Table 1 for a quick reference, and Figure 1 for an introductory example. Small networks of this kind can then be composed together to form arbitrary complex networks.

Converting an SBML model into a Petri net is done with the Petri net editor and simulator Snoopy [10], and involves two adjustments.

- As required for any discrete modelling approach, reversible reactions are modelled by two opposite transitions representing the two directions a reversible reaction can occur.

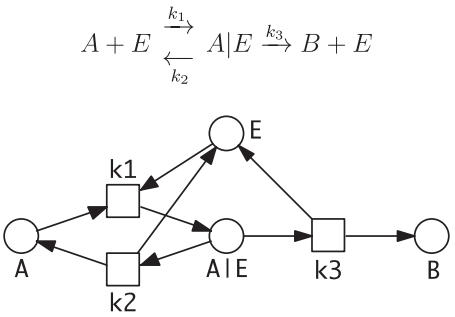

Figure 1. Petri net building block for mass action equation describing basic enzymatic reaction.

- Metabolites, which have been declared as boundary conditions, are associated with additional source and sink transitions called boundary transitions (Figure 2). This transforms a place-bordered net into a transition-bordered net, if all boundary places (i.e. source/sink places) have been declared as boundary conditions.

A boundary condition is a metabolite whose concentration is assumed to be constant despite any production or consumption by the network by assuming appropriate in/outflow from/to the environment. To transform a network model generated for constraint-based computation (exploring steady-state behaviour) into a general dynamic model that permits transient behaviour, boundary conditions can be basically treated in two ways. In the first approach, the boundary conditions are kept constant by the simulator. This is achieved for deterministic simulation by not generating ordinary differential equations for the boundary metabolites. In contrast in the stochastic case, because the simulator operates directly on the network, these conditions have to be flagged to be specially treated during simulation. The second approach is to generate explicit in/outflow transitions for each boundary metabolite, which maps the assumption underlying constraint-based approaches into a corresponding net structure, making the transformed network available for analysis and simulation by general purpose algorithms that do not need to be aware of boundary conditions. We prefer to use the second approach because we wish to be able to apply either continuous or stochastic simulation as appropriate.

Models can be interchangeably represented using SBML or Petri nets. We distinguish between strain-specific models, which may be closely related in their metabolic core while differing otherwise; model configurations—modelling the effect of different growth conditions on an individual strain; model variants—describing the effect of genetic modifications, which can act as designs in synthetic biology. In addition, model versions are created by the correction of modelling errors.

For all such models, the overall dynamic behaviour is expected to be a flow of metabolites through the network between the boundary conditions. Regarding metabolic networks, we are interested in maintaining flow through the network, and ensuring that all reactions and metabolites are sometimes active under the conditions for maximal growth that the model encodes.

A typical bacterium widely used both in modelling and experimentation is the K-12 strain of *E. coli*, which has >4000 genes, of which some 1400 are involved in metabolism [6]. When taking into account the compartmental structure of the organism (periplasm; cytosol; extracellular space), this results in a model comprising around 3000 reactions yielding about 4000 Petri net transitions, and the about 1200 unique metabolites yield about 2300 metabolites (Petri net places) respecting the compartmental structure. Although we can simulate and analyse models of this size, we have chosen to use a smaller example for illustration purposes in this article because it is easier for the reader to reproduce the results. See 'Discussion and conclusions' section for a discussion of how our techniques scale up to these large models.

Table 1. Analogies Petri nets—systems biology

Petri nets	Systems biology
Place	Metabolite
Transition	Reaction
Arc weight	Stoichiometry
Tokens	Mass
Token number	Concentration[a]
Marking	State
(Firing) rate	Flux
Incidence matrix	Stoichiometric matrix[b]
P-invariant	Mass conserving subnet
Minimal T-invariant	Elementary mode, extreme pathway[b]

[a]Up to obvious normalization.
[b]Up to reversible reactions.

Running example. We use a reduced version of the whole-genome metabolism of *E. coli* K-12 as a running example, which has been developed by Orth [11] to illustrate the basic structure of metabolic networks and their use in metabolic engineering (Figure 2). The reduction was originally done by hand [12] based on an early version of a GEM for *E. coli* K-12 [13] and subsequently used for comparison with the results of an automated procedure [14]. This model contains 94 reactions of which 46 are reversible, divided into 10 subsystems, and 92 metabolites of which 20 are boundary conditions; the corresponding Petri net model comprises 180 transitions ($94 + 46 + 2 \cdot 20$ boundary transitions) and 92 places.

The reduced model also contains a subset of the growth conditions of the full model, seven of which comprise the cut-down version of M9 used in the reduced model and are all activated by default, including oxygen resulting in an aerobic environment (CO_2, $H+$, H_2O, D-glucose, ammonium, phosphate, O_2). We considered the model under different growth conditions, and for simplicity, we report here two cases: the default seven minimal aerobic growth conditions (min-growth), and a version (enhanced-growth), which additionally included four deactivated sources that we selected to turn on (L-malate; L-glutamine; fumarate; D-fructose). Our expectation was that more metabolic reactions in the enhanced-growth model would be active compared with the min-growth model based on the assumption that there are pathways that are associated with specific nutritional sources.

Methods

In this section, we describe the elements of our workflow, which starting from an SBML model enables behaviour analysis over dynamic simulation traces; an overview is given in Figure 3.

Model preparation

There are several steps required in this stage, including three which involve graph-based static analysis (3, 4 and 5).

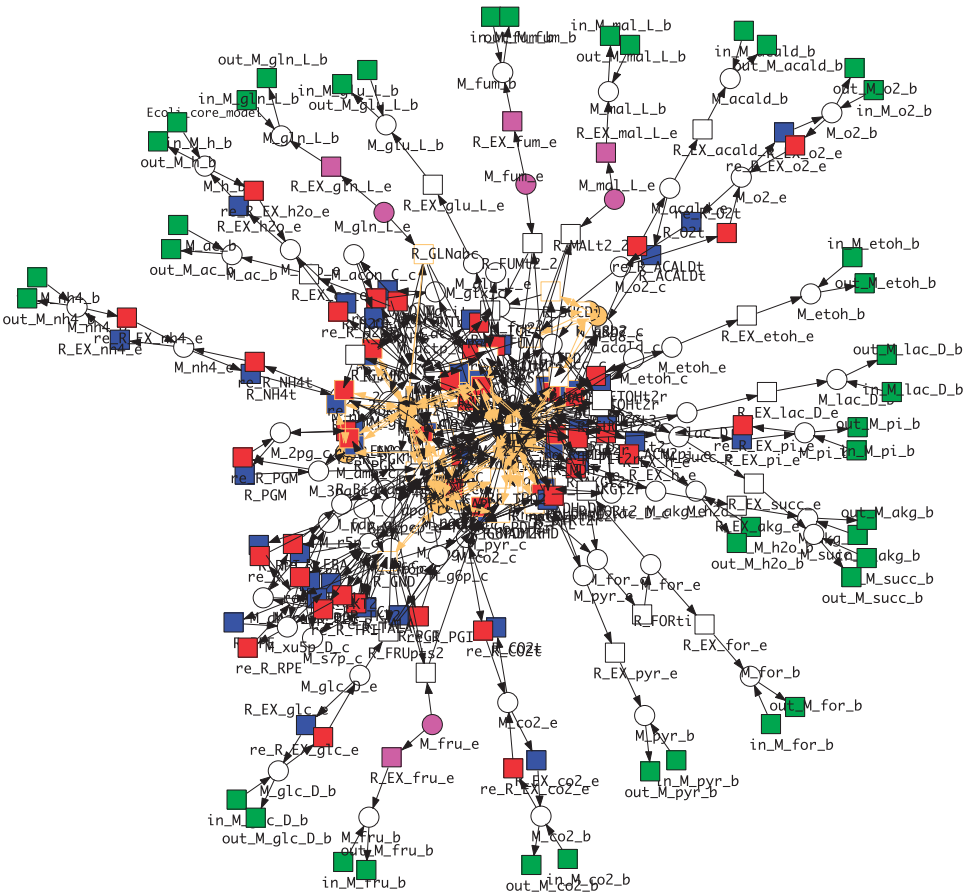

Figure 2. Petri net representation of reduced *E. coli* K-12 GEM according to [12, 14]; layout generated with Snoopy [10]. Colour code: green: generated boundary transitions, blue: reversible reactions, red: generated reverse direction for reversible reactions, yellow: P-invariants, pink: source places corresponding to the extra growth conditions, which we switch on in enhanced-growth.

(1) Rectifying typos: Assuming that models pass a standard SBML check (sbml.org/validator), there may be errors such as the incorrect naming of metabolites, usually by mixing them up, for example by confusing metabolites with similar formulae (e.g. H for H_2O). Although these are hard to detect, there are published approaches to achieve this, for example [20], and these are not dealt with further in this article.

(2) Converting SBML into internal format: We first need to ensure that the information regarding reversibility and boundary conditions conforms to the SBML standard regarding the use of the appropriate tags, which then allows the use of a standard SBML converter to the required internal format.

(3) Identifying the main connected subgraph and deleting all others: Models may contain disconnected subgraphs, possibly because of gaps in knowledge. Disconnected subgraphs cannot influence each other, and thus can be considered separately. In the case that the model contains one dominating large component, we remove any trivial small isolated subgraphs before simulation. It is the case that structurally disconnected subnets may be dynamically connected using for instance modifiers in the context of kinetics. However, we make use of only those parts of SBML, which typically occur in constraint-based models; these do not contain initialization of the metabolites or kinetic information. Thus, the situation that there are disconnected subnets, which are dynamically connected, never occurs in these models, which we consider. In general, disconnected components can be an indication of gaps in the model; gap filling is a large area in its own right; see [3] and references therein, and is beyond the scope of this article.

(4) Rectifying source and sink places: That is, boundary places, which have not been declared as boundary conditions, can be because of modelling gaps or errors, precluding those source places associated with growth conditions. These contradict the expected basic behaviour of sustained flow through the network under steady-state conditions. The ideal solution is to fill the gaps; otherwise, possible solutions are to declare them as boundary conditions, to delete them optionally with their associated reactions or to reverse one of their immediately neighbouring reactions.

Constraint-based methods implicitly ignore reactions, which can never contribute to steady-state behaviour. We take a more conservative approach, so that we do not generate a network specialized to one specific growth condition.

(5) Initializing metabolite concentrations: For dynamic simulation, we need to find appropriate initialization values for all metabolites. In the case of metabolic networks, all initialization values for metabolite concentrations are typically set to zero because the publicly available GEMs have been developed for FBA analysis, which does not require them. Options are to assign uniform values, impose some ratio over metabolite categories, e.g. [2], or extract specific values from the literature. Either all metabolites can be initialized to some non-zero value, or just the minimal set identified by P-invariant analysis (mass conservation).

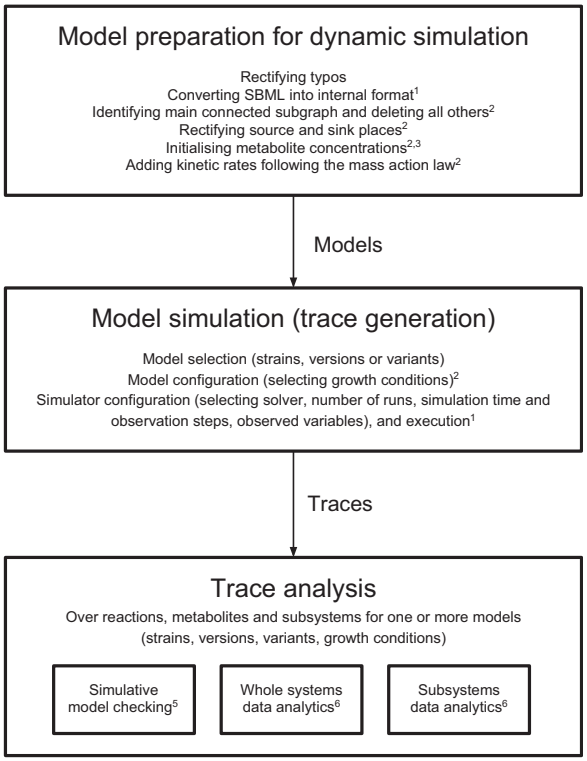

Figure 3. Workflow, exploiting the following tools: [1]Snoopy [10], [2]graph editing routines in GNU Prolog [15], [3]Charlie [16], [4]Marcie [17], [5]MC2 [18] and [6]R package [19].

A P-invariant defines a set of places, which induce a subnet conserving metabolic mass, i.e. the total mass of the metabolites in the subnet is constant. The (minimal) P-invariants of a network are all those (minimal) sets of metabolites that contribute to the mass conservation of that network. Owing to this conservation behaviour, P-invariants need to be initialized with non-zero mass for dynamic simulation; otherwise, all reactions with a substrate belonging to a mass-conserving subnet would never be able to occur. The P-invariants provide the minimal set of initialized metabolites required to obtain activity in the entire network, and we use it in this work because this makes it easier to distinguish between active and never-active subnets. Initializing all metabolites, which is the common practice, makes liveness analysis more cumbersome because non-activity will take longer to emerge.

(6) Adding kinetic rates: All reactions need to have an associated kinetic rate to permit dynamic simulation. Kinetic rates are typically state dependent and are described by kinetic laws, the most basic of which is mass action. The mass action law is defined by the mathematical product of the concentrations of the substrates of a reaction and a reaction-specific kinetic rate constant (also called a kinetic parameter). Note that kinetic rates and kinetic rate constants are related, but different notions; the former generally vary over time, while the latter are always constant. Other laws are approximative kinetic abstractions, Michaelis–Menten [21] or linlog [22], which however are in general not suitable for faithful stochastic simulation [23]. Thus, we elect to use the basic mass-action law here as a universal illustration. Because FBA analysis does not require these kinetic rates, the kinetic laws and associated rate constants are typically not specified in GEMs. In this case, we assign mass action laws to reactions, and some default rate constants. Options are to set rate constants as arbitrary uniform values (e.g. 1), or uniform for different categories (transport, etc.), or via literature or based on FBA values (however, there will be different steady-state rates under different target conditions). This implies a constant rate for inflow boundary conditions (as they do not have any substrate).

Running example. We encountered one typo, which we corrected, involving the reactions R_EX_h2o_e and R_EX_h_e, where the metabolites for water and hydrogen as products were incorrectly exchanged. Next, we added explicit SBML tags for the boundary conditions, identified by the metabolite naming convention, and adjusted the reversibility tag to be consistent with flux bounds. There is only one connected subgraph and no unintentional source or sink places.

There are five P-invariants involving 12 metabolites, computed with Charlie [16] (Supplementary Data B). For simplicity, we initialized all metabolites belonging to a P-invariant with the same value.

In the same way as for the large models, the running example did not contain reaction rate information, and we added mass-action kinetics with parameter 1 by default to all reactions. We used this model to test our techniques for behaviour analysis, which we report below; it will be the subject of future work to obtain differential rates by curation or optimization. A particular issue of interest in this respect is the evolution of behavioural properties of networks and their constituent subnets, with respect to reactions as well as metabolites over time.

Scalability. The only step in the preparation phase, which can be time-consuming, is the computation of the P-invariants because it is known that in the worst case, there can be exponentially many in terms of the net size. However, in practice, for these networks, the time required is manageable: computation of P-invariants for the running example on a standard desktop computer requires <1 min, and 15 min for the full-size *E. coli* K-12 GEM to detect 17 P-invariants involving 39 metabolites.

Model simulation

GEMs typically have an infinite state space, which precludes the use of exact analysis methods that build on an exhaustive description of the state space [24]. An obvious choice is thus dynamic simulation, i.e. the generation of a representative (finite) set of finite traces through the infinite state space.

Simulation efficiency. These large systems are highly stiff in nature, which causes severe numerical problems for continuous simulation of the set of ordinary differential equations induced by a biochemical reaction network [2] and unacceptably long runtimes for stochastic simulation algorithms (SSAs). For example, Gillespie's direct method [25] of the GEM for *E. coli* K-12 takes in the order of 90 min for 1 run of 1000 time points, or 62.5 days for 1000 runs on a single-core workstation; these figures increase by about 50% for τ-leaping SSA [26].

Discrete-time δ-leaping [4] is a method that can be efficiently applied even to large models, typically taking <1 s for 1 run or close to 14 min for 1000 runs for a GEM. It converts the underlying continuous-time Markov chain into an equivalent discrete-time Markov chain and improves the efficiency via discrete-time leaps, even though this results in an approximate simulation method. In SSA, the firing frequency depends solely on the rates, while in δ-leaping, it is a binomially distributed random variable. This means that for SSA, reactions with lower rates occur less frequently than reactions with higher rates; reactions with low rates (rare events) occur very infrequently, and are thus hard to

observe. In principle, this holds for δ-leaping too, but δ-leaping is much less sensitive to large differences in reaction rates (stiff models) in terms of runtime, so it is able to perform more simulation runs, and hence report more observations, than SSA in the same execution time.

The discrete-time leap method is able to reproduce the dynamic behaviour (including the occurrence of switches, oscillations and tipping points) of a stochastic model comparable with the Gillespie algorithm as long as the following condition is fulfilled for all transitions t:

$$0 \leq \frac{h_t(m)}{ed_t(m)} \cdot \delta \leq 1,$$

where $h_t(m)$ stands for the transition's rate and the $ed_t(m)$ is its enabledness degree. A violation of the condition in the above equation would not lead to negative values or incorrect markings (states), i.e. in all cases, a marking is reached that is part of the model's state space. However, the temporal behaviour of the model, simulated with δ-leaping, would not coincide anymore with the behaviour of exact SSAs. This may have two possible causes. First, the model's timescale is smaller than the chosen δ, i.e. reducing the δ would gain correct results. Secondly, some transitions' rate functions are not scaled correctly, i.e. stochastic reaction rates have to be scaled with respect to their reaction order [27].

The discrete-time leap method is supported by Snoopy and Marcie [17], both of which are publicly available at http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Software.

Types of traces. Traces are time series reporting the current variable values at the $n+1$ time points τ_i of a specified output grid $i=0,\ldots n$, typically splitting the simulation time into n equally sized time intervals. We consider two types of traces:

- Traces of metabolite concentrations, i.e. time series of the current concentration of the metabolites at the specified time points of the simulation run:
  $s(\tau_i) : P \to N$, with $s(\tau_0) = m_0$, that is, $s(\tau_i)$ are place vectors over all metabolite concentrations, indexed by the set of places P.
- Traces of reaction rates, i.e. time series of the number of occurrences, which each individual reaction had in total in the latest time interval,
  $v(\tau_i) : T \to N$, with $v(\tau_0) = 0$, that is, $v(\tau_i)$ are transition vectors over all reaction rates indexed by the set of transitions T.

Reading a reaction rate vector as Parikh vector immediately leads us to the state equation specifying the relation between both traces:

$$s(\tau_i) = s(\tau_{i-1}) + C \cdot v(\tau_i), i = 1 \ldots n,$$

where C is the incidence matrix of a Petri net (Supplementary Data A).

Thus, the metabolite trace can be derived from the rate trace, but generally not vice versa. In the stochastic setting, the reaction trace cannot be uniquely deduced from the metabolite trace because of alternative and parallel reactions, which specifically holds for individual traces. Therefore, we directly record the reaction traces during simulation. Often, we consider averaged traces to reduce stochastic noise, even though the average of a set of stochastic traces is itself stochastic (except in the case of that the number of traces is infinite). Thus, the individual values at each time point are non-negative real numbers instead of natural numbers. When model checking, rare events are more obvious in an averaged trace than in single traces.

Meta model. To facilitate simulating a model under various conditions, we have implemented a meta model, which takes advantage of in-built parameter selection in the simulation engines that we use—Marcie and Snoopy. This enables us to use one model, which can be configured for simulation under different conditions, rather than a set of model variants, one per condition (e.g. aerobic/non-aerobic, min-growth/max-growth, typos-fixed/typos-not-fixed). The meta model approach eliminates the danger of typographical errors, which can creep in when variants of a model are created. See Supplementary Data C for more details.

Running example. The analysis techniques discussed in the following were applied to δ-leaping simulation traces averaged over 1000 runs for 1000 time points to reduce the effects of biological noise, but they could also be applied to any kind of stochastic or continuous traces. The window of 1000 time steps has been determined pragmatically, based on a few initial exploratory simulations; there will always be a need to make a decision about the length of the simulation. We consider the two versions of the running example, min-growth and enhanced-growth, focusing on the differences in their behaviours.

Scalability. We use δ-leaping on the running example because of its scalability up to (unreduced) GEMs. The size of the generated average trace file from simulating the GEM of *E. coli* K-12 is 19 MB for reactions and 10 MB for metabolites, compared with 600 and 300 KB, respectively, for the running example. These traces are used in the model checking step for analysing transient behaviour. Simulation times on a standard desktop computer (four cores and eight threads) were 7 s for the running example (1000 runs, 1000 time points and observations, initialized with 100 tokens), and 184 s with the same conditions for the *E. coli* K-12 GEM.

Simulative model checking

Basic principles. Model checking permits us to determine if a model fulfils given properties specified in temporal logics, e.g. probabilistic linear-time logic, PLTL [18]; see examples below. In this research, we have used simulative model checking over time series traces of metabolite and reaction behaviours. In principle, this could be done over individual runs generated by a stochastic model, yielding probabilistic results, or over one trace averaging the individual runs—although this trace is still stochastic, model checking it will return a Boolean result instead of a probability. In this article, we use the latter approach, which technically belongs to a non-probabilistic subset of PLTL. For consistency with the model checker MC2 [18] that we used, we give the properties in PLTL format, with the results belonging to the set {0, 1} rather than in being in the range [0–1].

Model checking generally requires that the properties of interest have to be known, often motivated by observations in the wet lab, e.g. *The concentration of metabolite A is always above a certain threshold, let us say 10:*

$$P_{\geq 1}[\, G(A > 10)\,].$$

If one is not so sure about an appropriate threshold, one could use the established concept of a free variable $x [28, 29], i.e. which ranges over all possible values, to determine the probability distribution of the threshold, so that A fulfils the property:

$$P_{\geq 1}[\, G(A > \$x)\,].$$

In our setting, we do not know (yet) the PLTL properties certain observables (metabolites, reactions) are supposed to

exhibit, which brings us to a new scenario, where we wish to ask: *Which variables fulfil a given property pattern?*, expressed as:

$$P_{\geq 1}[\,G(\ll y \gg > 10)\,],$$

with «*y*» being a meta variable ranging over all entities (metabolites, reactions) in the model.

Properties of interest. We assume that there are generally desired behavioural characteristics. Liveness is a well-established notion for reactions (transitions) in Petri net theory. A reaction is live if for any point in time it exhibits future activity, i.e. it occurs. We extend this notion to metabolites (places): a metabolite is live if for any point in time at least one of the reactions involving the metabolite (as substrate or product) is live. A reaction network is reaction-live if all reactions in the network are live. Likewise, a network is metabolite-live if all metabolites in the network are live. If a network is reaction-live, then it is metabolite-live, however, generally not *vice versa*. Thus, metabolite-liveness is less strict than reaction-liveness for networks. A reaction is forever dead from a point in time after which it never exhibits activity. A metabolite is forever dead from a point in time after which all reactions involving the metabolite are dead.

The notions of liveness (respectively, deadness) above are defined over infinite traces. In our case, because we are using simulation generating finite traces, we need corresponding notions over time windows, and this is what we call (reaction or metabolite) activity. A reaction is dead over a period of time from t_1 to t_2 if it does not exhibit any activity from t_1 to t_2. Similarly, a reaction is active from t_1 to t_2 if it exhibits activity at some point between t_1 and t_2. A metabolite is dead over a period of time from t_1 to t_2 if all of the reactions involving the metabolite are dead in the given time window, and a metabolite is active over a period of time from t_1 to t_2 if at least one of the reactions involving the metabolite is active in the time window.

Property libraries. A library of appropriate property patterns now allows us to categorize all observables into (not necessarily disjunctive) sets fulfilling the individual property patterns. We compiled two such libraries of properties for reaction and metabolite behaviour, which are provided in Supplementary Data D accompanied by descriptions in natural language. The properties were derived from our extensive experience in model checking such models. We have previously used an automated approach using machine learning over sets of examples [30]. In the current case, the properties are so general that an automated approach is not fruitful, as it relies on the selection strategy for the example set. Of course, the current property libraries could be enhanced by automatically derived properties, or hand-crafted properties specific to the set of models under consideration.

One desired behavioural characteristic that we are interested in is that under conditions for maximal growth, all reactions and metabolites are active over the period of the trace. In the following, we give two examples from the reaction library, which define dead behaviour. Note the use of the meta variable «*x*», which ranges over all reactions.

(1) Never active reactions, i.e. always dead reactions:
$P_{\geq 1}[\,G(\ll x \gg = 0)\,]$,
which is equivalent to $P_{\geq 1}[\,\neg F(\ll x \gg \neq 0)\,]$.

(2) Reactions with changing activity and finally a steady state of zero activity (d— differential operator):
$P_{\geq 1}[\,F(d(\ll x \gg) \neq 0) \wedge F(G(\ll x \gg = 0 \wedge d(\ll x \gg) = 0))\,]$.

Running example. Model checking the min-growth and enhanced-growth models, we found that the number of reactions fulfilling property (1) reduced from 13 to 0, and for property (2) from 94 to 4, confirming our earlier expectation that more metabolic reactions in the enhanced-growth model would be active compared with the min-growth model. A closer analysis revealed that these four are rare events. In the 'Whole system data analytics' section, we introduce time blocks to better distinguish between rare and zero events.

Subnets. We distinguish two categories: property subnets are defined by sets of entities sharing a certain temporal logical property, the composition of which can vary over time, unlike functional subnets (subsystems), which are statically defined by sets of reactions contributing to the same biological function.

In this article, we focus on a specific class of property subnets, called dead subnets, which exhibit no activity from the current time point onwards. The existence of such subnets can be an indication of a modelling fault, missing information in the network structure (e.g. gaps because of unidentified genes), or unused parts of the network because of the set of environment conditions imposed (e.g. the growth conditions).

There are two classes of dead subnets: reaction dead subnets and metabolite dead subnets. A reaction dead subnet over a period of time from t_1 to t_2 is induced by reactions (transitions), which are dead over that period of time; it includes those reactions and their substrates and products. At least one of the substrates has to be dead, but not all the substrates and products are necessarily dead because they can be involved in alternative pathways. A metabolite dead subnet over a period of time from t_1 to t_2 is induced by metabolites (places), which are dead over that period of time; all reactions involving the metabolites are dead in that time window.

A metabolite that is always from some point in time in a steady state (a constant concentration of zero or above) can be so because either (i) its production matches its consumption, and these rates are non-zero, or (ii) it is neither produced nor consumed. We are interested in the latter category, as they induce dead subnets. A reaction that is in a steady state can be so because either (i) it has a steady non-zero activity, and is 'ticking over', or (ii) it is non-active, i.e. with zero activity. We are particularly interested in the latter category because reactions with zero activity belong to dead subnets, and we can directly monitor/observe reaction activity over time. This justifies why we are looking at reaction dead subnets because otherwise we could not distinguish between the two cases of zero and non-zero activity.

Running example. Figure 4 shows the development of the active subnet over time for the min-growth and enhanced-growth versions. Note that in the min-growth model, the active subnets initially increase in extent but then decrease, whereas in the enhanced-growth model, the active subnets increase over time until they cover the entire network, possibly suggesting that more than the minimal medium is required to maintain metabolic activity according to this model.

Scalability. The library comprises 53 properties for metabolites and 80 properties for reactions, 133 properties in total. The MC2 model checker [18] requires 7.5 s to process the reaction library, 1.6 s for the metabolites library for the running example (92 metabolites, 194 reactions) and 8 s (both metabolites and reactions) on a standard desktop computer. For the *E. coli* K-12 GEM (2133 metabolites, 4162 reactions), the time required is 2 m 3 s for reactions, 24 s for metabolites and 2 min 34 s for both libraries. Checking a model over all properties in both libraries is achieved using a script.

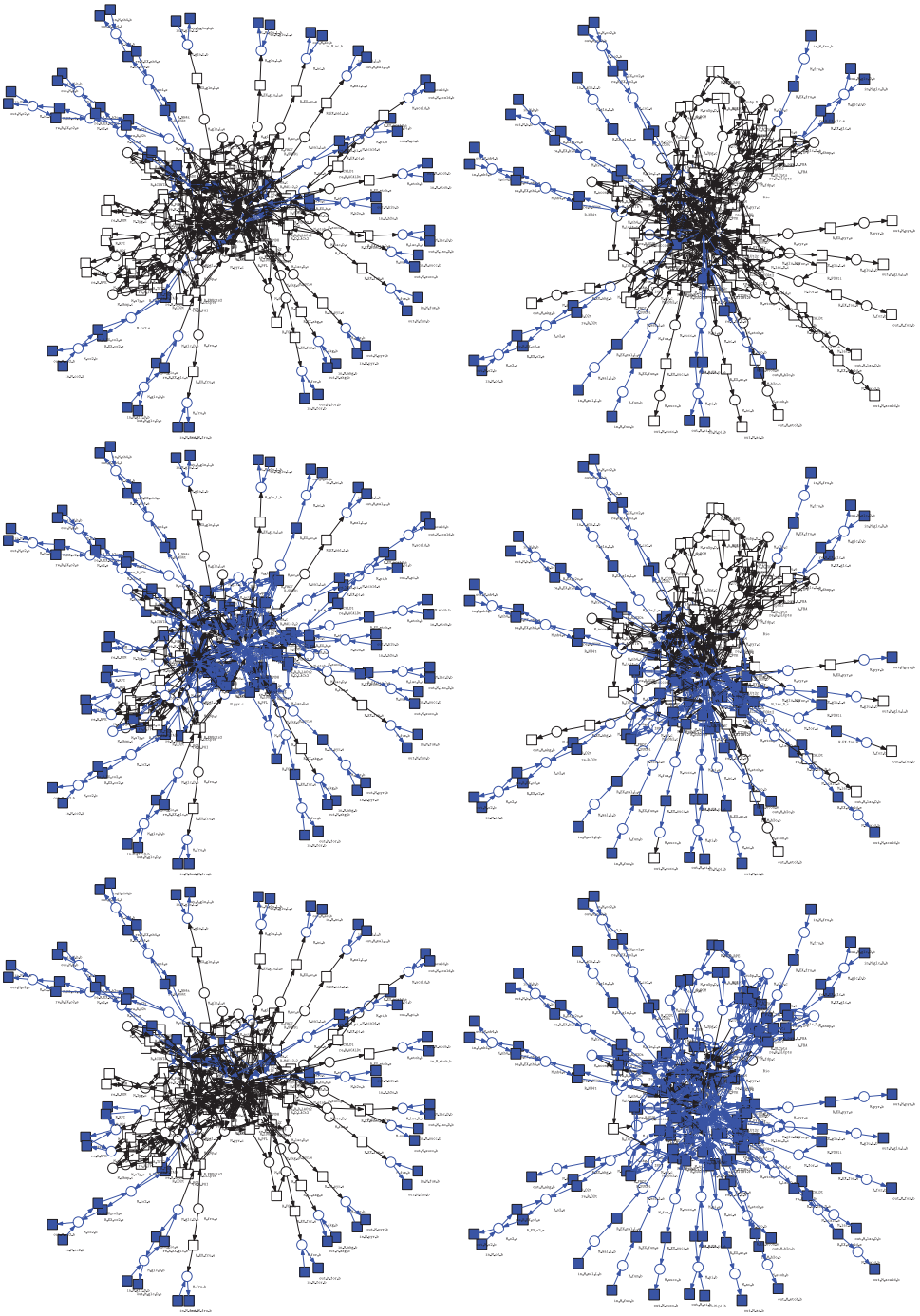

Figure 4. Dynamic simulation of the min-growth model (left column), and enhanced-growth model (right column); shown are snapshots at three time points—the beginning, middle and end of the simulation, with the active reactions highlighted in blue. Snoopy was used to both automatically generate layouts, and reactions coloured using activity over reaction traces identified in each time window. These results clearly show that the model predicts under enhanced-growth conditions that metabolic reaction activity increases over the time of simulation, while under min-growth conditions, the reaction activity decreases after an initial peak. This temporal aspect of the transient behaviour can only be observed using dynamic simulation.

In the following sections, we introduce methods to explore the evolution of dead subnets, including the use of observers, i.e. derived variables defined over the model variables, for example the total number of dead reactions.

Whole system data analytics

Data analytics is a well-established field of research; it can be applied to large data sets to identify trends, which may be buried under the huge amount of data. These techniques provide complementary insights, which are otherwise not obvious from the visual inspection or model checking over time series; this is because of the size and complexity of the models in terms of the number of reactions and metabolites.

Data analytics comprises a large number and huge variety of techniques. In the following, we discuss our general approach and a selection of techniques, which we found most applicable in our scenario, which are then applied to our running example.

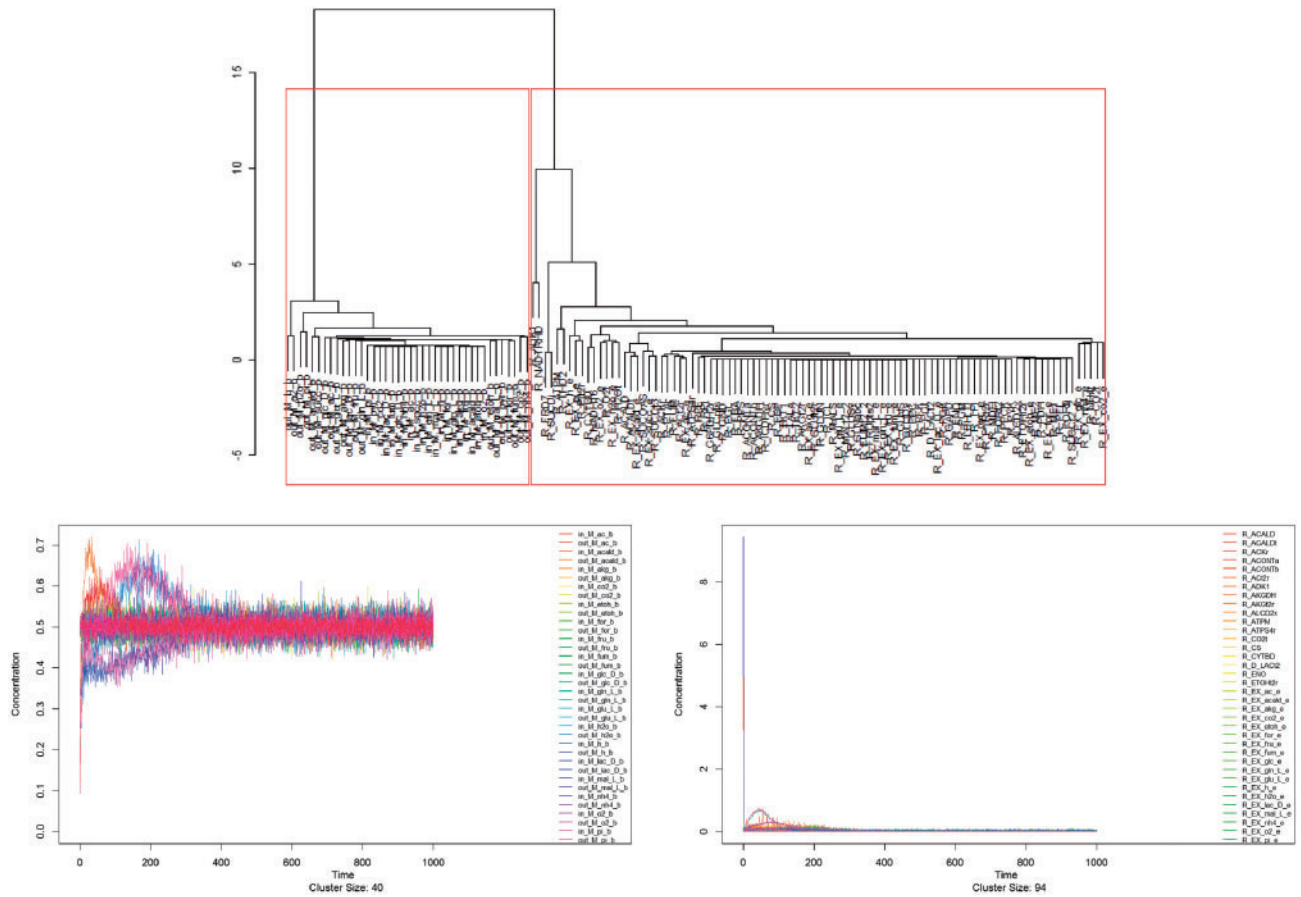

Figure 5. Hierarchical clustering using Euclidean distance as dissimilarity measure of the min-growth model based on reaction activity traces; the dendrogram shows two distinctive clusters of behaviour identified using the average silhouette width measure [33] in the clValid R package [19], which are illustrated in the consecutive figures. (Left) A cluster of 40 reactions comprising all boundary reactions plus some exchange and transport reactions, which maintain their activity throughout the simulation because they do not rely on the flow of metabolites through the network. (Right) A cluster of 94 reactions, which all reach a steady-state value of < 0.02 occurrences per time unit early in the simulation because of the minimal growth environment.

(1) Clustering is a learning technique, often unsupervised, for partitioning a set of entities, so that the entities in the same partition (cluster) are more similar to each other than to those in other partitions. This technique can be used to hierarchically cluster the reaction rate traces as well as metabolite traces, using, for example, Euclidean distance; see [31] for a survey of clustering techniques for time series data. Distances can be calculated over the raw data, or over derived values, e.g. derivatives, which obviously might yield different clustering results. We choose to illustrate clustering over raw stochastic data in this article; otherwise, smoothing would be required to compute meaningful derivatives.

(2) Time division blocks over traces and evolution of properties. To better investigate the evolution of properties over time, we can introduce a level of abstraction by dividing the simulation time into (equal) blocks. Each block can be seen as a mini time series, and treated accordingly, e.g. by model checking or clustering. Alternatively, values for the reaction and metabolite activity can be summed or averaged within each block. Analysis of these variables within the blocks can be achieved using a variety of visualization methods, e.g. scatter plots, density plots and bar charts. These are supported by the statistical package R [19]; the ggplot2 [32] package can be used to generate plots.

Running example. For the purpose of the following quantitive analyses, we have combined forward and reverse directions of reversible reactions because we wish to focus on the total metabolic flow carried out through the reactions in the network. For this, we have computed the absolute value of the difference between the individual rates of the two directions.

(1) Clustering. We used this technique to hierarchically cluster the reaction rate traces; see Figure 5 for the result for the min-growth model, and applied the average silhouette width measure [33] in the clValid R package [19] to identify the optimal partition of the data set.

This yielded two clusters: (i) one cluster of 40 reactions comprising all boundary reactions plus some exchange and transport reactions peaking at a maximum of 0.7 before reaching a steady state at 0.5. which maintain their activity throughout the simulation because they do not rely on the flow of metabolites through the network, and (ii) a larger cluster of 94 reactions, with a few members showing early high activity peaks, and then all showing low activity after 400 time units because of the minimal growth environment. Note that these clusters would not be obtained easily using model checking because the stochastic nature of the traces makes the detection of peaks and general tendencies (e.g. 'generally increasing') difficult. For reasons of space, we omit the same analysis for the enhanced-growth model but do treat both versions below.

(2) Time division blocks over traces and evolution of properties. For illustration, we take four blocks (i.e. quarters). Our

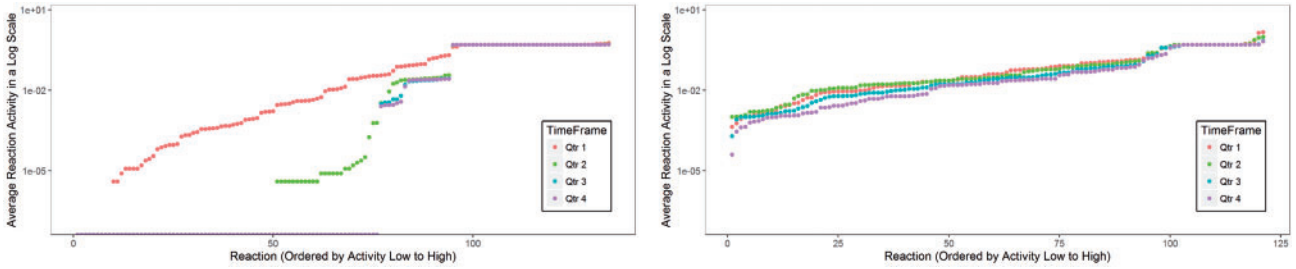

Figure 6. Scatter plots of progression of average reaction activity over time blocks (quarters). (Left) Min-growth model, (Right) enhanced-growth model; in the min-growth model, the number of dead reactions increases over the time blocks, whereas in the enhanced-growth model, there is less variation in activity in the quarters. This is in accordance with the observations in Figure 4.

analysis shows that in the min-growth model, the number of dead reactions increases over the time blocks, with by far the greatest increase between quarter 1 and quarter 2. In the enhanced-growth model, there is much less variation in activity in the quarters, with a slight overall higher activity in quarter 2. See the ordered average reaction activity of all reactions in the scatter plot in Figure 6. We also found that there was a peak at activity value 0.5 for all quarters for both model versions. These are primarily the boundary transitions, responsible for the inflow and outflow of the network; see the density plot in Figure 7 showing how often the values of average reaction activity occur in the four time intervals. The density of low activity reactions in the min-growth model is virtually identical in all four quarters; only the first two quarters exhibit any dead reactions in the enhanced-growth model as metabolism starts up, although in general there is a progressive shift towards both lower activity accompanied by mid-range activity as the system stabilizes. The bar charts in Figure 8 show how the number of reactions in each reaction category of zero, rare and non-rare varies between the time blocks.

Scalability. None of the algorithms involved causes any scalability issues.

Subsystem data analytics

Our general approach here comprises the following steps:

(1) Identifying subsystems using SBML annotations.

(2) Abstracting over reactions by representing subsystem behaviour by the averaged reaction activity of constituent reactions to give a set of derived variables.

(3) Clustering the subsystems by their average behaviour hierarchically using Euclidean distance and discrete wavelet transform. We have found that dynamic time warping [34] is too inefficient to be used in practice, not completing after 48 h on a standard workstation.

(4) Clustering the subsystems according to the degree of their structural inter-connectivity. For this, we defined a similarity metric for hierarchical clustering over two subsystems P and Q by

$$\left(\sum_{i=1}^{i=n} \frac{min(|X_i|, |Y_i|)}{|X_i \cup Y_i|}\right)/n,$$

where n is the number of connected subgraphs in the network $P \cdot Q$ induced by the union of their reactions, and for each such connected subgraph, X is the set of reactions in P and Y the set of reactions in Q, respectively. The basis of the measure is that two subsystems, which by their definition never overlap, can be connected by shared metabolites to form a connected subgraph induced by the union over their reactions. Note that in the case of two disconnected subsystems, the similarity metric yields zero because either $|X_i|$ or $|Y_i|$ is zero in each subgraph. Similarly, the highest possible value for similarity is 0.5 when two equal size subsystems are being considered.

(5) Pairwise comparing the clusterings by behaviour and structural inter-connectivity using two well-established measures: (i) the Fowlkes–Mallows (FM) index [35] over the dendrograms produced by hierarchical clustering, implemented in the dendextend package in *R* [36], (ii) the Mantel test [37] over the dissimilarity matrices, implemented in the ade4 package in *R* [38], producing a correlation and corresponding *P*-value indicating the significance of the results.

Running example. To obtain a functional view of model behaviour for our running example, we investigated the effect on subsystem behaviour of exposing the organism to enhanced-growth conditions. As before, in the simulation traces that we consider below, we have combined the rates of the forward and reverse directions of reversible reactions.

(1) Subsystems identified from the SBML model with number of reactions (ignoring reversibility) in brackets: citric acid cycle (8), glutamate metabolism (4), glycolysis/gluconeogenesis (12), oxidative phosphorylation (9), pentose phosphate pathway (8), pyruvate metabolism (6), anaplerotic reactions (6), inorganic ion transport and metabolism (2), exchange (20), and extracellular transport (19).

(2) Abstracting over reactions. The way in which the growth environment differentially affects subsystem behaviour is shown in Figures 1 and 2 in Supplementary Data E which plot the time series average activity of all reactions in a subsystem, for all subsystems under minimal and enhanced-growth conditions. In the case of all but one of the subsystems, the metabolic activity is greatly suppressed under min-growth compared with enhanced-growth conditions. The traces for inorganic ion transport and metabolism exhibit no effective difference between the two growth conditions, indicating that the two reactions involved (ammonia reversible transport and phosphate reversible transport via proton symport) make no contribution to the metabolic activity of the system according to the current model.

(3) Clustering by subsystem average behaviour. Both Euclidean distance and discrete wavelet transform [39] yielded the same clusters, suggesting a robustness in the analysis for these data; we present the Euclidean distance version in Figure 9 for the enhanced-growth model and in Supplementary Data E and Figure 3A for the min-growth model. Ignoring inorganic ion transport and metabolism because of the finding in the previous step, the result clearly shows that exchange and oxidative phosphorylation are outliers in both conditions. This

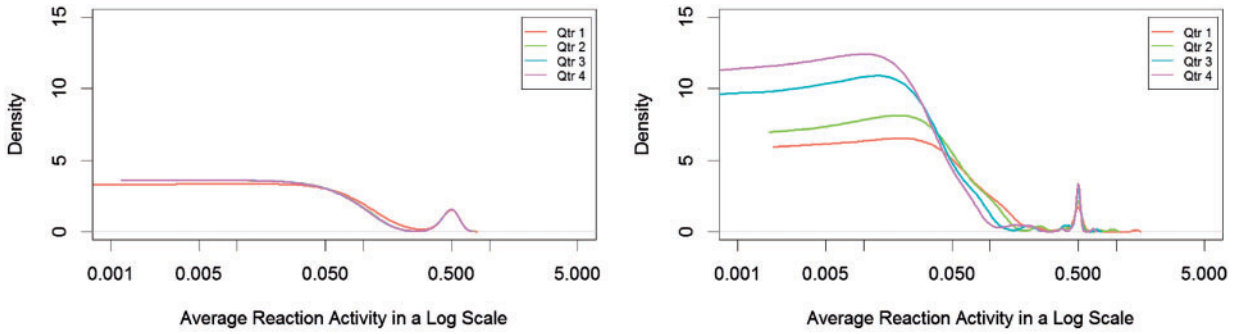

Figure 7. Variation in density of average reaction activity over time blocks (quarters). (Left) Min-growth model, (Right) enhanced-growth model. In both versions, as the block time progresses, the activity of reactions gradually decreases. Zero values are not displayed because of log plotting. The results for the min-growth model show that the density of low activity reactions is virtually identical in all four quarters. In contrast, under the enhanced-growth conditions, there is a progressive shift towards lower activity over the four quarters. However, because of the log scale on the X-axis, these low activity reactions do not dominate the overall activity. In both cases, the peaks at 0.5 correspond to the exchange and transport reactions; see Figure 5 (Left).

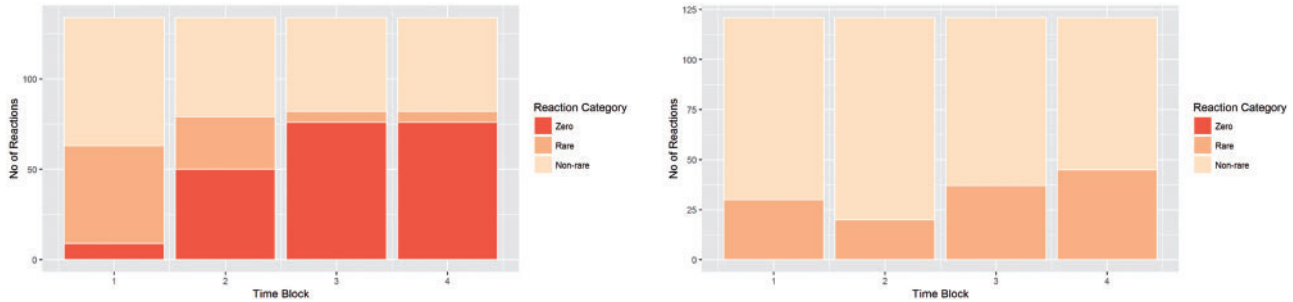

Figure 8. Comparison of variation in membership of reaction categories over time blocks (quarters). (Left) Min-growth model, (Right) enhanced-growth model. Reactions have been categorized into zero, rare and non-rare based on their average reaction activity (activity = 0; 0 < activity ≤ 0.01; activity > 0.01). The number of dead reactions in the min-growth model increases over the time blocks indicating the progression of deadness in the network, whereas the network remains alive in the enhanced-growth model.

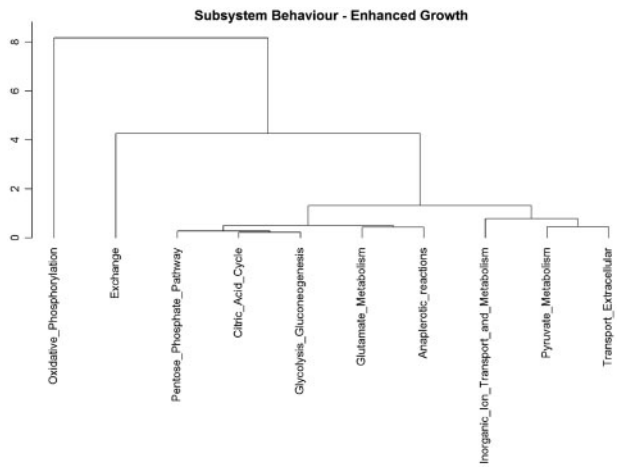

Figure 9. Hierarchical clustering of the subsystems in the enhanced-growth model based on averaged activity traces per subsystem using Euclidean distance. For more details, see Figure 3 in the Supplementary Data.

can be explained by the fact that oxidative phosphorylation generates ATP and thus powers, amongst other things, import/export systems. Also, the major pathways are grouped within one major cluster, together with extracellular transport.

(4) Clustering according to subsystem structural inter-connectivity. As there was no difference between the min-growth and enhanced-growth model in terms of reactions comprising the subsystems, we have only computed one structural diagram for subsystem structural inter-connectivity (Supplementary Data E, Figure 3C). The clusters clearly show that the core metabolic subsystems are closely interconnected in terms of metabolites, for example glutamate metabolism and the pentose phosphate pathway are a closely related pair. Additionally, the externally oriented reactions are clustered apart from the core metabolic subsystems, with exchange and extracellular transport more closely related than with inorganic ion transport and metabolism, which itself is in between the externally oriented reactions and the metabolic core.

(5) Pairwise comparison of the clusterings by behaviour (both min-growth and enhanced-growth model) and structural inter-connectivity. All three dendrograms are shown in Supplementary Data E, Figure 3. The FM values, Mantel correlation and Mantel *P*-value are shown in Table 2.

Comparison of the min-growth and enhanced-growth dendrograms resulted in the highest FM index value of 1, which indicates that there is significant evidence that the two trees are similar. This conclusion is supported by the results of the Mantel test because the observed correlation of 0.86 and *P*-value of 0.01 with an associated cut-off alpha of 0.05 suggest that the matrix entries have a strong positive linear association.

The low FM index values of 0.61 computed for behaviour against structure comparisons (min-growth – structure and enhanced-growth – structure) provide only weak evidence against the null hypothesis that the two trees are dissimilar. This is also supported by the results of the Mantel test, which

Table 2. Three pairwise comparisons of clusterings, using the FM index [35] and the Mantel test [37]

Clustering 1	k	Clustering 2	k	FM	Correlation	P-value	Relatedness
Min	3	Enhanced	3	1.00	0.86	0.01	Related
Min	2	Structure	2	0.61	0.02	0.32	Not related
Enhanced	2	Structure	2	0.61	0.03	0.33	Not related

Note: k, cut value; Correlation and P-value, Mantel test.

has a low correlation of 0.02 and 0.03 for the two conditions, respectively, and corresponding higher P-values of >0.3 for both comparisons.

In summary, both the FM index and Mantel test results suggest that for these subsystems, min-growth and enhanced-growth behaviours are related, while for both conditions, the behaviour is not significantly related to structure using the similarity metric defined above.

Scalability. None of the algorithms involved causes any scalability issues.

Discussion and conclusions

The analysis of the dynamic behaviour of bacterial GEMs currently presents considerable challenges because of the difficulties of simulating such large and complex networks. Moreover, such models cannot be checked without sophisticated tool support.

In this article, we have described a workflow comprising a set of methods and associated tools to analyse the dynamic behaviour of whole-genome bacterial metabolic models, illustrating these on a reduced version for ease of reproducibility and clarity. The workflow is applicable to full-scale GEMs, as illustrated by the discussion of scalability in the 'Methods' section. The focus of the running example is the analysis of two configurations of the reduced model, to compare the effects of growth conditions, with a special emphasis on reactions and functional subsystems. We introduced abstract views, which provide complementary insights, which are specifically useful for the differential behaviour analysis of sets of related models. In our example, this enables us to correlate the effect of environmental conditions with pathway activity.

The ability to partition a whole-genome metabolic model into its constituent subsystems facilitates the exploration of the behaviour of a subsystem in isolation as well as in combination with other subsystems. We generally expect to obtain different behaviour and more meaningful insights when analysing pathways in context rather than in isolation; high connectivity can compound this effect.

The models are currently not dynamically adaptive to environmental conditions in terms of changes in the expression of genes coding for enzymes, and hence changes in the concentration of the corresponding enzymes. However, the models do encode the reactions catalysed by the enzymes, i.e. products of gene regulation, for parts of the network that are activated because of environmental inputs, and hence in this sense do encode the relationship between environment, gene regulation and metabolic activity. The concentration of the enzymes is thus represented by the rate constants of the corresponding reactions. To incorporate dynamically changing availability of enzymes, the model would need to include dynamically changing rate constants, or even better explicitly model the enzymes, which is not done at present. The development of a model of such a complex system connecting gene expression to enzyme availability is an interesting topic, which has been explored by e.g. [40]; however, that work used constraint-based methods for the metabolic part, and Petri nets for gene regulation.

Our techniques to explore GEMs through transient behaviour and structural characteristics gave us insights into the functional behaviour of the GEM. The metabolic functional pathways become inactive early on in the min-growth model, whereas all of them exhibited activity throughout the behaviour of the enhanced-growth model. Structural analysis showed that the core metabolic subsystems are closely interconnected in terms of metabolites, while the externally oriented reactions are remote from the core metabolic subsystems.

Our techniques can be applied to large whole-genome-scale models, exploiting the scalability of our abstraction and approximation techniques. These are general approaches, and not bound to Petri net representations; note that some of what is done here with Snoopy or Marcie could be achieved with Copasi [41], as long as we use standard simulation algorithms. Moreover, the data analytics methods can be applied to any kind of simulation traces, not just approximative ones. The approach can be used to compare the behaviour of sets of related models, for example during model development, automatic repair, manual curation or target-based optimization.

Our overall longer-term goal is to build on our workflow to support the design process in synthetic biology and the sound implementation of valuable engineered organisms. Our tools are ready to be used for metabolic engineering as soon as we have more precise kinetic information available. The determination of these is a challenge, which we are currently addressing. In the course of this, we intend to incorporate constraint-based methods, for example to obtain steady-state fluxes for the derivation of kinetic rate constants, along the lines of [2, 42].

Model modification and configuration, and the behavioural analysis induced by these models, will play a crucial role in predicting the results of genetic engineering or forced evolution in the context of specific nutritional environments. Moreover, the dynamic approach will play a crucial role in the characterization and analysis of genome-scale signal transduction or gene regulatory networks as reliable models increasingly become available in the future.

Key Points

- Simulation of dynamic behaviour of large genome-scale models.
- Analysis of dynamic behaviour of large genome-scale models using model checking and data analytics.
- Analysis of the behaviour of functional subsystems.
- Workflow building on public domain tools, and supporting methodology.
- Illustrated on non-trivial public domain running example.

Supplementary data

Supplementary data are available at *BIB* online.

Funding

MH was partially funded by the Brunel University London, College of Engineering, Design and Physical Sciences via the SEED fund to promote multi-disciplinary research.

References

1. Karr JR, Sanghvi JC, Macklin DN, *et al*. A whole-cell computational model predicts phenotype from genotype. *Cell* 2012; **150**(2):389–401.
2. Smallbone K, Mendes P. Large-scale metabolic models: from reconstruction to differential equations. *Industrial Biotechnology* 2013;**9**(4):179–84.
3. Palsson BØ. *Systems Biology: Constraint-Based Reconstruction and Analysis*. Cambridge, UK: Cambridge University Press, 2015.
4. Rohr C. Simulative analysis of coloured extended stochastic Petri nets. PhD thesis, Department of Computer Science, Brandenburg Technical University, Cottbus, 2017.
5. Hucka M, Finney A, Sauro HM, *et al*. The Systems Biology Markup Language (SBML): a medium for representation and exchange of biochemical network models. *J Bioinformatics* 2003;**19**:524–31.
6. Monk JM, Charusanti P, Azizb RK, *et al*. Genome-scale metabolic reconstructions of multiple *Escherichia coli* strains highlight strain-specific adaptations to nutritional environments. *Proc Natl Acad Sci USA* 2013;**110**(50):20338–43.
7. King Z, Lu A, Dräger JA, *et al*. BiGG models: a platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids Res* 2016;**44**(D1):D515–22.
8. Mamiatis T, Fritsch EF, Sambrook J, Engel J. *Molecular Cloning–A Laboratory Manual*. New York, NY: Cold Spring Harbor Laboratory. 1982, 545 S., 1985.
9. Heiner M, Gilbert D, Donaldson R. Petri nets in systems and synthetic biology. In: *Formal Methods for Computational Systems Biology, SFM 2008*. (LNCS, Vol. 5016), Berlin-Heidelberg: Springer, 2008, 215–64.
10. Heiner M, Herajy M, Liu F, *et al*. Snoopy—a unifying Petri net tool. In: *Proceedings of the International Conference on Application and Theory of Petri Nets and Concurrency*, (LNCS, Vol. 7347). Berlin-Heidelberg: Springer, 2012, 398–407.
11. Orth JD. Systems biology analysis of Escherichia coli for discovery and metabolic engineering. PhD thesis, University Of California, San Diego, 2012.
12. Orth JD, Fleming RMT, Palsson BØ. Reconstruction and use of microbial metabolic networks: the core Escherichia coli metabolic model as an educational guide. *EcoSal plus* 2010;**4**(1). doi: 10.1128/ecosalplus.10.2.1.
13. Orth JD, Conrad TM, Na J, *et al*. A comprehensive genome-scale reconstruction of Escherichia coli metabolism–2011. *Mol Syst Biol* 2011;**7**(535):535.
14. Erdrich P, Steuer R, Klamt S. An algorithm for the reduction of genome-scale metabolic network models to meaningful core models. *BMC Syst Biol* 2015;**9**(1):48.
15. Diaz D, Codognet P. The GNU prolog system and its implementation. In: *Proceedings of the 2000 ACM Symposium on Applied Computing*, Vol. 2. New York, NY, USA: ACM, ACM Digital Library, 2000, 728–32.
16. Heiner M, Schwarick M, Wegener J. Charlie—an extensible Petri net analysis tool. In: *Proceedings of the International Conference on Applications and Theory of Petri Nets and* Concurrency (LNCS, Vol. 9115). Cham-Heidelberg-New York-Dordrecht-London: Springer, 2015, 200–211.
17. Heiner M, Rohr C, Schwarick M. MARCIE—model checking and reachability analysis done effiCIEntly. In: *Proceedings of the International Conference on Applications and Theory of Petri Nets and Concurrency (LNCS, Vol.* 7927). Berlin-Heidelberg: Springer, 2013, 389–99.
18. Donaldson R, Gilbert D. A model checking approach to the parameter estimation of biochemical pathways. In: *Proceedings of the International Conference on Computational Methods in Systems Biology* (LNCS/LNBI, Vol. 5307). Berlin-Heidelberg: Springer, 2008, 269–87.
19. Team RC. *R: a language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing, 2014, 3–36.
20. Chindelevitch L, Stanley S, Hung D, *et al*. Metamerge: scaling up genome-scale metabolic reconstructions with application to mycobacterium tuberculosis. *Genome Biol* 2012;**13**(1):r6.
21. Breitling R, Gilbert D, Heiner M, *et al*. A structured approach for the engineering of biochemical network models, illustrated for signalling pathways. *Brief Bioinform* 2008;**9**(5):404–21.
22. Heijnen JJ. Approximative kinetic formats used in metabolic network modeling. *Biotechnol Bioeng* 2005;**91**(5):534–45.
23. Sabouri-Ghomi M, Ciliberto A, Kar S, *et al*. Antagonism and bistability in protein interaction networks. *J Theor Biol* 2008; **250**(1):209–18.
24. Heiner M, Rohr C, Schwarick M, *et al*. A comparative study of stochastic analysis techniques. In: *Proceedings of the 8th International Conference on Computational Methods in Systems Biology*. New York, NY, USA: ACM, ACM Digital Library, 2010, 96–106.
25. Gillespie DT. A general method for numerically simulating the stochastic time evolution of coupled chemical species. *J Comput Phys* 1976;**22**:403–34.
26. Cao Y, Gillespie DT, Petzold LR. Adaptive explicit-implicit tau-leaping method with automatic tau selection. *J Chem Phys* 2007;**126**(22):224101.
27. Wilkinson DJ. *Stochastic Modelling for System Biology*, 1st edn. New York, NY: CRC Press, 2006.
28. Fages F, Rizk A. On the analysis of numerical data time series in temporal logic. In: *Proceedings of the International Conference on Computational Methods in Systems Biology* (LNCS/LNBI, Vol. 4695), Berlin-Heidelberg-New York: Springer, 2007, 48–63.
29. Donaldson R, Gilbert D. A model checking approach to the parameter estimation of biochemical pathways. In: *Proceedings of the International Conference on Computational Methods in Systems Biology*. Springer, 2008, 269–87.
30. Maccagnola D, Messina E, Gao Q, *et al*. A machine learning approach for generating temporal logic classifications of complex model behaviours. In: *Proceedings of the Winter Simulation Conference (WSC)*. IEEE, 2012, 1–12.
31. Liao TW. Clustering of time series data—a survey. *Pattern Recognition* 2005;**38**(11):1857–74.
32. Wickham H. *ggplot2: Elegant Graphics for Data Analysis*. Dordrecht-Heidelberg-London-New York: Springer, 2009.
33. Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *J Comput Appl Math* 1987;**20**:53–65.
34. Berndt DJ, Clifford J. Using dynamic time warping to find patterns in time series. In *AAAIWS'94 Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, Technical Report WS-94-03. Seattle, WA: AAAI Press, 1994, 359–70.
35. Fowlkes EB, Mallows CL. A method for comparing two hierarchical clusterings. *J Am Stat Assoc* 1983;**78**(383):553–69.

36. Galili T. *dendextend: Extending R's Dendrogram Functionality*, 2015.
37. Mantel N. The detection of disease clustering and a generalized regression approach. *Cancer Res* 1967;**27(2 Pt 1)**:209–20.
38. Chessel D, Dufour AB, Thioulouse J. The ade4 package-i-one-table methods. *R News* 2004;**4**(1):5–10.
39. Chaovalit P, Gangopadhyay A, Karabatis G, *et al*. Discrete wavelet transform-based time series analysis and mining. *ACM Comput Surv* 2011;**43**(2):6.
40. Fisher CP, Plant NJ, Moore JB, *et al*. QSSPN: dynamic simulation of molecular interaction networks describing gene regulation, signalling and whole-cell metabolism in human cells. *Bioinformatics* 2013;**29**:3181–90.
41. Hoops S, Sahle S, Gauges R, *et al*. Copasi—a complex pathway simulator. *Bioinformatics* 2006;**22**(24):3067–74.
42. Machado CD, Costa RS, Rocha M, *et al*. Model transformation of metabolic networks using a Petri net based framework. In: International Workshop on Biological Processes & Petri Nets (BioPPN 2010). Universidade do Minho, Portugal, 2010, 101–15. http://hdl.handle.net/1822/16761.