Bioinformatics
doi.10.1093/bioinformatics/btab335
Advance Access Publication Date: 10 May 2021
Original Paper
Accepted Version
Version of record: https://doi.org/10.1093/bioinformatics/btab335

Systems biology

Visual Exploration of Large Metabolic Models

Michael Aichem [1,*], Tobias Czauderna [2], Yan Zhu [3], Jinxin Zhao [3], Matthias Klapperstück [2], Karsten Klein [1], Jian Li [3] and Falk Schreiber [1,2]

[1]Department of Computer and Information Science, University of Konstanz, Konstanz, Germany
[2]Faculty of Information Technology, Monash University, Melbourne, Australia and
[3]Biomedicine Discovery Institute, Infection & Immunity Program and Department of Microbiology, Monash University, Melbourne, Australia.

*MA and TC: contributed equally to this paper and should be both considered first author.

Associate Editor: Alfonso Valencia

Received on July 29, 2020; revised on March 1, 2021; accepted on April 30, 2021

Abstract

Motivation: Large metabolic models, including genome-scale metabolic models (GSMMs), are nowadays common in systems biology, biotechnology and pharmacology. They typically contain thousands of metabolites and reactions and therefore methods for their automatic visualisation and interactive exploration can facilitate a better understanding of these models.

Results: We developed a novel method for the visual exploration of large metabolic models and implemented it in LMME (Large Metabolic Model Explorer), an add-on for the biological network analysis tool VANTED. The underlying idea of our method is to analyse a large model as follows. Starting from a decomposition into several subsystems, relationships between these subsystems are identified and an overview is computed and visualised. From this overview, detailed subviews may be constructed and visualised in order to explore subsystems and relationships in greater detail. Decompositions may either be predefined or computed, using built-in or self-implemented methods. Realised as add-on for VANTED, LMME is embedded in a domain-specific environment, allowing for further related analysis at any stage during the exploration. We describe the method, provide a use case, and discuss the strengths and weaknesses of different decomposition methods.

Availability: The methods and algorithms presented here are implemented in LMME, an open-source add-on for VANTED. LMME can be downloaded from www.cls.uni-konstanz.de/software/lmme and VANTED can be downloaded from www.vanted.org. The source code of LMME is available from GitHub, at https://github.com/LSI-UniKonstanz/lmme.

Contact: michael.aichem@uni-konstanz.de

1 Introduction

Metabolic models have gained increasing interest over the last decades. Large metabolic models, including genome-scale metabolic models (GSMMs), which represent the complete metabolism of an organism and are usually based on its genomic information, are nowadays common in systems biology, biotechnology and pharmacology. Their analysis and simulation provides deeper insight into the molecular mechanisms of the organism under investigation and is, for example, important to predict targets for gene manipulations and to understand the metabolic effects of drugs. Since the first GSMM was developed for the bacterium *Haemophilus influenzae* in 1999 (Edwards and Palsson, 1999), many GSMMs have been created for bacteria, archaea as well as eukaryota. For example, Path2Models (Büchel *et al.*, 2013), a branch of the BioModels Database (Malik-Sheriff *et al.*, 2019), contains a large amount of metabolic (and other) models automatically generated from pathway resources, including more than 2,600 genome-scale metabolic reconstructions.

Large metabolic models typically contain thousands of metabolites and reactions. Automatic visualisation and interactive exploration methods can facilitate a better understanding of metabolic models and help to find errors in a model more easily, to support model comparisons, and to solve similar tasks. Visualisations have already proven to be useful to investigate biological data and processes, and many different approaches exist for all kinds of biological data (Gehlenborg *et al.*, 2010;

Kerren *et al.*, 2017), including biological networks such as metabolic networks (Kohlbacher *et al.*, 2014). To provide interactive exploration, the decomposition of the metabolic network into sub-networks (pathways) is an important step as it allows to break the large and complex network into several much smaller parts. Such a decomposition can be done manually, for example, the pathways in the KEGG database (Kanehisa *et al.*, 2012) are manually curated, or by using an algorithm, such as the one presented by Schuster *et al.* (2002). Decompositions also offer new ways for analysing data related to the metabolic network, such as over-representation analysis (Khatri *et al.*, 2012), using the given or computed decomposition of the network as starting point.

This paper describes a novel method for the interactive visualisation of large metabolic models, including GSMMs, which is based on model decompositions. Starting from a decomposition into subsystems, their relationships are identified and an interactive overview is computed and visualised. This overview may be analysed, compared, or used to select parts to be recombined and shown in greater detail. Integrating more than one decomposition as alternatives into this conceptual pipeline may facilitate the understanding and eventually lead to broader conclusions. As one possible way, we discuss decomposition-based over-representation analyses. As a proof of concept, the method has been implemented in LMME (Large Metabolic Model Explorer), an add-on for the open-source framework VANTED (Junker *et al.*, 2006). Implementing it as a VANTED add-on allows the analyst to make use of all the other functionality of VANTED (such as mapping of *omics data onto metabolic networks and general network analysis algorithms) in addition to layout, interactive exploration and decomposition-based over-representation analysis. We introduce the methodological background and describe the method in detail, followed by a use case and a discussion of the strengths and weaknesses of different decomposition methods.

2 System and Methods

2.1 Background

2.1.1 Model sources

Metabolic models can be built from scratch, obtained from supplementary materials of publications and from databases such as the BioModels Database (Malik-Sheriff *et al.*, 2019), BiGG Models (King *et al.*, 2015a), the Human Metabolic Atlas (Robinson *et al.*, 2020), Model SEED (Henry *et al.*, 2010) and the Virtual Metabolic Human Database (Brunk *et al.*, 2018). According to the review by Gu *et al.* (2019), GSMMs have been reconstructed for more than 6,200 organisms.

These models are typically given as a file using the SBML (Hucka *et al.*, 2003) notation, a machine-readable XML-based format for the representation of biochemical models. Often, the models are also given in form of an SBGN (Le Novère *et al.*, 2009) map, a graphical representation standard for biochemical models.

2.1.2 Layout

Algorithms for the layout of metabolic pathways and networks have been presented first in the mid 1990s and early 2000s, for example, the mixed layout approach of Karp and Paley (1994) (which depicts (sub-)pathways of different topology using combined linear, circular, tree and hierarchical layout algorithms), an extended layered approach (Schreiber, 2002) (which provides hierarchical layout for different node sizes, consideration of co-substances and special layout of open and closed cycles) and the algorithm of Becker and Rojas (2001) (which emphasises cyclic structures). Most layout methods available nowadays are based on simple force-based layout methods (in particular for large networks with thousands of elements), while manual layouts such as given in KEGG (Kanehisa *et al.*, 2012) are still important. Only a few advanced methods for automated conversion are available, see for example the approach by Czauderna *et al.* (2013), which is using a constraint-based method.

2.1.3 Interactive exploration

There are many tools and web-based systems which visually represent metabolic pathways and networks (partly including the possibility to design or customise pathways), allow to search through the visualisation and provide mechanisms to map data onto pathways. Examples include ArrayXPath (Chung *et al.*, 2004), CellDesigner (Funahashi *et al.*, 2008), Cytoscape (Shannon *et al.*, 2003), Escher (King *et al.*, 2015b), iPath (Darzi *et al.*, 2018), PathVisio (Kutmon *et al.*, 2015), Omix (Droste *et al.*, 2011), VANTED (Junker *et al.*, 2012), VisAnt (Granger *et al.*, 2016) and WikiPathways (Slenter *et al.*, 2017).

However, there are only a few tools which provide interactive layout and exploration methods for metabolic models. The tool ModelExplorer (Martyushenko and Almaas, 2019) is intended to check a model for inconsistencies during the construction and refinement processes. It allows to visually explore reactions that may not be able to carry flux during simulations, showing it within its local neighbourhood in the network. The web-based application provided by the Metabolic Atlas project (Robinson *et al.*, 2020) also offers visual exploration of models, but is so far restricted to two available models. A large overview map is provided and one can investigate individual pathways and compartments in detail. MetExplore (Cottret *et al.*, 2018) offers web-based exploration of large models providing a set of interactive, interconnected tables, together with a visualisation component. Subsets of pathways and reactions can be filtered and visualised together, showing their interconnections. However, these tools do not directly provide a summary of the overall network structure and the relationships between pathways or subsystems in general.

Using a graph summary that allows to investigate details on demand is an established technique when dealing with the exploration of large graphs (Pienta *et al.*, 2015), that has also been adopted by corresponding applications for metabolic models. KGML-ED (Klukas and Schreiber, 2007) provides the opportunity to create an overview graph, where nodes correspond to KEGG pathways. Any node can then be expanded to show the contained reactions and metabolites, which allows for a detailed investigation of the interaction between pathways. During the process, nodes may also be collapsed again to prevent from loosing track of the overall structure. Another tool, GLIEP (Jusufi *et al.*, 2012), uses a different approach to investigate the interconnections between different pathways. Having a detailed view of a particular pathway, one can directly see which of the contained metabolites serve as an interface to other pathways. The other pathways may then be selected and be shown in detail. In addition, glyphs (graphical markers) are used to show the distribution of the connected pathways across pathway categories.

Our concept deviates from what the mentioned tools offer as follows. While these, if any, only allow to explore predefined decompositions obtained from respective model annotations or established online databases (such as KEGG), the main purpose of our method is to allow the exploration of large models through different decompositions of the model. The approach is complemented by the possibility to extend the range of possible decomposition methods. This counteracts the situation that there have been developed many decomposition methods for this kind of networks, but hardly any applications to integrate them. To the best of the authors' knowledge, LMME is the first tool providing such an exploration approach that includes different decomposition methods for large metabolic models.

2.1.4 Decomposition

With the increasing size of metabolic models there is an increasing need for automatic methods to compute decompositions of them. Accordingly, a variety of methods have been developed and published over the last two

decades. There is for example a very recently published web service, called GEMtractor, that allows a model to be trimmed by removing currency metabolites (abundant metabolites, see Subsection 2.3) and transformed to related alternative representations (Scharm *et al.*, 2020).

Available decomposition methods on the one hand may only take the network structure into account to derive a decomposition, e.g. by identifying hub metabolites using local (Schuster *et al.*, 2002) or global (Holme *et al.*, 2003) criteria, that are then interpreted as connections between different subsystems. On the other hand, there are methods that include more domain-specific data, e.g. metabolic flux measurements (Yoon *et al.*, 2007), in order to derive a decomposition. The interested reader may find several of these methods in the reviews by Rezvan and Eslahchi (2017) and Singh and Lercher (2020).

The pathways that may be found in the literature often overlap and intertwine to a large extent (Holme *et al.*, 2003). To complement these traditional decompositions, the need for unbiased decomposition methods has been stated several times (Holme *et al.*, 2003; Papin *et al.*, 2004). While, for the available methods, it is generally assumed that metabolites may belong to more than one subsystem, e. g. as interfaces (Schuster *et al.*, 2002; Ma *et al.*, 2004), reactions are in most of these approaches only assigned to at most one subsystem (see for example Schuster *et al.*, 2002; Holme *et al.*, 2003; Ma *et al.*, 2004). To be consistent, we always assign a reaction to exactly one subsystem in our proposed method. As the KEGG database sometimes assigns a reaction to more than one pathway, we used a heuristic approach to end up with a single assignment (for more information, see Sec. 3.1).

2.1.5 Over-representation analysis

Over-representation analysis (ORA) is a standard analysis tool for many years now (Khatri *et al.*, 2012), even though there are still some shortcomings, see also the performance evaluation of Marco-Ramell *et al.* (2018) which in particular mentions the completeness of metabolite and pathway databases as an issue. The method has been originally developed for gene sets and starting with MSEA (Xia and Wishart, 2010) in 2010, several similar techniques for metabolite sets have been developed. The underlying idea is to test a family of metabolite sets in order to find sets that contain an unexpectedly high proportion of metabolites that have been measured with significantly altered concentrations. The ratio between the number of significant metabolites and the number of reference metabolites (e.g., the total set of metabolites that have been measured during an experiment) is computed and compared to the ratios within the individual metabolite sets using statistical methods. This reflects the following idea: if, for example, 27 out of 100 measured metabolites were significant, one would also expect to have around 27% of the metabolites within every metabolite set being significant. However, if there is a metabolite set containing 33 metabolites and 20 of them are significant, this would be a starting point for a detailed investigation.

Khatri *et al.* (2012) describe three generations of pathway analysis approaches: over-representation analysis (ORA) approaches as first generation, functional class scoring (FCS) approaches as second generation, and pathway-topology (PT)-based approaches as third generation. There are also approaches that combine methods originating from different of the mentioned generations, like the tool netGO (Kim *et al.*, 2020), which has been published very recently.

As the focus of our tool is on exploring the structure of large models, we decided to initially only provide a simple statistical approach, while, once having computed a decomposition of interest, the analysis beyond that can be performed using other tools that are specifically designed for that purpose.

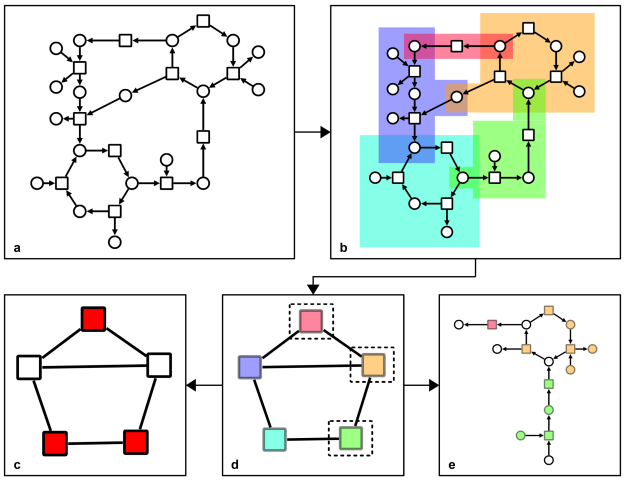

Fig. 1. A schematic overview of the method. Subfigure (a) shows an example metabolic network (base graph) that may be decomposed into five subsystems (pathways) as indicated by the background colours in subfigure (b). The corresponding overview graph, preserving the colours, is depicted in subfigure (d). The user may then either perform an over-representation analysis, providing a visual mapping like the one shown in subfigure (c) (red colour means that the p-value of the subsystem was significant), or investigate selected subsystem(s) as shown in subfigure (e) (again preserving the colours).

2.2 Approach

The methodological approach described in this paper mainly consists of two phases: the *decomposition phase* and the *exploration phase*, see Fig. 1.

In the decomposition phase, a model is decomposed into several subsystems (pathways) that, in a graph-theoretic sense, are (overlapping) subnetworks of the original network. This may take the topological network structure into account, as well as integrate more domain-specific data. For further details see Sec. 3. Using the decomposition, an overview layout of the resulting subsystems and their relationships is computed, presenting one node per subsystem, connected by edges that reflect the connections between the respective subsystems (Fig. 1(d)).

In the subsequent exploration phase, users can explore the model. Employing the view of the overall model structure (Fig. 1(d)) one can identify subsystems of interest, which can be selected and investigated in more detail on demand (Fig. 1(e)). This is an iterative process that may be repeated for any subsystem of interest. Moreover, one can select more than one subsystem at a time in order to investigate the interplay between these subsystems, see Sec. 3 for more details. In addition, the overview graph can be used for analysis, for example, for over-representation analysis (Fig. 1(c)).

The approach is implemented as an add-on for VANTED (Rohn *et al.*, 2012), so the analysis and visualisation infrastructure provided by VANTED can be used to further investigate both the overview graph and the detailed subsystem graphs, see Sec. 4 for more details.

2.3 Theoretical Model

This section outlines the theoretical model of our approach and defines the terms that will be used in the remainder of the article.

A *graph* $G = (V, E)$ consists of a set of vertices V and a set of edges E. The graph may either be *directed* (i. e., the edge set consists of ordered pairs of vertices) or undirected (i. e., unordered pairs of vertices). A vertex is a *neighbour* of another vertex if they have a common edge. The *degree* of a vertex is the number of edges that this vertex belongs to. As *base graph*, we refer to the bipartite graph that represents the large metabolic model of

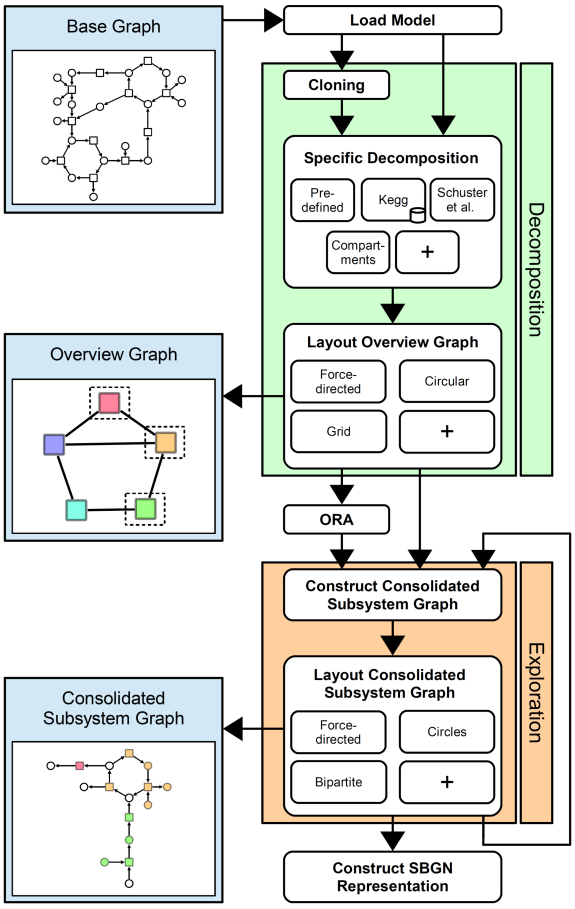

Fig. 2. A flow chart describing the overall workflow. For some of the steps, the corresponding graphs are shown next to the chart. The symbol + represents further optional algorithms.

interest (Fig. 1(a)). Its vertex set is partitioned into the set of metabolites and the set of reactions, which are connected via directed edges.

A *decomposition* is a set of subgraphs of the base graph (Fig. 1(b)). Each of these subgraphs is referred to as a *subsystem* and it is assumed that it represents a meaningful subunit within the overall system. For any reaction vertex that is contained in a subsystem, also every adjacent metabolite vertex is contained in the same subsystem. Therefore, a subsystem may already be determined by the reactions it contains. Accordingly, a decomposition can be constructed by assigning every reaction to a subsystem.

A *transport reaction* is a reaction in which metabolites are transported from one cellular compartment to another. On the network level, several definitions for transport reactions exist. The one we used for our implementation is a reaction that involves metabolites from at least two different compartments. The set of all transport reactions may already constitute a subsystem itself - the *transporter subsystem*.

An *interface* between two subsystems is a metabolite that participates in two reactions that each belong to one of the two subsystems.

The *overview graph* is the graph that is constructed as follows: There is one vertex per subsystem that has been derived in a particular decomposition while an edge connects two subsystems if the base graph contains an interface between these two subsystems (Fig. 1(d)).

Metabolite vertices in the base graph with many neighbours may be *cloned*. For this purpose, a metabolite vertex of degree d is replaced by d copies of itself, each of which are adjacent to exactly one of the edges that the original vertex has been adjacent to. Cloning metabolites in our case is mainly done for two reasons. First, it keeps the graphs readable in further stages, as it decreases their density. Second, it avoids meaningless interfaces between subsystems in the overview graph. For example, if two subsystems both include ATP (which is very likely), this does not necessarily mean that these subsystems have a meaningful relationship. Following the terminology of Huss and Holme (2007), we distinguish between *currency metabolites* (the abundant metabolites) and *commodity metabolites* (the non-currency metabolites).

A *consolidated subsystem graph* is a bipartite graph that is constructed as union of several subsystems (Fig. 1(e)). Unifying in this case ensures that interfaces are only contained once in the resulting graph.

3 Algorithm

This section gives a detailed description of the algorithms and the corresponding user workflows. We first explain the decomposition phase and then the exploration phase. Fig. 2 gives a visual overview of the overall possible workflows using our tool.

3.1 Decomposition Phase

1. Load a metabolic model in the SBML format into VANTED (either as file, or via direct access from VANTED to the BioModels Database).
2. Select a decomposition method (see step 4 for details).
3. Cloning is possible for all currently available decomposition methods: Select a degree threshold, such that all metabolites having at least this degree are cloned. However, as the degree alone is not always sufficient to discriminate between currency metabolites and commodity metabolites, the list of cloned metabolites can be manually edited.
4. Compute the decomposition. Currently, the following four different methods are provided in LMME.

KEGG decomposition: Users specify the name of an SBML note on an SBML reaction that contains the KEGG identifier (ID) of a reaction. In addition, the user sets a threshold t for the minimum number of reactions per subsystem (the role of t is described below). Let W denote the set of all reactions that provide such an ID (working set). Using the KEGG Rest API, for any reaction $R \in W$, its ID is used to retrieve a list C_R of candidate pathways that R might belong to. Having computed C_R for all $R \in W$, the following heuristic approach is used to assign each $R \in W$ to a subsystem. We iteratively repeat the following steps until W is empty:

1. For the KEGG pathway P that currently is contained in C_R for the most reactions $R \in W$, create a new subsystem S_P.
2. For each reaction R that has P as candidate pathway (i.e. $P \in C_R$), assign R to the new subsystem S_P and remove R from the working set W.
3. For any KEGG pathway P that is now contained in C_R for less than t reactions $R \in W$ (meaning that P is a candidate pathway only for less than t reactions), remove P from C_R. If C_R is now empty, also remove R from W.

The reason for using the threshold t is that some reactions exist in more than one pathway and therefore some of the candidate pathways may actually not be part of the model at all. For example, the model described in Sec. 3.4 contains a reaction (*L-glutamate-5-semialdehyde:NADP+ 5-oxidoreductase (phosphorylating)*), which according to its KEGG ID belongs to the pathways *Arginine and proline metabolism* and *Carbapenem biosynthesis*. However, while there are 36 reactions in the model belonging

to the former, there are only 3 that belong to the latter. So, *Carbapenem biosynthesis* is most probably not part of the model. Instead, the model only shares a few reactions. Consequently, if a pathway occurs as candidate in less than t reactions, it is very likely that the respective pathway is not part of the model but it just occurred as alternative in a few of the reactions by chance. The threshold itself is very sensitive. An ad hoc test with the model described in Sec. 3.4 revealed that $t = 1$ results in 71 subsystems, $t = 5$ results in 46 subsystems and $t = 10$ results in 35 subsystems. Hence, there are 36 subsystems that contain between 1 and 9 reactions. The threshold may be considered as a measure of confidence for the existence of the actual subsystems in the model. The default value is $t = 5$.

Decomposition by Schuster *et al.* (2002): This method computes the decomposition solely from the graph structure of the model. It temporarily removes vertices with degree above a specified threshold, followed by a computation of the resulting connected components within the graph. These are then interpreted as subsystems, while the temporarily removed vertices are understood as interfaces.

Compartment decomposition: The compartment affiliation of the metabolites is read from the SBML file and the reactions are accordingly classified into either being inside one of the compartments or as being a transport reaction. The decomposition consists of the transporter subsystem and one subsystem per compartment contained in the model.

Predefined decomposition: The user specifies the name of an SBML note on an SBML reaction (Hucka *et al.*, 2019), containing a subsystem assignment, that is then used to classify the reactions.

In case there are unclassified reactions after the decomposition procedure, these reactions will be assigned to a *default subsystem*. This ensures that the model itself remains complete and no entities are lost. In addition to the derived decomposition, the transporter subsystem can be computed optionally, into which any transport reaction will be inserted, independent of whether it has been classified before. This may reduce the size and complexity of the default subsystem by introducing another subsystem that is determined by its function.

After a decomposition is computed, for each pair of subsystems their interfaces are determined and from this information the overview graph is constructed. It is then laid out using a force-directed layout, a grid layout, or a circular layout. However, VANTED offers some additional layout algorithms that can easily be accessed and executed afterwards.

Once the overview graph is constructed, it is shown on the left side of the application window, reserving the right side for the detail view of a consolidated subsystem graph (see Fig. 3).

3.2 Exploration Phase

There are two main directions for model investigation: Analysing the overview graph or selecting one or more subsystems for a detailed look at their consolidated subsystem graph.

To analyse the relationships between the derived subsystems, the number of interfaces can be mapped to the edge thickness in the overview graph (Fig. 3). However, if the relationships are not of interest at all, the edges can be hidden in order to reduce visual clutter in the drawing. In addition, other analytic features provided by VANTED can be used to further investigate the overview graph. This includes computing attributes like centrality values and mapping the results to visual variables such as the node size or colour. Whenever a single subsystem node in the overview graph is selected, additional information such as the number of reactions and metabolites that are contained in this subsystem is shown. Whenever an edge between two subsystem nodes is selected, the list of interfaces that confirm this relationship is shown (Fig. 3 top right and in the overlay bottom right). These features may guide the decision for subsystems that may be analysed in the detail view.

Selected subsystems in the overview graph can be used to construct and visualise a consolidated subsystem graph (CSG) out of them. The CSG is shown on the right side of the application window. A colour mapping between the overview graph and the subsystems view ensures a user can keep track of which metabolites and reactions belong to which subsystem. Interfaces remain uncoloured to emphasise their role in the CSG. Different layout algorithms for the drawing of the CSG can be selected. Besides the force-directed algorithm, there are two layouts which make use of the bipartition of the CSG: a layout that consists of two concentric circles, and a layout that consists of two straight lines made up of nodes. Using further layout algorithms provided by VANTED as well as implementing new ones is also possible.

Finally, the consolidated subsystem graph drawing can be translated into SBGN-PD (Rougny *et al.*, 2019) or subsequently translated into SBGN-AF (Mi *et al.*, 2015).

3.3 Over-representation analysis based on network decomposition

As soon as the overview graph has been constructed, the user can also run an over-representation analysis. For an explanation of this method, see Subsec. 2.1.5. We implemented a one-tailed Fishers exact test (assuming a hyper-geometric distribution) using the false discovery rate (Benjamini and Hochberg, 1995) (FDR) to correct for multiple testing. These are very common techniques for ORA (Khatri *et al.*, 2012). To perform this analysis, the user additionally provides a list of metabolites that have been measured with significantly altered concentrations. In addition, the user may provide another list containing the reference set for the calculation (a superset of the former, e. g. the total set of metabolites that have been measured during the experiment), or decides to use the set of all metabolites that exist in the model as a reference set. Using Fishers exact test and FDR, a corrected p-value for every subsystem is then computed and all subsystems having $p \leq 0.05$ are coloured while the others remain uncoloured (see Fig. 1(c) for a schematic view).

While many of the previously published ORA tools only allow for the comparison against the predefined metabolite sets of corresponding online databases, our approach allows the analysis to be done for any decomposition that has been derived before (including user-specific implementations).

3.4 Use Case

To demonstrate the usage of LMME, we now describe an exemplary use case. We used *iPAO1*, a GSMM for *Pseudomonas Aeruginosa PAO1*, that has been developed by Zhu *et al.* (2018). A copy of the model can be downloaded from the LMME webpage. We applied all four currently available decomposition methods to the model and computed some properties of the resulting decompositions: the number of subsystems, the number of connections between the subsystems (i. e. the number of edges in the overview graph), the number of reactions contained in the transporter and default subsystems, as well as the minimum, the median, the maximum, the mean, and the standard deviation of the number of reactions over all subsystems. The detailed results are shown in Table 1.

iPAO1 contains 3,022 metabolites and 4,365 reactions. For all decompositions, we chose the cloning threshold to be its default value 15 and chose the transporter subsystem option. This resulted in 1,854 transport reactions and 2,511 remaining reactions that were distributed across the remaining subsystems. The KEGG decomposition was the only method that introduced a default subsystem, while all remaining did not end up with any unclassified reactions. For the KEGG decomposition, the decomposition-specific reaction threshold was chosen to be its default value $t = 5$, while for the decomposition by Schuster *et al.* (2002), the

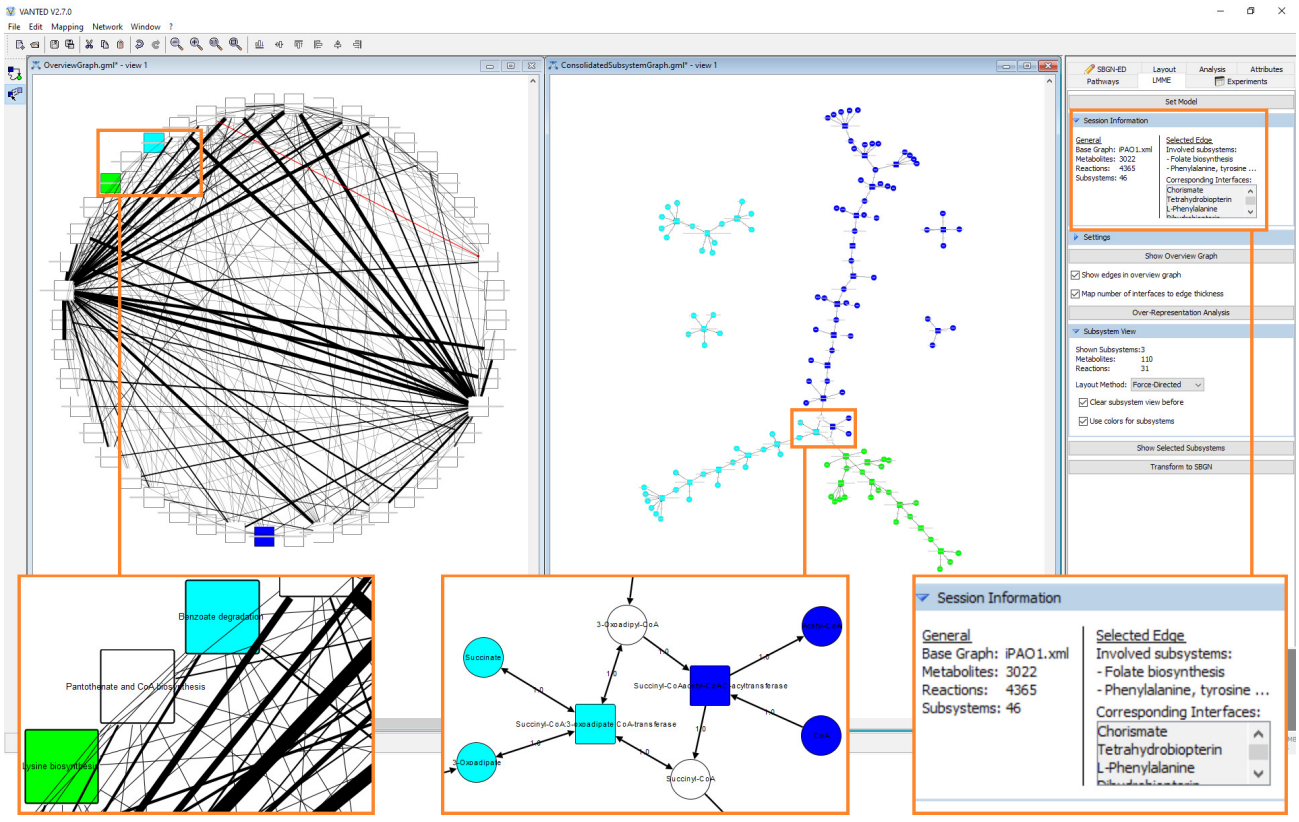

Fig. 3. A screenshot of LMME. On the left side, the three subsystems Benzoate degradation (light blue), Lysine biosynthesis (green) and Phenylalanine metabolism (blue) have been chosen to be shown as consolidated subsystem graph on the right side. Consistent colour coding is used to link the two views. One edge has been selected in the overview graph (for presentation purposes this edge has been drawn red). Corresponding information is shown in the session information panel (top right and in the overlay bottom right). The number of interfaces between two subsystems has been mapped to the edge thickness. The two highly connected subsystem nodes represent the default subsystem (left) and the transporter subsystem (right). For both views, an overlay has been added at the bottom, to show the appearance when zoomed in.

decomposition-specific splitting threshold was chosen to be its default value 8.

Fig. 3 shows the tools appearance, when performing a KEGG decomposition with the circular layout method chosen for the overview graph. From the 4,365 available reactions, 1,027 provide a KEGG reaction ID and may therefore be assigned to a subsystem that corresponds to a KEGG pathway. The execution results in a decomposition consisting of 46 subsystems. The default subsystem contains 1,645 reactions, while the transporter subsystem contains 1,854. The remaining 866 reactions are accordingly distributed across the remaining 44 subsystems. The reason why only 866 out of the annotated 1,027 reactions were finally assigned to a KEGG induced subsystem is that the remaining ones were either assigned to the transporter subsystem or belonged to a pathway that did not exceed the threshold $t = 5$. For more information, see the caption of Fig. 3.

Finally, we used a metabolomics dataset from an experiment studying metabolomic changes in response to drug treatments (Mahamad Maifiah, 2017), which was collected as follows. Metabolomics samples were collected at 0, 0.25, 1, 4, and 24h of a polymyxin B ($1mg/L$) time-kill experiment with an initial PAO1 inoculum size of 108 CFU/mL. Intracellular metabolites were then extracted and used for LC-MS analyses as previously described (Han *et al.*, 2019). Metabolomic data analyses were then performed using IDEOM (Creek *et al.*, 2012). Significantly changed metabolites of treated samples relative to untreated control samples at each time point were identified by One-way Analysis of Variance (ANOVA) ($p < 0.05$, $FDR \leq 0.05$) for multiple comparison and post hoc analysis using Tukey's Honestly Significant Difference (Tukey's HSD) with MetaboAnalyst 3.0 (Xia *et al.*, 2015).

Finally, using the KEGG ID, the processed data collected at 1h was mapped to those species in the model that provided a KEGG ID and we performed an ORA with this data on each of the four resulting decompositions. The resulting overview graphs are shown in Fig. 4.

4 Implementation

The method that was presented in this paper is implemented in the software LMME. LMME is open-source, developed in Java and realised as an add-on for VANTED. It uses several of VANTED's core features, such as drawing graphs, reading and processing SBML files, sending http requests to the KEGG API, and several graph data structures and algorithms. For translation to SBGN, the SBGN-ED add-on (Czauderna *et al.*, 2010) of VANTED also has to be installed. It also offers an SBGN-PD to SBGN-AF translation (Vogt *et al.*, 2013).

A custom decomposition method can be implemented by extending the abstract class `MMDecompositionAlgorithm`. A respective developers guide is available on the LMME webpage.

5 Discussion

When exploring a systems structure and functional mechanisms, having to view the detailed functional cascades within the entire context of

Table 1. Properties of the resulting decompositions after applying the different methods to iPAO1 (Zhu et al., 2018). For each resulting decomposition, we computed the number of subsystems, the number of connections between the subsystems (i. e. the number of edges of the overview graph), the number of reactions in the transporter and default subsystems, as well as the minimum, the median, the maximum, the mean, and the standard deviation (SD) of the number of reactions over all subsystems.

			Number of reactions						
Method	Subsystems	Connections	Transporter subsystem	Default subsystem	Minimum[a]	Median	Maximum[a]	Mean	SD
KEGG decomp.	46	237	1,854	1,645	3[b]	14	85	94.9	353.9
Schuster *et al.* (2002)	322	398	1,854	0	1	1	1,916	13.6	148
Compartment decomp.	4	3	1,854	0	392	1,059.5	1,716	1,091.3	695.5
Predefined decomp.	106	476	1,854	0	1	6.5	550	41.2	190.4

[a] Not considering the transporter and default subsystems.
[b] This falls below the threshold $t = 5$ as some reactions have been classified as transport reactions afterwards.

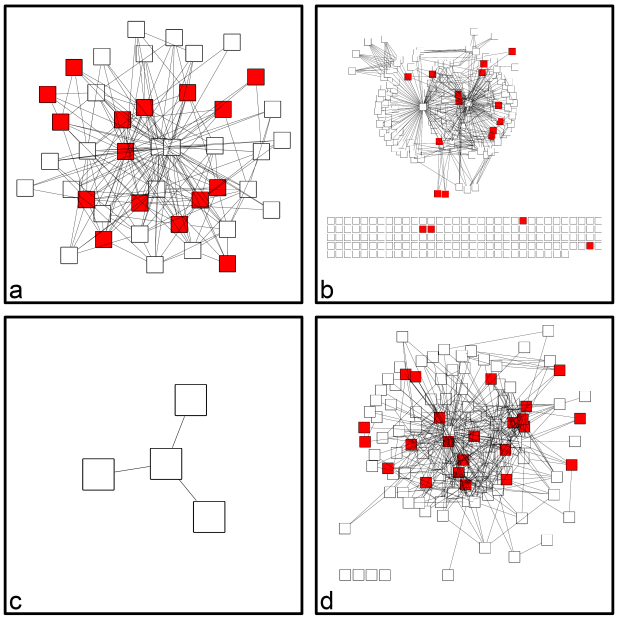

Fig. 4. Resulting overview graphs of the model discussed in Sec. 3.4. An over-representation analysis with the same underlying metabolomics dataset has been performed on all four decompositions: KEGG decomposition (a), decomposition by Schuster et al. (2002) (b), compartment decomposition (c), and predefined decomposition (d).

thousands of network entities might be challenging, time-consuming and cluttered. So, the idea is to reduce the size of the detailed view. To enable this as freely and unbiasedly as possible, it is crucial to offer several different decomposition methods and paradigms. In the following, we briefly discuss the strengths of individual methods in terms of the size of decompositions (number of resulting subsystems) and the size of individual subsystems (number of reactions contained), their confidence (to what extent the derived subsystems represent meaningful substructures of the total system), the underlying paradigm and the customisability.

Decomposition size: The number of resulting subsystems may be an important factor. While compartment decompositions result in very few subsystems (4 in our use case), the KEGG decomposition results in many more (46 in our use case), but is still bounded by the number of pathways available at KEGG. Computational methods on the other hand, may vary in their size, also being not bounded (e. g. 322 mostly small subsystems in our use case).

Sizes of subsystems: According to the means and medians computed in our use case, the subsystems resulting from the compartment are very large, while the KEGG and predefined decompositions seem to have provided more compact subsystems. The latter may, however, keep the successive exploration of the connections between different subsystems more manageable.

Confidence: The pathways available at KEGG are established, so the resulting decomposition has a high confidence (in terms of common metabolic pathways). In addition, the compartment decomposition naturally comes with a very high confidence (in terms of spatial separation of pathways). Computational methods, however, may have a low confidence, especially those that only consider the network structure. In particular, this can be seen in our use case, where the decomposition by Schuster *et al.* (2002) results in a median subsystem size of 1, meaning that at least half of the resulting subsystems only contain a single reaction. In this case, the decomposition may not be considered in its entirety but rather used to find individual meaningful subsystems that are part of it (Rezvan and Eslahchi, 2017)

Paradigm: While the compartment decomposition is a spatial decomposition, the KEGG decomposition is a functional one. Different computational methods in addition, may aim for both paradigms, or even further ones.

Customisability: While the compartment and KEGG decomposition methods are quite limited in their customisability, computational methods have their strengths here. Through parameters, users may be able to control the results, e. g. varying the size. However, this may change the confidence.

The importance of the individual factors may heavily depend on the research question at hand.

A researcher having metabolomics data available may now run over-representation analyses as shown in Fig. 4. The interpretation of the results can now happen in the context of the respective decomposition method, e. g. regarding its underlying paradigm: While we see that the spatial compartment decomposition in Fig. 4(c) shows no particular location having significant metabolite concentrations, the functional KEGG decomposition in Fig. 4(a) shows several functional units having significant metabolite concentrations.

Some methods derive a large default subsystem. One could apply another decomposition method in order to split the default subsystem. In a similar fashion, the refinement or coarsening of existing decompositions by respective algorithms may be of interest. These aspects are currently part of our research and will become part of a future release of LMME.

6 Conclusion

We presented a novel method for the visual exploration of large metabolic models based on decomposition, and an implementation that provides automatic layout, several decomposition methods and over-representation

analysis. Our approach is implemented in LMME, an add-on for the VANTED framework and publicly available as open-source software.

With the development of our method, we address the need for exploration approaches that facilitate the understanding of large metabolic models, containing thousands of metabolites and reactions. By allowing researchers to investigate different decompositions of a model, we hope to facilitate the understanding of the overall biochemical mechanisms and structures present in the system at hand.

We have pointed out the main workflow of our corresponding tool LMME, explained the available decomposition and layout methods, and have shown its application to a large model of *Pseudomonas Aeruginosa* (Zhu *et al.*, 2018).

A large amount of decomposition methods can be found in the literature, and we provide a basic subset that will cover a broad range of use cases and can be directly employed by the users of LMME. However, by providing the source code of the tool and the corresponding API, we hope to find contributors in the community to have a steadily growing set of available decomposition methods. We also think that LMME can serve as a test bed during the development of new methods. We also plan to extend the range of available decomposition methods ourselves in the future.

Data availability

The source code of LMME is available from GitHub, at `https://github.com/LSI-UniKonstanz/lmme`.

The *iPAO1* model that was used to demonstrate LMME can be downloaded from the LMME webpage at `https://www.cls.uni-konstanz.de/software/lmme/getting-started/`.

The metabolomics dataset underlying the use case was provided by Mohd Hafidz Mahamad Maifiah, Yan Zhu and Jian Li by permission and has not yet been published. It will be shared on reasonable request to the corresponding author with permission of Mohd Hafidz Mahamad Maifiah, Yan Zhu and Jian Li.

Funding

This work has been supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project-ID 251654672 – TRR 161 and the National Health and Medical Research Council (NHMRC) – Project-ID APP1127948. J. L. is an Australian NHMRC Principal Research Fellow.

References

Becker, M. Y. and Rojas, I. (2001). A graph layout algorithm for drawing metabolic pathways. *Bioinformatics*, **17**(5), 461–467.

Benjamini, Y. and Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, **57**(1), 289–300.

Brunk, E. *et al.* (2018). Recon3d enables a three-dimensional view of gene variation in human metabolism. *Nature Biotechnology*, **36**, 272–281.

Büchel, F. *et al.* (2013). Path2models: large-scale generation of computational models from biochemical pathway maps. *BMC Systems Biology*, **7**, 116.

Chung, H.-J. *et al.* (2004). ArrayXPath: mapping and visualizing microarray gene-expression data with integrated biological pathway resources using Scalable Vector Graphics. *Nucleic Acids Research*, **32**(suppl_2), W460–W464.

Cottret, L. *et al.* (2018). MetExplore: collaborative edition and exploration of metabolic networks. *Nucleic Acids Research*, **46**(W1), W495–W502.

Creek, D. J. *et al.* (2012). IDEOM: an Excel interface for analysis of LC–MS-based metabolomics data. *Bioinformatics*, **28**(7), 1048–1049.

Czauderna, T. *et al.* (2010). Editing, validating, and translating of SBGN maps. *Bioinformatics*, **26**(18), 2340–2341.

Czauderna, T. *et al.* (2013). Conversion of kegg metabolic pathways to SBGN maps including automatic layout. *BMC Bioinformatics*, **14**, 250.

Darzi, Y. *et al.* (2018). iPath3.0: interactive pathways explorer v3. *Nucleic Acids Research*, **46**(W1), W510–W513.

Droste, P. *et al.* (2011). Visualizing multi-omics data in metabolic networks with the software omix—a case study. *Biosystems*, **105**(2), 154–161.

Edwards, J. S. and Palsson, B. O. (1999). Systems properties of the *haemophilus influenzae* Rd metabolic genotype. *The Journal of Biological Chemistry*, **274**, 17410–17416.

Funahashi, A. *et al.* (2008). CellDesigner 3.5: A versatile modeling tool for biochemical networks. *Proceedings of the IEEE*, **96**(8), 1254–1265.

Gehlenborg, N. *et al.* (2010). Visualization of omics data for systems biology. *Nature Methods*, **7**(3), S56–S68.

Granger, B. R. *et al.* (2016). Visualization of metabolic interaction networks in microbial communities using visant 5.0. *PLoS computational biology*, **12**(4), e10048750.

Gu, C. *et al.* (2019). Current status and applications of genome-scale metabolic models. *Genome Biology*, **20**(1), 121.

Han, M.-L. *et al.* (2019). Comparative metabolomics and transcriptomics reveal multiple pathways associated with polymyxin killing in pseudomonas aeruginosa. *mSystems*, **4**(1).

Henry, C. S. *et al.* (2010). High-throughput generation, optimization and analysis of genome-scale metabolic models. *Nature Biotechnology*, **28**, 977–982.

Holme, P. *et al.* (2003). Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, **19**(4), 532–538.

Hucka, M. *et al.* (2003). The Systems Biology Markup Language (SBML): A medium for representation and exchange of biochemical network models. *Bioinformatics*, **19**(4), 524–531.

Hucka, M. *et al.* (2019). The systems biology markup language (sbml): Language specification for level 3 version 2 core release 2. *Journal of Integrative Bioinformatics*, **16**(2), 20190021.

Huss, M. and Holme, P. (2007). Currency and commodity metabolites: their identification and relation to the modularity of metabolic networks. *IET Systems Biology*, **1**(5), 280–285.

Junker, A. *et al.* (2012). Creating interactive, web-based and data-enriched maps using the Systems Biology Graphical Notation. *Nature Protocols*, **7**(3), 579–593.

Junker, B. H. *et al.* (2006). VANTED: A system for advanced data analysis and visualization in the context of biological networks. *BMC Bioinformatics*, **7**, 109.

Jusufi, I. *et al.* (2012). Guiding the interactive exploration of metabolic pathway interconnections. *Information Visualization*, **11**(2), 136–150.

Kanehisa, M. *et al.* (2012). KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic Acids Research*, **40**(Database issue), D109–D114.

Karp, P. D. and Paley, S. M. (1994). Automated drawing of metabolic pathways. In H. Lim, C. Cantor, and R. Bobbins, editors, *Intl. Conf. Bioinf. Genome Res.*, pages 225–238.

Kerren, A. *et al.* (2017). Biovis explorer: A visual guide for biological data visualization techniques. *PLoS One*, **12**(11), e0187341.

Khatri, P. *et al.* (2012). Ten years of pathway analysis: current approaches and outstanding challenges. *PLoS Computational Biology*, **8**(2), e1002375.

Kim, J. *et al.* (2020). netGO: R-Shiny package for network-integrated pathway enrichment analysis. *Bioinformatics*, **36**(10), 3283–3285.

King, Z. A. *et al.* (2015a). BiGG Models: A platform for integrating, standardizing and sharing genome-scale models. *Nucleic Acids*

Research, **44**(D1), D515–D522.

King, Z. A. *et al.* (2015b). Escher: a web application for building, sharing, and embedding data-rich visualizations of biological pathways. *PLoS Computational Biology*, **11**(8).

Klukas, C. and Schreiber, F. (2007). Dynamic exploration and editing of KEGG pathway diagrams. *Bioinformatics*, **23**(3), 344–350.

Kohlbacher, O. *et al.* (2014). Multivariate networks in the life sciences. In *Multivariate network visualization*, pages 61–73. Springer.

Kutmon, M. *et al.* (2015). PathVisio 3: An extendable pathway analysis toolbox. *PLoS Computational Biology*, **11**(2), 1–13.

Le Novère, N. *et al.* (2009). The Systems Biology Graphical Notation. *Nature Biotechnology*, **27**, 735–741.

Ma, H.-W. *et al.* (2004). Decomposition of metabolic network into functional modules based on the global connectivity structure of reaction graph. *Bioinformatics*, **20**(12), 1870–1876.

Mahamad Maifiah, M. H. (2017). *Deciphering the modes of action of polymyxins and the synergistic combinations against multidrug-resistant Gram-negative bacteria: a systems pharmacology approach*. Ph.D. thesis, Monash University.

Malik-Sheriff, R. S. *et al.* (2019). BioModels—15 years of sharing computational models in life science. *Nucleic Acids Research*, **48**(D1), D407–D415.

Marco-Ramell, A. *et al.* (2018). Evaluation and comparison of bioinformatic tools for the enrichment analysis of metabolomics data. *BMC Bioinformatics*, **19**(1), 1.

Martyushenko, N. and Almaas, E. (2019). Modelexplorer - software for visual inspection and inconsistency correction of genome-scale metabolic reconstructions. *BMC Bioinformatics*, **20**, 56.

Mi, H. *et al.* (2015). Systems Biology Graphical Notation: Activity Flow language level 1 version 1.2. *Journal of Integrative Bioinformatics*, **12**(2), 265.

Papin, J. A. *et al.* (2004). Hierarchical thinking in network biology: the unbiased modularization of biochemical networks. *Trends in Biochemical Sciences*, **29**(12), 641 – 647.

Pienta, R. *et al.* (2015). Scalable graph exploration and visualization: Sensemaking challenges and opportunities. In *2015 International Conference on Big Data and Smart Computing (BIGCOMP)*, pages 271–278.

Rezvan, A. and Eslahchi, C. (2017). Comparison of different approaches for identifying subnetworks in metabolic networks. *Journal of Bioinformatics and Computational Biology*, **15**(06), 1750025. PMID: 29187029.

Robinson, J. L. *et al.* (2020). An atlas of human metabolism. *Science Signaling*, **13**(624).

Rohn, H. *et al.* (2012). VANTED v2: a framework for systems biology applications. *BMC Systems Biology*, **6**, 139.

Rougny, A. *et al.* (2019). Systems Biology Graphical Notation: Process Description language level 1 version 2.0. *Journal of Integrative Bioinformatics*, **16**(2), 20190022.

Scharm, M. *et al.* (2020). GEMtractor: extracting views into genome-scale metabolic models. *Bioinformatics*, **36**(10), 3281–3282.

Schreiber, F. (2002). High quality visualization of biochemical pathways in BioPath. *In Silico Biology*, **2**(2), 59–73.

Schuster, S. *et al.* (2002). Exploring the pathway structure of metabolism: decomposition into subnetworks and application to *Mycoplasma pneumoniae*. *Bioinformatics*, **18**(2), 351–361.

Shannon, P. *et al.* (2003). Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Research*, **13**(11), 2498–2504.

Singh, D. and Lercher, M. J. (2020). Network reduction methods for genome-scale metabolic models. *Cellular and Molecular Life Sciences*, **77**, 481–488.

Slenter, D. N. *et al.* (2017). WikiPathways: a multifaceted pathway database bridging metabolomics to other omics research. *Nucleic Acids Research*, **46**(D1), D661–D667.

Vogt, T. *et al.* (2013). Translation of SBGN maps: Process Description to Activity Flow. *BMC Systems Biology*, **7**, 115.

Xia, J. and Wishart, D. S. (2010). MSEA: a web-based tool to identify biologically meaningful patterns in quantitative metabolomic data. *Nucleic Acids Research*, **38**(suppl_2), W71–W77.

Xia, J. *et al.* (2015). MetaboAnalyst 3.0–making metabolomics more meaningful. *Nucleic Acids Research*, **43**(W1), W251–W257.

Yoon, J. *et al.* (2007). Modular decomposition of metabolic reaction networks based on flux analysis and pathway projection. *Bioinformatics*, **23**(18), 2433–2440.

Zhu, Y. *et al.* (2018). Genome-scale metabolic modeling of responses to polymyxins in Pseudomonas aeruginosa. *GigaScience*, **7**(4). giy021.