RESEARCH

Identifying Frequent Patterns in Biochemical Reaction Networks – a Workflow

Fabienne Lambusch[1], Dagmar Waltemath[2], Olaf Wolkenhauer[2,3], Kurt Sandkuhl[1], Christian Rosenke[4] and Ron Henkel[5]

Abstract

Background: Computational models in biology encode molecular and cell biological processes. These models often can be represented as biochemical reaction networks. Studying such networks, one is mostly interested in systems that share similar reactions and mechanisms. Typical goals of an investigation include understanding of the parts of a model, identification of reoccurring patterns, and recognition of biologically relevant motifs. The large number and size of available models, however, require automated methods to support researchers in achieving their goals. Specifically for the problem of finding patterns in large networks only partial solutions exist.

Results: We propose a workflow that identifies frequent structural patterns in biochemical reaction networks encoded in the Systems Biology Markup Language. The workflow utilises a subgraph mining algorithm to detect frequent network patterns. Once patterns are identified, the textual pattern description can automatically be converted into a graphical representation. Furthermore, information about the distribution of patterns among the selected set of models can be retrieved. The workflow was validated with 575 models from the curated branch of BioModels. In this paper, we highlight interesting and frequent structural patterns. Further, we provide exemplary patterns that incorporate terms from the Systems Biology Ontology. Our workflow can be applied to a custom set of models or to models already existing in our graph database MaSyMoS.

Conclusions: The occurrences of frequent patterns may give insight into the encoding of central biological processes, evaluate postulated biological motifs, or serve as a similarity measure for models that share common structures.

Availability: https://github.com/FabienneL/BioNet-Mining
Contact: fabienne.lambusch@uni-rostock.de

Keywords: workflow; pattern detection; SBML; reaction networks

Background

Modeling is an integral part of computational biology [1]. Its increasing impact is reflected in the rapidly growing number and complexity of computational models [2, 3]. Such models encode a wide range of biological processes (including cell cycle processes, apoptosis, mitogen-activated protein kinase and many more [4]) and thereby enable computer-based analysis of complex biological systems. We observe that many models reassemble large biochemical reaction networks. They may have been semi-automatically generated using data driven approaches, for example, to construct models from metabolic networks [5, 6]. Models may also prove a theory or

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.1479v4 | CC BY 4.0 Open Access | rec: 30 Jul 2018, publ: 30 Jul 2018

concept, for example to mathematically describe interactions between biological entities [7] or generic oscillatory networks of transcriptional regulators [8]. Published models are often provided in standard formats such as the Systems Biology Markup Language (SBML) [9]. A resource of curated SBML models is BioModels [10]. Release 29 of this open repository contains 575 curated SBML models.

In order to reuse an SBML model, scientists require computational tools for model exploration, coupling, merging, or combination. Support is also needed during model curation, i. e., for validation and semantic annotation of models [3]. As models evolve over time, management strategies must be implemented to ensure model exchangeability, stability and result validity; and to foster communication between project partners [11, 12, 13, 14]. All these tasks require means to compare the characteristics of different models to answer questions such as: “What are frequently used structures to represent biochemical processes?”; “What are characteristic patterns in the class of cell cycle models?”; “Do frequent patterns reflect well-known motifs in Systems Biology, such as the ones proposed by Tyson [15]?”; “Does the network contain cycles and how many?”. An automated retrieval of reoccurring patterns will enable new kinds of analysis. Current approaches for network analysis, however, provide key figure values for the network topology [16, 17, 18], but they do not detect actual patterns.

We present a five-step workflow for the discovery of structural patterns in biological networks: (1) import models, (2) export networks, (3) create labeled graphs from networks, (4) execute graph mining, (5) visualise and distribute pattern. The workflow implementation imports a set of SBML-encoded models in graph-representation. It then extracts all reaction networks belonging to these models. Based on the network structures converted into a standard graph format, a mining algorithm identifies frequently occurring patterns. Finally, the patterns are visualised, and their distribution among the model set is computed. We show exemplary patterns, purely structural and also incorporating SBO-annotations, which were detected in curated SBML-models by means of the proposed workflow.

Methods

Data mining is a common technique for the extraction of implicit, non-obvious information from huge data sets [19]. The mining of frequent patterns has its roots in the early 90’s, when it had been used to examine the customers’ buying behavior. Sales could be increased by detecting patterns in frequent combinations of bought products [20]. We focus on graph-based approaches in data mining, because our models are represented by reaction networks. Approaches for identifying patterns in graphs are, for example, based on set-similarity [21], hypergraph analysis [22] or require specific types of edges and vertices, e. g., the existence of taxonomic relationships [23, 24]. For this work, we chose frequent subgraph mining (FSM) [25], which addresses the problem: Given a set of graphs, find those subgraphs within the graphs that pass a given frequency threshold [26]. To decide whether a graph is embedded in another, FSM algorithms require subgraph isomorphism testing [25]. This is known as an NP-complete task. Thus, FSM techniques rely on prior knowledge, heuristics and further domain-dependent strategies to improve the performance. A variety of FSM algorithms have already been implemented [27]. It should be noted that most FSM algorithms are used in a domain-specific manner. For example, an FSM algorithm exists specifically for molecular databases with structures of atoms and bonds [28].

For our application domain, we decided to use gSpan [29]. GSpan takes a set of graphs as input, in this case a set of reaction networks, and produces all frequent connected subgraphs according to a given frequency threshold, i. e., gSpan searches for structures that occur in at least a certain number of graphs within the set. While other algorithms supply only approximate results, gSpan fulfills our requirement for exact results. [30] evaluate and compare the performance of the subgraph miners MoFa, gSpan, FFSM and Gaston. For this purpose, [30] developed a tool called the “Parallel and Sequential Mining Suite” (ParSeMiS). ParSeMiS is based on Java and implements algorithms such as gSpan, Gaston, and Dagma. In addition, [31] described a detailed approach to graph mining using the gSpan algorithm.

Current network analysis mostly focuses on network diameter and network efficiency [16], on the topological and dynamical properties that control the behavior of the network [17], or on the degree of tolerance against errors in scale-free networks [18]. These approaches provide key figure values for the network topology, but they do not detect actual patterns. On the other hand, biologists have an interest in classifying models by their function. While analysing the function of patterns requires knowledge of a domain expert, frequently occurring patterns can be determined automatically. [32] discuss the biological significance of network patterns and present several algorithms to identify such patterns. These algorithms are compared and classified. Searches for frequent patterns were already performed in the Kyoto Encyclopedia of Genes and Genomes (KEGG, [33]). [34] describe a method to compare chemical structures of the KEGG LIGAND database by identifying their common patterns. The considered chemical structures are mostly metabolic compounds. The atoms and covalent bonds are represented as graphs, where the maximum common subgraph is searched for all possible pairs of compounds. The procedure is applied to detect frequent patterns in 9383 compounds and to cluster these compounds according to their similarity. [35] propose an algorithm to discover frequent patterns within a set of metabolic pathways in the KEGG PATHWAY database. The algorithm performs frequent subgraph mining on the metabolic pathways that are represented as directed graphs. The authors show exemplary results for detected patterns. As subgraph isomorphism testing is NP-complete, the computational cost for the algorithm is reduced by utilising the sparse nature of metabolic pathways and unique node labelling.

In the field of business informatics, [36] propose a method to extract occurring subgraphs in a repository of business process graphs, compute the distance between the user’s process model and the extracted patterns, and recommend a ranked list of patterns. By remodelling the process graphs to be represented uniformly, they can even find large patterns or rather the ones only contained in a few networks.

Results

We designed a five-step workflow to retrieve frequent patterns within reaction networks of SBML-models (see Figure 1). To store and access models, our worklow utilises a graph database. Network structures are extracted to detect occurring

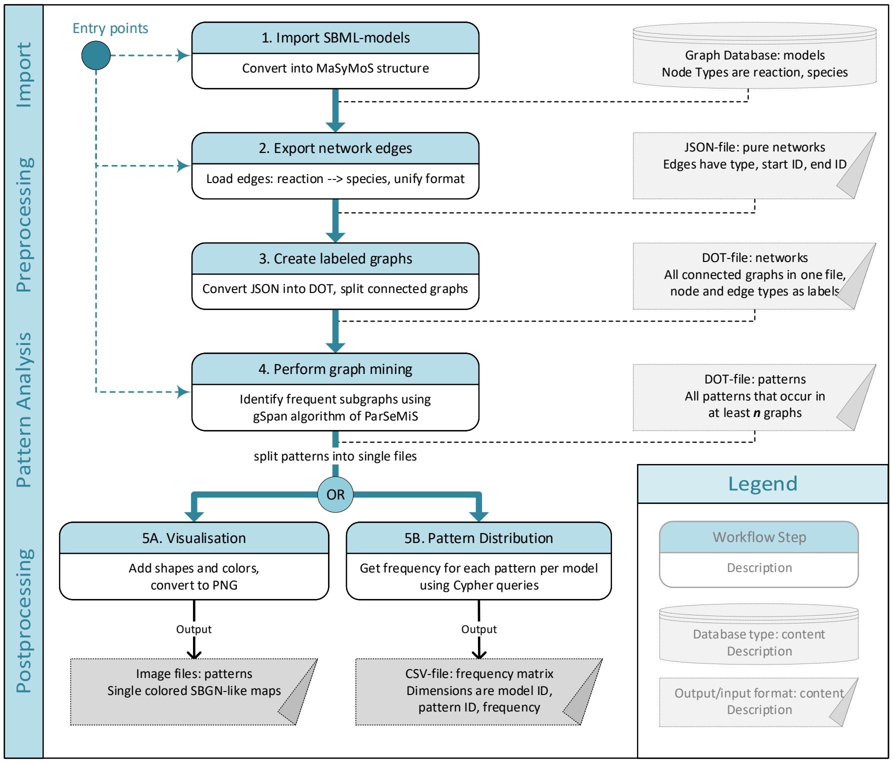


Figure 1 Pattern identification workflow. Numbered, oval boxes describe workflow steps. Rectangular boxes describe data produced by each step and taken as an input for the following step. Starting at multiple entry points is possible depending on the available data and format. The first step is to import models into the MaSyMoS graph database. From there the reaction networks are extracted and converted to a uniform structure. The converted reaction networks are used as an input for the subgraph mining step to identify patterns. Subsequently, two options to further process the pattern descriptions are available. On the one hand, images representing the patterns can be generated (output on the bottom left). On the other hand, patterns can be fed back to the database to create a feature matrix showing the distribution of identified patterns among the models (output on the bottom right).

patterns by means of the frequent subgraph mining algorithm gSpan. The generated patterns can be visualised using SBGN-compliant glyphs. Furthermore, the pattern distribution among all models can be computed. The workflow has different entry points, which can be chosen depending on the available data. Below, we explain the single steps in detail.

Step 1: Import SBML-models

The workflow may either be applied to models already existing in the graph database MaSyMoS [14] or to a custom set of models. The published instance of MaSyMos is shipped together with a database[1] containing all curated models of BioModels Release 29[2]. Additional SBML-models can be imported into a local MaSyMoS instance. In MaSyMos, the SBML-structure is mapped onto a custom graph structure which preserves network information: the species and reactions are represented by nodes and their relations are represented as edges between them. A species can take

[1] https://github.com/FabienneL/BioNet-Mining/tree/master/data

[2] ftp://ftp.ebi.ac.uk/pub/databases/biomodels/releases/2015-04-16/

the role of a reactant, modifier, or product. Relations between species and reactions are bidirectional. Of particular importance are the relations “a reaction `HAS` participants” and “a species `IS` participant” in a reaction. When applying the workflow to a custom set of models, the representation of networks must be graph-based and comply to the structure available in MaSyMoS.

Step 2: Export network edges

Using a query interface for MaSyMoS and the query language Cypher[3], our script retrieves all reaction networks of the SBML-models that are present in the database. The corresponding Cypher-query is shown in Listing 1. By adapting the script, a custom model set can be used.

Listing 1 Cypher-query to export the reaction networks of SBML-models stored in the MaSyMoS database. All structures connecting reactions and species are exported. The output is a set of 3-tuples consisting of the reaction's identifier the role type and the species' identifier.

```
MATCH (reaction:SBML_REACTION)-[edge]->(species:SBML_SPECIES)
RETURN ID(reaction),TYPE(edge),ID(species)
```

We only query the nodes with their edges and do not incorporate further information, such as the associated model, publication, etc. For this reason, unconnected reaction networks belonging to the same model will not further be associated with each other. The query result is a set of typed directed edges with a reaction as start node and a species as end node. Each result entry is a 3-tuple containing a reaction ID, role type (reactant, modifier, product), and a species ID. The resulting set of tuples is provided as JSON output. An example is shown in Listing 2.

Listing 2 Exemplary output (JSON) for the query in Listing 1. It contains a table with columns defining a reaction's identifier an edge type and a species' identifier. Consequently the table entries are 3-tuples each representing an edge with start node end node and role type. In this example the IDs 100233 and 100229 represents reactions cdc2k dephosphorylation and cdc2k phosphorylation. ID 100186 is the species cdc2k.

```
{
  ‘columns’ : [ ‘ID(reaction)’, ‘TYPE(edge)’, ‘ID(species)’ ],
  ‘data’ :
     [
        [ 100233, ‘HAS_PRODUCT’, 100186 ],
        [ 100229, ‘HAS_REACTANT’, 100186 ],
        ...
     ]
}
```

Step 3: Create labeled graphs

For later analysis, the JSON-file must first be converted into a graph representation format. We provide this information in the graph description language DOT [4]. The associated framework Graphviz [5] offers manifold opportunities to process graphs

[3] https://neo4j.com/developer/cypher-query-language/

[4] http://www.graphviz.org/content/DOT-language

[5] http://www.graphviz.org/

by providing a collection of tools using DOT-files as input. We use a few of the tools in later steps of the workflow.

To translate the JSON-file into DOT-format, we convert each 3-tuple into a graph with a start node, an end node and an edge between them. The start and end node are characterised by their unique identifier and their node type (species or reaction, stored in a DOT-label). An edge is defined by the identifiers of its start and end node, its direction and its type (representing the role, stored in a DOT-label). As it is more natural for the order of nodes in the visualisation that reactants and modifiers are ingoing for a reaction and products outgoing, the edge directions are adjusted. Thus, possible edge labels are IS_REACTANT, IS_MODIFIER, or HAS_PRODUCT. An example for the entries resulting from the two edges of the exemplary JSON-file converted into DOT-format is shown in Listing 3.

Listing 3 Exemplary DOT-format after converting the output (JSON) shown in Listing 2. This transitory format defines one digraph (directed graph) containing all exported nodes and edges. For each 3-tuple from Listing 2 two nodes and one edge is defined. Consequently the created digraph may contain nodes multiple times (shown here the node with ID 100186) because one node can be part of several edges. In this example the IDs 100233 and 100229 represents reactions cdc2k dephosphorylation and cdc2k phosphorylation. ID 100186 is the species cdc2k.

```
digraph {
        100233   [label=SBML_Reaction];
        100186   [label=SBML_Species];
        100233 -> 100186 [label=HAS_PRODUCT]

        100229   [label=SBML_Reaction];
        100186    [label=SBML_Species];
        100186 -> 100229 [label=IS_REACTANT];

        ...
 }
```

The resulting DOT-file defines one graph with nodes and edges from all reaction networks. It should be noted that the file my contain nodes multiple times, because we create one entry for a node each time it occurs as a start or end node in an edge. Consequently, we bundle all connected nodes with their corresponding edges as one graph each and eliminate redundant nodes. This is possible by means of a Graphviz tool to split a graph into its connected components. Then, each connected reaction network represents its own graph in the new DOT-file and has no redundant nodes anymore. As an SBML-model can contain entities that are not explictly connected (e.g. only connected by rules), it is possible to have more graphs defined in the DOT-file than models used as input for the workflow. As mentioned before, unconnected reaction networks belonging to the same model will not further be associated with each other. Listing 4 shows the final DOT-format for our example.

Listing 4 Exemplary DOT-format neccessary for the subgraph mining process created by splitting the digraph from Listing 3. Here each connected reaction network is represented by one digraph and has no redundant nodes. In this example the IDs 100233 and 100229 represents reactions cdc2k dephosphorylation and cdc2k phosphorylation. ID 100186 is the species cdc2k.

```
digraph {
 100233   [label=SBML_Reaction];
 100186   [label=SBML_Species];
 100233 -> 100186 [label=HAS_PRODUCT];
 100229   [label=SBML_Reaction];
 100186 -> 100229 [label=IS_REACTANT];
 ...
 }
digraph {
        ...
 }
...
```

Step 4: Perform graph mining

The created DOT-file is the input for the graph mining and the basis for finding frequent patterns in the set of reaction networks. The frequency of patterns is equal to the number of reaction networks, in which a pattern occurs. Each pattern is thus counted only once for each network, even if it occurs multiple times in a model's reaction network. We use the implementation of the gSpan algorithm in the software tool ParSeMiS to calculate frequencies: Given the user-specified values *min* (minimum frequency) and *max* (maximum frequency), the mining finds all subgraphs that occur in at least *min* and at most *max* of the graphs. We call these subgraphs frequent patterns. It does not matter for the algorithm, how often a pattern occurs within one graph, only the number of graphs is relevant. Consequently, the frequencies are values between one and the total number of graphs in the DOT-file. As already mentioned, one model may have several unconnected reaction graphs. Therefore, the number of defined graphs can be higher than the number of models used as input.

The result of the subgraph mining is one DOT-file containing all subgraphs having a frequency within the given interval. For each pattern in the DOT-file the frequency of its occurrence and the names of the corresponding models are attached as a comment. An exemplary output is shown in Listing 5.

Listing 5 Exemplary pattern mining results (DOT-format). The output contains for each detected pattern one directed graph. The digraph numbering denoted by '...' can be discarded. First all nodes are defined starting with 'Node_0'. Second the edges are defined. Following a digraph's definition a comment (introduced by #) contains the number of graphs in which the described pattern occurs. The following square brackets can also be discarded.

```
digraph '560' {
 Node_0 [label='SBML_REACTION'];
 Node_1 [label='SBML_SPECIES'];
 Node_2 [label='SBML_REACTION'];
 Node_3 [label='SBML_REACTION'];
 Node_4 [label='SBML_SPECIES'];
 Node_0 -> Node_1 [label='HAS_PRODUCT'];
 Node_1 -> Node_2 [label='IS_REACTANT'];
 Node_1 -> Node_3 [label='IS_REACTANT'];
 Node_4 -> Node_0 [label='IS_REACTANT'];
}# => 398[ , , ... ,]
digraph '560' {
        ...
 }# => 436[ , , ... ,]
...
```

To find those subgraphs within the graphs that pass a given frequency threshold requires subgraph isomorphism testing [25]. Because this is known as an NP-complete task [26], the minimum frequency must be chosen carefully to obtain results. If the minimum frequency is set too low, the computation will not succeed due to capacity limitations (memory or time).

Step 5: Pattern post-processing

The generated patterns may be used in various ways. Here, we illustrate two possible options to further process them: the first option is the visualisation; the second option is the computation of frequencies for each pattern per model. In both cases, the DOT-file is split into multiple DOT-files each containing one pattern. The name of a DOT-file comprises the pattern's frequency and an identifier. The identifier is used to distinguish between several patterns occurring with the same frequency.

Step 5A: Visualisation

The visualisation follows the standardized Systems Biology Graphical Notation (SBGN) [37]. Node and edge labels are expressed by the visualised shape suggested by SBGN. Furthermore, the contour, fill color and size of nodes, and the stroke width, direction, arrowhead and size of edges are set. Consequently, textual display of node and edge labels is disregarded. For each DOT-file an image-file is rendered. The standard image-format is PNG, but other formats such as PDF are supported by the DOT framework.

Step 5B: Pattern Distribution

To compute the frequencies of patterns per model, a Cypher query is generated for each DOT-file. An example is shown in Listing 6. The query describes the graph representing the pattern. Further, a restriction is added that nodes are not allowed to be equal. The output is a JSON-file that lists all distinct model IDs, the model names that contain the pattern, and how often a pattern is present in each of those models. Subsequently, the queries are executed on the MaSyMoS database and the results stored as JSON-files. All JSON-files are then processed to create a CSV-file representing a frequency matrix. Here, the first two columns specify the model. The following columns define the patterns. Each row contains a model ID in the first column, a model name in the second column, and the frequency of each pattern in the following columns. Thus, each row can be seen as a feature vector for one model.

Listing 6 Exemplary Cypher-code to query MaSyMoS for the distribution of a certain pattern. The examplary pattern here represents a chain with two reaction nodes and one species node. The species takes a role as product in the first reaction and a role as reactant in the second reaction. Furthermore it is defined that nodes are not allowed to be equal. The result is a set of 3-tuples each containing a model identifier a model name (stored as attribute in the associated document) and the number of occurences of the pattern.

```
{
 ‘query’:
 ‘MATCH (m:SBML_MODEL)-->(d:DOCUMENT), m-[HAS_REACTION]->Node_0,
  Node_0-[:HAS_PRODUCT]->Node_1, Node_1-[:IS_REACTANT]->Node_2
  WHERE Node_0<>Node_1 AND Node_0<>Node_2 AND Node_1<>Node_2
```

```
   RETURN DISTINCT ID(m), d.FILENAME, COUNT(Node_0)
   AS sum ORDER BY sum DESC',
  'params':{}
}
```

Exemplary Application

Using the aforementioned combination of tools and methods, we exemplarily analyzed two data sets on a cluster node (180GB RAM, 16 Intel(R) Xeon(R) CPU X5650 @ 2.67GHz). Graph-pattern identification is an NP-complete task, thus memory and CPU are the limiting constraints.

Data Set

For the pattern detection, we incorporated publicly available models from BioModels. The stored reaction networks are encoded in SBML. BioModels contains two types of models: curated and non-curated. We here chose only models from the curated branch as those models are ensured to accurately represent the work described in the reference publication. Furthermore, curated models are syntactically and semantically validated and annotated with ontology terms, and they comply with the MIRIAM standard [38]. Specifically, we analyzed SBML-models from two different releases of BioModels. Release 1 (in the following referred to as R1) is the first release of the repository. It contains 30 curated models. Release 29 (in the following referred to as R29) is one of the latest releases. It contains 575 curated models. We chose these two releases to take the evolution of BioModels into account.

As we decided to perform subgraph analysis with an FSM algorithm, we translated the biological reaction network into a graph representation using the MaSyMoS database. For the reaction network, the MaSyMoS graph structure distinguishes two types of nodes (i. e.,labeled *species* and *reaction*) and three types of edges (labeled *is_reactant*, *has_product*, and *is_modifier*).

Quantitative Analysis

First, we performed a key figure analysis to calculate the quantities of node types and edges in the networks. In our data set, 557 out of 575 models in R29 contain species, and 499 models contain reactions. The remaining models only define rules, but do not form a network. The data set contains a total of 18852 reaction nodes and 16843 species nodes.

Data set R1 contains only 30 curated models. These models contain a total of 736 reactions and 425 species. The big difference in numbers between R1 and R29 are due to the rapid growth of models, as previously reported [2]. On average, a model from R29 has 30.2 species and 37.7 reactions. In R1, a model has 14.6 species and 25.4 reactions on average. For both datasets most models contain three up to eleven species. In addition, most models have three up to twelve reactions. However, there are a few outliers with more than 100 reactions and species. Figure 2 shows the correlation between species (and their respective role as reactants, products and modifier) and reactions. As the figure states, most reactions have two or three participating species. The most frequently encoded reaction has two species as reactants and one species as product. The second most frequently encoded reaction has one species as reactant and one as product.

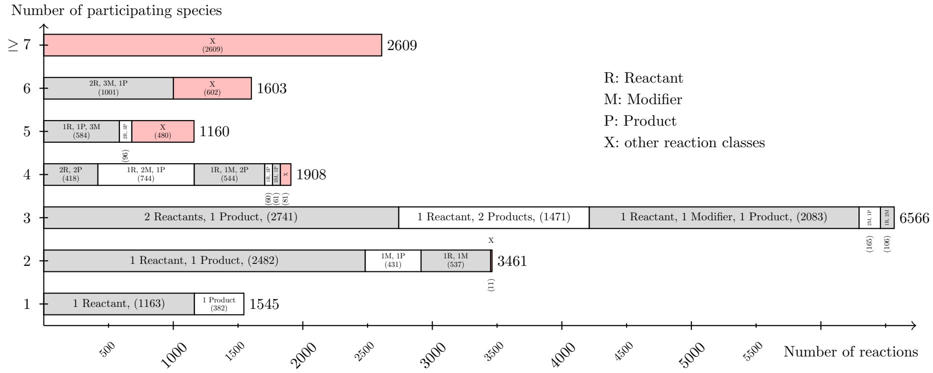


Figure 2 Listing of the node degree for reaction nodes in the data set R29 of curated models in BioModels Database. For each number of species (from 1 to 6, and more then 6) participating in a reaction, the figure lists the number of reaction nodes identified with a particular combination of its species relations (reaction class). The figure sums up smaller reaction classes displayed by X. It becomes obvious that most reactions have two or three participating species.

Exemplary Patterns

We identified a subset of patterns shared by at least a certain number of models. For data set R29, we were able to identify 37 patterns in total. Each identified pattern is shared by at least 350 out of 575 models. For the much smaller data set R1, we identified 190 patterns. Here, each pattern is shared by at least 20 out of 30 models. For R29, the identified patterns contain between one and six entities (species or reactions) whereas patterns for R1 contain between one and eleven entities. It was not possible to scale down the number of models that share a pattern due to memory limitations.

Common types of reactions

From the quantitative analysis and the statistics shown in Figure 2, we expected to see patterns having one reaction and three species (participating as product, reactant or modifier). Surprisingly, the pattern identification shows that no such patterns are shared by at least 350 models in R29 or by at least 20 models in R1, respectively. Subsequently, we searched for expected structures having one reaction and three species in the MaSyMoS database. The specific combination of two reactants as a reaction's input and one product as a reaction's output only occurs in 314 models, despite being the most frequently encoded reaction class according to Figure 2. Same holds for all other possible reaction classes with three species for R29 and R1, respectively. One can conclude that such types of reactions are often used, but are not equally distributed across models.

Species as a reaction modifier:

Generally, species in R29 most often take part in a reaction as a modifier (33209 times), and less as a product (23630) or reactant (25595). However, only four out of 37 retrieved patterns (R29) contain species that act as a modifier. One of those four patterns is shown in Figure 3. A further investigation reveals the unequal distribution of modifiers among the models. Ten models together count for 20620 modifier usages. Among those ten models, five models are derivations of the aforementioned semi-automatically created models of metabolic networks [6].

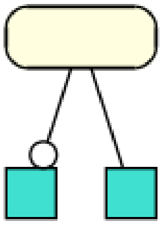


Figure 3 Species as modifier: This pattern occurred in 351 models of data set R29 and shows a species taking part in a reaction as a modifier and a reactant

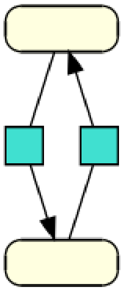
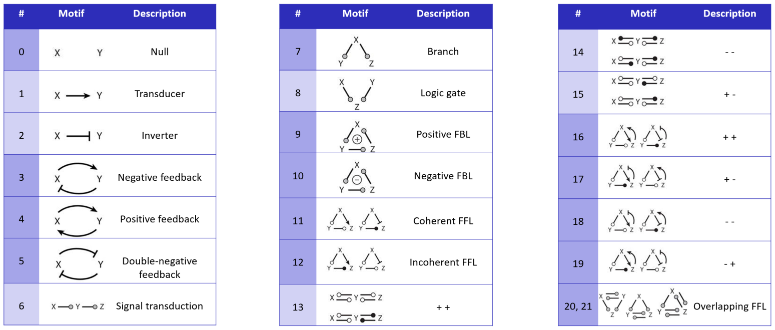


Figure 4 Cycle: The smallest biologically meaningful cycle (2 species and 2 reactions). It is contained in 330 models of data set R29 and in 25 models of data set R1.



#	Motif	Description
0	X Y	Null
1	X → Y	Transducer
2	X ⊣ Y	Inverter
3	X Y	Negative feedback
4	X Y	Positive feedback
5	X Y	Double-negative feedback
6	X —○ Y —○ Z	Signal transduction

#	Motif	Description
7	X Y Z	Branch
8	X Y Z	Logic gate
9	X ⊕ Y Z	Positive FBL
10	X ⊖ Y Z	Negative FBL
11	X Y Z X Y Z	Coherent FFL
12	X Y Z X Y Z	Incoherent FFL
13	X Y Z X Y Z	+ +

#	Motif	Description
14	X Y Z X Y Z	- -
15	X Y Z X Y Z	+ -
16	X Y Z X Y Z	+ +
17	X Y Z X Y Z	+ -
18	X Y Z X Y Z	- -
19	X Y Z X Y Z	- +
20, 21	X Y Z X Y Z X Y Z	Overlapping FFL

Figure 5 Functional motifs postulated by [15]: A gray circle in a motif indicates an interaction that may be either + or -. All white circles in a motif must have the same sign, either + or -, and they must be of opposite sign to any black circle in the same motif. We grouped this motifs by structure. For example, motifs 3-5 are grouped as they are all cycles of two species and two reactions. An analogous group is built by motifs 9-12. The groups are depicted by alternating colors.

Motifs

Biologists have an interest in classifying models by their function. Tyson and Novak [15], for example, were interested in the mechanisms of information processing. They showed that complex networks could be decomposed into simple patterns, each fulfilling specific functions within a cell. These patterns were postulated as common motifs in biochemical reaction networks. It remains an open question how and how frequently these motifs are encoded in a model. Figure 5 shows the network motifs that were postulated by [15]. The structure of motifs 3-5 can be represented as a graph with two species and two reactions forming a cycle. Such motifs can encode, for example, the production and degradation of a protein, or positive or negative feedbacks. While analysing the function of patterns requires knowledge of a domain expert, frequently occurring patterns can be determined automatically. Using our workflow, we identified one pattern that represents the structure of motifs 3-5; it occurs in 26 models of R1 (shown in Figure 4). However, this pattern is not among the 37 patterns retrieved using dataset R29. A subsequent query in MaSyMoS, for the exact pattern, reveals that it the structure indeed only contained in 342 models. Surprisingly, the query retrieved more than 45,000 occurrences of this cycle in R29. To investigate further, we ordered the results by model. Again, the answer is the distribution of the pattern: two models by [5] (generated semi-automatically) count for approximately 10,000 cycles each. Together with our observations regarding the usage of species as modifiers in reactions, we can assume that semi-automatically generated models have a distinguishable network structure. BioModels contains two prominent examples of such models [5, 6].

Pattern identification (semantics-aware)

Our workflow currently does not consider semantic annotations and thus cannot provide information about the intended semantics of reactions and species. Consequently, we cannot distinguish all of the postulated motifs. For example, the pattern describing a simple cycle (*cmp.* Figure 4) could be corresponding to motif 3, motif 4 or motif 5.

To regard semantics, we adapted Step 2 of the described workflow. The network extraction was refined to additionally receive for the species and reactions the SBO-annotation [39] of these entities. These annotations reflect the biological role of each species and reaction. Two downsides of this approach have to be considered: First, only 116 out of 575 (R29) models have reaction networks annotated with SBO terms. Second, as [40] states, the specificity of SBO-annotations varies among models. Taken together, the remaining reaction networks are less complex, allowing us to retrieve 176 patterns contained by at least 12 out of 116 valid models. Structure-wise, Figure 6 and 7 are equivalent to Figure 4, but they now include semantics, i. e. the role of each participating species and reaction. Figure 7 is an identified pattern that describes a biochemical reaction (SBO:176) between simple chemicals (SBO:247) and Figure 7 describes the phosphorylation (SBO:216) and de-phosphorylation (SBO:330) of a polypeptide chain (SBO:252).

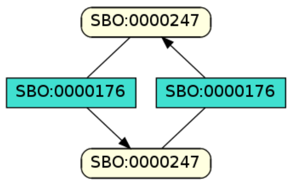


Figure 6 Simple chemical and biochemical reaction.

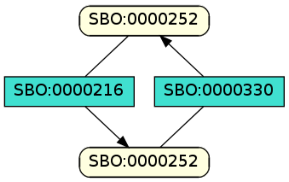


Figure 7 Phosphorylation and de-phosphorylation of a polypeptide chain.

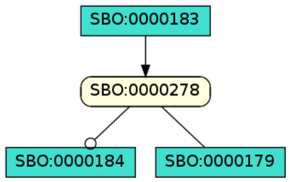


Figure 8 Transcription, translation and degradation of messenger RNA.

A brief analysis of all retrieved SBO-based patterns reveals structures similar to Figure 9. In fact, all but one pattern with at least four entities contain a combination of simple chemical (SBO:247), biochemical reaction (SBO:176), or phosphorylation (SBO:216), de-phosphorylation (SBO:330) and polypeptide chain (SBO:216). The one outsider pattern encodes the transcription, translation and degradation of messenger RNA (*cmp.* Figure 8).

Feature matrix

Current approaches for model clustering only incorporate semantic annotation and meta-information [40, 41]. Our work is a first step towards creating structural similarity measures for biological models. We hypothesize that these similarity scores

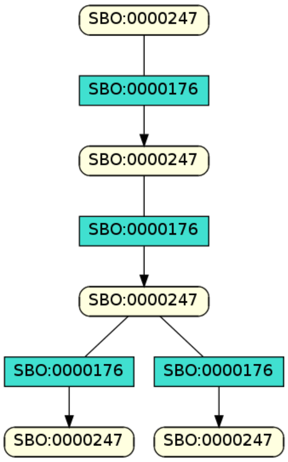


Figure 9 Exemplary pattern, a combination of simple chemical and biochemical reaction.

can help to distinguish models, for example, to classify them by a certain modeling technique (theoretical, data driven, or hybrid). Having identified patterns at hand, it is easy to generate a vector for each model holding the number of occurrences for each pattern within a model. Using the approach of term frequency and inverse document frequency with a vector space model, well studied in the field of information retrieval, one can draw conclusions about the similarity of models based on shared pattern. However, it is not feasible to use all identified patterns for such a model comparison. Instead, patterns should be weighted according to their biological significance. Also it seems fruitful to incorporate information about the uniqueness of a pattern, i.e. does the pattern contain other identified patterns itself. Such an analysis would lead to an approach similar to eTVSM [42].

Discussion

The presented workflow can be used to test hypotheses about reoccurring patterns in domain-specific model sets. It can furthermore help to calculate structure-based similarities of model (see [43] for a discussion of possible measures). Based on the calculated similarities, models can then be classified. Finally, the evolution of a model can be studied through the evolution of the network. Here, our workflow can help to identify stable regions in the network.

The visualisation of identified patterns follows the SBGN standard. By providing a standards-compliant visualisation of the detected pattern, they are more easily comparable to other works, for example to the already existing SBGN bricks [44].

The workflow can be adapted and extended. It is possible to adapt the preprocessing steps to enable pattern detection in CellML-encoded models, or even in other model representation formats. The preprocessing could also be adapted to better incorporate semantic information, such as the knowledge about mathematical concepts encoded in the Systems Biology Ontology (SBO) [39].

Current approaches for clustering of a model set could be extended towards structural approaches. The consideratoin of patterns will further enable search for models that share similar structures, improve the mapping of similar models onto each other [45], and lead to recommender systems that support the modeling process. In addition with with already existing similarity measures [43], this work will impact the reuse and reproducibility of scientific modeling results. A number of approaches exist to compare models based on the encoding format, the XML tags, or semantic annotations. It is, for example, interesting to study models regarding function, structure and behavior [46]; regarding their temporal evolution [47, 48]; or regarding their dynamics [49]. In this paper, we propose a first step towards to a new structural analysis by providing a workflow to retrieve frequent patterns.

In the future, we need to incorporate better information about the role of a reaction (e. g. promoter or inhibitor). The use of annotations, specifically from SBO, will enable us to identify motifs more precisely. SBO provides terms for the functional role of a species or reaction but is to broad. For example, a species can simply be annotated as a “simple chemical” (SBO:247). Most species and reactions in our data sets contain such annotations, but some networks are still not annotated. The consideration of annotation will also lower the computational costs for the search for sub-models, because valuable semantic knowledge can be incorporated to reduce the number of potential alignments.

Conclusion

The increasing amount of published models and the growing size of encoded reaction networks demand automated methods for model analysis. Pattern detection in biological networks, being one such method, is of great scientific interest. In this paper, we present a workflow that addresses the problem of obtaining common patterns in SBML-encoded models by applying a frequent subgraph mining algorithm. Our workflow implementation loads a custom set of SBML models into a graph database and delivers information about frequent patterns in that set of models. For the pattern detection it uses a Java-based gSpan implementation. Identified patterns can be fed back into the graph database to retrieve further information, for example, about the pattern distribution. The presented workflow is openly available and can be adapted to other model encoding formats. It can also be extended to support further types of pattern analysis. When being integrated with available model repositories, information retrieved from our workflow can improve model search, comparison, and provenance.

Competing interests

The authors declare that they have no competing interests.

Author’s contributions

FL conceived and designed the experiments, performed the experiments, analyzed the data, wrote the paper, prepared figures and tables, performed the computation work.
DW analyzed the data, wrote the paper, reviewed drafts of the paper.
OW contributed to figures and reviewed drafts of the paper.
KS reviewed drafts of the paper.
CR contributed to figures and reviewed drafts of the paper.
RH conceived and designed the experiments, performed the experiments, analyzed the data, wrote the paper, prepared figures and tables, performed the computation work.

Acknowledgements

The authors like to thank Faiz Muhammad Khan for the many fruitful discussions and Martin Scharm for helping with scripting issues.

Author details

[1]Business Information Systems, University of Rostock, 18051, Rostock, Germany. [2]Systems Biology and Bioinformatics, University of Rostock, 18051 Rostock, Germany. [3]Stellenbosch Institute for Advanced Study (STIAS), Wallenberg Research Centre, Stellenbosch University, Private Bag X1, Matieland 7602 , Stellenbosch, South Africa. [4]Visual Computing and Computer Graphics, University of Rostock, 18051 Rostock, Germany. [5]Scientific Databases and Visualization, Heidelberg Institute for Theoretical Studies, Heidelberg, Germany.

References

1. Finkelstein, A., Hetherington, J., Li, L., Margoninski, O., Saffrey, P., Seymour, R., Warner, A.: Computational challenges of systems biology. Computer **37**(5), 26–33 (2004)
2. Henkel, R., Endler, L., Peters, A., Le Novère, N., Waltemath, D.: Ranked retrieval of computational biology models. BMC Bioinformatics **11**(1), 423 (2010)
3. Chelliah, V., Juty, N., Ajmera, I., Ali, R., Dumousseau, M., Glont, M., Hucka, M., Jalowicki, G., Keating, S., Knight-Schrijver, V., *et al.*: Biomodels: ten-year anniversary. Nucleic Acids Research **43**(D1), 542–548 (2014)
4. Juty, N., Ali, R., Glont, M., Keating, S., Rodriguez, N., Swat, M.J., Wimalaratne, S., Hermjakob, H., Le Novère, N., Laibe, C., *et al.*: Biomodels database: Content, features, functionality, and use. CPT: Pharmacometrics & Systems Pharmacology **2**(4), 1–14 (2015)
5. Stanford, N.J., Lubitz, T., Smallbone, K., Klipp, E., Mendes, P., Liebermeister, W.: Systematic construction of kinetic models from genome-scale metabolic networks. PLOS ONE **8**(11), 79195 (2013)
6. Smallbone, K., Messiha, H.L., Carroll, K.M., Winder, C.L., Malys, N., Dunn, W.B., Murabito, E., Swainston, N., Dada, J.O., Khan, F., *et al.*: A model of yeast glycolysis based on a consistent kinetic characterisation of all its enzymes. FEBS letters **587**(17), 2832–2841 (2013)
7. Tyson, J.J.: Modeling the cell division cycle: cdc2 and cyclin interactions. Proceedings of the National Academy of Sciences **88**(16), 7328–7332 (1991)
8. Elowitz, M.B., Leibler, S.: A synthetic oscillatory network of transcriptional regulators. Nature **403**(6767), 335–338 (2000)
9. Hucka, M., Bergmann, F.T., Hoops, S., Keating, S.M., Sahle, S., Schaff, J.C., Smith, L.P., Wilkinson, D.J.: The systems biology markup language (sbml): language specification for level 3 version 1 core. Nature (2010)
10. Li, C., Donizelli, M., Rodriguez, N., Dharuri, H., Endler, L., Chelliah, V., Li, L., He, E., Henry, A., Stefan, M.I., *et al.*: Biomodels database: An enhanced, curated and annotated resource for published quantitative kinetic models. BMC Systems Biology **4**(1), 92 (2010)
11. Bechhofer, S., Buchan, I., De Roure, D., Missier, P., Ainsworth, J., Bhagat, J., Couch, P., Cruickshank, D., Delderfield, M., Dunlop, I., *et al.*: Why linked data is not enough for scientists. Future Generation Computer Systems **29**(2), 599–611 (2013)
12. Waltemath, D., Henkel, R., Hälke, R., Scharm, M., Wolkenhauer, O.: Improving the reuse of computational models through version control. BIOINFORMATICS **29**(6), 742–748 (2013)
13. Waltemath, D.: Management of simulation studies in computational biology. In: Mosig, A., Rahnenführer, J., Eisenacher, M., Rahmann, S. (eds.) Invited Presentations, Junior Research Groups and Research Highlights at GCB 2015 vol. 3, p. 1668. PeerJ Inc., San Francisco, USA (2015)
14. Henkel, R., Wolkenhauer, O., Waltemath, D.: Combining computational models, semantic annotations and simulation experiments in a graph database. DATABASE **2015**, 130 (2015)
15. Tyson, J.J., Novák, B.: Functional motifs in biochemical reaction networks. Annual review of physical chemistry **61**, 219 (2010)
16. Zhang, Z., Zhang, J.: A big world inside small-world networks. PLOS ONE **4**(5), 5686 (2009)
17. Barabási, A.-L., Oltvai, Z.N.: Network biology: understanding the cell's functional organization. Nature Reviews Genetics **5**(2), 101–113 (2004)
18. Albert, R., Jeong, H., Barabási, A.-L.: Error and attack tolerance of complex networks. Nature **406**(6794), 378–382 (2000)
19. Chen, M.-S., Han, J., Yu, P.S.: Data mining: an overview from a database perspective. IEEE Transactions on Knowledge and Data Engineering **8**(6), 866–883 (1996)
20. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent pattern mining: current status and future directions. Data Mining and Knowledge Discovery **15**(1), 55–86 (2007)
21. Ramon, J., Bruynooghe, M.: A polynomial time computable metric between point sets. Acta Informatica **37**(10), 765–780 (2001)
22. Zass, R., Shashua, A.: Probabilistic graph and hypergraph matching. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2008). IEEE
23. Melnik, S., Garcia-Molina, H., Rahm, E.: Similarity flooding: A versatile graph matching algorithm and its application to schema matching. In: 18th International Conference on Data Engineering, pp. 117–128 (2002). IEEE
24. Chirita, P.A., Ghita, S., Nejdl, W., Paiu, R.: Semantically enhanced searching and ranking on the desktop. In: Workshop on The Semantic Desktop Next Generation Personal Information Management and Collaboration Infrastructure (2005). ISWC
25. Lakshmi, K., Meyyappan, T.: Frequent Subgraph Mining Algorithms - A Survey And Framework For Classification (2012)
26. Keyvanpour, M.R., Azizani, F.: Classification and analysis of frequent subgraphs mining algorithms. Journal Of Software **7** (2012)
27. Kuramochi, M., Karypis, G.: Frequent subgraph discovery. In: Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference On, pp. 313–320 (2001). IEEE
28. Borgelt, C., Berthold, M.R.: Mining molecular fragments: Finding relevant substructures of molecules. In: IEEE International Conference on Data Mining, pp. 51–58 (2002). IEEE
29. Yan, X.Y.X., Han, J.H.J.: gSpan: graph-based substructure pattern mining. 2002 IEEE International Conference on Data Mining, 2002. Proceedings. (2002)

30. Wörlein, M., Meinl, T., Fischer, I., Philippsen, M.: A quantitative comparison of the subgraph miners mofa, gspan, ffsm, and gaston. In: 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 392–403. Springer, Berlin Heidelberg (2005)
31. Priyadarshini, S., Mishra, D.: An approach to graph mining using gspan algorithm. In: International Conference on Computer and Communication Technology, pp. 425–430 (2010)
32. Wong, E., Baur, B., Quader, S., Huang, C.H.: Biological network motif detection: Principles and practice. Briefings in Bioinformatics **13**(2), 202–215 (2011)
33. Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., Hattori, M.: The kegg resource for deciphering the genome. Nucleic Acids Research **32**(suppl 1), 277–280 (2004)
34. Hattori, M., Okuno, Y., Goto, S., Kanehisa, M.: Development of a chemical structure comparison method for integrated analysis of chemical and genomic information in the metabolic pathways. Journal of the American Chemical Society **125**(39), 11853–11865 (2003)
35. Koyutürk, M., Grama, A., Szpankowski, W.: An efficient algorithm for detecting frequent subgraphs in biological networks. BIOINFORMATICS **20**(suppl 1) (2004)
36. Li, Y., Cao, B., Xu, L., Yin, J., Deng, S., Yin, Y., Wu, Z.: An efficient recommendation method for improving business process modeling. IEEE Transactions on Industrial Informatics **10**(1), 502–513 (2014)
37. Le Novère, N., Hucka, M., Mi, H., Moodie, S., Schreiber, F., Sorokin, A., Demir, E., Wegner, K., Aladjem, M.I., Wimalaratne, S.M., Bergman, F.T., Gauges, R., Ghazal, P., Kawaji, H., Li, L., Matsuoka, Y., Villéger, A., Boyd, S.E., Calzone, L., Courtot, M., Dogrusoz, U., Freeman, T.C., Funahashi, A., Ghosh, S., Jouraku, A., Kim, S., Kolpakov, F., Luna, A., Sahle, S., Schmidt, E., Watterson, S., Wu, G., Goryanin, I., Kell, D.B., Sander, C., Sauro, H., Snoep, J.L., Kohn, K., Kitano, H.: The Systems Biology Graphical Notation. Nature Biotechnology **27**(8), 735–741 (2009)
38. Le Novère, N., Finney, A., Hucka, M., Bhalla, U.S., Campagne, F., Collado-Vides, J., Crampin, E.J., Halstead, M., Klipp, E., Mendes, P., *et al.*: Minimum information requested in the annotation of biochemical models (miriam). Nature Biotechnology **23**(12), 1509–1515 (2005)
39. Courtot, M., Juty, N., Knüpfer, C., Waltemath, D., Zhukova, A., Dräger, A., Dumontier, M., Finney, A., Golebiewski, M., Hastings, J., *et al.*: Controlled vocabularies and semantics in systems biology. Molecular Systems Biology **7**(1), 543 (2011)
40. Alm, R., Waltemath, D., Wolfien, M., Wolkenhauer, O., Henkel, R.: Annotation-based feature extraction from sets of sbml models. Journal of Biomedical Semantics **6**(1), 20 (2015)
41. Schulz, M., Krause, F., Le Novère, N., Klipp, E., Liebermeister, W.: Retrieval, alignment, and clustering of computational models based on semantic annotations. Molecular Systems Biology **7**(1), 512 (2011)
42. Polyvyanyy, A., Kuropka, D.: A quantitative evaluation of the enhanced topic-based vector space model (2007)
43. Henkel, R., Hoehndorf, R., Kacprowski, T., Knüpfer, C., Liebermeister, W., Waltemath, D.: Notions of similarity for systems biology models. Briefings in Bioinformatics, 090 (2016)
44. Junker, A., Sorokin, A., Czauderna, T., Schreiber, F., Mazein, A.: Wiring diagrams in biology: towards the standardized representation of biological information. Trends in biotechnology **30**(11), 555 (2012)
45. Rosenke, C., Waltemath, D.: How can semantic annotations support the identification of network similarities? In: Paschke, A., Burger, A., Romano, P., Marshall, M.S., Splendiani, A. (eds.) Proceedings of the 7th International Workshop on Semantic Web Applications and Tools for Life Sciences, p. 11. CEUR Workshop Proceedings, Aachen (2014)
46. Knüpfer, C., Beckstein, C., Dittrich, P., Novère, N.L.: Structure, function, and behaviour of computational models in systems biology. BMC Systems Biology **7**(1), 43 (2013)
47. Scharm, M., Wolkenhauer, O., Waltemath, D.: An algorithm to detect and communicate the differences in computational models describing biological systems. BIOINFORMATICS, 484 (2015)
48. Scharm, M., Waltemath, D., Mendes, P., Wolkenhauer, O.: Comodi: an ontology to characterise differences in versions of computational models in biology. Journal of Biomedical Semantics **7**(1), 46 (2016)
49. Cooper, J., Scharm, M., Mirams, G.R.: The cardiac electrophysiology web lab. Biophysical journal **110**(2), 292–300 (2016)