Internal Report

Deliverable 1.2:

End-to-end Supervised Knowledge Graph Decomposition

Institution Version 0.1 FINAL

Abstract: Knowledge graphs are networks with annotated nodes and edges, representing different relations between the network nodes. Learning from such graphs is becoming increasingly important as numerous real-life systems can be represented as knowledge graphs, where properties of selected types of nodes or edges are learned. This paper presents a fully autonomous approach to targeted knowledge graph decomposition, advancing the state-of-the-art HINMINE network decomposition methodology. In this methodology, weighted edges between the nodes of a selected node type are constructed via different typed triplets, each connecting two nodes of the same type through an intermediary node of a different type. The final product of such a decomposition is a weighted homogeneous network of the selected node type. HINMINE is advanced by reformulating the supervised network decomposition problem as a combinatorial optimization problem, and by solving it by a differential evolution approach. The proposed approach is tested on node classification tasks on two real-life knowledge graphs. The experimental results demonstrate that the proposed end-to-end learning approach is much faster and as accurate as the exhaustive search approach.

Document administrative information	
Project acronym:	HinLife
Project number:	J7-7303
Deliverable number:	D1.2
Deliverable full title:	End-to-end Supervised Knowledge Graph Decomposition
Document identifier:	HinLife -del-D1.2-supervised-graph-decomposition-final-v0.1
Lead partner short name:	JSI
Report version:	0.1, final
Report preparation date:	01/10/2018
Lead author:	Blaž Škrlj
Co-authors:	Jan Kralj, Nada Lavrač
Status:	Final

Journal of Intelligent Information Systems manuscript No.
(will be inserted by the editor)

CBSSD: Community-Based Semantic Subgroup Discovery

Blaž Škrlj · Jan Kralj · Nada Lavrač

Received: date / Accepted: date

Abstract Modern data mining algorithms frequently need to address learning from heterogeneous data and knowledge sources, including ontologies. A data mining task where ontologies are used as background knowledge in data analysis is referred to as semantic data mining. The specific task we address is semantic subgroup discovery, allowing for ontology terms to be used in induced subgroup describing rules. This paper proposes Community-Based Semantic Subgroup Discovery (CBSSD) as means to advance ontology-based subgroup identification by taking into account also the structural properties of induced complex networks related to the studied phenomenon. Following the idea of multi-view learning, which builds on using different sources of information to obtain better models, the proposed CBSSD approach can leverage different types of nodes of the induced complex network, such as simultaneously using information from multiple levels of a biological system. The approach was tested on ten data sets consisting of genes related to complex diseases, as well as core metabolic processes. The experimental results show that the CBSSD approach is scalable, applicable to large complex networks, and that it can be used to identify significant combinations of terms, which could not be uncovered by standard term enrichment analysis approaches.

Keywords Semantic data mining, bioinformatics, community detection, network analysis, term enrichment analysis

Blaž Škrlj
Jožef Stefan International Postgratuate School, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: blaz.skrlj@ijs.si

Jan Kralj
Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
E-mail: jan.kralj@ijs.si

Nada Lavrač
Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
University of Nova Gorica, Glavni trg 8, 5271 Vipava, Slovenia
E-mail: nada.lavrac@ijs.si

1 Introduction

Modern machine learning approaches are capable of using continuously increasing amounts of information to explain complex phenomena in numerous fields, including biology, sociology, mechanics and electrical engineering. As there can be many distinct types of data associated with a single phenomenon, novel approaches strive towards the integration of different, heterogeneous data and knowledge sources, as data used in building predictive or descriptive models [11].

In such settings, prior knowledge can play an important role in the development and deployment of learning algorithms in real world scenarios. Background knowledge can come in many forms, which introduces additional complexity to the modeling process, yet can have a great impact on the model's performance. For example, Bayesian methods can be leveraged to incorporate knowledge about prior states of a system, i.e. prior distributions of random variables being modeled. Such methods are in widespread use, e.g., in the field of phylogenetics, where Bayesian inference is used for reconstruction of evolutionary trees [20]. Background knowledge can also be encoded more explicitly, as an additional knowledge source to be used in learning the models. Machine learning research that relies on the use of explicitly encoded background knowledge includes relational data mining [22] and inductive logic programming (ILP) [39]. In ILP, background knowledge is used along with the examples to derive hypotheses in the form of logical rules, which explain the positive examples.

A special form of background knowledge are *ontologies*, which can be used to guide the rule learning process. *Semantic subgroup discovery* (SSD) [37, 59] is a field of rule learning, which uses ontologies as background knowledge in the subgroup discovery process, aimed at inducing rules from classification data. Here, class labels denote the groups for which descriptive rules are to be learned.

In this work we use the formalism of *complex networks* to represent the studied interactions [13]. They consist of nodes (i.e. proteins) and edges (i.e. interactions between proteins). Real world networks often contain communities, or other topological structures of interest, which correspond to functional properties of the network [55, 21]. We propose a methodology, where iteratively constructed complex networks are used as input to identify relevant subgroups by network partitioning, followed by semantic subgroup discovery. We experimentally demonstrate that new knowledge can be obtained using existing, freely accessible heterogeneous data in the form of complex networks and ontologies.

Community-based Semantic Subgroup Discovery is to our knowledge one of the first attempts, where we address the issue of learning from complex networks by using semantic subgroup discovery. Further, the developed approach is scalable, and offers the opportunity to investigate interaction between different semantic (GO) terms.

This paper is a significant extension of our previous work [53]. We more thoroughly describe the theoretical background and contribute to a better understanding of representing network partitions in a machine learning setting. Next, in addition to community-based network partitioning, we investigate also component-based partitioning. Moreover, we also perform a quantitative evaluation of the proposed CBSSD methodology compared to standard enrichment analysis approaches.

After presenting the background and related work in Section 2, the subsequent sections present the theoretical (Section 3), as well as empirical aspects of the pro-

posed CBSSD methodoloy. We test the use of the new approach on 10 different life science data sets, i.e. expert defined gene sets, where the CBSSD methodology is quantitatively compared to the existing enrichment analysis approaches (Section 5). Section 6 demonstrates the qualitative utility of the proposed CBSSD methodology on two real world data sets from the life science domain. The experimental evaluation of the methodology is followed by a discussion on the obtained results in Section 7, which also presents the plans for further work.

2 Background and related work

This section introduces relevant concepts from the fields of complex networks, enrichment analysis, semantic data mining, subgroup discovery and multi-view learning.

2.1 Complex networks

Many natural phenomena can be described using graphs. They can be used to model physical, biological, chemical and mechanical systems [47, 61]. Complex networks are graphs with distinct, non-trivial, real world topological properties [13]. Real world networks can be characterized with distinct statistical properties regarding their node degree distribution, component distribution or connectivity [55].

Despite extensive efforts to understand complex networks from a physical standpoint, methods for associating the distinct topological features of real-life networks with existing knowledge remain poorly investigated. Such methodology can provide valuable insights into functional organization of otherwise incomprehensible quantity of topological structures, which commonly occur in, e.g., biological or transportation networks.

Complex networks are commonly used in modeling systems, where extensive background knowledge is not necessarily accessible. Motif finding, community detection and similar methods can provide valuable insights into the latent organization of the observed network [18]. Such networks are also known to include many *communities*, i.e. smaller, distinct units of a network that correspond to subsets of network nodes with dense connections between nodes within the subset and sparse connections between nodes in the subset and other nodes in the network [21] Communities can be detected with random walk-based sampling, spectral graph properties or other network properties [42, 36]. In this work we focus mostly on two community detection algorithms, the Louvain algorithm and the InfoMap algorithm.

2.1.1 Louvain algorithm

The Louvain algorithm [9, 10], defined on undirected networks, is based on the network modularity measure Q [12] defined for a network partitioned into communities as follows:

$$Q = \frac{1}{2m} \sum_{v=1}^{n} \sum_{w=1}^{n} \left[A_{v,w} - \frac{k_v k_w}{2m} \right] \delta(c_v, c_w) \tag{1}$$

where n represents the number of nodes and m the number of edges, $[A_{v,w}]_{v,w=1}^{n}$ denotes the adjacency matrix (i.e. $A_{v,w}$ is 1, when u and v are connected by an

edge, and 0 otherwise), k_v denotes the degree of the v-th node and c_v denotes the community the v-th node is assigned to. The $\delta(c_v, c_w)$ represents the Krönecker delta function, which amounts to 1 when $c_v = c_w$ and 0 otherwise. The value $\frac{k_v k_w}{2m}$ represents the average fraction of edges between nodes v and w in a random graph with the same node degree distribution as the considered graph. The modularity value Q will be high if most connections in the graph are between nodes assigned to the same community. The Louvain algorithm discovers the partitioning of nodes into communities for which the value Q is maximized using a greedy, non-exact procedure that runs in $\mathcal{O}(n \log(n))$. We refer the interested reader to [10] for more information on the algorithm.

2.1.2 InfoMap algorithm

Many real world networks contain different types of nodes (i.e. node layers). When connections between different types of nodes are taken into account, new form of dynamics can emerge, which yields e.g., otherwise non-detectable community structure [29]. To account for such heterogeneity, our methodology can also account for such an organization without additional simplification of the network. For such tasks, we leverage the state-of-the-art InfoMap algorithm for multilayer community detection [49].

The InfoMap algorithm is based in the idea of minimal description length of the walks performed by a random walker traversing the network. we describe its movements using words from m *community* codebooks (describing movements within a given community) and one *index* codebook (describing movements between communities). Assuming the codebooks are constructed using Huffman coding [33] (a form of optimal lossless compression), let H_i represent the frequency-weighted average length of encoded random walks in community codebook i and H_q represent the frequency-weighted average length of codewords in the index codebook. The value of $L(M)$ therefore corresponds to the average length of the entire codeword describing the movement of the random walker. The idea is that the network partition that gives the shortest description length best captures the community structure of the network with respect to the dynamics on the network—such partition intuitively traps random walkers within individual communities.

The objective of the InfoMap algorithm is to minimize the community description length, a property corresponding to lengths of codewords (i.e. binary codes of nodes that encode the node) describing the movement of a random walker traversing the network. Its main objective function is formulated as the map equation:

$$L(M) = q_{\curvearrowright} H_q + \sum_{i=1}^{m} p^i_{\circlearrowright} H_i \tag{2}$$

where M is a partitioning of the network into communities, and $m = |M|$. The value $q_{\curvearrowright}$ represents the total probability that the random walker enters any of the communities M. For a community $i \in M$, $p^i_{\circlearrowright}$ represents the total probability that a node, visited by the random walker, is in community i, plus the probability that the

random walker exits community i. The values H_q and H_i are calculated as

$$H_q = -\sum_{i=1}^{m} \frac{q_{i\curvearrowleft}}{q_{\curvearrowleft}} \log\left(\frac{q_{i\curvearrowleft}}{q_{\curvearrowleft}}\right)$$
$$H_i = -\frac{q_{i\curvearrowright}}{p^i_{\circlearrowleft}} \log\left(\frac{q_{i\curvearrowright}}{p^i_{\circlearrowleft}}\right) - \sum_{\alpha\in i} \frac{p_\alpha}{p^i_{\circlearrowleft}} \log\left(\frac{p_\alpha}{p^i_{\circlearrowleft}}\right)$$

where $q_{i\curvearrowleft}, q_{i\curvearrowright}$ are rates at which the random walker enters and exists community i and p_α is the probability that the random walker will be at node α.

In this work we also investigate how multiplex variation of the InfoMap algorithm can be used for network partitioning. Here, connections between different types of nodes can be taken into account, which often yields different community partitioning compared to the standard InfoMap. Detailed description of the multiplex variation of the InfoMap is given in Appendix A.

2.2 Knowledge graphs

Apart from complex networks, this work relies heavily on the notion of knowledge graphs. Compared to complex networks, knowledge graphs consist of relation-labeled edges, such as the following example:

$$protein \xrightarrow{\text{interactsWith}} protein \xrightarrow{\text{annotatedWith}} domain. \tag{3}$$

Knowledge graphs are commonly used as a source of knowledge for understanding other phenomena, where annotations are not accessible (e.g., real-world complex networks). As knowledge graphs consist of defined relations between defined entities (nodes of knowledge graphs), inductive logic programming algorithms can be used to traverse and learn more general, interpretable rules.

Knowledge graphs, built by domain experts are also known as *ontologies* [28]. The challenge of incorporating domain ontologies in the data mining process has been addressed in the work on *semantic data mining* (SDM) [38].

It remains an open question as to whether it is possible to implement computationally feasible semantic data mining approaches, which can leverage both complex networks, as well as knowledge graphs related to the studied phenomenon, to simultaneously learn descriptions in form of rules of different generality, i.e. as general as possible, and/or as specific as possible. The obtained rules can offer additional context as they, compared to standard statistical tests, consist of multiple terms.

2.3 Enrichment analysis

Enrichment analysis (EA) techniques are statistical methods used to identify explanations for a set of entities based on over- or under-representation of their attribute values, which can be referred to as *differential expression*. In gene expression analysis, sets of differentially expressed genes are considered, and the approach is referred to as *gene set enrichment*. For example, Schipper et al [51] used miRNA expression profiles to obtain sets of genes, which were further studied to understand Alzheimer's disease in terms of transcription/translation and synaptic activity.

In life sciences, gene enrichment analysis is widely used with the Gene Ontology (GO) [3] to profile the biological role of genes, such as differentially expressed cancer genes in microarray experiments [57]. While standard EA looks at individual ontology terms (*term enrichment*) to provide explanations in terms of concepts/terms of a single ontology, researchers are increasingly combining several ontologies and data sets to uncover novel associations. Such efforts are needed, as different aspects of e.g., biological systems are studied by different research communities, resulting in multiple ontologies, each describing different aspects of a system from a different perspective. The ability to detect patterns in data sets that use sources other than the Gene Ontology can yield valuable insights into diseases and their treatment.

Enrichment of sets of genes can be studied also based on topological properties of complex networks. Here, a set of nodes—representing some network community, certain network component or some other topological structure that emerges in real-life networks—can be considered as a set of terms to be studied with enrichment analysis methods. Such statistics-based enrichment is widely used in fields such as social science and bioinformatics. For example, Alexeyenko et al [2] demonstrate an extension of gene set enrichment by using gene-gene interactions, List et al [40] propose a component-based enrichment approach and Dong et al [19] propose LEGO, a network-informed enrichment approach where network-based gene weights are used.

2.4 Semantic data mining

Semantic data mining can discover complex rules describing subgroups of data instances that are connected to terms (annotations) of an ontology, where the ontology is referred to as background knowledge used in the learning process. An example SDM problem is to find subgroups of enriched genes in a biological experiment, where background knowledge is the Gene Ontology [3]. We begin the discussion on semantic data mining by describing RDF graph formalism, used to represent semantic information. Next, we discuss three different use cases of how RDF-based knowledge was used to aid data mining approaches.

Formally speaking, semantic data mining (SDM) [58] is a field of machine learning that employs curated domain knowledge in the form of ontologies as background knowledge used in the learning process. An ontology can be represented as a data structure consisting of semantic triplets $T(S, P, O)$, which represent the subject, its predicate and the object. Such triplets form directed acyclic graphs. Resource Description Format (RDF) hypergraph is a data model commonly used to operate at the intersection of data and the ontologies.

There are many approaches, which use background knowledge in the form of ontologies to obtain either more accurate or more general results. First, knowledge in the form of ontologies can represent constraints, specific to a domain. It has been empirically and theoretically demonstrated that using background knowledge as a constraint can improve classification performance [4]. The RDF framework provides also the necessary formalism to leverage the graph-theoretic methods for ontology exploration. Network mining approach was used to discover indirectly associated biomedical terms. Here, Liu et al [41] developed a methodology, used to discover and suggest corrections for misinformation in biomedical ontologies.

Semantic clustering is an emerging field, where semantic similarity measures are used to determine the clusters using the background knowledge, in a manner simi-

lar to, for example, k-means family of clustering algorithms. Semantic clustering is frequently used in the area of document clustering [31].

Large databases in the form of RDF triplets exist for many domains. For example, the Bio2RDF project [5] aims at integrating all major biological databases and joining them under a unified framework, which can be queried using SPARQLQ—a specialized query language. The BioMine methodology is another example of large-scale knowledge graph creation, where biological terms from many different databases are connected into a single knowledge graph with millions of nodes [23]. Despite such large amounts of data being freely accessible, there remain many new opportunities to fully exploit their potential for knowledge discovery.

2.5 Semantic subgroup discovery

Semantic subgroup discovery (SSD) [37, 59] is a field of subgroup discovery, which uses ontologies as background knowledge in the subgroup discovery process, aimed at inducing rules from classification data. Here class labels denote the groups for which descriptive rules are to be learned. In semantic subgroup discovery, ontologies are used to guide the rule learning process. For example, the Hedwig algorithm [1, 59] accepts as input a set of class labeled training instances, one or several domain ontologies, and the mappings of instances to the relevant ontology terms. Rule learning is guided by the hierarchical relations between the considered ontology terms. Hedwig is capable of using an arbitrary ontology to identify latent relations explaining the discovered subgroups of instances. The result of the Hedwig algorithm are descriptions of target class instances as a set of rules of the form TargetClass $\leftarrow$ Explanation, where the rule condition is a logical conjunction of terms from the ontology. A detailed description of the Hedwig algorithm is given in Appendix B.

2.6 Multi-view learning

Multi-view learning represents the idea of learning using different approaches and different data sources. It is becoming an increasingly relevant topic, as systems such as multi-scale biological networks, transportation routes, or deep neural networks can only be understood when different aspects are studied simultaneously [63]. In a common multi-view setting, the data corresponding to the studied system comes in different forms. One of the possible goals is to learn a joint representation using all available sources of information (e.g., audio, video and sound). Further, different approaches are necessary to process different types of data or yield results of different generality. The latter is one of the main aspects of this work. Extensive collections of biological information have been previously analysed using ideas from multi-view learning. For example, Alexeyenko et al [2] propose a method, which apart from single genes computes enrichment of subsets of genes. The recently introduced KeyPathwayMinerWeb [40] offers similar functionality when focusing on network's components, i.e. connected subgraphs. Finally, the EnrichNet approach developed by Glaab et al [27] offers a web-based interface for qualitative exploration of expression profiles alongside biological pathways, i.e. networks of interacting proteins.

The discussed methods extend the standard term enrichment paradigm with different, network-based views, yet are application-specific, and as such not nec-

essarily flexible enough for modern heterogeneous biological networks. One of the goals of this work is to offer a general computational framework for learning from network partitions using arbitrary background knowledge collections. Furthermore, we demonstrate its use on biological networks, where multiple different aspects (e.g., gene and protein interactions, publications and domain annotations) are used to learn from complex networks.

3 Theoretical background and setting

This section discusses the relationship between network analysis and rule learning, starting with the preliminaries on rule learning and network partitions, and followed by our own contribution to explaining the problem setting.

3.1 Theoretical background

3.1.1 Rule learning preliminaries

A supervised machine learning task can be defined as follows: Given a set of classes T and a set of class labeled data instances D, the goal is to approximate the mapping $\Theta : D \rightarrow T$, which can explain/predict instances $d \in D$. In this work, we focus on rule learning algorithms.

Definition 1 Let $\mathfrak{R}$ denote a set of all rules that can be learned from given D and T. In rule learning, best rules $r_{1,\ldots,n} \in \mathfrak{R}$ are found by optimizing a predefined success criterion evaluated using a scoring function ϵ, that assigns each identified rule r_i a corresponding score, i.e. $\epsilon : r_i \rightarrow \mathbb{R}$.

In this work we focus on subgroup discovery, a subfield of supervised descriptive rule induction [45]. Here, learner Θ is given the data set D labeled with target classes from T, and comparable to supervised learning, aims at identifying and describing interesting subsets of D, corresponding to the selected target $t \in T$. Instead of a predictive model, the final result of descriptive learning are sets of rules, explaining a subset of positive examples of selected class t. In general, the optimal set of rules is obtained by maximizing rule quality, for a single rule defined as follows:

$$r_{opt} = \arg\max_{r_i \in \mathfrak{R}} \left[\epsilon(r_i) \right].$$

This criterion is e.g., used when coverage-based approaches are considered [24].

In this work we follow a different, recently introduced rule learning paradigm [60], which does not use a covering approach. instead, subgroup describing rules are learned using a specialized beam search procedure [59], and the output is a set of b rules in the final beam of size $b{=}|Beam|$.

For an interested reader we here explain the formulation for rule induction used by the Hedwig algorithm, described in detail in Appendix B. The presented formulation consists of two distinct objectives; rule uniqueness and rule quality, which together form the joint scoring function as follows:

$$\mathfrak{R}_{opt} = \arg\max_{\mathfrak{R}} \frac{\sum_{r \in \mathfrak{R}} \epsilon(r)}{\sum_{\substack{r_i, r_j \in \mathfrak{R} \\ i \neq j}} |Cov(r_i) \cap Cov(r_j)| + 1} \tag{4}$$

where $\mathfrak{R}$ represents a set of rules being optimized, $r \in \mathfrak{R}$ represents a single rule, and $Cov(r_i)$ denotes the set of examples covered by r_i. In Hedwig, a set of rules (a beam of size b) is iteratively refined during the learning phase using a selected refinement heuristic, such as for example lift or weighted relative accuracy.

The term $\sum_{r \in \mathfrak{R}} \epsilon(r)$ corresponds to the quality of individual rules. Simultaneously, the rules shall not overlap, which is achieved by introduction of the following term: $\sum_{\substack{r_i, r_j \in \mathfrak{R} \\ i \neq j}} |Cov(r_i) \cap Cov(r_j)| + 1$. Here, Hedwig aims to minimize the intersection of instances, covered by rules r_i and r_j.

Essentially, we want to maximize rule quality of the set of rules (the numerator), while at the same time having the rules cover different parts of the example space (minimize the denominator).

The beam search-based algorithm used in this work hence yields multiple different rules that represent different subgroups of the data set being learned on.

3.1.2 Network partition preliminaries

Let G represent a complex network, i.e. a graph with non-trivial topological properties. The set of network's nodes is denoted as N. In this work we address the issue of learning from n different partitions of G. A partition is a subnetwork, which can for example represent a functional community, a component or a convex subgraph [43].

Definition 2 (Trivial network partition) If a network is partitioned into a single partition set P, we term this partition a trivial partition, i.e., $|P| = 1; \forall p \in P_1 | p \in N$. On the contrary, if $|P| > 1$, the partition is non-trivial.

In this work we focus on non-trivial network partitions, where partitions can be *overlapping*, as defined below.

Definition 3 (Overlapping network partition) An overlapping network partition consists of at least two partitions $P_x \in P$ and $P_y \in P$, which include the same node:

$$\exists n \in N | (n \in P_x) \wedge (n \in P_y); (P_x \neq P_y).$$

3.2 Explaining the problem setting

This section formally presents the problem of learning from network partitions as a rule learning problem.

3.2.1 Representing network partitions as classes in a rule learning setting

To understand the connection between network's partitions P and a representation, useful for different down-stream machine learning approaches, e.g., for rule learning or subgroup discovery, we need to establish a relationship between the partitions P and the corresponding classes T.

A non-overlapping network partition can be described as a surjective mapping between the nodes and their corresponding partitions, whereas overlapping partitions are described as one-to-many mappings.

Proposition 1 *The upper bound for the number of classes, needed to represent an overlapping partition P is*

$$|T| = |P|.$$

Proof Let the $s : N \rightarrow P$ denote a mapping between the set of nodes N and the set of partitions P. The cardinality of the set of all mappings $|\bigcup_{i=1}^{i=|N|} s(n_i)|$ is thus equal to $|P|$. □

This observation is useful for studying a more general case, where all possible partitions are accounted for. To prove a general case for overlapping partitions, a relation between a node and its corresponding partitions needs to be defined. The number of all non-empty partitions of a set is known as the *Bell number* [25].

Definition 4 (Bell number) Let B_i denote the i-th Bell number and $B_0 = 1$. The k-th Bell number is then defined via recurrent relation:

$$B_{k+1} = \sum_{i=0}^{k} \binom{k}{i} B_i. \tag{5}$$

Proposition 2 *A network with $n = |N|$ nodes can be partitioned into $B_n - 1$ unique non-trivial partitions, where n denotes the n-th Bell number.*

Proof Each network with n nodes can be partitioned into $1 + \text{nt}$ partitions, consisting of 1 trivial partition and nt non-trivial partitions. Consequently, the number of non-trivial partitions $n\text{t}$ for networks with more than a single node equals $\sum_{i=0}^{i=n-1} \binom{n-1}{i} B_i - 1 = B_n - 1$. □

Corollary 1 *Take a network with n nodes. Having defined the maximum number of possible partitions $|P|$, and given Proposition 1, it immediately follows that the maximum number of classes $|T|$ assigned to a node corresponds to the number of all non-trivial partitions that it is part of, which equals $|T| = B_n - 1$. A node can be present in all possible partitions simultaneously, as long as they differ by at least one node.*

Example 1 Consider a network consisting of two nodes $\{a, b\}$. There are two possible partitions of this network, as $B_2 = \sum_{i=0}^{i=2-1} \binom{2-1}{i} B_i = 1 + 1 = 2$. The two partitions are: $\{\{a\}, \{b\}\}$, and $\{\{a, b\}\}$. The latter is a trivial partition and as such it is not relevant for various downstream learning tasks such as rule learning. According to Proposition 2, there remains a single relevant partition of nodes $\{a, b\}$ into two sets: $\{a\}$ and $\{b\}$.

Finally, we prove that considering all relevant partitions takes exponential time.

Proposition 3 *Considering all non-trivial partitions of a network with n nodes is exponential in terms of n.*

Proof As B_n is clearly a strictly increasing function of n, we can assume without loss of generality that n is even. Let $B_n - 1$ denote the set of non-trivial partitions. It follows that:

$$\begin{aligned} B_{n+1} - 1 &\geq B_n + nB_{n-1} - 1 \geq nB_{n-1} - 1 \\ &\geq n(n-2)(n-4)\cdots 2 - 1 = 2^{\frac{n}{2}} \cdot \left(\frac{n}{2}\right)! - 1 \end{aligned}$$

□

Corollary 2 *Overlapping network partitions are at least exponential in terms of n, as there are at least as many possible overlapping partitions, as there are non-overlapping partitions.*

Consequently, for a network with n nodes, a naïve approach for learning from its partitions would need to consider $B_n - 1$ possible classes, which would result in at least exponential time complexity in terms of n. For example, exhaustive rule learning for a network with $|N| = 20$ would need to consider the following number of possible classes: $B_{20} - 1 = 5,832,742,205,056$.

In the following sections, we propose a computationally feasible approach that considers as classes only the relevant partitions derived from a network's topology.

4 Proposed CBSSD methodology

This section presents the proposed approach to semantic subgroup discovery from complex networks, named CBSSD (Community-Based Semantic Subgroup Discovery). The proposed implementation focuses mostly on learning from the lists of nodes associated with the studied phenomenon, yet can be also applied to learn from complex networks directly. An overview of the CBSSD methodology is illustrated in Figure 1.

4.1 Steps of the CBSSD methodology

The methodology consists of four main steps, described in this section: network construction, network partitioning via community detection or other methods, appropriate background knowledge representation, and semantic subgroup discovery.

Step 1: Constructing a network of associations

The first step of the CBSSD methodology takes as input a list of input data instances, along with any complex network to which the input list of instances can be mapped. In the first step of the methodology, we automatically induce a network based on the input list. As an alternative first step of the CBSSD methodology, existing networks (such as for example the human proteome network) can be used instead of automatically induced networks. Both options are demonstrated and tested in Section 5.

To automatically induce a network from an input list of biological entities (such as proteins or genes), we levarage the Biomine methodology [23] for network construction; individual terms are used as seeds for crawling the BioMine knowledge graph, which includes millions of term associations across main biological databases, such as UniProt [15], Kegg [35], and GenBank [7].

To construct the graph from BioMine, we introduced a network generator function Γ, which takes as input a node of interest and yields a graph corresponding to the node's neighborhood in the BioMine. The final knowledge graph G_f is constructed incrementally, by querying one term at a time. This results in a set of graphs $\{\Gamma(v_1), \ldots, \Gamma(v_n)\}$, where $\{v_1, \ldots, v_n\}$ are the input query terms. Setting

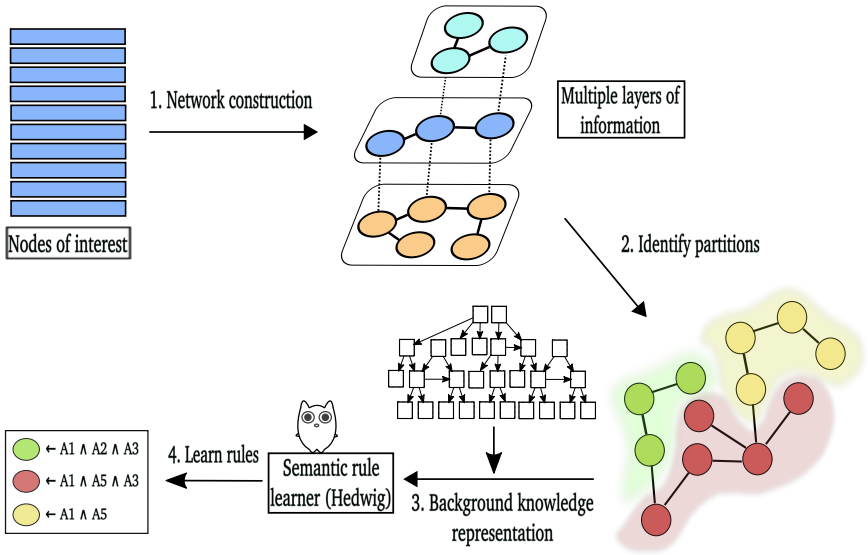

Fig. 1 Schematic representation of the proposed CBSSD procedure. Complex graph's communities are used to identify possible subgroups corresponding to the input node list. The subgroups are further explained using semantic subgroup discovery with background knowledge. The *Ai* terms in the rules denote different semantic terms, corresponding to individual communities (colored circles).

$\Gamma(v_i) = (V_i, E_i)$ for each i (where V_i is the set of nodes of the graph G_i and E_i is a set of edges), we construct a single final graph from the graphs by merging the nodes and edges, i.e. we construct the graph $G_f = (V_f, E_f)$ by setting $V_f = \bigcup_{i=1}^{n} V_i$ and $E_f = \bigcup_{i=1}^{n} E_i$.

Step 2: Partitioning a complex network

In the second step of the CBSSD methodology, the network constructed in the first step is partitioned. As shown in Section 3, there exist $B_n - 1$ relevant network partitions for non-overlapping partitions, and even more when partitions overlap. As exhaustive partition analysis is not computationally feasible due to exponential time complexity, we leverage two different community detection algorithms.

The first community detection algorithm we use in this step is the Louvain algorithm which is useful for large networks. The Louvain algorithm used is not capable of multiplex community detection, which is of relevance, as interaction coupling between protein-protein and gene-gene interaction layers can be considered. For this task, we leverage the second community detection algorithm, the multiplex variation of the InfoMap algorithm (described in Appendix A). For completeness, our approach also includes a variant of the InfoMap algorithm which detects communities in homogeneous networks, i.e. networks consisting of single node types. The community detection algorithm to be used is application specific, yet our initial experiments show that for larger homogeneous networks, the Louvain algorithm performs faster.

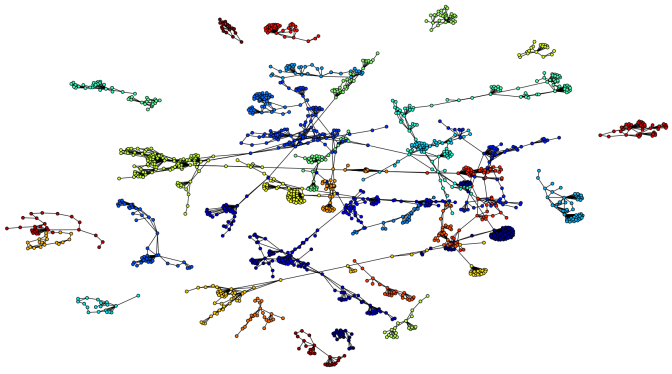

Fig. 2 An example community partition. Different colors correspond to different communities. The network represents communities detected on a BioMine network used in our previous study [53]. It can be observed, that multiple communities emerge, which were shown to correspond to different functional processes.

We tested this claim on the IntAct network, described in detail in Section 5.1. This step is considered multi-view learning, as heterogeneous networks consisting of multiple layers of different types of information are used to partition input instances. More specifically, the community detection on heterogeneous networks can be viewed as multi-view clustering, which aims to obtain a partition of the data in multiple views that often provide complementary information to each other [63].

Apart from community-based partitioning, we also consider network components, i.e. connected subnetworks present in real world complex networks [54]. Similarly to community detection, the result of component-based partitioning is a set of a network's components further used for learning.

For this step, the constructed knowledge graph can be interpreted either as an undirected graph (in biological context this makes sense as long as we are interested only in associations), or as a heterogeneous network. In our experiments, we used *codes_for* relation to associate individual proteins from the protein-protein interaction layer with genes from the gene-gene network. The community detection procedure returns sets of nodes $\{C_1, C_2, \ldots C_n\}$ that represent individual communities. Each node in the network belongs to exactly one community (i.e. the communities are non-overlapping). An example community partition is depicted in Figure 2[1].

Step 3: Background knowledge representation

The goal of the CBSSD algorithm is to discover semantic descriptions of identified communities. To this end, each community C_i (discovered in Step 2 of the CBSSD algorithm) becomes a class label T_i—the nodes from the input list are labelled with the

[1] Visualization was plotted with the Py3Plex library (`https://github.com/SkBlaz/Py3Plex`)

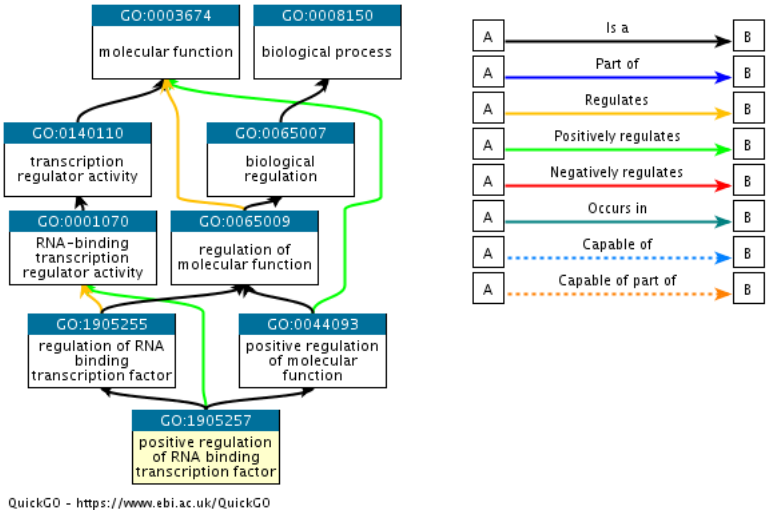

Fig. 3 Example GO hierarchy related to RNA binding factors. Connections between terms are directed.

community they belong to. In this way, input nodes are grouped into distinct classes, yet no additional nodes present in the detected communities are added as instances, as they could introduce unnecessary noise in the semantic subgroup discovery step.

Semantic rule learning requires the data to be encoded in the form of RDF triplets $T(S, P, O)$, where S is the subject, P the predicate and O the object. The experimental data from the previous step was converted into RDF triplets as required by Hedwig, the algorithm used in the rule discovery process [59]. Hedwig is capable of leveraging the background knowledge in the form of ontologies to guide the rule construction process. It does so by using the hierarchical relations between the ontology terms. Rules are initially constructed using more general terms and further refined using more specific terms. Our main source of background knowledge in this study is the Gene Ontology (GO) [3] database, one of the largest semantic resources for biology. It includes tens of thousands of terms, which together form a directed acyclic graph, directly usable by semantic subgroup discovery tools. An example hierarchy taken from the GO is displayed in Figure 3.

For Hedwig to perform rule construction, two conditions must be met. First, individual node names from the community detection step need to have the corresponding GO term mappings, and second, the whole gene ontology must be provided as a source of background knowledge. This requires that the nodes, corresponding to the discovered communities are encoded in the form of semantic triplets. Such encoding is achieved by treating each observed community as an individual target class, where all of its nodes are considered instances of this class. The key aspect of the rule generation procedure is the definition of the predicate, which will be used for finding suitable rule conjunctions.

To summarize, the output of this step is a list of nodes from the complex network. The nodes are (1) labelled by classes that correspond to the communities they belong to and (2) annotated with corresponding GO terms, which enable semantic rule induction described in the next step.

Step 4: Semantic rule induction

The result of this step (and the final result) of the CBSSD methodology are then rules of the form TargetClass ← Explanation, where TargetClass corresponds to one of the partitions, discovered in step 2, and Explanation is a conjunct of one or more terms from the background knowledge, prepared in step 3. The subgroup discovery is carried out by the Hedwig algorithm. Individual rules are learned by maximization of the criterion, introduced in section 3 (Equation 4). By convention, we use the *subClassOf* predicate when constructing the background knowledge base. Further, *is_a* predicate is used to map individual nodes to their semantic term annotations. Individual rules' p-values are determined by the Fisher's exact test (FET), a non-parametric, contingency table-based procedure, where a difference in coverage between two rules is leveraged to select the better one. We refer the interested reader to [59, 60] for a comprehensive treatment of the statistical rule evaluation, as used by the Hedwig algorithm.

4.2 Final formulation of the CBSSD approach

The CBSSD approach can be formalized as Algorithm 1. First, individual input terms are used to construct the heterogeneous network related to the studied phenomenon. Partitions are identified ($PartitionDetection$ step) and the input term list is partitioned according to the presence of individual terms within specific partitions ($PartitionFunction$). Finally, background knowledge in the form of ontologies is used to discover meaningful rules of individual partitions ($runHedwig$).

In Algorithm 1, I represents the input node list, O the ontology used in the semantic learning process, Γ a graph generator, and S represents the knowledge graph, which is incrementally constructed from the input list. The stopping criterion for evaluating individual sets of rules can be any rule significance heuristics, such as, for example, the chi-squared metric, entropy-based measures or similar [45].

Input: nodes of interest I annotated by ontologies (Ξ),
network generator (Γ)
Output: Rule sets
$V, E := \varnothing$; ▷ `Network construction (optional)`
foreach *node* $v \in I$ **do**
  $V := \Gamma(v)_{nodes} \cup V$;
  $E := \Gamma(v)_{edges} \cup E$;
end
$S := (V, E)$;
$C_{1...n} := PartitionDetection(S)$; ▷ `Partition detection`
$P_{1...n} := PartitionFunction(I, C_{1...n})$; ▷ `Partition representation`
RuleSets := $runHedwig(P_{1...n}, \Xi)$; ▷ `Rule induction`
return RuleSets

Algorithm 1: Pseudocode of the CBSSD approach.

There are two computationally expensive steps in the current implementation of the CBSSD approach, the community detection and the semantic subgroup discovery. The community detection algorithms used [49, 9] were previously proven to scale well

to millions of nodes and edges. The subgroup discovery part performed by Hedwig uses an efficient beam search, where only a set of rules is propagated through search space and continuously upgraded. A parallel beam search could potentially speed up the rule discovery, yet we leave the development of such algorithm for further work. Note that Hedwig already [1, 59] uses efficient parallelism with bitsets for determining the coverage of conjuncts of rules.

Individual parts of the CBSSD framework are parameterized as follows.

- Parameters of network construction (step one
  - Node batch size, denoting the number of nodes used to query the BioMine network
  - Types of nodes and edges kept in the final network
- Parameters of partition detection (step two)
  - Partition detection algorithm with corresponding parameters, e.g., number of iterations, type of community detection etc.
- Parameters of background knowledge representation step three
  - Generalization predicate used
- Parameters of rule induction (step four)
  - Search heuristic used (e.g., lift, gain, WRAcc etc.)
  - Beam size
  - Depth (maximum number of conjunctions)

If not otherwise stated, we use the Hedwig's default parameter settings. In the next section, we discuss the quantitative evaluation of the proposed approach.

5 Learning from the proteome: A quantitative scale up study

In this section we present a quantitative experimental setting, where the properties of the CBSSD algorithm are studied.

We begin the experimental evaluation of the proposed approach by investigating how different combinations of background knowledge and different complex networks used for term partitioning influence the explanatory capabilities of CBSSD. In this section we quantitatively demonstrate that the proposed approach can discover significant patterns, invisible to conventional enrichment approaches. The following sections are as follows. First, we discuss the experimental setting used. Next, we discuss evaluation measures, used to compare the two methods. Finally, we present the experimental findings in form of critical distance diagrams.

5.1 Experimental setting

To perform the experiments, we first downloaded the current version of binary protein interaction-based proteome from the IntAct database [46], which at the time of writing consists of more than 350,000 nodes and approximately 3.8 million edges. In IntAct, the nodes represent individual proteins, and the (undirected) edges represent their interactions. The edges are weighted, where the edge weights correspond to experimental reliability of the interactions between the corresponding proteins, and take values between 0 and 1.

A subset of the IntAct network is used to test the scalability of CBSSD, and to assess the difference between the term enrichment and the proposed semantic rule induction approaches. In this work we have filtered the network, keeping only the edges with reliability > 0.2 and eliminating isolated nodes. The filtered IntAct network—which is illustrated in Figure 4—consists of 100,000 nodes and 850,000 edges. Note that this is an order of magnitude larger than the automatically constructed BioMine knowledge graph, which consists of a union of input-specific subnetworks (as discussed in Section 4).

As the CBSSD leverages background knowledge in the form of ontologies, we additionally test the CBSSD's performance when either the reduced GO (GO Slim) [14] or the whole Gene Ontology is used [3]. The two ontologies contain biological terms describing different biological functions, components and processes.

We compare the proposed methodology against the Fisher's exact test-based term enrichment, as used in DAVID [32] and similar tools for gene set enrichment. Here, the Fisher's exact test is used to determine the significance of a term. This test is based on the hypergeometric distribution, where the p value is defined as follows:

$$p = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}} = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!n!} \tag{6}$$

where a represents the count of query genes within a pathway, b the number of all known genes present in the pathway, c the number of genes not present in the

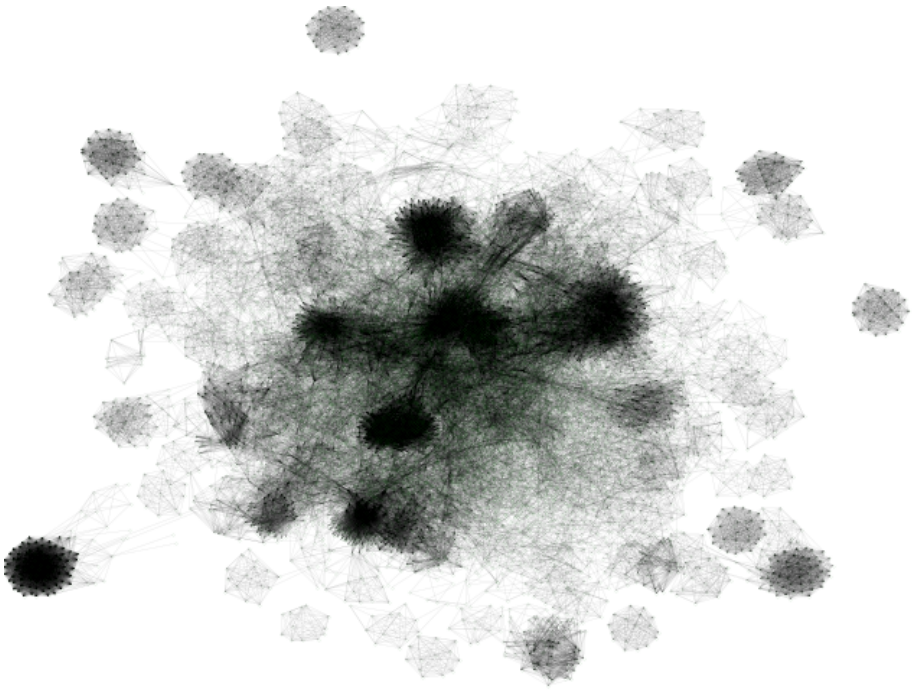

Fig. 4 Part of the human (IntAct) proteome (above the 0.2 reliability threshold) used in this study. It can be observed that densely connected subnetworks emerge, thus the network could contain potentially interesting communities and components.

Table 1 Description of different term enrichment and semantic subgroup discovery approaches compared, where "Terms" denote EASE-based single term enrichment (i.e. term enrichment using EASE score defined in Equation 7), and "Rules" denote the CBSSD approach.

Algorithm	Description
Rules(IntAct+GO)	CBSSD with IntAct proteome and whole Gene Ontology
Rules(IntAct+GOslim)	CBSSD with IntAct proteome and reduced Gene Ontology
Rules(BMN+GO)	CBSSD with BioMine and whole Gene Ontology
Rules(BMN+GOslim)	CBSSD with BioMine and reduced Gene Ontology
Terms(IntAct+GO)	TE (EASE) with Intact proteome and whole Gene Ontology
Terms(IntAct+GOslim)	TE (EASE) with IntAct proteome and whole Gene Ontology
Terms(BMN+GOslim)	TE (EASE) with BioMine and reduced Gene Ontology
Terms(BMN+GO)	TE (EASE) with BioMine and whole Gene Ontology

pathway, and d the number of all known genes not present in the pathway, and $n = a + b + c + d$. Additionally, DAVID uses a more conservative EASE score [30], where a is replaced by $a - 1$. This leads to modifying Equation (6) as follows:

$$p_{\text{EASE}} = \frac{\binom{a-1+b}{a-1}\binom{c+d}{c}}{\binom{n}{a-1+c}} = \frac{(a-1+b)!(c+d)!(a-1+c)!(b+d)!}{(a-1)!b!c!d!n!} \tag{7}$$

This correction provides more robust results for cases when only a handful of genes are used as an input. We systematically investigate whether the selected genes are associated with disease pathways as well as with basic metabolic processes. If not stated otherwise, we consider terms or rules to be significant at significance level < 0.05. As both approaches (CBSSD and EASE) evaluate many terms, we correct the p-values obtained during learning by the Benjamini-Hochberg multiple test correction [6].

Different term enrichment and semantic subgroup discovery settings used in the experiments are summarized in Table 1. The CBSSD's running times can differ significantly, therefore we parameterize the beam size—one of the parameters determining the CBSSD's runtime—as follows. We run the parallel implementation of the term enrichment for each data set and measure the running times. We adapt the CBSSD's beam size so that execution takes approximately the same amount of time for each data set. The final beam size requiring a similar amount of time to EASE-based enrichment averaged over all input lists was 300. On the BioMine network, we used the InfoMap algorithm for community detection as it can leverage the heterogeneous structure of the network. On the much larger IntAct network, we used the Louvain algorithm for its performance.

Apart from the community detection algorithms, we additionally explored learning from component-based partitions.

5.2 Evaluation measures

We describe six different quantitative measures ϵ used in evaluating each of the aforementioned approaches. The measures are divided into two main groups; Weighted relative accuracy (WRAcc)-based measures and Information content (IC)-based measures. First, we investigate how different approaches behave when measured with the weighted relative accuracy (WRAcc). For a given rule $r_i \in \mathfrak{R}$ for class C, WRAcc of the rule is calculated as follows. Given the number of examples N, the number N_C

of examples of a given class C (i.e. the number of positive examples), the number of all covered examples $Cov(r_i)$ by rule r_i, the number of correctly classified positive examples $TP(r_i)$ (true positives), the WRAcc for class C is defined as follows:

$$WRAcc(r_i) = \frac{Cov(r_i)}{N}\left(\frac{TP(r_i)}{Cov(r_i)} - \frac{N_C}{N}\right). \tag{8}$$

The WRAcc defined for a rule represents the rule's accuracy for explaining the target class, weighted by the number of instances belonging to that class. The higher the WRAcc, the better a rule explains the target class under consideration. We compute three variations of WRAcc for each approach:

1. Maximum WRAcc—the maximum WRAcc score of any rule in the whole rule set.
2. Average (mean) WRAcc—the mean WRAcc score of WRAcc scores of all rules in the rule set.
3. Minimum (worst case) WRAcc—the minimum WRAcc score of any rule in the whole rule set.

The three metrics indicate different properties of the tested approaches. Minimum and maximum WRAcc represent the worst and best rule learned by an approach. The mean WRAcc represents an average performance. Individual terms, which are the main result of EASE-based term enrichment, are considered as single term rules for WRAcc calculation. We consider such representation relevant, as single term results are commonly interpreted one by one, should no additional software be used for term summarization.

Next, we compute the information content of individual rules. Information content for a single term rule (standard term enrichment) is defined as:

$$IC_{\text{term}} = -\log(p(\text{term})). \tag{9}$$

This definition can be extended to rules r_i where the condition is a conjunct of several terms, i.e. $\text{term}_1 \wedge \text{term}_2 \wedge \cdots \wedge \text{term}_k$. In this case, assuming that the probability of one term annotating a gene is independent of another term annotating the gene, and

$$IC_{\text{r}_\text{i}} = -\log(p(\text{term}_1 \wedge \text{term}_2 \wedge \cdots \wedge \text{term}_k)) = \sum_{i=1}^{k} -\log(p(\text{term}_i)). \tag{10}$$

This strong assumption is partially due to Hedwig's capability to generalize similar (dependent) terms, and thus reduce term dependencies. Similarly to the WRAcc measure, we compute three variations of the information score:

4. Maximum IC (best case) —the maximum IC score of any rule in the whole rule set.
5. Average (mean) IC—the average IC score of rules in the rule set.
6. Minimum IC (worst case) —the minimum IC score of any rule in the whole rule set.

Table 2 Gene lists used for the evaluation of gene enrichment and subgroup discovery approaches.

Name	Short description	No. UniProt IDs
Protein secretion	Genes involved in protein secretion pathway.	686
Unfolded protein response	Genes up-regulated during unfolded protein response, a cellular stress response related to the endoplasmic reticulum.	116
Coagulation	Genes encoding components of blood coagulation system; also up-regulated in platelets.	141
DNA repair	Genes involved in DNA repair.	158
Epigenetics TF	All known epigenetic transcription factors related to cancer.	153
Fatty acids	Genes encoding proteins involved in metabolism of fatty acids.	159
Hypoxia	Genes up-regulated in response to low oxygen levels (hypoxia).	205
SNP-BS	Genes, containing SNPs within protein binding sites.	466
Diabetes	A gene list containing diabetes-related genes.	513
miRNA	A gene list containing miRNA targets.	1296

To statistically evaluate the difference between results, we first computed the significance scores using the Friedman's test, followed by the Nemenyi post-hoc correction. The results are presented according to the classifier's average ranks along a horizontal line [17]. The obtained critical distance diagrams are interpreted as follows. If one or more classifiers are connected with a bold line, one can conclude that their performance is approximately the same with a 5% risk (no significant difference was detected). The classifiers are ranked for each data set separately; we assume that the data sets are independent.

5.3 Experimental data

We used ten different data sets, using previously analyzed gene and protein lists as input queries. All lists apart from SNP-BS and Diabetes were obtained from the download section of the GSEA project[2] [56]. The SNP-BS list represents the results of a recent study, where sequence variants were studied in the context of protein binding sites [62]. The Diabetes protein list represents a UniProt query with keyword diabetes. Entries from the UniProt [15] database, the largest database of proteins sequences, correspond to individual proteins. The lists are summarized in Table 2.

The lists used correspond to genes, present in different biological processes, both in terms of underlying network organization, as well as functional annotation. All gene accessions were converted to the corresponding UniProt identifiers for easier evaluation.

[2] `http://software.broadinstitute.org/gsea/msigdb/collections.jsp`

5.4 Experimental results

In this section we present the experimental findings. We first discuss the community-based partitioning, followed by the component-based one.

5.4.1 CBSSD with community detection: WRAcc results

We evaluate the performance based on three WRAcc measures introduced in Section 5.2, as well as the computational costs associated with different approaches. We begin by investigating the rule WRAcc. The critical distance diagram showing results for all approaches and statistical significance of them being different is depicted in Figure 5.

It can be observed that the maximum WRAcc scores mostly correspond to rule-based approaches. Here, the best performing approach leverages smaller BioMine network along with GO Slim—reduced ontology. The BioMine network appears to have had a noticeable effect on performance, as it serves as the background network for the top three approaches. The top approach (BMN+GOslim) noticeably outperforms the two Term enrichment approaches, based on the IntAct network. Similarly, the best term enrichment approach (BMN+GOslim) significantly outperforms the two term enrichment approaches, based on the IntAct network.

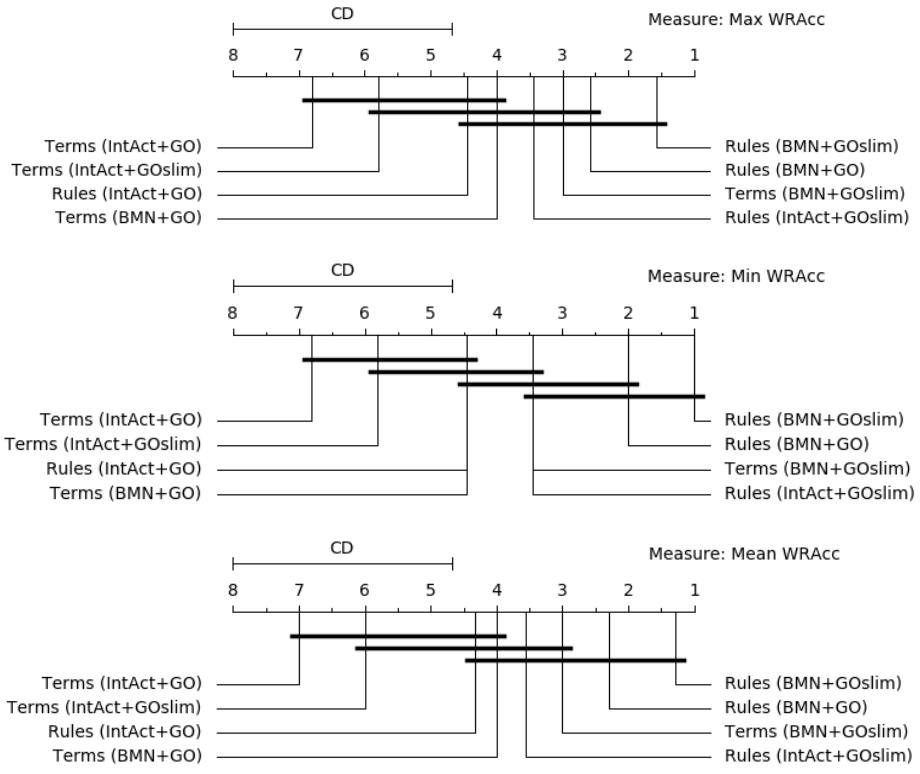

Fig. 5 WRAcc results for enrichment based on communities.

A very similar classifier rankings can be observed for all three CD diagrams. The CBSSD approach also results in a rule with the worst WRAcc measure (Figure 5, second diagram).

The average WRAcc ranks are similar to the maximum WRAcc results. Compared to maximum WRAcc and mean WRAcc, different approaches differ the most when minimum WRAcc is considered. Although the ranks of individual algorithms are the same, the *Rules (BMN + GOslim)* approach outperform all term-based approaches but *Terms (BMN+GOslim)*. All three diagrams indicate, BioMine (BMN)-based community partitioning yields rules with high WRAcc when reduced ontology (GOslim) is considered.

5.4.2 CBSSD with community detection: IC results

We continue the performance investigation when information content is considered. As the final result we obtained 3 different critical distance diagrams, corresponding to information content, shown in Figure 6.

Maximum information content corresponds to CBSSD's results (Rules), although there is no significant difference between the best CBSSD result and the best term enrichment (*Terms (IntAct+GOslim)*), which here leverages the IntAct network as the source for obtaining the network's partitions. The minimum IC results suggest

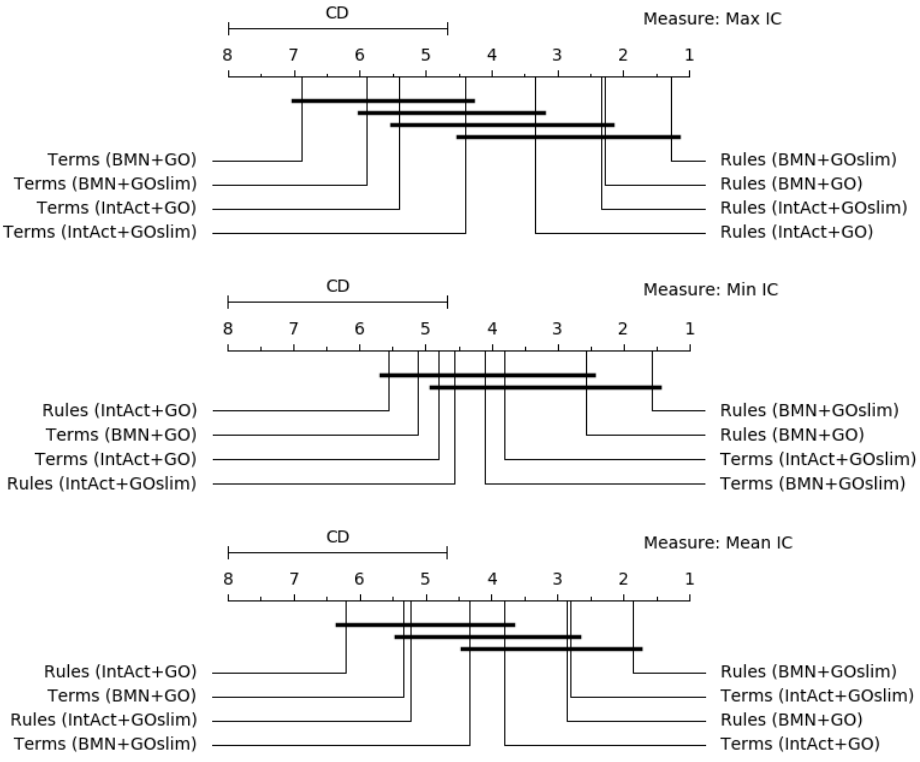

Fig. 6 IC results for enrichment based on communities.

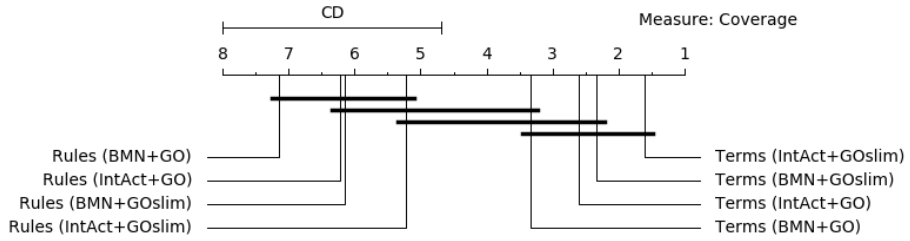

Fig. 7 Coverage results for enrichment based on communities.

some form of uniform distribution in terms of worst IC. A similar classifier ranking is observed when mean IC is considered. Interestingly, the best approach remains the one which leverages reduced GOslim ontology.

5.4.3 Coverage results for enrichment based on communities

We additionally report the rankings of the compared approaches with respect to rule coverage. As shown in Figure 7, in terms of coverage, term-based enrichment generally outperforms CBSSD variations. This is not surprising, since term-based enrichment corresponds to rules with only one term and has therefore larger coverage. Detailed overview of quantitative results is presented in Section 7. We continue the discussion with the results, obtained using component-based network partition function.

5.4.4 CBSSD with component partitioning: WRAcc results

The results presented in this section are structured similarly to the previous section. First, we present the WRAcc-related results, followed by the IC-based results, and conclude with an examination of the overall coverage.

The critical distance diagrams representing WRAcc-based comparisons are presented in Figure 8.

We observe a similar algorithm distribution compared to community-based partitioning in terms of absolute ranks. The best maximum WRAcc scores were obtained by rules and terms, based on the BioMine induced network. Similar rankings are obtained when mean WRAcc is considered. In both diagrams, the combination of a rule learner, BioMine network and the reduced ontology significantly outperform the IntAct-based approaches (*Rules (BMN + GOslim* dominates). A similar ranking of algorithms is obtained when the best minimum WRAcc is considered, i.e. the rule-based approaches are among the top three. We discuss the results obtained in this section in more detail in Section 7.

5.4.5 CBSSD with component partitioning: IC results

Similarly to the community-based partitioning, we further investigate the information content (IC) of individual approaches when component-based partitioning is

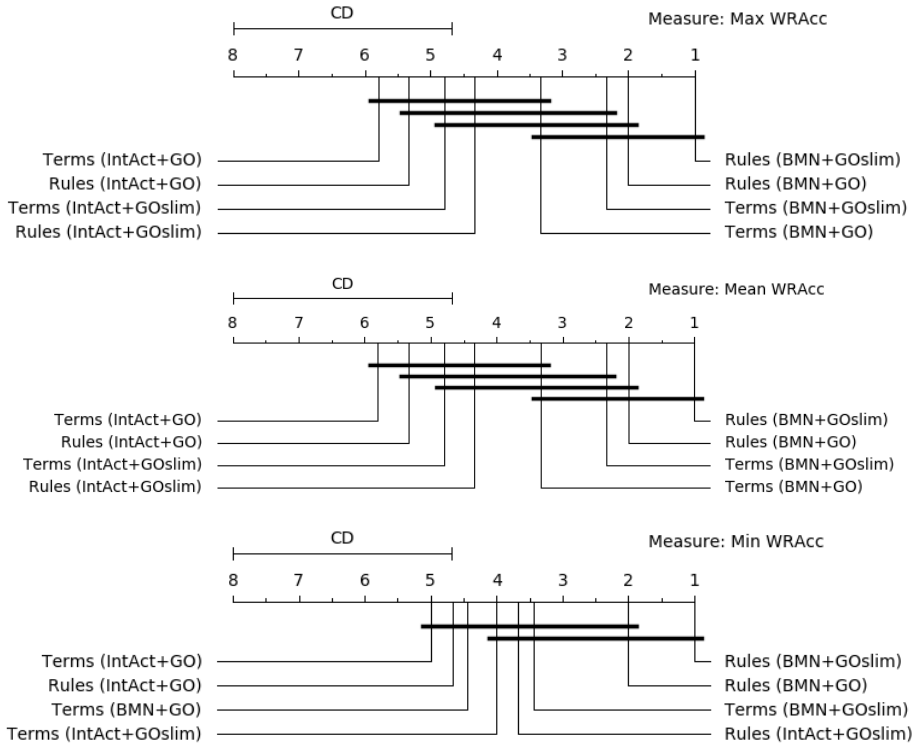

Fig. 8 WRAcc results for enrichment based on components.

considered (Figure 9). The maximum IC results similarly to the WRAcc based measurement yield the rule-learning, augmented with the BioMine network and GO Slim ontology as the best approach (*Rules (BMN + GOslim)*).

Interestingly, the use of IntAct network in terms of IC for all three score variations (min, max, mean) yielded better results, compared to WRAcc in previous section. Three out of four best performing approaches in terms of mean IC leverage GO Slim as the background knowledge database, which indicates reduced ontologies have high potential for explanatory tasks.

5.4.6 CBSSD with component partitioning: Coverage results

Similarly to community-based network partition, term enrichment outperformes rule learning coverage-wise (Figure 10). This result indicates the network partition does not influence the algorithm's performance in terms of coverage. The difference in coverage is possibly due to different types of rules compared (exclusively single term rules—EASE-based enrichment vs. multi conjunct rules).

A possible explanation for the observed result is that finding interesting higher order rules is a challenging task, and compared to terms, fewer rules are identified. We further observe that using GO Slim (reduced ontology) as the background knowledge,

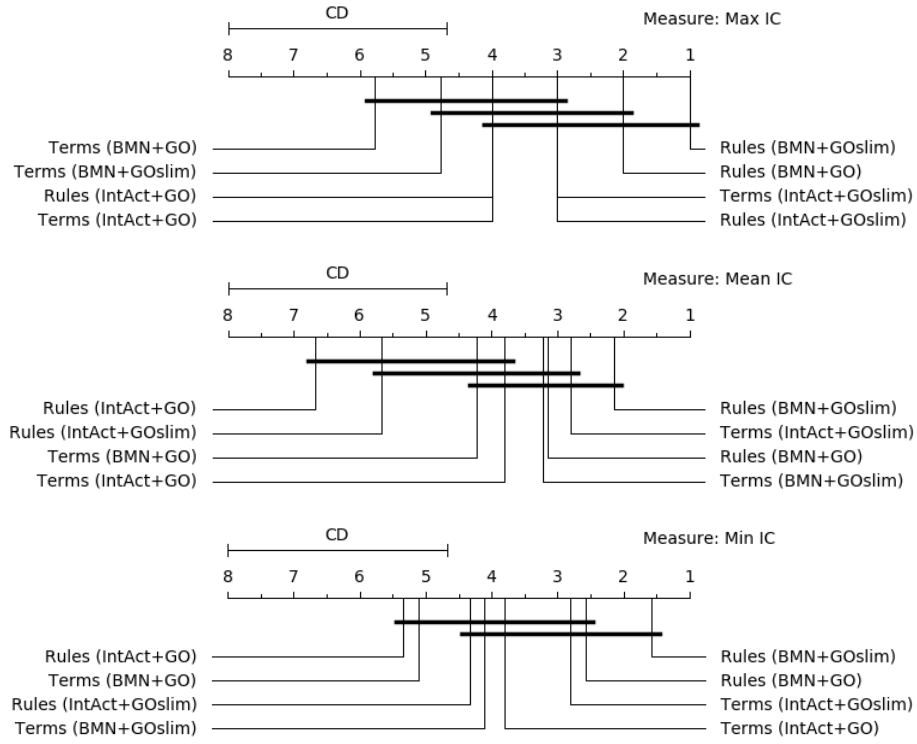

Fig. 9 IC results for enrichment based on components.

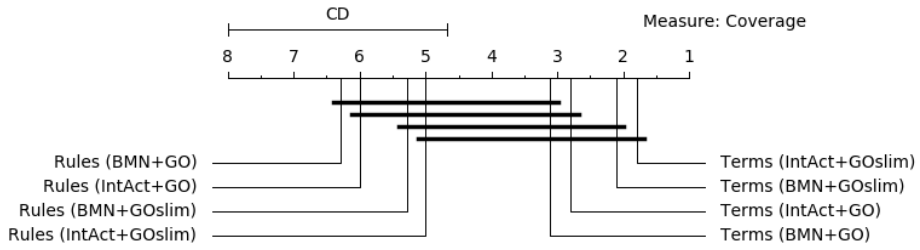

Fig. 10 Coverage results for enrichment based on components.

rules which cover larger portion of the input set emerged. This result is expected, as GO Slim consists of less, more general terms compared to whole GO.

5.4.7 Results summary

The presented quantitative results indicate the dominance of term-based approaches in terms of coverage. Using community based partition on networks with different types of nodes has proven beneficial both in terms of WRAcc and information

content. We conclude that using multiple different types of nodes—multiple views—increases the partition qualities and results in better rules.

6 Using CBSSD in two knowledge discovery tasks

This section demonstrates the use of the proposed methodology on two real world data sets from the life science domain. First, we consider the properties of amino-acid variants within protein binding sites, followed by cancer related transcription factors identified in the context of epigenetics.

6.1 Discovery of properties of proteins with single amino-acid variants present in the binding sites

Sequence variants are nucleotide or amino acid substitutions that can lead to unstable protein interaction complexes and thus influence the organism's phenotype (e.g., induce a disease state). There are two main types of variants: polymorphisms or germ-line variants that are heritable, and somatic mutations that appear in somatic tissues without previous genetic encoding. Although it was demonstrated that variants within biological interactions can be associated with disease occurrence [8, 62, 52, 34], currently there are no studies of this phenomenon aimed at discovering new subgroups of proteins associated with variants within interaction sites at a more general level.

We use the results from a previous enrichment analysis study [8] for comparison with the proposed CBSSD methodology. Enrichment analysis in the context of this study is concerned with the identification of single significant terms, associated with the studied phenomenon. The results are compared based on the terms appearing in both approaches, i.e. terms found as a result of enrichment analysis as well as as a result of semantic subgroup discovery. As the two compared approaches are fundamentally different, the intersection of both results is expected to be only a few highly significant terms).

More than 300 UniProt terms for which variants were found within protein binding sites were used as the input query list (found in supplementary material of [8]). A BioMine knowledge graph with more than 1,650 nodes and 2,300 edges was constructed. The resulting network is shown in Figure 11.[3]

Triplet construction consists of first mapping the nodes from the knowledge graph to the associated ontology terms, followed by the construction of the background knowledge. In this application, the Gene ontology [3] was used in both steps. Semantic subgroup discovery was conducted for more than 20 communities, and as the main result more than 100 rules of various lengths were obtained. The most significant and longest rules were manually inspected to identify possible overlap with previous pathway enrichment studies done on the same input data set. Different beam sizes were experimented with in the procedure (from 10 to 50).

The obtained rule sets for the identified communities were further inspected. We directly compared the ontology terms present in the rules with the terms identified as significant in our previous study [8]. For this naïve comparison, conjuncts were

[3] Plotted with the Py3Plex library (`https://github.com/SkBlaz/Py3Plex`).

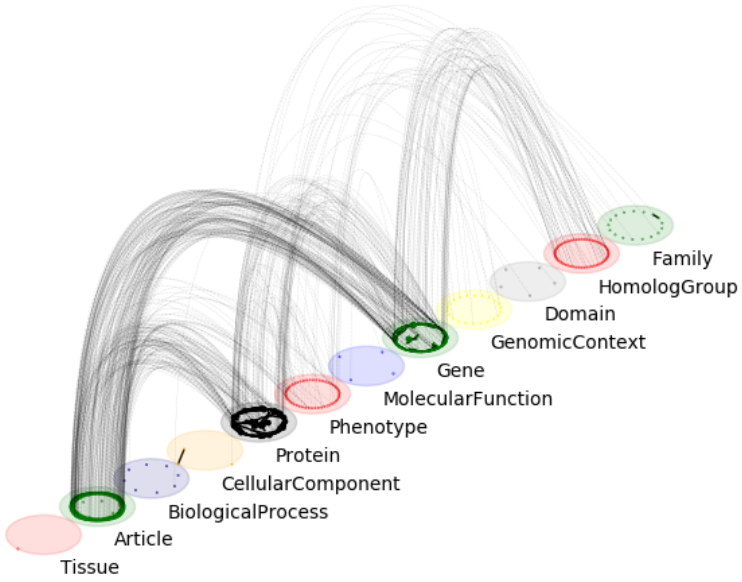

Fig. 11 The BioMine network associated with polymorphisms located within protein interaction sites.

considered as individual entries, as we were only interested in term presence (not coverage). There were 13 gene ontology terms present in both approaches (Table 3).

Although only 13 terms were found with both procedures, the identified terms were among the most significant ones detected in the enrichment analysis setting. This indicates, that both procedures identified a strong signal related to DNA and cell cycle related processes. As semantic subgroup discovery was conducted for separate communities, the results were expected to be more detailed and comprehensive. This was indeed the case: given that many CBSSD rules consist of two conjuncts,

Table 3 Gene ontology terms, found both in enrichment and semantic rule learning process. Terms marked with * emerged as the most relevant to semantic subgroup discovery.

Gene ontology term	Meaning
GO:0000077	DNA damage checkpoint*
GO:0000086	Mitotic cell cycle*
GO:0003677	DNA binding*
GO:0004871	Signal transducer activity*
GO:0005730	Nucleolus*
GO:0005814	Centriole
GO:0016020	membrane
GO:0016605	PML body
GO:0030018	Z-disc
GO:0035264	Multicellular organism growth
GO:0045892	Negative regulation of transcription (DNA)
GO:0000122	Negative regulation of transcription (RNA)
GO:0000785	Chromatin

these rules are potentially more informative than the ones identified by ontology enrichment analysis. As iron binding proteins were present in the protein list (this was known from the previous study [8]), rule $R = \textit{GO:0034618} \wedge \textit{GO:0006874}$ appeared as one of the most significant rules ($p < 0.1$). Ontology terms in this rule represent arginine binding and cellular calcium homeostasis—both processes described by terms annotating nodes from the input list representing a term combination not detected with conventional enrichment analysis. The key UniProt term found for this rule was P41180 (CASR), which represents the extracellular calcium-sensing receptor [26]. As CASR is indeed critical for calcium homeostasis discovery (GO:0006874), it confirms the validity of our CBSSD approach. The second term (GO:0034618), representing arginine binding is not so directly associated with the CASR protein. To further investigate the context within which GO:0034618 occurs, we queried the gene ontology database directly for similar proteins, already associated with this term. The majority of proteins annotated with this term correspond to acetylglutamate kinase, an enzyme that participates in the metabolism of amino acids (e.g., urea cycle). A possible interpretation of this association is that the CASR protein induces hormonal response, which could effectively lead to increased amino-acid metabolism, providing the molecular components necessary for establishment of homeostasis. This association serves as a possible candidate for further experimental testing and demonstrates the hypothesis generation capabilities of proposed approach.

Another interesting rule emerged from the first identified community, i.e. the rule $\textit{GO:0030903} \wedge \textit{GO:0000006}$ was found for UniProt entries Q96SN8 (CDK5 regulatory subunit-associated protein 2), O94986 (Centrosomal protein), Q9HC77 (Centromere protein J) and O43303 (Centriolar coiled-coil protein). It can be observed that all the identified proteins are connected with nucleus-related processes. Term *GO:0030903* corresponds to notochord development, which is a stage in cell division—a term directly associated with the identified proteins. The second term, *GO:0000006*, corresponds to high-affinity zinc uptake transmembrane transporter activity, a process related to enzyme system responsible for cell division and proliferation. Although this rule does not imply any new hypothesis, it demonstrates the generalization capability of the proposed approach.

Many terms are specific to either semantic rule discovery based on community detection or enrichment analysis. This discrepancy appears due to the fact that community detection splits the input term list into smaller lists, which can be described by completely different terms than the list as a whole. As the proposed methodology splits the input list, it is not sensible to compare it with conventional approaches, which operate on whole lists. Both approaches cover approximately the same percentage of input terms. The CBSSD's coverage is 12.02% with 218 GO terms, whereas the term coverage for conventional enrichment is 12.3% with 881 GO terms. The term discrepancy serves only as a proof of fundamental difference between the two approaches. Nevertheless, we demonstrate that our approach is a useful complementary methodology to the well established enrichment analysis.

6.2 Grouping of cancer-related epigenetic factors

Epigenetics is a field where processes such as methylation are studied in the context of the influence of environment on the phenotype. Epigenetic factors are actively researched and are constantly updated in databases such as emDB [44], where in-

formation such as gene expression, tissue information and variant information is publicly accessible. We tested the developed approach on the list of many currently known epigenetic factors related to cancer. The epigenetics data set was chosen for two main reasons: first, to demonstrate the CBSSD's performance on a data set, to our knowledge not yet used in semantic subgroup discovery, and second, this data set serves to further test the developed methodology in the context of different biological process. The 153 distinct UniProt terms were used as input for the BioMine knowledge graph construction. The final graph consisted of approximately 4,500 nodes and 5,500 edges, respectively. The obtained knowledge graph is significantly larger than the one used in the previous case study (properties of SNVs in binding sites) and thus demonstrates the capabilities of the developed approach on larger graphs.

Using InfoMap, more than 50 communities were identified. These communities were further inspected. For the community including UniProt term Q8WTS6 (Histone-lysine N-methyltransferase), many interesting rules were detected by the CBSSD approach. For example, rule *GO:1990785* $\wedge$ *GO:0000975* $\wedge$ *GO:0000082* (with $p = 0.09$) indicates that the protein is indeed highly associated with epigenetic processes. Term *GO:1990785* describes water-immersion restraint stress, term *GO:0000975* regulatory region DNA binding and term *GO:0000082* transition of mitotic cell cycle. All three terms describe the *Q8WTS6* entry, as it effects the DNA's topological properties (coil formation) and is responsible for transcriptional activation of genes, which code for collagenases, enzymes crucial to mitotic cell cycle (wall formation).

To further analyze CBSSD's generalization capabilities, we plotted all the rules (discovered by CBSSD) for the individual communities (identified by InfoMap) against all the GO terms identified as enriched by the DAVID Bioinformatics Suite [32]. As this experiment is conducted using only the terms, previously identified as significant by DAVID, CBSSD's significance threshold was relaxed to $p = 0.5$. This relaxation was introduced to enable the discovery of more interesting patterns, which would otherwise be considered noise or false positive results.

The semantic landscape obtained in this experiment is depicted in Figure 12. For an expert defined list of genes coding for cancer-related epigenetic regulators, the rows of the visualized matrix correspond to enriched GO terms discovered by DAVID, while the columns represent the terms present in rules discovered by the CBSSD approach. In particular, each column represents a community detected by the InfoMap algorithm, while the matrix cells of the given column represent all the terms appearing in any of the rules describing the given community. The number of columns equals the number of communities detected by InfoMap. The red rectangles represent the terms present in any of the rules composed of a conjunction of at least GO terms. The green rectangles correspond to terms identified by DAVID and appearing in simple (single term) CBSSD rules. Rows, located in the uppermost part of the matrix represent the most general GO terms.

It can be observed (see the enlarged inset image in Figure 12) that only a couple of previously identified GO terms correspond to multi-term CBSSD rules (red rectangles), where by multi-term rules we denote rules consisting of conjuncts of several GO terms. Terms, such as *GO:0000118* represent very high level terms, associated with majority of epigenetics-related processes. Such terms are most commonly included in more complex rules, consisting of conjuncts of several GO terms. Only a handful of GO terms serve as a basis for more complex rules (this is observed by seeing only a few lines in the matrix containing red rectangles). For example,

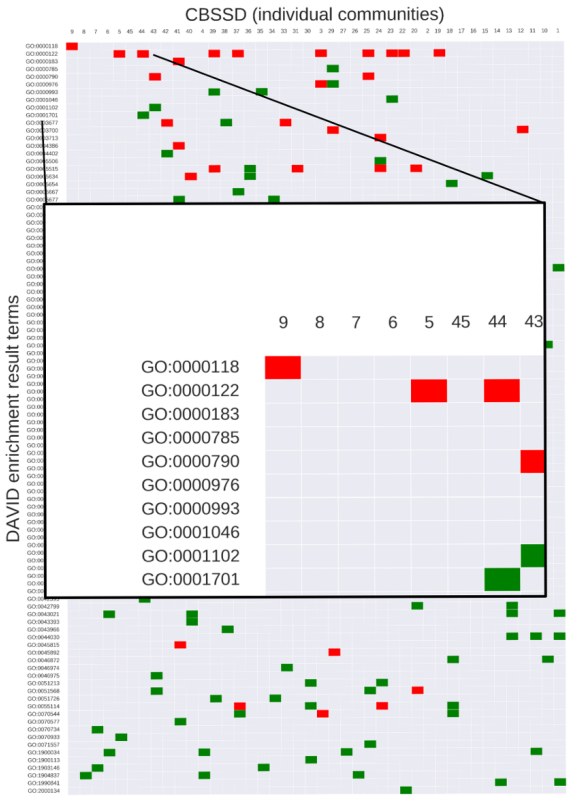

Fig. 12 Visualizing the GO terms appearing in CBSSD discovered rules compared to the GO terms discovered by the DAVID Bioinformatics Suite on an expert defined list of genes coding for cancer-related epigenetic regulators.

one of these terms is *GO:0000118*, which represents the Hystone deacetylase complex, one of the key mechanisms for hystone structure regulation. Other terms involved in multi-term rules include *GO:0000112*, representing negative regulation of transcription from RNA polymerase II promoter, a mechanism by which many epigenetic regulators influence the transcription patterns, *GO:0000183*, representing chromatin silencing at rDNA, *GO:0000785* and *GO:0000790*, representing chromatin in general, *GO:0000976*, representing transcription regulatory region sequence-specific DNA binding and *GO:0001046*, which represents core promoter sequence-specific DNA binding. The described terms are all fundamentally associated with epigenetic regulation, which proves that CBSSD is able to use the more general terms to construct meaningful rules.

Overall, 27% of all significant terms identified via conventional enrichment analysis by DAVID were also found with the CBSSD algorithm. Such low percentage is expected, as CBSSD builds upon individual subsets of the larger set of terms found in conventional enrichment analysis. This result implies that higher level terms are similar in both approaches, yet CBSSD identified latent patterns, which can not be

detected via conventional enrichment analysis. The higher level terms appear to form the base for more complex rules. Similar behavior was reported as a result of the SegMine methodology [48], which similarly to CBSSD, yields explanatory power of rules in order to find enriched parts of input term lists.

Coverage-wise, both conventional enrichment, as well as CBSSD perform the same, as the CBSSD's coverage is 96.7% with 230 GO terms, whereas the term coverage for conventional enrichment is 96.7% with 360 GO terms. Similarly to the case study one, CBSSD needed fewer GO terms to cover approximately the same percentage of input term list.

7 Discussion and further work

The quantitative evaluation of different enrichment settings indicates that the rules discovered by CBSSD can represent patterns, otherwise missed by conventional enrichment analysis approaches. In terms of coverage, conventional enrichment approaches dominate. A possible explanation for such behaviour is that more significant terms are identified (compared to rules), and shorter rules (1 term) imply more general rules and larger coverage. Further, the probability of a random rule, composed of multiple terms is smaller compared to single terms discovered by conventional enrichment approaches. The larger the number of terms in a single rule, the smaller the probability the rule will emerge as significant.

The results imply that rule learning through semantic subgroup discovery can be used in parallel with term enrichment in order to maximize the number of interesting patterns found.

With regard to WRAcc, we demonstrate that automatically induced BioMine networks yield better rule sets compared to IntAct network. This result serves as an additional confirmation that the community-based heterogeneous network partitioning yields better rules. Understanding the meaning of topological structures, which emerge from large complex networks remains an open problem. We demonstrated that larger networks (IntAct) can also be used as input for CBSSD.

The issue we did not address in this study is the process of obtaining the input (i.e. the gene list) at the first place. We believe this step is entirely problem specific, and can as such not be implemented in the existing CBSSD methodology. In the limit, all known proteins can be used as the input. In such a scenario, the CBSSD approach would yield enrichment of a network's partitions in terms of all nodes. In this work we do not focus on this task, yet current state-of-the-art high-throughput experimental methods already yield large, species-specific interaction networks, which could benefit from the generalized version of the CBSSD that would consider all the nodes. Recent improvements in the sequencing technology offer extensive amounts of gene-gene interaction networks coming from the field of metagenomics. We leave the case studies related to this topics for further work.

We believe that approaches concerned with network analysis could benefit by using CBSSD methodology. As it is currently not well understood for example, how protein-ligand binding sites can be understood via structural similarity analysis [54, 50], multi-conjunct descriptions of topological features, which emerge in such networks could offer novel insights.

Semantic data mining is an emerging field, where background knowledge in the form of ontologies can be used to generalize the rules emerging from the learning

process. In this study, we demonstrate how such an approach can be used to induce rules describing the communities and components, detected on an automatically constructed knowledge graph. Our implementation was tested on two data sets from the life science domain, where the validity of the most significant rules was manually inspected in terms of biological context. This approach works for up to 6,000 nodes of interest in reasonable time (e.g., in a day), but for more (e.g., 10,000 nodes), whole graphs should be used from the beginning, if possible. As the number of rules produced can be large, adequate rule visualization techniques for elegant result inspection are still to be developed.

The proposed CBSSD methodology is to our knowledge one of the first attempts, where we address the issue of learning from complex networks by leveraging semantic subgroup discovery. Further, the developed approach is scalable, and offers the opportunity to investigate interaction between different semantic (GO) terms.

We currently see CBSSD as a complementary methodology to enrichment analysis, as it is capable of describing latent patterns beyond the ones expected by domain experts.

Further work regarding CBSSD includes incorporation of ontology, as well as network reduction techniques to speed the rule discovery even more. Further, it remains an open problem as to how the obtained results can be visualized. Finally, CBSSD will be extended to other forms of symbolic learning, as for example association rules similarly remain poorly investigated in the context of learning from complex networks.

Availability

The Community-based subgroup discovery reference implementation is freely available at `https://github.com/SkBlaz/CBSSD`. It relies on primitives for ontology processing, network construction and analysis available as part of `https://github.com/SkBlaz/Py3plex`.

Acknowledgments

We are grateful to Marko Robnik-Šikonja and to anonymous reviewers for valuable comments and suggestions that helped us to refine the paper. This research was funded by the Slovenian Research Agency funded research projects *HinLife: Analysis of Heterogeneous Information Networks for Knowledge Discovery in Life Sciences* (J7-7303) and *Semantic Data Mining for Linked Open Data* (financed under the ERC Complementary Scheme, N2-0078).

References

1. Adhikari PR, Vavpetič A, Kralj J, Lavrač N, Hollmén J (2016) Explaining mixture models through semantic pattern mining and banded matrix visualization. Machine Learning 105(1):3–39

2. Alexeyenko A, Lee W, Pernemalm M, Guegan J, Dessen P, Lazar V, Lehtiö J, Pawitan Y (2012) Network enrichment analysis: extension of gene-set enrichment analysis to gene networks. BMC Bioinformatics 13(1):226, DOI 10.1186/1471-2105-13-226, URL https://doi.org/10.1186/1471-2105-13-226
3. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, et al (2000) Gene ontology: Tool for the unification of biology. Nature genetics 25(1):25–29
4. Balcan N, Blum A, Mansour Y (2013) Exploiting structures and unlabeled data for learning. In: ICML'13 Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28, pp 1112–1120
5. Belleau F, Nolin MA, Tourigny N, Rigault P, Morissette J (2008) Bio2rdf: Towards a mashup to build bioinformatics knowledge systems. Journal of Biomedical Informatics 41(5):706–716
6. Benjamini Y, 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) pp 289–300
7. Benson DA, Cavanaugh M, Clark K, Karsch-Mizrachi I, Lipman DJ, Ostell J, Sayers EW (2012) Genbank. Nucleic Acids Research 41(D1):D36–D42
8. Škrlj Blaž, Janez K, Tanja K (2017) Identification of sequence variants within experimentally validated protein interaction sites provides new insights into molecular mechanisms of disease development. Molecular Informatics 36(9):1700017, DOI 10.1002/minf.201700017
9. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment 2008(10):P10008
10. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. Journal of statistical mechanics: theory and experiment 2008(10):P10008
11. Chen G, Wang X, Li X (2014) Fundamentals of complex networks: models, structures and dynamics. John Wiley & Sons
12. Clauset A, Newman ME, Moore C (2004) Finding community structure in very large networks. Physical review E 70(6):066111
13. Cohen R, Havlin S (2010) Complex Networks: Structure, Robustness and Function. Cambridge University Press
14. Consortium GO (2004) The gene ontology (go) database and informatics resource. Nucleic acids research 32(suppl_1):D258–D261
15. Consortium U, et al (2017) Uniprot: the universal protein knowledgebase. Nucleic acids research 45(D1):D158–D169
16. De Domenico M, Lancichinetti A, Arenas A, Rosvall M (2015) Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. Physical Review X 5(1):011027
17. Demšar J, Curk T, Erjavec A, Gorup Č, Hočevar T, Milutinovič M, Možina M, Polajnar M, Toplak M, Starič A, et al (2013) Orange: data mining toolbox in python. The Journal of Machine Learning Research 14(1):2349–2353
18. Ding D, Sun X (2017) A comparative study of network motifs in the integrated transcriptional regulation and protein interaction networks of shewanella. Network 8:9
19. Dong X, Hao Y, Wang X, Tian W (2016) Lego: a novel method for gene set overrepresentation analysis by incorporating network-based gene weights. Scientific

reports 6:18871
20. Drummond AJ, Rambaut A (2007) Beast: Bayesian evolutionary analysis by sampling trees. BMC Evolutionary Biology 7(1):214
21. Duch J, Arenas A (2005) Community detection in complex networks using extremal optimization. Physical Review E 72(2):027104
22. Džeroski S, Lavrač N (eds) (2001) Relational Data Mining. Springer
23. Eronen L, Toivonen H (2012) Biomine: predicting links between biological entities using network models of heterogeneous databases. BMC Bioinformatics 13(1):119, DOI 10.1186/1471-2105-13-119, URL http://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-13-119
24. Fürnkranz J, Gamberger D, Lavrač N (2012) Foundations of rule learning. Springer Science & Business Media
25. Gardner M (1978) Bells-versatile numbers that can count partitions of a set, primes and even rhymes. Scientific American 238(5):24
26. Garrett JE, Capuano IV, Hammerland LG, Hung BC, Brown EM, Hebert SC, Nemeth EF, Fuller F (1995) Molecular cloning and functional expression of human parathyroid calcium receptor cdnas. Journal of Biological Chemistry 270(21):12919–12925
27. Glaab E, Baudot A, Krasnogor N, Schneider R, Valencia A (2012) Enrichnet: network-based gene set enrichment analysis. Bioinformatics 28(18):i451–i457, DOI 10.1093/bioinformatics/bts389, URL http://dx.doi.org/10.1093/bioinformatics/bts389, /oup/backfile/content_public/journal/bioinformatics/28/18/10.1093_bioinformatics_bts389/2/bts389.pdf
28. Guarino N, Oberle D, Staab S (2009) What Is an Ontology? In: Handbook on Ontologies, Springer, pp 1–17
29. Hmimida M, Kanawati R (2015) Community detection in multiplex networks: A seed-centric approach. NHM 10(1):71–85
30. Hosack DA, Dennis G, Sherman BT, Lane HC, Lempicki RA (2003) Identifying biological themes within lists of genes with ease. Genome biology 4(10):R70
31. Hotho a, Staab S, Stumme G (2003) Ontologies improve text document clustering. Third IEEE International Conference on Data Mining pp 2–5, DOI 10.1109/ICDM.2003.1250972
32. Huang DW, Sherman BT, Tan Q, Kir J, Liu D, Bryant D, Guo Y, Stephens R, Baseler MW, Lane HC, et al (2007) David bioinformatics resources: Expanded annotation database and novel algorithms to better extract biology from large gene lists. Nucleic Acids Research 35(2):W169–W175
33. Huffman DA (1952) A method for the construction of minimum-redundancy codes. Proceedings of the IRE 40(9):1098–1101, DOI 10.1109/JRPROC.1952.273898
34. Kamburov A, Lawrence MS, Polak P, Leshchiner I, Lage K, Golub TR, Lander ES, Getz G (2015) Comprehensive assessment of cancer missense mutation clustering in protein structures. Proceedings of the National Academy of Sciences 112(40):E5486–E5495
35. Kanehisa M, Goto S (2000) KEGG: Kyoto encyclopedia of genes and genomes. Nucleic Acids Research 28(1):27–30
36. Kuncheva Z, Montana G (2015) Community detection in multiplex networks using locally adaptive random walks. In: Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on, IEEE, pp 1308–1315

37. Langohr L, Podpečan V, Petek M, Mozetič I, Gruden K, Lavrač N, Toivonen H (2012) Contrasting subgroup discovery. The Computer Journal 56(3):289–303
38. Lavrač N, Vavpetič A (2015) Relational and semantic data mining. In: Proceedings of the Thirteenth International Conference on Logic Programming and Nonmonotonic Reasoning, Lexington, KY, USA, pp 20–31
39. Lavrač N, Džeroski S (1994) Inductive Logic Programming: Techniques and Applications
40. List M, Alcaraz N, Dissing-Hansen M, Ditzel HJ, Mollenhauer J, Baumbach J (2016) Keypathwayminerweb: online multi-omics network enrichment. Nucleic Acids Research 44(W1):W98–W104, DOI 10.1093/nar/gkw373, URL http://dx.doi.org/10.1093/nar/gkw373, /oup/backfile/content_public/journal/nar/44/w1/10.1093_nar_gkw373/3/gkw373.pdf
41. Liu H, Dou D, Jin R, LePendu P, Shah N (2013) Mining biomedical ontologies and data using rdf hypergraphs. In: Machine Learning and Applications (ICMLA), 2013 Proceedings of the 12th International Conference on, IEEE, vol 1, pp 141–146
42. Malliaros FD, Vazirgiannis M (2013) Clustering and community detection in directed networks: A survey. Physics Reports 533(4):95–142
43. MARC T, ŠUBELJ L (2018) Convexity in complex networks. Network Science p 1–28, DOI 10.1017/nws.2017.37
44. Nanda JS, Kumar R, Raghava GP (2016) dbem: A database of epigenetic modifiers curated from cancerous and normal genomes. Scientific reports 6:19340
45. Novak PK, Lavrač N, Webb GI (2009) Supervised descriptive rule discovery: A unifying survey of contrast set, emerging pattern and subgroup mining. Journal of Machine Learning Research 10(Feb):377–403
46. Orchard S, Ammari M, Aranda B, Breuza L, Briganti L, Broackes-Carter F, Campbell NH, Chavali G, Chen C, Del-Toro N, et al (2013) The mintact project—intact as a common curation platform for 11 molecular interaction databases. Nucleic acids research 42(D1):D358–D363
47. Palla G, Derényi I, Farkas I, Vicsek T (2005) Uncovering the overlapping community structure of complex networks in nature and society. arXiv preprint physics/0506133
48. Podpečan V, Lavrač N, Mozetič I, Novak PK, Trajkovski I, Langohr L, Kulovesi K, Toivonen H, Petek M, Motaln H, et al (2011) Segmine workflows for semantic microarray data analysis in orange4ws. BMC Bioinformatics 12(1):416
49. Rosvall M, Axelsson D, Bergstrom CT (2009) The map equation. The European Physical Journal-Special Topics 178(1):13–23
50. Sardiu ME, Gilmore JM, Groppe B, Florens L, Washburn MP (2017) Identification of topological network modules in perturbed protein interaction networks. Scientific Reports 7:43845
51. Schipper HM, Maes OC, Chertkow HM, Wang E (2007) Microrna expression in alzheimer blood mononuclear cells. Gene regulation and systems biology 1:GRSB–S361
52. Schröder NW, Schumann RR (2005) Single nucleotide polymorphisms of toll-like receptors and susceptibility to infectious disease. The Lancet Infectious Diseases 5(3):156–164
53. Škrlj B, Kralj J, Vavpetič A, Lavrač N (2018) Community-Based Semantic Subgroup Discovery. In: Appice A, Loglisci C, Manco G, Masciari E, Ras ZW (eds) New Frontiers in Mining Complex Patterns, Springer International Publishing,

Cham, pp 182–196
54. Škrlj B, Kunej T, Konc J (2018) Insights from ion binding site network analysis into evolution and functions of proteins. Molecular Informatics
55. Strogatz SH (2001) Exploring complex networks. Nature 410(6825):268
56. Subramanian A, Tamayo P, Mootha VK, Mukherjee S, Ebert BL, Gillette MA, Paulovich A, Pomeroy SL, Golub TR, Lander ES, et al (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. Proceedings of the National Academy of Sciences 102(43):15545–15550
57. Tipney H, Hunter L (2010) An introduction to effective use of enrichment analysis software. Human Genomics 4(3):1
58. Vavpetič A, Lavrač N (2012) Semantic subgroup discovery systems and workflows in the sdm-toolkit. The Computer Journal 56(3):304–320
59. Vavpetič A, Novak PK, Grčar M, Mozetič I, Lavrač N (2013) Semantic data mining of financial news articles. In: Proceedings of the International Conference on Discovery Science, Springer, pp 294–307
60. Vavpetič A (2017) Semantic subgroup discovery. PhD thesis, Jožef Stefan International Postgraduate School
61. Vrabič Rok HD, Butala P (2012) Discovering autonomous structures within complex networks of work systems. CIRP Annals-Manufacturing Technology 61(1):423–426
62. Škrlj B, Kunej T (2016) Computational identification of non-synonymous polymorphisms within regions corresponding to protein interaction sites. Computers in Biology and Medicine 79:30–35, DOI 10.1016/j.compbiomed.2016.10.003, URL `https://doi.org/10.1016/j.compbiomed.2016.10.003`
63. Zhao J, Xie X, Xu X, Sun S (2017) Multi-view learning overview: Recent progress and new challenges. Information Fusion 38:43 – 54, DOI https://doi.org/10.1016/j.inffus.2017.02.007, URL `http://www.sciencedirect.com/science/article/pii/S1566253516302032`

Appendix A Multiplex InfoMap algorithm

As Community-Based Semantic Subgroup Discovery operates on multilayer networks, i.e. networks consisting of multiple node types, we provide additional explanation how the derived InfoMap algorithm is used for community detection in multilayer networks, initially presented in [16]. Let M denote a partition of state nodes i assigned to communities $l = 1, 2, \ldots, m$. Each node is a part of a layer denoted here with Greek letters (e.g., α). For example, $q_{ij}^{\alpha\beta}$ corresponds to the transition rate between the node i in α layer and node j in the β layer. The transition rates with which a random walker enters ($q_{l\curvearrowleft}$) and exits ($q_{l\curvearrowright}$) a community can be defined as

$$q_{l\curvearrowleft} = \sum_{\{i,\alpha\}\in J\neq V, \{j,\beta\}\in V} q_{ij}^{\alpha\beta} \tag{11}$$

$$q_{l\curvearrowright} = \sum_{\{i,\alpha\}\in V, \{j,\beta\}\in J\neq V} q_{ij}^{\alpha\beta} \tag{12}$$

where J and V denote two different layers and i, j denote two different nodes. The codewords are based on physical node visits. For codebook l the physical node visits

are denoted as

$$p_{i\in l} = \sum_{\{i,\alpha\}\in l} p_i^{\alpha}. \tag{13}$$

The $p_{\circ}$ is defined as the sum of the exit rates, $p_{\circ} = \sum_l q_{l\curvearrowright}$. The normalized visit probability distribution is redefined as $P^l = \{p_{i\in l}/p_{l\circ}\}$. The Q distribution is similarly redefined to $Q = \{q_{l\curvearrowright}/q_{\curvearrowright}\}$. The multilayer map equation can be defined as follows:

$$L(M) = q_{l\curvearrowright} H_q + \sum_{i=1}^{m} p_{\circlearrowright}^{i} H_i \tag{14}$$

Although community detection represents one of the most commonly used network partition methods used in analysis of complex networks, the proposed approach is by no means limited to learning from communities. Currently, the proposed implementation also supports partition based on a network's components—connected sub-networks. Further, arbitrary network partition function can also be specified as part of the input. In this work we mostly focus on community-based partitions, as they have been proven to correspond to causal patterns in systems, ranging from biological networks, transportation to social networks [47].

Appendix B The Hedwig algorithm

In this section we describe in detail the two main procedures used in the Hedwig semantic rule induction algorithm [59, 60]. The Hedwig algorithm is capable of using domain ontologies to formulate a generalized hypothesis. Its result are descriptive rules that describe individual parts of the input data set. Initially, a RDF-based hierarchy is used to construct the hierarchical relations between instances, further used in the rule induction step. The two key procedures of the Hedwig semantic subgroup discovery algorithm are presented in Algorithms 2 and 3, respectively.

Input : Input examples E, background knowledge B, target class value c, beam size k, p-value threshold α
Output: Set of rules

rules ← [`default_rule(`*E, c, B*`)`]

while `improvement(`*rules*`)` **do**
  ; ▷ `Add specializations of each rule to the beam`
  for *rule* ∈ *rules* **do**
    `extend(`*rules,* `specialize(`*rule, B*`))`
  end
  rules ← `best(`*rules, k*`)` ; ▷ `Select the top` *k* `rules`
end
rules ← `validate(`*rules,* α`)` ; ▷ `Significance testing`

return *rules*

Algorithm 2: Hedwig's `induce`(E, B, c, k, α) procedure.

```
Input  : Rule to specialize rule, background knowledge B
Output: Set of specializations of rule

specializations ← []
; ▷ Predicates that can be specialized
eligible_preds ← eligible(predicates(rule))

for predicate ∈ eligible_preds do
    ; ▷ Specialize by traversing the subClassOf hierarchy
    for subclass ∈ subclasses(predicate, B) do
        new_rule ← swap(rule, predicate, subclass)
        if can_specialize(new_rule) then
            append(specializations, new_rule)
        end
    end
    ; ▷ Specialize by negating
    new_rule ← negate(rule, predicate)
    if can_specialize(new_rule) then
        append(specializations, new_rule)
    end
end
if rule ≠ default_rule then
    ; ▷ Specialize by adding a new unary predicate
    new_predicate ← next_non_ancestor(eligible_preds)
    new_rule ← append(rule, new_predicate)
    if can_specialize(new_rule) and non_redundant(new_rule) then
        append(specializations, new_rule)
    end
end
; ▷ Specialize by adding new binary predicates
if is_unary(last(predicates(rule))) then
    extend(specializations, specialize_binary(new_rule))
end
return specializations
```

Algorithm 3: Hedwig's `specialize`(*rule*, *B*) procedure.