Journal of Intelligent Information Systems (2019) 53:265–304
https://doi.org/10.1007/s10844-019-00545-0

CBSSD: community-based semantic subgroup discovery

Blaž Škrlj[1,2] · Jan Kralj[2] · Nada Lavrač[2,3]

Received: 29 June 2018 / Revised: 13 January 2019 / Accepted: 17 January 2019 /
Published online: 26 January 2019

Abstract
Modern data mining algorithms frequently need to address the task of learning from heterogeneous data, including various sources of background knowledge. A data mining task where ontologies are used as background knowledge in data analysis is referred to as semantic data mining. A specific semantic data mining task is semantic subgroup discovery: a rule learning approach enabling ontology terms to be used in subgroup descriptions learned from class labeled data. This paper presents Community-Based Semantic Subgroup Discovery (CBSSD), a novel approach that advances ontology-based subgroup identification by exploiting the structural properties of induced complex networks related to the studied phenomenon. Following the idea of multi-view learning, using different sources of information to obtain better models, the CBSSD approach can leverage different types of nodes of the induced complex network, 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 demonstrate that the CBSSD approach is scalable, applicable to large complex networks, and that it can be used to identify significant combinations of terms, which can not be uncovered by contemporary term enrichment analysis approaches.

Keywords Semantic data mining · Ontologies · Community detection · Network analysis · Bioinformatics · Term enrichment analysis

✉ Nada Lavrač
nada.lavrac@ijs.si

Blaž Škrlj
blaz.skrlj@ijs.si

Jan Kralj
jan.kralj@ijs.si

[1] Jožef Stefan International Postgratuate School, Jamova 39, 1000 Ljubljana, Slovenia

[2] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

[3] University of Nova Gorica, Glavni trg 8, 5271 Vipava, Slovenia

1 Introduction

Modern machine learning approaches are capable of using continuously increasing amounts of information to explain complex systems in numerous fields, including biology, sociology, mechanics and electrical engineering. As there can be many distinct types of data associated with a single system, novel approaches strive towards the integration of different, heterogeneous data and knowledge sources used as data in learning predictive or descriptive models (Chen et al. 2014).

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 performance of the learned model. The incorporated background knowledge can be implicit or explicit. For example, Bayesian methods can be leveraged to incorporate implicit 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 (Drummond and Rambaut 2007). 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 (RDM) (Džeroski and Lavrač 2001) and inductive logic programming (ILP) (Muggleton 1991; Lavrač and Džeroski 1994), where the background knowledge is used along with the training 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 construction process. A data mining task where ontologies are used as background knowledge in data analysis is referred to as semantic data mining (SDM), a field at the intersection of inductive rule learning and semantic web. We refer the interested reader to Dou et al. (2015), Lavrač and Vavpetič (2015), and Ławrynowicz (2017) for a detailed overview of the field of semantic data mining. A specific semantic data mining task is semantic subgroup discovery (SSD) (Langohr et al. 2012; Vavpetič et al. 2013), which is a rule learning approach enabling ontology terms to be used in subgroup descriptions learned from class labeled data. Here, class labels denote the target groups for which individual descriptive rules are learned.

In this work, in addition to ontologies, we use the formalism of complex networks to represent the interactions in the studied phenomenon (Cohen and Havlin 2010). Complex networks consist of nodes (e.g., proteins) and edges (e.g., interactions between proteins). Real world networks often contain communities, or other topological structures of interest, which correspond to functional properties of the network (Strogatz 2001; Duch and Arenas 2005).

This paper presents the Community-Based Semantic Subgroup Discovery (CBSSD) approach as means to advance ontology-based subgroup identification by taking into account also the structural properties of induced complex networks related to the studied system. To the best of our knowledge, Community-based Semantic Subgroup Discovery is the first approach to learning from complex networks by using semantic subgroup discovery. In the CBSSD methodology, iteratively constructed complex networks are used as input to identify relevant subgroups through network partitioning, followed by semantic subgroup discovery.

We experimentally demonstrate that new knowledge can be obtained using the existing, freely accessible heterogeneous data in the form of complex networks and ontologies.

The experimental results demonstrate that the CBSSD approach is scalable, and offers the opportunity to investigate the interactions between different semantic terms (e.g., genes and gene groups encoded in the gene Ontology). The approach is applicable to large complex networks and can be used to identify significant combinations of terms, which can not be uncovered by contemporary term (gene) enrichment analysis approaches.

This paper is a significant extension of our previous work (Škrlj et al. 2018a). It more thoroughly describes the theoretical background and contributes 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 network partitioning. Moreover, we also perform a quantitative evaluation of the CBSSD methodology by comparing it to standard enrichment analysis approaches.

After presenting the related work in Section 2, Section 3 presents the background technologies. The CBSSD methodology is presented in Section 4. In Section 5 we evaluate the use of the CBSSD approach on ten different life science data sets, i.e. expert defined gene sets, where the CBSSD methodology is quantitatively compared to existing enrichment analysis approaches. Section 6 demonstrates the utility of the CBSSD methodology on two real world data sets from the life science domain. The experimental evaluation of the methodology is followed by discussing the results and presenting the plans for further work in Section 7.

2 Related work

This section introduces the relevant concepts and presents the related work in the fields of complex networks, knowledge graphs and ontologies, enrichment analysis, semantic data mining, semantic 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 (Palla et al. 2005; Vrabič Rok and Butala 2012). Complex networks are graphs with distinct, non-trivial real world topological properties (Cohen and Havlin 2010). Real world networks can be characterized with the statistical properties regarding their node degree distribution, component distribution or connectivity (Strogatz 2001).

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 an active research field on its own. 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 (Ding and Sun 2017). 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 (Duch and Arenas 2005). Communities can be detected with random walk-based sampling, spectral graph properties or other network

properties (Malliaros and Vazirgiannis 2013; Kuncheva and Montana 2015). In this work we focus on two community detection algorithms, the Louvain algorithm and the InfoMap algorithm.

2.1.1 The Louvain algorithm

The Louvain algorithm (Blondel et al. 2008), defined on undirected networks, is based on the network modularity measure Q (Clauset et al. 2004), 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. Function $\delta(c_v, c_w)$ represents the Krönecker delta function, which amounts to 1 when $c_v = c_w$ and 0 otherwise. The value of $\frac{k_v k_w}{2m}$ 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. Modularity value Q is high if most connections in the graph are between the nodes assigned to the same community. The Louvain algorithm discovers the partitioning of nodes into communities for which value Q is maximized using a greedy, non-exact procedure that runs in $\mathcal{O}(n \log(n))$. We refer the interested reader to Blondel et al. (2008) for more information on the algorithm.

2.1.2 The InfoMap algorithm

Real world networks may 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 may yield some new—otherwise non-detectable—community structure (Hmimida and Kanawati 2015). Our methodology can take into account such heterogeneity by leveraging the state-of-the-art InfoMap algorithm for multilayer community detection (Rosvall et al. 2009).

The InfoMap algorithm is based on 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 an individual community) and one *index codebook* (describing movements between the communities). Assuming the codebooks are constructed using Huffman coding (Huffman 1952) (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, and let $L(M)$ correspond 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 thus to minimize the community description length, a property corresponding to the 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:

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

where M is a partitioning of the network into communities, and $m = |M|$. Value $q_{\curvearrowleft}$ represents the total probability that the random walker enters any of the communities M. For a given community $i \in M$, $p^i_{\circlearrowleft}$ 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. Values H_q and H_i are calculated as follows:

$$H_q = -\sum_{i=1}^{m} \frac{q_{i\curvearrowleft}}{q_{\curvearrowleft}} \log\left(\frac{q_{i\curvearrowleft}}{q_{\curvearrowleft}}\right)$$

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

where $q_{i\curvearrowleft}, q_{i\curvearrowright}$ are the 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 a 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, while the computational complexity of the aforementioned algorithms is given in Appendix C.1.2.

2.2 Knowledge graphs and ontologies

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.

Large databases in the form of RDF triplets exist for many domains. For example, the Bio2RDF project (Belleau et al. 2008) 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 (Eronen and Toivonen 2012). Despite such large amounts of data being freely accessible, there remain many new opportunities to fully exploit their potential for knowledge discovery.

Knowledge graphs, built by domain experts are referred to as *ontologies* (Guarino et al. 2009). Ontologies are used in the so-called enrichment analysis approaches presented in Section 2.3, as well as in semantic data mining and semantic subgroup discovery approaches, presented in Sections 2.4 and 2.5, respectively.

2.3 Enrichment analysis

Enrichment analysis (EA) techniques are statistical methods used to identify explanations based on over- or under-representation of a given set of features. The features can, for example, represent gene expression or similar measurements, where the rows (instances) represent the individuals. For example, Schipper et al. (2007) 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) (Ashburner et al. 2000) to profile the biological role of genes, such as differentially expressed cancer genes in microarray experiments (Tipney and Hunter 2010). 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. (2012) demonstrate an extension of gene set enrichment by using gene-gene interactions, List et al. (2016) propose a component-based enrichment approach and Dong et al. (2016) propose LEGO, a network-informed enrichment approach where network-based gene weights are used.

2.4 Semantic data mining

The challenge of incorporating domain ontologies in the data mining process is addressed in *semantic data mining* (SDM) research (Dou et al. 2015; Lavrač and Vavpetič 2015; Ławrynowicz 2017). 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 (Ashburner et al. 2000). 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) (Vavpetič and Lavrač 2012) 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 (Balcan et al. 2013). 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. (2013) 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 similar to, for example, k-means family of clustering algorithms. Semantic clustering is frequently used in the area of document clustering (Hotho a and Stumme 2003).

It remains an open question whether it is possible to implement computationally feasible semantic data mining approaches, which can leverage both complex networks, as well as automatically constucted knowledge graphs or expert-curated ontologies related to the studied phenomenon, to simultaneously learn descriptions in the 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—compared to standard statistical tests such as the ones used in enrichment analysis—they consist of multiple terms.

2.5 Semantic subgroup discovery

Semantic subgroup discovery (SSD) (Langohr et al. 2012; Vavpetič et al. 2013) is a field of subgroup discovery, which uses ontologies as background knowledge in the subgroup discovery process, aimed at inducing rules from classification data, where 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 (Adhikari et al. 2016; Vavpetič et al. 2013) 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 ← 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 using different representations of the same instance during learning. Multi-view learning has become 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 (Zhao et al. 2017). In a common multi-view setting, data instances can be represented by more than a single representation. 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. The abundance of genomic information raised the need to develop general frameworks, used to fuse information from different sources (Lanckriet et al. 2004). Futher, more recent "multi-omics" approaches rely on merging biological information from e.g, the protein, gene, as well as phenotype levels to prioritize novel biomarkers (Leonavicius et al. 2019). Extensive collections of biological information have been previously analysed using ideas from multi-view learning. For example, Alexeyenko et al. (2012) propose a method, which apart from single genes computes enrichment of subsets of genes. Recently introduced KeyPathwayMinerWeb (List et al. 2016) offers similar functionality when focusing on network's components, i.e. connected subgraphs. Finally, the EnrichNet approach developed by Glaab et al. (2012) 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 necessarily 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 Background technology: formal definitions

In this section we discuss the technological background of this work through formal definitions. We start by presenting supervised rule learning and semantic subgroup discovery preliminaries, followed by network partition detection preliminaries. To make this section more compact, an interested reader can find detailed theoretical formulations and complexity analyses in the appendices.

3.1 Rule learning and semantic subgroup discovery 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 learned from given data D and class labels 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, $\epsilon : r_i \rightarrow \mathbb{R}$, that assigns each identified rule r_i a corresponding score in $\epsilon(r_i) \in \mathbb{R}$.

In this work we focus on subgroup discovery, a subfield of supervised descriptive rule induction (Novak et al. 2009). 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 given target class $t \in T$. Instead of a predictive model, the final result of descriptive learning are sets of rules, each 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:

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

Class-labeled rules are usually learned in coverage-based approaches (Fürnkranz et al. 2012). In this work we follow a different, recently introduced rule learning approach, which does not use a covering approach. In the selected approach implemented in the Hedwig algorithm (Vavpetič et al. 2013; Vavpetič 2017), subgroup describing rules are learned using a specialized beam search procedure, 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 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. Term $\sum_{r \in \mathfrak{R}} \epsilon(r)$ corresponds to the quality of individual rules r. Simultaneously, the rules shall not overlap, which is achieved by introducing of the following term: $\sum_{\substack{r_i, r_j \in \mathfrak{R} \\ i \neq j}} |Cov(r_i) \cap Cov(r_j)| + 1$. As Hedwig aims to maximize the numerator in order to maximize rule quality of a set of rules, while at the same time having the rules cover different parts of the example space, which is achieved by minimizing the denominator, minimizing the intersection of instances covered by different rules r_i and r_j.

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 beam search-based algorithm used in this work hence yields multiple different rules that represent different subgroups of the data set being learned on. The Hedwig semantic subgroup discovery algorithm is used in this work, as it was previously successfully applied in the biomedical domain (Adhikari et al. 2016), as it supports RDF-encoded inputs, and is hence suitable for working with collections of background knowledge ontologies. As part of this work, we rewrote Hedwig from Python2 to Python 3, making it one of the first algorithms in Python 3, capable of semantic rule induction.

3.2 Network partitioning 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 (Marc and Lovro 2018).

Definition 2 (**Trivial network partitioning**) A network partitioning P is trivial if $P = \{N\}$, i.e. if all nodes are placed into the same partition.

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

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

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

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 partitioning can be described as a surjective mapping between the nodes and their corresponding partitions, whereas overlapping partitions are described as one-to-many mappings. The number of classes, needed to represent all partitions P is equal to $|P|$ (the proof is available in Appendix C.2).

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. We observe that the number of non-trivial partitions grows exponentially with the number of nodes, hence it is computationally unfeasible to represent all possible partitions as target classes. The proof of this claim using *Bell numbers* (Gardner 1978) is also given as part of Appendix C.2.

Consequently, for a network with n nodes, a naïve approach for learning from its partitions would need to consider $B_n - 1$ possible classes (where B_n represents the n-th Bell number), 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 The CBSSD methodology

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

4.1.1 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 this step,

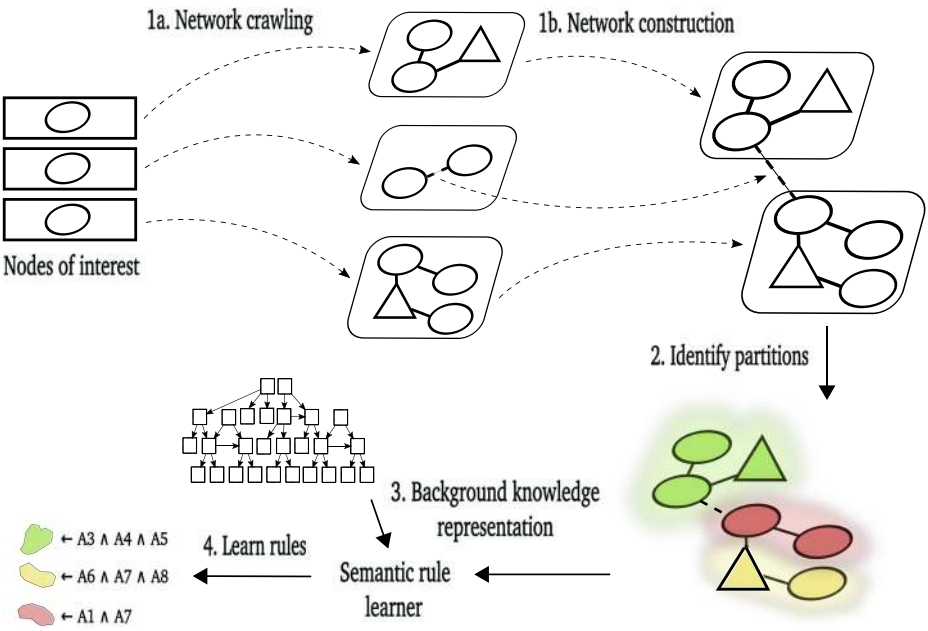

Fig. 1 Schema of the CBSSD methodology, consisting of four main steps: network construction (**1a** and **1b**), network partitioning (**2**), background knowledge representation (**3**), and semantic subgroup discovery (**4**). Triangles and circles represent different types of nodes (e.g., genes and proteins) and colors represent different network partitions. In the network construction step (**1a** and **1b**), for each node of interest either new nodes and links are added or just new links are introduced (e.g., for the 2nd node of interest, the sub-network does not introduce any new node but only adds a connection (dotted edge) between two existing nodes). In the rule learning step (**4**), subgroup describing rules for each target class (node partition of a given color) are formed as conjunctions of ontology terms *Ai* explaining the given network partition

based on the input list, the network is automatically induced by crawling a given knowledge graph. As an alternative to this 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.

The network construction step of CBSSD leverages the BioMine methodology (Eronen and Toivonen 2012) as follows. To automatically induce an association network from the input list of biological entities (such as proteins or genes), individual terms are used as seeds for crawling the BioMine knowledge graph (Eronen and Toivonen 2012), which includes millions of term associations across main biological databases, such as UniProt (Consortium U et al. 2017), Kegg (Kanehisa and Goto 2000), and GenBank (Benson et al. 2012).

To construct the relevant associations from the BioMine knowledge graph, we introduce a network generator function Γ, which takes as input a node (a term) of interest and yields a graph corresponding to the node's neighborhood in the BioMine knowledge graph. The BioMine crawler yields smaller networks, associated with input nodes (e.g., genes). Note that the final network 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 $\Gamma(v_i) = (V_i, E_i)$ for each i (where V_i is the set of nodes of graph G_i and E_i is the set of edges of G_i), we construct a single final graph obtained from the graphs G_i by merging the nodes and edges, i.e. we construct the final graph $G_f = (V_f, E_f)$ by setting

$V_f = \bigcup_{i=1}^{n} V_i$ and $E_f = \bigcup_{i=1}^{n} E_i$. Note that if a gene is connected to the same partner node in two of the smaller (input) networks, in the final network the connection to the gene remains, where the information that two of such edges were identified is lost.

4.1.2 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 the 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 that is useful for large networks. Note however that the Louvain algorithm 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. We tested this claim on the IntAct network; this network is described in detail in Section 5.1.

This step of the methodology is considered as multi-view learning, as heterogeneous networks consisting of multiple layers of different types of information are used to partition the input instances. More specifically, 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 (Zhao et al. 2017).

Apart from community-based partitioning, we also consider network components, i.e. connected subnetworks present in real world complex networks (Škrlj et al. 2018b). Similarly to community detection, the result of component-based partitioning is a set of network's components used as labels for the process of subgroup discovery via rule 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 use *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 Fig. 2.

4.1.3 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 labeled with the 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.

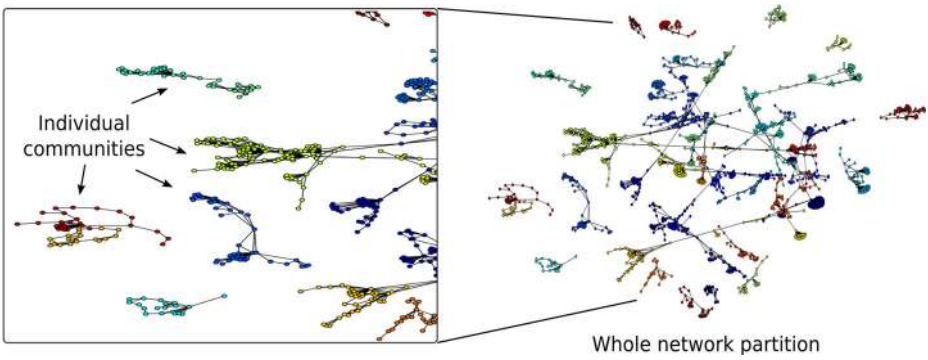

Fig. 2 An example community partition. Different colors correspond to different communities. The network represents the communities detected on the BioMine network used in our previous study (Škrlj et al. 2018a). It can be observed, that multiple communities emerge, which were shown to correspond to different functional processes

As the Hedwig semantic subgroup discovery algorithm (Vavpetič et al. 2013) is used for rule learning in Step 4 of the CBSSD methodology, the experimental data from the previous step had to be converted into RDF triplets as required by Hedwig, where the RDF triplets have the form $T(S, P, O)$, where S is the subject, P the predicate and O the object.

Our main source of background knowledge in this study is the Gene Ontology (GO) (Ashburner et al. 2000) 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 Fig. 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) labeled 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. By convention, we use the *subClassOf* predicate when constructing the background knowledge base, while the *is_a* predicate is used to map individual nodes to their semantic term annotations.

4.1.4 Step 4: semantic rule induction

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. The rules are initially constructed using more general terms and further refined using more specific ontology terms.

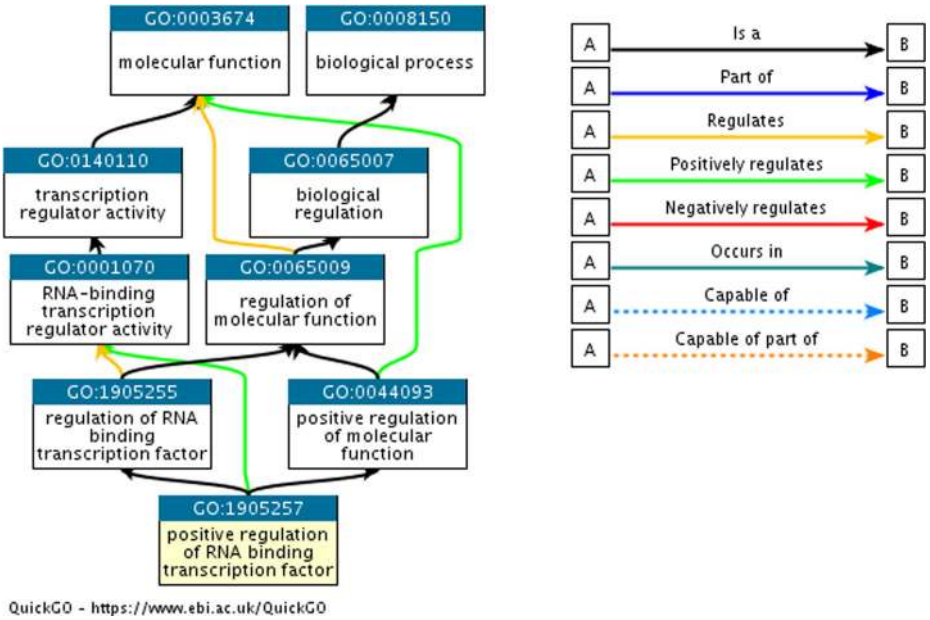

Fig. 3 Example GO hierarchy related to RNA binding factors. The connections between the terms are directed

The result of the final step of the CBSSD methodology are rules of the form TargetClass ← Explanation, where TargetClass corresponds to one of the partitions, discovered in Step 2, and Explanation is a conjunction of one or more terms from the background knowledge, prepared in Step 3. Semantic subgroup discovery perfomed by Hedwig results in individual rules, learned by maximization of the criterion introduced in Section 3 (4). 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 Vavpetič et al. (2013) and Vavpetič (2017) for a comprehensive treatment of 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.[1] 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 (*learnRules*).

In Algorithm 1, I represents the input node list, Ξ 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

[1]Note that if such a network of interest is already given, this step can be omitted.

sets of rules can be any rule significance heuristics, such as, for example, the chi-squared metric, entropy-based measures or similar (Novak et al. 2009).

Algorithm 1 Pseudocode of the CBSSD approach.

Input: nodes of interest I annotated by ontology (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 := learnRules($P_{1...n}$, Ξ) ; ▷ Rule induction
return RuleSets

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 (Rosvall et al. 2009; Blondel et al. 2008) 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 (Adhikari et al. 2016; Vavpetič et al. 2013) already 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 1)
  - node batch size, i.e. 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 2)
  - partition detection algorithm with corresponding parameters, e.g., number of iterations, type of community detection etc.
- Parameters of background knowledge representation (Step 3)
  - generalization predicate used
- Parameters of rule induction (Step 4)
  - 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. The next section presents a quantitative evaluation of the CBSSD approach.

5 Quantitative CBSSD evaluation on ten life science data sets

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 CBSSD approach by investigating how different combinations of background knowledge and different complex networks used for term partitioning influence the explanatory potential of CBSSD. In this section we quantitatively demonstrate that the CBSSD approach can discover significant patterns, which can not be uncovered by conventional enrichment approaches. The following sections are as follows. First, we discuss the experimental setting. Next, we discuss the evaluation measures used to compare the different results. Finally, we present the experimental findings in the form of critical distance diagrams.

5.1 Experimental setting

We test the proposed method by using two different types of complex networks—the automatically constructed BioMine network (full CBSSD methodology) described in the previous section, as well as a homogeneous protein-protein interaction network (CBSSD without network construction step) described next. To perform the experiments, we first downloaded the current version of binary protein interaction-based proteome from the IntAct database (Orchard et al. 2013), 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 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 Fig. 4—consists

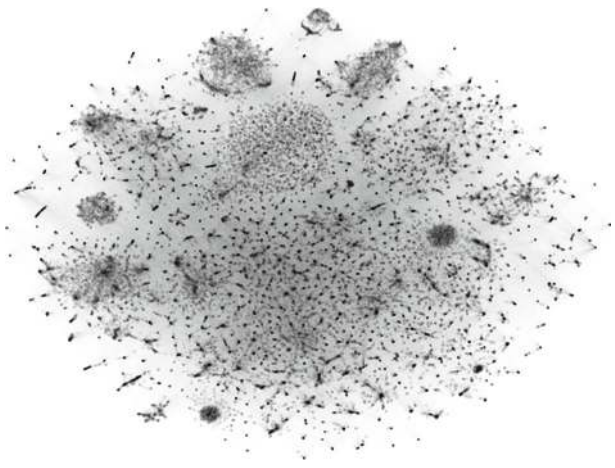

Fig. 4 Part of the human (IntAct) proteome (above the 0.2 reliability threshold) used in this study. It can be observed that densely connected subnetworks emerge, thus the network could contain potentially interesting communities and components

of approximately 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) (Consortium GO 2004) or the whole Gene Ontology is used (Ashburner et al. 2000). The two ontologies contain biological terms describing different biological functions, components and processes.

We compare the CBSSD methodology against the Fisher's exact test-based term enrichment, as used in DAVID (Huang et al. 2007) 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:

$$p = \frac{\binom{a+b}{a}\binom{c+d}{c}}{\binom{n}{a+c}} = \frac{(a+b)!(c+d)!(a+c)!(b+d)!}{a!b!c!d!n!} \tag{5}$$

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 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 (Hosack et al. 2003), where a is replaced by $a - 1$. This leads to modifying (5) as follows:

$$p_{\text{EASE}} = \frac{\binom{a-1+b}{a-1}\binom{c+d}{c}}{\binom{n}{a-1+c}} = \frac{(a-1+b)!(c+d)!(a-1+c)!(b+d)!}{(a-1)!b!c!d!n!} \tag{6}$$

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 (Benjamini and Hochberg 1995).

Different term enrichment and semantic subgroup discovery settings used in the experiments are summarized in Table 1. The CBSSD's run 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 run 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.

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 (6)), 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

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:

$$WRAcc(r_i) = \frac{Cov(r_i)}{N}\left(\frac{TP(r_i)}{Cov(r_i)} - \frac{N_C}{N}\right). \tag{7}$$

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. Max WRAcc: maximum WRAcc score of any rule in the whole rule set (best case)
2. Mean WRAcc: average WRAcc score of WRAcc scores of all rules in the rule set
3. Min WRAcc: minimum WRAcc score of any rule in the whole rule set (worst case)

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:

$$IC_{\text{term}} = -\log(p(\text{term})). \tag{8}$$

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

$$IC_{\text{r}_\text{i}} = -\log(p(\text{term}_1 \wedge \text{term}_2 \wedge \cdots \wedge \text{term}_k)) = \sum_{i=1}^{k} -\log(p(\text{term}_i)). \tag{9}$$

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. Max IC: maximum IC score of any rule in the whole rule set (best case)
5. Mean IC: average IC score of all rules in the rule set
6. Min IC: minimum IC score of any rule in the whole rule set (worst case)

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 (Demšar et al. 2013). 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.

As the CBSSD considers both the complex network (used for partitioning) as well as the ontology (used for rule learning) as free parameters, we compare the performance by varying the complex network, as well as the type of background knowledge used. The rationale for such comparisons is that we are interested in observing how background knowledge (ontology), as well as the complex network used influence the learning outcome (rules).

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] (Subramanian et al. 2005). The SNP-BS list represents the results of a recent study, where sequence variants were studied in the context of protein binding sites (Škrlj and Kunej 2016). The Diabetes protein list represents a UniProt query with keyword diabetes. Entries from the UniProt (Consortium U et al. 2017) database, the largest database of proteins sequences, correspond to individual proteins. The rationale for using the selected data sets is that they represent diverse biological processes and diseases, the number of UniProt ID's per process varies, and are freely accessible. 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.

After a series of initial tests using the epigenetics and DNA repair input lists, we observed that beam size of 50 combined with the depth of 5 yielded promising results, performing at a time scale similar to parallel implementation of conventional enrichment, hence this setting was used for the experimental evaluation. Other fixed parameter values include Support=0.1

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

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

and FDR=0.05. Further, we observed that the Hedwig's execution time notably increases with increased depth of the search, which we additionally discuss in Appendix C.

5.4 Experimental results

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

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 WRAcc of rules. The critical distance diagram showing the results for all approaches and statistical significance of them being different is depicted in Fig. 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.

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 (Fig. 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.

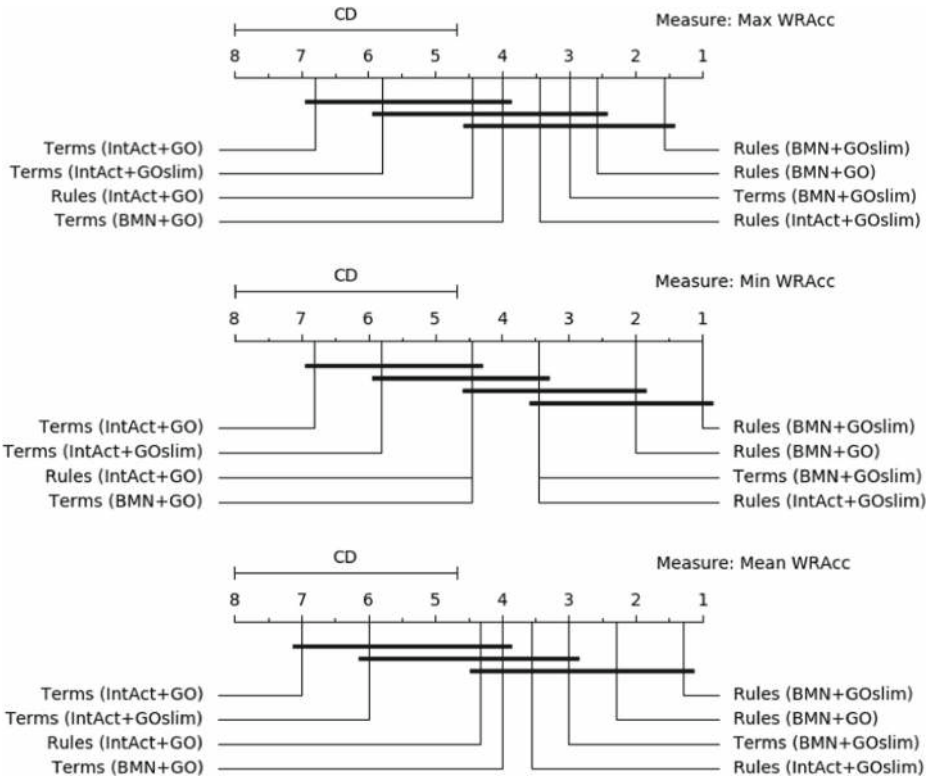

Fig. 5 WRAcc results for enrichment based on communities

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 Fig. 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 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 Fig. 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

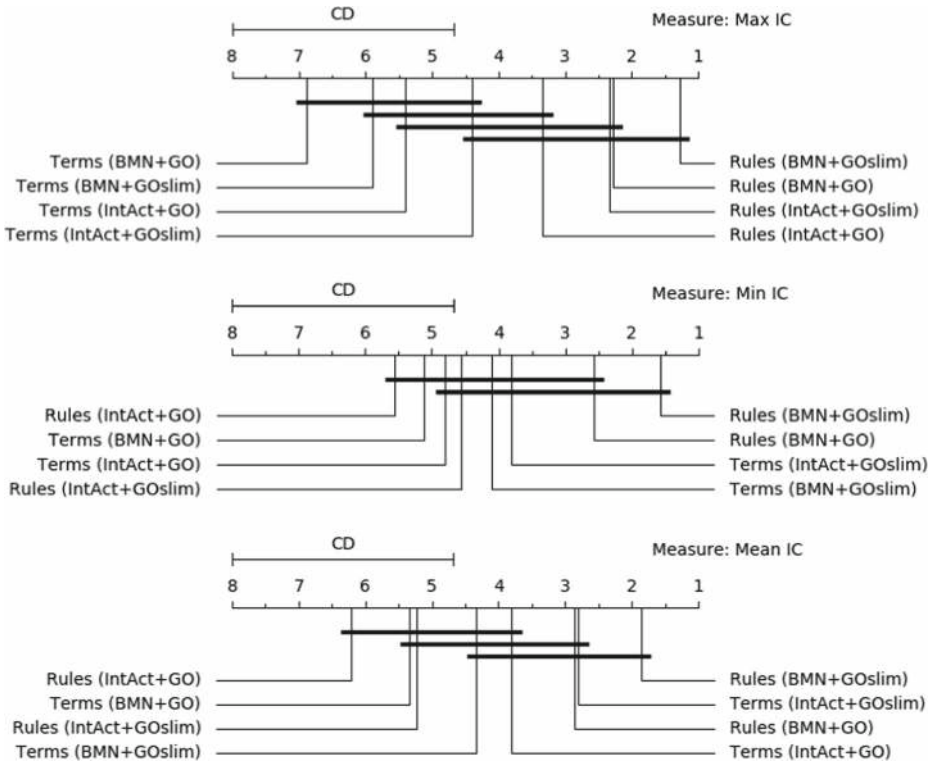

Fig. 6 IC results for enrichment based on communities

results is presented in Section 7. We continue the discussion with the results, obtained using component-based network partitioning 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 Fig. 8.

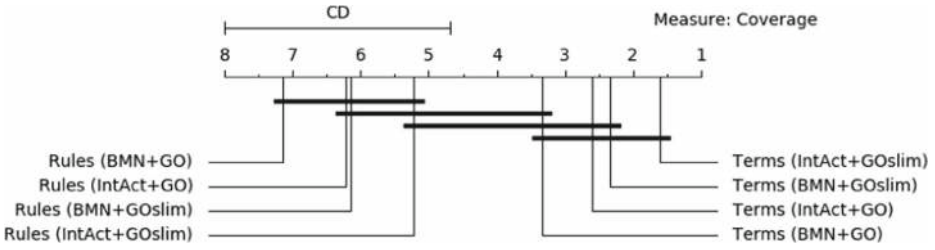

Fig. 7 Coverage results for enrichment based on communities

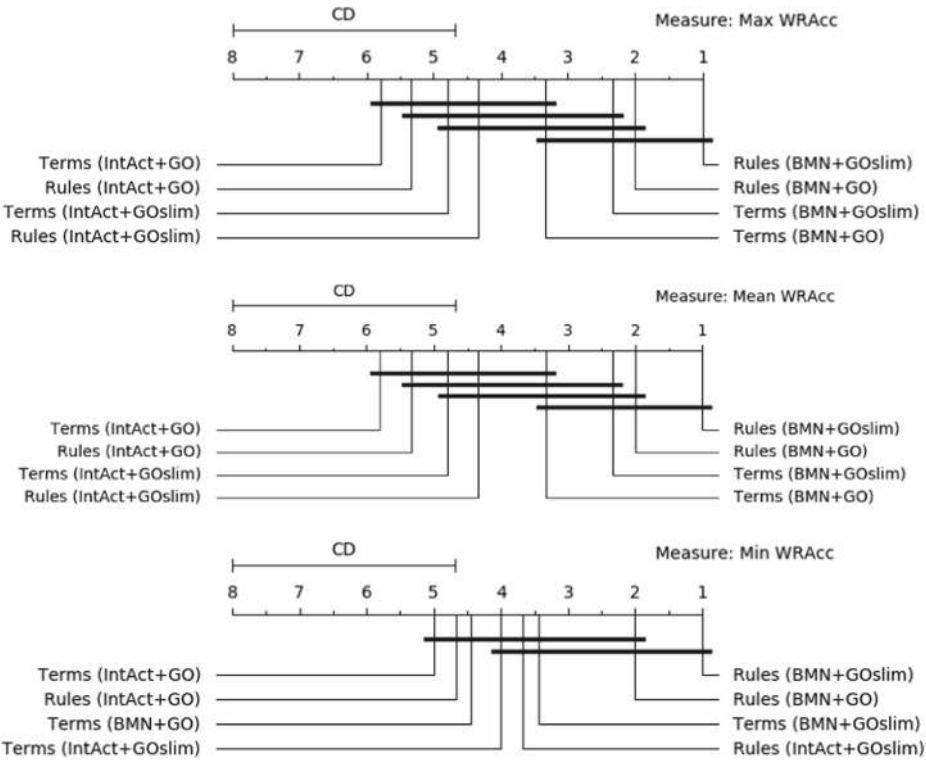

Fig. 8 WRAcc results for enrichment based on components

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 considered (Fig. 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.

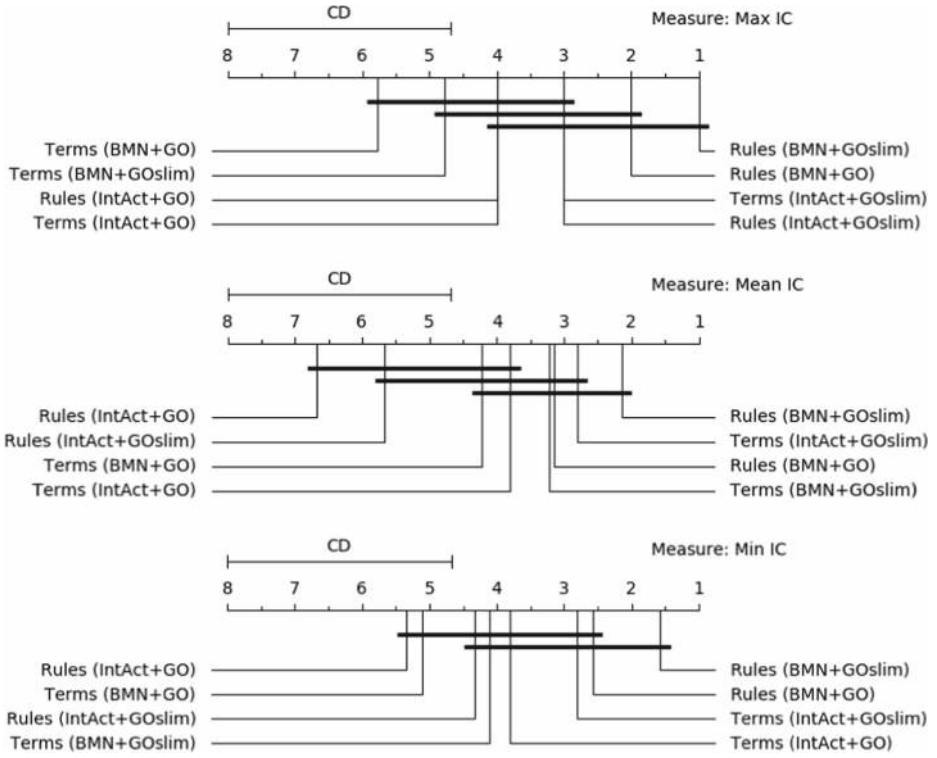

Fig. 9 IC results for enrichment based on components

5.4.6 CBSSD with component partitioning: coverage results

Similarly to community-based network partitioning, term enrichment outperformes rule learning coverage-wise (Fig. 10). This result indicates the network partitioning 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, rules which

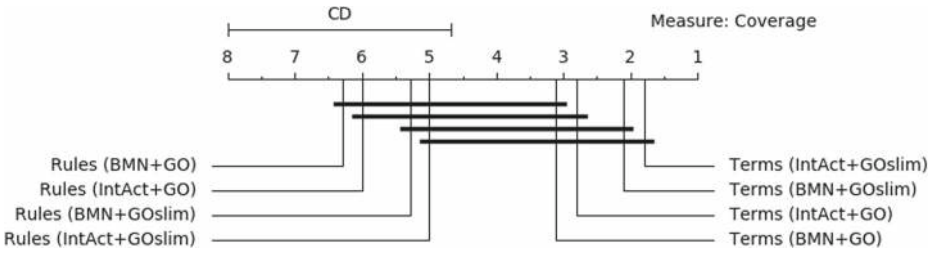

Fig. 10 Coverage results for enrichment based on components

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. The run time of individual experiments was observed to be between 10 minutes and up to ten hours. For example, testing the epigenetics input list by first incrementally constructing the BioMine subgraph, followed by component-based partition detection and semantic rule learning, finished in 14 minutes on an Intel(R) Core(TM) i7 (870), 2.93GHz processor with 16GB of RAM and an Ubuntu 18.04 operating system. Here, majority of the time was spent on rule induction. On the contrary, community detection (C++ implementation of InfoMap, 500 iterations) on the IntAct network took almost three hours on the same machine, followed by hours of rule induction. We added approximate analysis of the computational complexity of individual steps, as well as additional experimental time measurements as part of Appendix C.

The average number of rules that Hedwig found over all data sets is 7. Note that this number is expected to vary when the beam size and the search depth are tuned.

6 Qualitative CBSSD evaluation on two life science use cases

This section demonstrates the use of the CBSSD 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 (Škrlj et al. 2017; Škrlj and Kunej 2016; Schröder and Schumann 2005; Kamburov et al. 2015), 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 (Škrlj et al. 2017) for comparison with the 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).

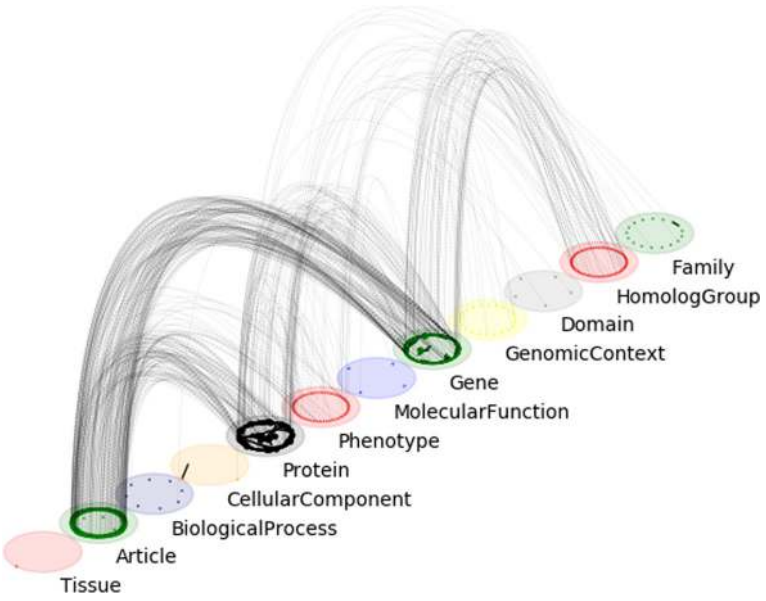

Fig. 11 The BioMine network associated with polymorphisms located within protein interaction sites

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 Škrlj et al. (Škrlj et al. 2017)). A BioMine knowledge graph with more than 1,650 nodes and 2,300 edges was constructed. The resulting network is shown in Fig. 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 (Ashburner et al. 2000) 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 (Škrlj et al. 2017). For this naïve comparison, conjuncts were 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, these rules are potentially more

[3]Plotted with the Py3Plex library (https://github.com/SkBlaz/Py3Plex) (Škrlj et al. 2019).

Table 3 Gene ontology terms, found both in enrichment and semantic rule learning process

Gene ontology term	Meaning	Information content
GO:0000077	DNA damage checkpoint*	9.58
GO:0000086	Mitotic cell cycle*	8.06
GO:0003677	DNA binding*	5.63
GO:0004871	Signal transducer activity*	7.61
GO:0005730	Nucleolus*	6.22
GO:0005814	Centriole	8.18
GO:0016020	membrane	5.40
GO:0016605	PML body	8.37
GO:0030018	Z-disc	8.16
GO:0035264	Multicellular organism growth	8.55
GO:0045892	Negative regulation of transcription (DNA)	6.76
GO:0000122	Negative regulation of transcription (RNA)	6.36
GO:0000785	Chromatin	8.43

Terms marked with * emerged as the most relevant to semantic subgroup discovery

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 Škrlj et al. (Škrlj et al. 2017)), rule $R = GO{:}0034618 \wedge GO{:}0006874$ appeared as one of the most significant rules. 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 (Garrett et al. 1995). 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 the CBSSD approach.

Another interesting rule emerged from the first identified community, i.e. the rule $GO{:}0030903 \wedge GO{:}0000006$ 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 CBSSD 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 CBSSD 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 (Nanda et al. 2016), where information 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 (Huang et al. 2007). 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 shown in Fig. 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

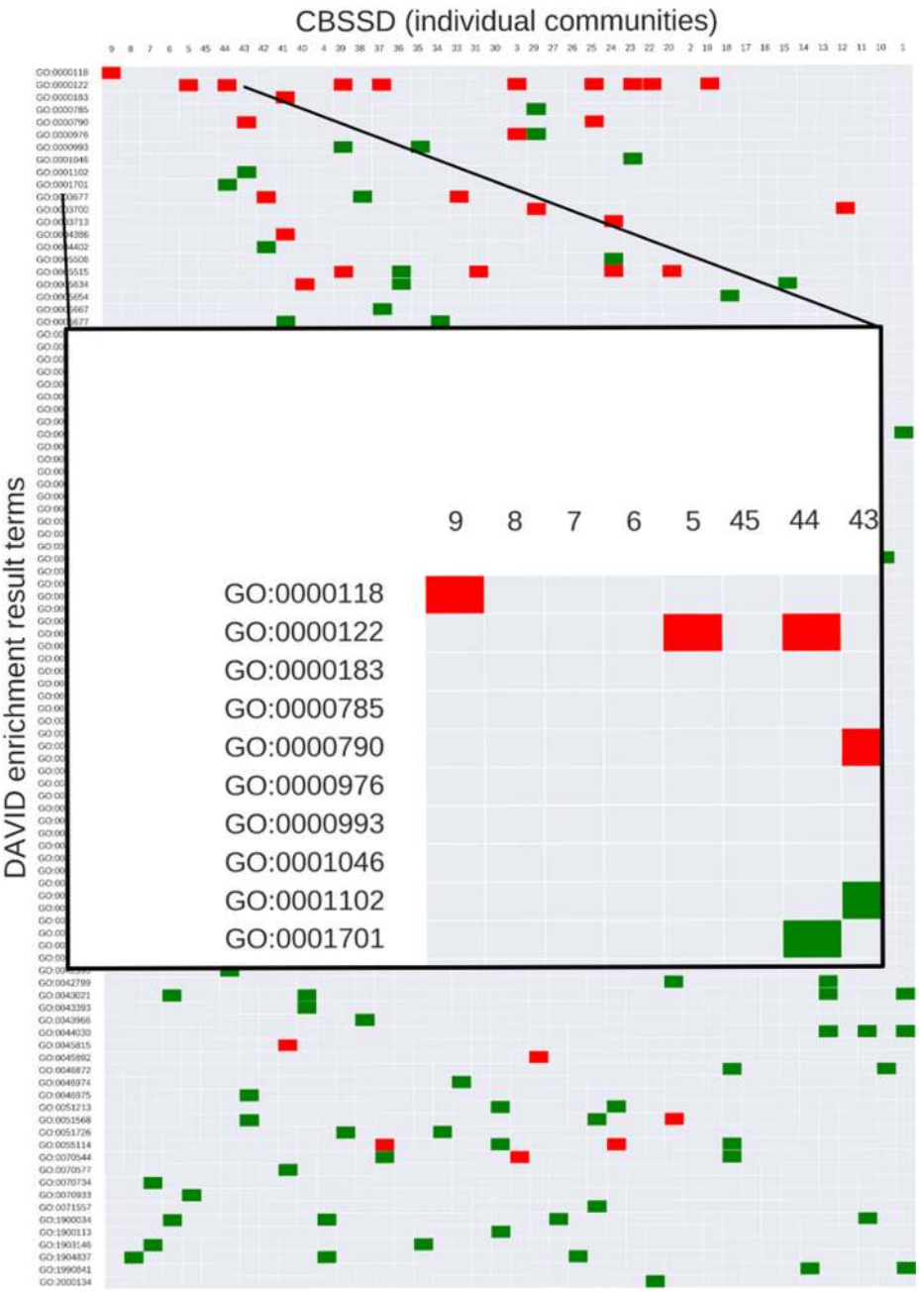

Fig. 12 Visualization of the GO terms appearing in CBSSD discovered rules compared to the GO terms discovered by the DAVID Bioinformatics Suite on an expert defined list of genes coding for cancer-related epigenetic regulators

represent the terms present in the 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 two 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 Fig. 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. The 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, 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 (term conjuncts), 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 (Podpečan et al. 2011), 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 the 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 behavior 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 (Škrlj et al. 2018b; Sardiu et al. 2017), multi-conjunct descriptions of topological features, which emerge in such networks could offer novel insights.

The CBSSD approach depends on the community detection scheme. In this work we explored three different approaches to partitioning a network: Louvain and InfoMap community detection, as well computation of connected components. Even though majority of communities found by Louvain and InfoMap are similar, joining the results of the two (or more) algorithms could be used to increase the robustness of the proposed approach. One of such approaches would be, for example, sorting the communities by size and only keeping the nodes which occur with a predefined frequency in similar communities, obtained by different approaches.

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.

Recent state-of-the-art approaches for learning from complex networks can also take into account many properties, associated with individual nodes. Current implementation of CBSSD does not yet offer such functionality, yet can be extended, for example, by HINMINE (Kralj et al. 2018), a recently proposed approach for decomposition of heterogeneous

networks. Here, node properties could be encoded as a new type of nodes, and as such be taken into account when enumerating directed paths of length two between the nodes of interest, which are used obtain weights between a pre-defined type of nodes.

In this work we explore how rules can be learned using the whole, as well as a reduced version of the gene ontology. As the whole ontology consists of three main parts (Biological Process, Molecular Function, Cellular Component), among which many GO terms are highly correlated, the learned rules can be potentially less informative. This problem is partially addressed by using Hedwig, as it attempts to specialize knowledge in a top-down manner, potentially bypassing redundant intermediary terms. Nonetheless, apart on GOSlim, CBSSD was not tested on any expert-preprocessed version of the Gene Ontology. We believe that the recently introduced NetSDM approach (Kralj et al., 2018) could also be used to reduce the search space and minimize term redundancy. We consider these experiments as part of further work.

The results of qualitative evaluation (cancer-related terms) indicate that different terms can be identified by CBSSD, when compared to conventional enrichment. We believe the main reason for this is that CBSSD takes into account the properties of a given complex network, hence learns rules which explain parts of the input e.g., gene list, and not the whole list, as done by conventional enrichment. The results imply that taking into account also the interactions between the considered genes offers potentially new insights, not necessarily similar to the ones obtained by e.g., the DAVID suite.

The 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 in form of term conjuncts 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 as both ontologies, as well as the networks used contain a lot of redundant information. 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, such as association rules which are also poorly investigated in the context of learning from complex networks.

7.1 Availability

The Community-based Semantic Subgroup Discovery (CBSSD) 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 The work of the first author was funded by the Slovenian Research Agency through a young researcher grant (TSP). The work of other authors was supported by the Slovenian Research Agency (ARRS) core research programme *Knowledge Technologies* (P2-0103) and two ARRS 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). We are grateful to Marko Robnik-Šikonja and to anonymous reviewers for valuable comments and suggestions that helped us to refine the paper. We also gratefully acknowledge the support of NVIDIA Corporation for the donation of Titan-XP GPU used in GPU-based network community detection performed in this work.

Appendix A: The 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 De Domenico et al. (2015). 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 follows:

$$q_{l\curvearrowleft} = \sum_{\{i,\alpha\}\in J\neq V,\{j,\beta\}\in V} q_{ij}^{\alpha\beta} \tag{10}$$

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

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 follows:

$$p_{i\in l} = \sum_{\{i,\alpha\}\in l} p_i^{\alpha}. \tag{12}$$

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 $Q = \{q_{l\curvearrowleft}/q_{\curvearrowleft}\}$. The multilayer map equation can be defined as follows:

$$L(M) = q_{l\curvearrowleft} H_q + \sum_{i=1}^{m} p_{\circlearrowleft}^{i} H_i \tag{13}$$

Although community detection represents one of the most commonly used network partitioning methods used in analysis of complex networks, the CBSSD approach is by no means limited to learning from communities. Currently, the implementation also supports partition based on a network's components—connected sub-networks. Further, arbitrary network partitioning 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 (Palla et al. 2005).

Appendix B: The Hedwig algorithm

In this section we describe in detail the two main procedures used in the Hedwig semantic rule induction algorithm (Vavpetič et al. 2013; Vavpetič 2017). 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.

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

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 3 Hedwig's `specialize`($rule$, B) 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
```

Appendix C: Computational complexity of CBSSD

This section first presents theoretical and empirical findings regarding computational complexity of the CBSSD approach, followed by theoretical treatment of the network partitioning problem.

C.1 Computational complexity of CBSSD

We first present the computational complexity of the BioMine crawler, followed by the complexity of network partition detection and semantic rule induction algorithm Hedwig.

C.1.1 Complexity of the BioMine crawler

In the first step, the full CBSSD methodology first incrementally queries the BioMine network in order to obtain a problem-specific subnetwork, further used in learning. We observe that the largest amount of time is spent by querying the API and waiting for response, hence we can conclude the time spent for obtaining the BioMine network is linear in terms of requests made. The number of requests was on average 130. Overall, this step for all experiments took a maximum of 30 minutes, and is not the bottleneck of the entire methodology.

C.1.2 Complexity of partition detection

The time spent on partition detection on a complex network depends largely on both the method used for detecting partitions and the network itself. For example, computation of a network's connected components is significantly (up to 20 times) faster than, for example, a single run of the InfoMap algorithm with 500 iterations. Similarly, the Louvain algorithm was notably (up to ten times) faster than Infomap. The time complexity of the Louvain algorithm, i.e. $\mathcal{O}(|N|\log(|N|))$, is also theoretically faster than the time complexity of Infomap, which is $\mathcal{O}(|E|)$, when denser networks are considered, however, the number of iterations is the most important parameter of both algorithms with respect to the execution times. The time required for community detection was also notably longer compared to e.g., network construction phase, yet for the proteome network, which is the largest network considered, it did not take more than four hours in the worst case (Infomap, 500 iterations).

C.1.3 Complexity of semantic rule learning

The semantic rule learning step took the largest portion of the overall execution time. Here, the Hedwig algorithm first internally processes the whole ontology, and next uses beam search for rule discovery. In this section we first focus on the time complexity of the rule discovery step, and finally consider spatial requirements.

We first present a general formulation of the Hedwig's execution time. Next, we comment on how the execution time behaves with respect to rule depth and the beam size.

Let d denote the rule depth and n the number of terms in the ontology. The total number of all rules in the search space is on the order of n^d, necessitating a beam search strategy. Let b represent the beam size and w the maximum number of descendants of an ontology term, taking into account only the "SubClassOf" relation. Hedwig considers, at each step of the algorithm, at most $b \cdot d \cdot w$ specializations using the SubClassOf relation (each discovered in constant time) and, additionally, at most $b \cdot d$ specializations obtained by extending the rules

(each discovered in $O(n)$ time). Therefore, in total, Hedwig examines at most $b \cdot (d \cdot w + 1)$ possible specializations, discovered in $O(b \cdot (d \cdot w + n))$ time.

The number of iterations needed for a rule to converge depends largely on the topology of the background knowledge. We measured the average number of iterations empirically over different CBSSD input settings and found that it is approximately 10. As the majority of considered communities are small, the number of terms annotating them is also small and requires a small number of iterations for rules to converge. Only for a handful of the largest communities the number of iterations, needed to find the final set of rules, increased to more than 50.

As the main bottleneck of the current CBSSD implementation we identified Hedwig's spatial requirements, as with larger ontology, the initialization phase, as well as the rule induction phase took considerably longer compared to e.g., GOslim—the prunned version of the original Gene Ontology.

C.2 Theoretical foundations of network partitioning

In this section, we present proofs related to representing network partitions as classes in a supervised learning setting. The number of classes $|T|$ needed to represent $|P|$ partitions is defined as follows.

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

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

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:

$$B_{k+1} = \sum_{i=0}^{k} \binom{k}{i} B_i. \tag{14}$$

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 $1 +$ nt partitions, consisting of 1 trivial partition and nt non-trivial partitions. Consequently, the number of non-trivial partitions nt 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:

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

□

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.*

References

Adhikari, P.R., 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.

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.

Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, *25*(1), 25–29.

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* (Vol. 28, pp. 1112–1120).

Belleau, F., Nolin, M.A., Tourigny, N., Rigault, P., Morissette, J. (2008). Bio2rdf: towards a mashup to build bioinformatics knowledge systems. *Journal of Biomedical Informatics*, *41*(5), 706–716.

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)*, 289–300.

Benson, D.A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Sayers, E.W. (2012). Genbank. *Nucleic Acids Research*, *41*(D1), D36–D42.

Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, *2008*(10), P10008.

Chen, G., Wang, X., Li, X. (2014). *Fundamentals of complex networks: models, structures and dynamics*. Wiley.

Clauset, A., Newman, M.E., Moore, C. (2004). Finding community structure in very large networks. *Physical Review E*, *70*(6), 066111.

Cohen, R., & Havlin, S. (2010). *Complex networks, structure, robustness and function*. Cambridge University Press.

Consortium GO (2004). The gene ontology (GO) database and informatics resource. *Nucleic Acids Research*, *32*(suppl_1), D258–D261.

Consortium U et al. (2017). Uniprot: the universal protein knowledgebase. *Nucleic Acids Research*, *45*(D1), D158–D169.

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.

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.

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.

Dong, X., Hao, Y., Wang, X., Tian, W. (2016). Lego: a novel method for gene set over-representation analysis by incorporating network-based gene weights. *Scientific Reports*, *6*, 18871.

Dou, D., Wang, H., Liu, H. (2015). Semantic data mining: a survey of ontology-based approaches. In *Proceedings of the 2015 IEEE 9th international conference on semantic computing (IEEE ICSC 2015)* (pp. 244–251). IEEE.

Drummond, A.J., & Rambaut, A. (2007). Beast: Bayesian evolutionary analysis by sampling trees. *BMC Evolutionary Biology*, *7*(1), 214.

Duch, J., & Arenas, A. (2005). Community detection in complex networks using extremal optimization. *Physical Review E*, *72*(2), 027104.

Džeroski, S., & Lavrač, N. (Eds.) (2001). *Relational data mining*. Berlin: Springer.

Eronen, L., & Toivonen, H. (2012). Biomine: predicting links between biological entities using network models of heterogeneous databases. *BMC Bioinformatics*, *13*(1), 119.

Fürnkranz, J., Gamberger, D., Lavrač, N. (2012). *Foundations of rule learning*. Springer.

Gardner, M. (1978). Bells-versatile numbers that can count partitions of a set, primes and even rhymes. *Scientific American*, *238*(5), 24.

Garrett, J.E., Capuano, I.V., Hammerland, L.G., Hung, B.C., Brown, E.M., Hebert, S.C., Nemeth, E.F., Fuller, F. (1995). Molecular cloning and functional expression of human parathyroid calcium receptor cDNAs. *Journal of Biological Chemistry*, *270*(21), 12919–12925.

Glaab, E., Baudot, A., Krasnogor, N., Schneider, R., Valencia, A. (2012). Enrichnet: network-based gene set enrichment analysis. *Bioinformatics*, *28*(18), i451–i457. /oup/backfile/content_public/journal/bioinformatics/28/18/10.1093_bioinformatics_bts389/2/bts389.pdf.

Guarino, N., Oberle, D., Staab, S. (2009). *What is an ontology?* (pp. 1–17). Berlin: Springer.

Hmimida, M., & Kanawati, R. (2015). Community detection in multiplex networks: a seed-centric approach. *American Institute of Mathematical Sciences*, *10*(1), 71–85.

Hosack, D.A., Dennis, G., Sherman, B.T., Lane, H.C., Lempicki, R.A. (2003). Identifying biological themes within lists of genes with ease. *Genome Biology*, *4*(10), R70.

Hotho a, S.taab.S., & Stumme, G. (2003). Ontologies improve text document clustering. In *Proceedings of the Third IEEE international conference on data mining* (pp. 2–5).

Huang, D.W., Sherman, B.T., Tan, Q., Kir, J., Liu, D., Bryant, D., Guo, Y., Stephens, R., Baseler, M.W., Lane, H.C., 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.

Huffman, D.A. (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, *40*(9), 1098–1101.

Kamburov, A., Lawrence, M.S., Polak, P., Leshchiner, I., Lage, K., Golub, T.R., Lander, E.S., Getz, G. (2015). Comprehensive assessment of cancer missense mutation clustering in protein structures. *Proceedings of the National Academy of Sciences*, *112*(40), E5486–E5495.

Kanehisa, M., & Goto, S. (2000). KEGG: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, *28*(1), 27–30.

Kralj, J., Robnik-Šikonja, M., Lavrač, N. (2018). HINMINE: heterogeneous information network mining with information retrieval heuristics. *Journal of Intelligent Information Systems*, *50*(1), 29–61.

Kuncheva, Z., & Montana, G. (2015). Community detection in multiplex networks using locally adaptive random walks. In *2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)* (pp. 1308–1315): IEEE.

Lanckriet, G.R., De Bie, T., Cristianini, N., Jordan, M.I., Noble, W.S. (2004). A statistical framework for genomic data fusion. *Bioinformatics*, *20*(16), 2626–2635.

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.

Lavrač, N., & Džeroski, S. (1994). *Inductive logic programming: techniques and applications*. Ellis Horwood.

Lavrač, N., & Vavpetič, A. (2015). Relational and semantic data mining. In *Proceedings of the thirteenth international conference on logic programming and nonmonotonic reasoning* (pp. 20–31). Lexington.

Ławrynowicz, A. (2017). *Semantic data mining: an ontology-based approach*. IOS Press.

Leonavicius, K., Nainys, J., Kuciauskas, D., Mazutis, L. (2019). Multi-omics at single-cell resolution: comparison of experimental and data fusion approaches. *Current Opinion in Biotechnology*, *55*, 159–166.

List, M., Alcaraz, N., Dissing-Hansen, M., Ditzel, H.J., Mollenhauer, J., Baumbach, J. (2016). Keypath-wayminerweb: online multi-omics network enrichment. *Nucleic Acids Research*, *44*(W1), W98–W104.

Liu, H., Dou, D., Jin, R., LePendu, P., Shah, N. (2013). Mining biomedical ontologies and data using RDF hypergraphs. In *Proceedings of the 12th international conference on machine learning and applications (ICMLA)* (Vol. 1, pp. 141–146). IEEE.

Malliaros, F.D., & Vazirgiannis, M. (2013). Clustering and community detection in directed networks: a survey. *Physics Reports*, *533*(4), 95–142.

Marc, T., & Lovro, Š. (2018). Convexity in complex networks. *Network Science*, 1–28. https://doi.org/10.1017/nws.2017.37.

Muggleton, S. (1991). Inductive logic programming. *New Generation Computing*, *8*(4), 295–318.

Nanda, J.S., Kumar, R., Raghava, G.P. (2016). dbem: a database of epigenetic modifiers curated from cancerous and normal genomes. *Scientific Reports*, *6*, 19340.

Novak, P.K., Lavrač, N., Webb, G.I. (2009). Supervised descriptive rule discovery: a unifying survey of contrast set, emerging pattern and subgroup mining. *Journal of Machine Learning Research*, *10*, 377–403.

Orchard, S., Ammari, M., Aranda, B., Breuza, L., Briganti, L., Broackes-Carter, F., Campbell, N.H., 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.

Palla, G., Derényi, I., Farkas, I., Vicsek, T. (2005). Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, *435*, 814–818.

Podpečan, V., Lavrač, N., Mozetič, I., Novak, P.K., 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.

Rosvall, M., Axelsson, D., Bergstrom, C.T. (2009). The map equation. *The European Physical Journal-Special Topics*, *178*(1), 13–23.

Sardiu, M.E., Gilmore, J.M., Groppe, B., Florens, L., Washburn, M.P. (2017). Identification of topological network modules in perturbed protein interaction networks. *Scientific Reports*, *7*, 43845.

Schipper, H.M., Maes, O.C., Chertkow, H.M., Wang, E. (2007). Microrna expression in alzheimer blood mononuclear cells. *Gene Regulation and Systems Biology*, *1*, GRSB–S361.

Schröder, N. W., & Schumann, R.R. (2005). Single nucleotide polymorphisms of toll-like receptors and susceptibility to infectious disease. *The Lancet Infectious Diseases*, *5*(3), 156–164.

Škrlj, B., Kralj, J., Vavpetič, A., Lavrač, N. (2018a). Community-based semantic subgroup discovery. In Appice, A., Loglisci, C., Manco, G., Masciari, E., Ras, Z. W. (Eds.) *New frontiers in mining complex patterns* (pp. 182–196): Springer International Publishing.

Škrlj, B., Kunej, T., Konc, J. (2018b). Insights from ion binding site network analysis into evolution and functions of proteins. *Molecular Informatics*, *37*(6–7), 1700144.

Škrlj, B., Kralj, J., Lavrač, N. (2019). Py3plex: a library for scalable multilayer network analysis and visualization. In Aiello, LM., Cherifi, C., Cherifi, H., Lambiotte, R., Lió, P., Rocha, L. M. (Eds.) *Complex networks and their applications VII* (pp. 757–768): Springer International Publishing.

Strogatz, S.H. (2001). Exploring complex networks. *Nature*, *410*(6825), 268.

Subramanian, A., Tamayo, P., Mootha, V.K., Mukherjee, S., Ebert, B.L., Gillette, M.A., Paulovich, A., Pomeroy, S.L., Golub, T.R., Lander, E.S., 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.

Tipney, H., & Hunter, L. (2010). An introduction to effective use of enrichment analysis software. *Human Genomics*, *4*(3), 1.

Vavpetič, A. (2017). *Semantic subgroup discovery*. PhD thesis, Jožef Stefan International Postgraduate School.

Vavpetič, A., & Lavrač, N. (2012). Semantic subgroup discovery systems and workflows in the SDM-toolkit. *The Computer Journal*, *56*(3), 304–320.

Vavpetič, A., Novak, P.K., Grčar, M., Mozetič, I., Lavrač, N. (2013). Semantic data mining of financial news articles. In *Proceedings of the international conference on discovery science* (pp. 294–307). Springer.

Vrabič Rok, H. D., & Butala, P. (2012). Discovering autonomous structures within complex networks of work systems. *CIRP Annals-Manufacturing Technology*, *61*(1), 423–426.

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.

Škrlj, B., Konc, J., Kunej, T. (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.

Zhao, J., Xie, X., Xu, X., Sun, S. (2017). Multi-view learning overview: recent progress and new challenges. *Information Fusion*, *38*, 43–54.