Bioinformatics, 38(6), 2022, 1600–1606
https://doi.org/10.1093/bioinformatics/btab876
Advance Access Publication Date: 4 January 2022
Original Paper

OXFORD

Systems biology

Robust disease module mining via enumeration of diverse prize-collecting Steiner trees

Judith Bernett [1,†], Dominik Krupke[2,†], Sepideh Sadegh[1,3], Jan Baumbach[3,4], Sándor P. Fekete[2,5], Tim Kacprowski [5,6], Markus List [1] and David B. Blumenthal [7,*]

[1]Chair of Experimental Bioinformatics, TUM School of Life Sciences, Technical University of Munich, 85354 Freising, Germany, [2]Department of Computer Science, TU Braunschweig, 38106 Braunschweig, Germany, [3]Institute for Computational Systems Biology, University of Hamburg, 22607 Hamburg, Germany, [4]Department of Mathematics and Computer Science, University of Southern Denmark, 5230 Odense, Denmark, [5]Braunschweig Integrated Centre of Systems Biology (BRICS), 38106 Braunschweig, Germany, [6]Division Data Science in Biomedicine, Peter L. Reichertz Institute for Medical Informatics, Technical University of Braunschweig and Hannover Medical School, 38106 Braunschweig, Germany and [7]Department Artificial Intelligence in Biomedical Engineering (AIBE), Friedrich-Alexander University Erlangen-Nürnberg (FAU), 91052 Erlangen, Germany

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors contributed equally.
Associate Editor: Lenore Cowen

Received on October 6, 2021; revised on November 29, 2021; editorial decision on December 27, 2021; accepted on December 31, 2021

Abstract

Motivation: Disease module mining methods (DMMMs) extract subgraphs that constitute candidate disease mechanisms from molecular interaction networks such as protein–protein interaction (PPI) networks. Irrespective of the employed models, DMMMs typically include non-robust steps in their workflows, i.e. the computed subnetworks vary when running the DMMMs multiple times on equivalent input. This lack of robustness has a negative effect on the trustworthiness of the obtained subnetworks and is hence detrimental for the widespread adoption of DMMMs in the biomedical sciences.

Results: To overcome this problem, we present a new DMMM called ROBUST (robust disease module mining via enumeration of diverse prize-collecting Steiner trees). In a large-scale empirical evaluation, we show that ROBUST outperforms competing methods in terms of robustness, scalability and, in most settings, functional relevance of the produced modules, measured via KEGG (Kyoto Encyclopedia of Genes and Genomes) gene set enrichment scores and overlap with DisGeNET disease genes.

Availability and implementation: A Python 3 implementation and scripts to reproduce the results reported in this article are available on GitHub: https://github.com/bionetslab/robust, https://github.com/bionetslab/robust-eval.

Contact: david.b.blumenthal@fau.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Over the last decades, high-throughput molecular profiling technologies have generated an immense amount of omics data, enabling the generation of detailed interaction networks. Motivated by the possibility to uncover the pathobiology of complex diseases, the field of network medicine has emerged to untangle these connections and to pinpoint the molecular basis of complex diseases (Barabási *et al.*, 2011; Roy *et al.*, 2014). This task is complicated by the fact that molecular omics data such as gene expression data are generally noisy and overdetermined. Disease-causing alterations such as mutations typically have a cascading effect on the expression of genes and proteins that form the nodes of most interaction networks with typically hundreds or thousands of differentially expressed genes. Additionally, not all of the genes triggering a certain disease might be differentially expressed in an experiment because the expression profiles are limited to a snapshot of the cell state. Therefore, the discovery of disease genes using simple statistical tests is infeasible. Consequently, disease module mining methods (DMMMs) have been developed to combine analyses of gene expression profiles with mining of prior knowledge encoded in protein–protein interaction (PPI) and other networks.

DMMMs try to identify significantly enriched subnetworks by projecting the expression data on a molecular interaction network. Since solving the underlying mathematical models to optimality is typically NP-hard (Ideker *et al.*, 2002), heuristic algorithms are used in practice, where different weight and scoring metrics are applied to the network components. Using these algorithms, subnetworks can be identified that are significantly associated with a certain disease, even when some of the individual nodes have a negligible score. Various DMMMs have been proposed in the past years (Batra *et al.*, 2017; Lazareva *et al.*, 2021). They have enabled new insights into complex diseases like Type-2 diabetes (Fernández-Tajes *et al.*, 2019; Sharma *et al.*, 2018), pulmonary arterial hypertension (Samokhin *et al.*, 2018), coronary heart disease (Wang and Loscalzo, 2018) and asthma (Sharma *et al.*, 2015).

Despite these success stories, existing DMMMs are known to be subject to several limitations. For instance, Levi *et al.* (2021) have shown that most DMMMs do not fully exploit the information contained in the gene expression data. Lazareva *et al.* (2021) have demonstrated that most DMMMs mainly learn from the node degrees rather than from the biological knowledge encoded in the edges of the PPI networks.

Here, we draw attention to an additional issue which has not been addressed yet: existing DMMMs lack robustness and are subject to random bias. The reason for this is that all DMMMs we are aware of include non-robust steps in their workflows, although this aspect is often not explicitly mentioned and sometimes not immediately obvious. For instance, for some methods, changing the order of the input data leads to dramatically different results. Other methods show variations of the resulting subnetworks when run multiple times on identical input. This lack of robustness is a major limitation, because reliable output is crucial to achieve a widespread adoption of DMMMs in the biomedical research community: Biomedical scientists without a strong background in computer science or mathematics often find it difficult to trust in tools that do not reliably produce the same output and, when confronted with non-robust disease modules, are therefore often less inclined to invest time and money in downstream wet lab validation. Note that simply ordering the input in some canonical but biologically meaningless way (e.g. by sorting based on gene or protein names) does not resolve this problem but merely hides it.

A straightforward approach for robustifying any DMMM is to run the DMMM n times on shuffled input and then to return the subgraph induced by nodes contained in many of the returned modules. However, this naïve approach has the disadvantage that the runtime increases by a factor of n. Moreover, it is not guaranteed to be effective because the modules obtained for the shuffled input might not be sufficiently diverse to ensure robustness (see Section 3.2 for results showing that this is a real and not only a hypothetical problem).

To address this issue, we present a new DMMM called ROBUST (robust disease module mining via enumeration of diverse prize-collecting Steiner trees). Unlike the naïve approach, ROBUST ensures robustness by enumerating pairwise diverse rather than merely pairwise non-identical disease modules. Large-scale tests on data for 829 diseases show that, unlike all tested competitors, ROBUST achieves almost perfect robustness (best-possible robustness already at the first quartile). ROBUST is also faster than its competitors and manages to compute disease modules for up to 400 seeds in <30 s. Tests on gene expression data for amyotrophic lateral sclerosis (ALS), non-small cell lung cancer (LC), ulcerative colitis (UC), Chron's disease (CD) and Huntington's disease (HD) demonstrate that, in most settings, ROBUST outperforms its competitors in terms of the returned modules' functional relevance, which we measured via KEGG (Kanehisa *et al.*, 2016) gene set enrichment w.r.t. known disease-associated pathways and overlap with DisGeNET (Piñero *et al.*, 2020) disease genes. Finally, a case study in multiple sclerosis (MS) shows how ROBUST can be used for hypothesis generation.

2 Materials and methods

2.1 Modeling disease modules via generalized Steiner trees

Two strategic decisions have to be made when designing a new DMMM. First, one has to decide which input should be expected. In addition to a PPI network, existing DMMMs use various types of input data such as normalized expression data (Larsen *et al.*, 2020; Ma *et al.*, 2011; Nacu *et al.*, 2007), gene scores (Barel and Herwig, 2020; Reyna *et al.*, 2018), sorted lists of genes (Breitling *et al.*, 2004), indicator matrices of differentially expressed genes (List *et al.*, 2016) or binary input in the form of sets of disease-associated or differentially expressed seed genes (Ding *et al.*, 2018; Ghiassian *et al.*, 2015; Levi *et al.*, 2021; Sadegh *et al.*, 2020). For ROBUST, we chose the latter option for the following reasons:

- Sets of seed genes are very user-friendly input. They can be computed via standard differential gene expression analysis or be obtained from public databases such as OMIM (Online Mendelian Inheritance in Man) (Amberger *et al.*, 2019) or DisGeNET (Piñero *et al.*, 2020), which provide disease–gene associations obtained from genome-wide association studies (GWAS).
- Levi *et al.* (2021) have shown that DMMMs using seed sets as input tend to outperform DMMMs operating on non-binary input data.

Nonetheless, there might be scenarios where binarization is not desirable because of the arbitrariness in selecting the cutoff and the resulting loss of information. In such settings, ROBUST is not applicable.

The second question is how the disease module mining problem should be modeled mathematically. Informally, our model can be viewed as a generalized minimum-weight Steiner tree (MWST) model (relaxation is explained below). Recall that an MWST for a weighted network $G = (V, E, w)$ and a set of seed nodes $S \subseteq V$ is a tree $T = (V_T, E_T)$ with $S \subseteq V_T \subseteq V$, $E_T \subseteq E$, and minimum total weight $\sum_{e \in E_T} w(e)$. Steiner trees have been used for disease module mining before, e.g. by the DMMMs DOMINO (Levi *et al.*, 2021) and MuST (Sadegh *et al.*, 2020). Computing MWSTs is NP-hard, but approximation algorithms exist, e.g. the classical 2-approximation by Kou *et al.* (1981) or the currently best 1.39-approximation by Byrka *et al.* (2013).

From a biological point of view, using MWSTs to model the disease module mining problem is promising. Functionally related genes or proteins tend to be close to each other in the molecular interaction network, and it could be shown that pairwise shortest paths of known disease genes show a considerable left shift in their distribution compared to the random expectation (Menche *et al.*, 2015). A reasonable hypothesis is hence that the shortest paths between these disease genes overlap with causal molecular pathways (Barabási *et al.*, 2011). Since MWSTs can be viewed as generalizations of shortest paths to settings with more than two endpoints, a disease module constructed using MWSTs can be expected to cover a large fraction of the disease-relevant molecular pathways.

As mentioned above, we use a generalized MWST model, which means that we do not strictly enforce $S \subseteq V_T$ but allow that some seeds are left uncovered (see Section 2.3 for the formal specification of our model). This is because, in the context of disease module mining, the seeds are potentially noisy due to false positives in GWAS or differential gene expression analysis. Moreover, we are eventually interested in the subgraph $G[V_T]$ induced by the node set of the tree $T = (V_T, E_T)$ rather than in T itself. The reason is that also edges between nodes from V_T which are not contained in E_T might pinpoint to causal disease mechanisms and are hence potentially of interest.

2.2 Ensuring robustness via enumeration with diversity

The main limitation ROBUST is designed to overcome is the lack of robustness of existing DMMMs. However, our generalized MWST model alone does not ensure this. For a given seed set S, the PPI network G typically contains multiple near-optimal generalized Steiner trees. If we simply returned the subgraph induced by the node set of one cheap generalized Steiner tree, the output would hence again depend on the random storage order of the input, hampering the robustness of our approach.

Algorithm 1: ROBUST

Input: Graph $G = (V, E)$, seeds $S \subseteq V$, parameters $n \in \mathbb{N}$, $\alpha \in (0, 1]$, $\beta \in [0, 1)$, $\tau \in (0, 1]$.
Output: Robust disease module for seeds S.
1 $\mathcal{T} \leftarrow$ `enumerate_diverse`(G, S, n, α, β);
2 $M \leftarrow \{v \in V | |\{V_T \in \mathcal{T} | v \in V_T\}| \geq \tau \cdot |\mathcal{T}|\}$;
3 **return** $G[M]$;

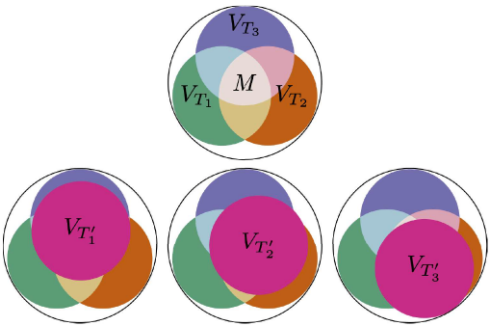

Fig. 1. Visualization of how enumeration with diversity ensures robustness. The bounding circle represents the set of all nodes contained in any near-optimal generalized Steiner tree. The sets V_{T_i} represent the node sets of three pairwise diverse near-optimal generalized Steiner trees. For $\tau > 2/3$, the set M corresponds to the intersection of the sets V_{T_i}. The sets $V_{T_i'}$ represent the node sets of three randomly sampled near-optimal generalized Steiner trees. Nodes contained in M are contained in almost all of the sets $V_{T_i'}$

To address this problem, our DMMM ROBUST is designed to provide a solution for the following problem specification: *Given a weighted network $G = (V, E, w)$ and a set of seed nodes $S \subseteq V$, compute an induced subgraph $G[M]$, where $M \subseteq V$ contains nodes that appear in many diverse near-optimal generalized Steiner trees for S.* ROBUST's overall approach is detailed in Algorithm 1 and visualized in Figure 1. Instead of computing just one near-optimal generalized Steiner tree, we enumerate up to n of them and ensure that their node sets are pairwise diverse (see Section 2.3). We then return the subgraph induced in G by nodes contained in at least $100 \cdot \tau$ % pairwise diverse Steiner trees, where both $n \in \mathbb{N}$ and $\tau \in (0, 1]$ are hyper-parameters.

Two aspects should be highlighted at this point: First, the subgraph $G[M]$ is in general not connected and its connected components hence potentially represent disjoint or complementary disease mechanisms. To allow separate downstream analyses, our implementation therefore labels $G[M]$'s connected components via node attributes in the output file. Second, note that the above specification of the problem solved by ROBUST is imprecise, as we did not formally define the qualifiers 'diverse', 'near-optimal' and 'generalized'. Several possible formal specifications are discussed in the Supplementary Material.

2.3 Enumerating cheap and diverse generalized Steiner trees

Let us consider how a set of diverse, low-weight networks that connect most of the seed nodes (generalized Steiner trees) can be computed. Naïvely, we can compute a Steiner tree T on G to obtain our first network. To obtain a different network, we simply remove a Steiner node or edge in T from G and compute a new Steiner tree T' that now differs in at least one position from T. As mentioned above, the currently best-known algorithm for the MWST problem is the 1.39-approximation by Byrka *et al.* (2013), but even with a better algorithm, the results may not be as hoped: If the removed edge or node is in a dense part of the graph, it can easily be circumvented and the resulting solution will nearly be the same. If it is in a sparse part and an important connection, the resulting solution will be expensive. This naïve approach is employed by the DMMM MuST (Sadegh *et al.*, 2020), which uses the 2-approximation for MWST by Kou *et al.* (1981), iteratively removes Steiner edges and eventually returns the union of all computed Steiner trees.

The alternative we propose is to not just focus on one seed node which is removed but instead to make all of the previously used nodes less attractive depending on how often they have been used. To achieve this, we use prize-collecting Steiner trees (PCSTs) where we assign every seed a high value and every other node a low but not negligible value. If a non-seed node is returned in a solution, we decrease its value to make it less attractive for future solutions. The seeds' high values encourage a PCST algorithm to include them in the solution. The low decreasing values of the other nodes encourage the algorithm to integrate less often used nodes. By keeping the values below the edge costs, this approach avoids randomly integrating nodes at the cost of a more expensive network. More details on the seed values are provided in the Supplementary Material.

Computing an optimal PCST is unfortunately also NP-hard. Therefore, we use the primal–dual approximation algorithm by Goemans and Williamson (1995) and an implementation by Hegde *et al.* (2015). It has a guaranteed approximation factor of at most two and a runtime complexity of $O(d|E| \log |V|)$, where d refers to the encoding size in bits for the weight and values. In practice, this algorithm yields very natural solutions, because it is based on a linear programming relaxation which captures a lot of structure and not only the objective value (see Supplementary Material for details). The implementation is remarkably fast and solves instances with multiple hundreds of thousand edges within seconds (often even less than a second), as shown by Hegde *et al.* (2014).

Let `pcst_apx` be a PCST algorithm that receives a graph $G = (V, E)$, positive edge weights $w : E \to \mathbb{R}_{>0}$, and non-negative node values $p : V \to \mathbb{R}_{\geq 0}$ and returns a tree $T = (V_T, E_T)$ with $V_T \subseteq V$ and $E_T \subseteq E$, minimizing $\sum_{e \in E_T} w(e) + \sum_{v \in V \setminus V_T} p(v)$. Note that, in the context of disease module mining, edges are typically unweighted, i.e. $w(e) = 1$ holds for all $e \in E$. However, some DMMMs use edge weights that penalize edges toward high-degree nodes in the PPI network (Sadegh *et al.*, 2020). While we have designed ROBUST with unweighted edges in mind, we here present the more general weighted version.

Our proposed algorithm is defined in Algorithm 2. As input, it expects a graph $G = (V, E)$, a seed set S, edge weights w, a number of desired trees n, as well as tuning parameters $\alpha \in (0, 1]$ and $\beta \in [0, 1)$ explained below. The first step is to define the initial values p to be passed to `pcst_apx`. To give the algorithm a high incentive to integrate all seeds, we determine their value based on the diameter of the graph and the maximum edge weight (line 1). The initial values of the non-seeds are defined as α times the minimum edge weight (line 1). Note that, in the unweighted case, both the minimum and the maximum edge weight equals 1.After initializing the node values, the algorithm repeatedly calls `pcst_apx` to compute a new PCST (V_T, E_T) dissimilar to the ones computed before until n PCSTs have been computed (line 2) or V_T is identical to the node set of a PCST computed before (line 4). Dissimilarity is ensured by multiplying the values of all non-seeds contained in V_T by a factor $\beta \in [0, 1)$, thereby making them less attractive for subsequent calls to `pcst_apx` (line 5). If connecting a seed $s \in S$ to the remaining seeds would incur a very high cost, it might happen that V_T—and therefore also the final disease module $G[M]$ returned by ROBUST—does not contain s. This can be viewed as an unsupervised data cleaning step built into ROBUST: It automatically discards seeds that are very badly connected to the remaining seeds and are hence potentially unreliable. An alternative which enforces full Steiner trees is described in the Supplementary Material.

2.4 Influence of the hyper-parameters

ROBUST has four hyper-parameters: the desired number of PCSTs n, the threshold τ, and the tuning parameters α and β. The effects of these parameters can be summarized as follows:

Algorithm 2: `enumerate_diverse`

Input: Graph $G = (V, E)$, seeds $S \subseteq V$, edge weights $w : E \to \mathbb{R}_{\geq 0}$, parameters $n \in \mathbb{N}$, $\alpha \in (0, 1]$, $\beta \in [0, 1)$.
Output: Set $\mathcal{T}$ of diverse PCST node sets.

for $v \in S$ **do** $p(v) \leftarrow 2 \cdot \mathrm{diam}(G) \cdot \max_{e \in E} w(e)$;
for $v \in V \backslash S$ **do** $p(v) \leftarrow \alpha \cdot \min_{e \in E} w(e)$;
$\mathcal{T} \leftarrow \emptyset$;
while $|\mathcal{T}| < n$ **do**
  $(V_T, E_T) \leftarrow$ `pcst_apx`(G, w, p);
  if $V_T \in \mathcal{T}$ **then break**
  $\mathcal{T} \leftarrow \mathcal{T} \cup \{V_T\}$;
  for $v \in V_T \backslash S$ **do** $p(v) \leftarrow \beta \cdot p(v)$
return $\mathcal{T}$

- Intuitively, the desired number of trees $n \in \mathbb{N}$ controls the extent to which the disease module computed by ROBUST covers the space of all near-optimal generalized Steiner trees. Setting n to a rather large value is hence desirable but detrimental for the runtime.
- The threshold $\tau \in (0, 1]$ provides a tradeoff between robustness and explorativeness. The larger τ, the more robust but less explorative the disease module computed by ROBUST.
- The parameter $\alpha \in (0, 1]$ modifies the initial values for integrating non-seeds into the tree. This implicitly represents the allowed diversion from the cheapest Steiner tree. For $\alpha = 0$, the algorithm would only return the best Steiner tree it can find but not allow any diversion from it. The larger α, the more diverse and but also more expensive the returned Steiner trees are allowed to become.
- The parameter $\beta \in [0, 1)$ modifies the decrease of the values for integrating non-seeds into the trees. Setting $\beta = 0$ will only give a value to a non-seed until its first appearance in one tree. This can quickly exhaust the available non-seeds and then has the same problem as $\alpha = 0$. A too high value, on the other hand, might not be able to reduce the values sufficiently to make the other non-seeds more attractive. Hence, more trees need to be generated to achieve diversity.

3 Results and discussion

3.1 Compared methods

We compared ROBUST to the state-of-the-art DMMMs DIAMOnD (Ghiassian *et al.*, 2015), MuST (Sadegh *et al.*, 2020) and DOMINO (Levi *et al.*, 2021). These methods were selected for the following reasons:

- They all expect binary input (i.e. lists of differentially expressed or disease-associated seed genes) and are hence directly comparable to our method ROBUST.
- DOMINO has been shown to outperform other DMMMs in two independent studies (Lazareva *et al.*, 2021; Levi *et al.*, 2021) and can hence be considered to be one of the best available methods.
- Based on the number of citations, DIAMOnD is arguably one of the most widely used DMMMs.
- MuST serves as a baseline model for extracting disease modules via Steiner trees without the improvements of ROBUST.

Moreover, we compared ROBUST to a baseline implementing the naïve approach at robustification outlined in Section 1. More precisely, instead of enumerating diverse PCSTs as detailed above, we enumerate multiple Steiner trees by simply shuffling the input data and running the classical 2-approximation algorithm by Kou *et al.* (1981) several times. In the sequel, this naïve baseline is called R-MuST (randomized MuST). An AIMe report (Matschinske *et al.*, 2021) with further details on the empirical evaluation is available at https://aime-registry.org/report/VM0hhS.

3.2 Robustness

3.2.1 Protocol and data used for robustness tests

The methods were tested on a human PPI network obtained from IID (Integrated Interactions Database) (Kotlyar *et al.*, 2019) filtered for experimentally validated interactions. The network consists of 329 215 edges between 17 666 proteins. Sets of disease-associated seed genes were constructed for 929 diseases by merging disease–gene associations from OMIM (Amberger *et al.*, 2019) and DisGeNET (Piñero *et al.*, 2020). For the hyper-parameter evaluation of ROBUST (Section 3.2.2), 100 of the seed sets were used and subsequently excluded for the comparison of ROBUST to its competitors (Section 3.2.3).

Robustness was measured by running each DMMM `ALG` 20 times on each seed set S. In each iteration, the input PPI network was randomly permuted before running `ALG`, yielding 20 disease modules. Let $M_i^{\mathtt{ALG},S}$ be the node set of the ith disease module computed by `ALG` on S. Then, we quantified `ALG`'s robustness on U using the mean Jaccard index

$$r_S(\mathtt{ALG}) := \binom{20}{2}^{-1} \sum_{i=1}^{19} \sum_{j=i+1}^{20} \frac{|M_i^{\mathtt{ALG},S} \cap M_j^{\mathtt{ALG},S}|}{|M_i^{\mathtt{ALG},S} \cup M_j^{\mathtt{ALG},S}|},$$

i.e. $r_S(\mathtt{ALG}) \in [0, 1]$ and large values of $r_S(\mathtt{ALG})$ indicate that `ALG` is robust to random storage order on the seed set S.

3.2.2 Effect of hyper-parameters

For testing ROBUST's robustness w.r.t. the hyper-parameters, we varied $\alpha \in \{0.25, 0.5, 0.75\}$, $\beta \in \{0.1, 0.3, \ldots, 0.9\}$, $\tau \in \{0.1, 0.2, \ldots, 0.9\}$ and $n \in \{5, 10, \ldots, 30\}$. Supplementary Figure S1 shows the full results. While increasing α significantly deteriorated the robustness, increasing β marginally improved it. Therefore, we focus on the results for $\alpha = 0.25$ and $\beta = 0.9$, which are shown in Figure 2. Unsurprisingly, we observe that the robustness improved when increasing the threshold τ and the number of trees n. However, for $n = 30$, we observe large robustness coefficients already for small values of τ. Keeping τ small is *prima facie* desirable as it allows ROBUST to compute more exploratory disease modules. Moreover, ROBUSTS's runtime requirements increase only very moderately with increasing n (see Section 3.4). For these reasons, we selected the hyper-parameter setup $(\alpha, \beta, n, \tau) = (0.25, 0.9, 30, 0.1)$ for all further experiments. Note (i) that the 100 seed sets used to select these hyper-parameters were not used for evaluating ROBUST's robustness in comparison to its competitors and (ii) that we tuned the hyper-parameters only for robustness and not for functional relevance.

3.2.3 Robustness in comparison to competitors

Figure 3 shows the distribution of the 829 mean Jaccard indices for each of the compared DMMMs. Of all the tested DMMMs, only ROBUST yielded almost perfectly robust disease modules with a robustness coefficient of $r_S = 1.0$ at the first quartile. ROBUST is clearly superior to its precursor MuST, to the naïve baseline implementation R-MuST, and, most importantly, also to the state-of-the-art DMMM DOMINO. DIAMOnD yielded remarkably high robustness coefficients, especially considering that its hyper-parameters were not tuned for robustness. Nonetheless, its robustness coefficients were still statistically significantly lower than the ones obtained for ROBUST (Table 1). The superiority of ROBUST to MuST and R-MuST shows that it is indeed necessary to ensure diversity when enumerating (generalized) Steiner trees. Merely enumerating pairwise different trees does not yield the desired robustness. More generally, the rather bad performance of R-MuST shows that the above-mentioned naïve approach for robustification

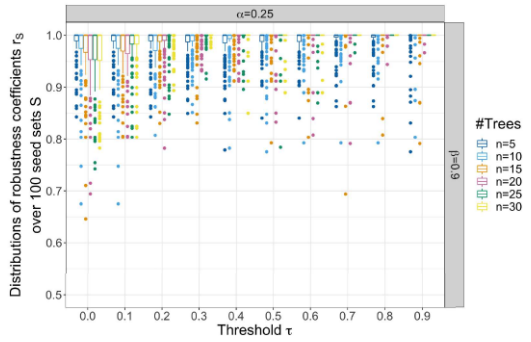

Fig. 2. Effect of the number of trees n and the threshold τ on the robustness of ROBUST for $\alpha = 0.25$ and $\beta = 0.9$

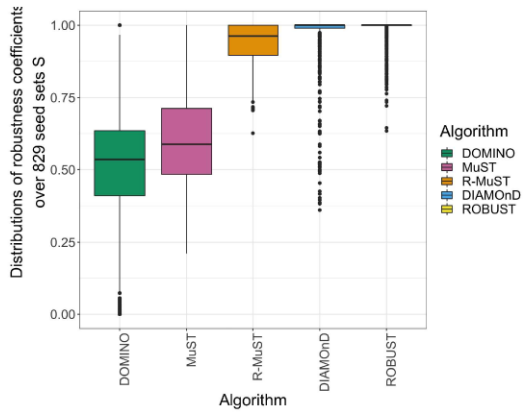

Fig. 3. Robustness of ROBUST with hyper-parameter setup $(\alpha, \beta, n, \tau) = (0.25, 0.9, 30, 0.1)$ in comparison to its competitors. Like ROBUST, MuST and R-MuST were setup to run with $(n, \tau) = (30, 0.1)$

Table 1. *P*-values obtained by comparing the robustness coefficients from two DMMMs via the Mann–Whitney *U* test (alternative hypothesis: DMMM 1 yields larger robustness coefficients than DMMM 2)

DMMM 1	DMMM 2	*P*-value
ROBUST	DOMINO	1.668×10^{-278}
ROBUST	MuST	3.847×10^{-226}
ROBUST	R-MuST	4.460×10^{-68}
ROBUST	DIAMOnD	6.796×10^{-6}

Note: Note that the *P*-values should be interpreted carefully, because the Mann–Whitney *U* test is a rank-based test and can hence yield very small *P*-values even if the quantitative differences between the compared populations are small.

(run DMMM n times on shuffled input and then return subgraph induced by nodes contained in many modules) is not guaranteed to be effective.

3.3 Functional relevance

3.3.1 Protocol and data used for functional relevance tests

Functional relevance tests were conducted by implementing custom wrappers for the DMMM test suite introduced by Lazareva *et al.* (2021). In the test suite, gene expression datasets with case/control information for five complex diseases are used, namely ALS, non-small cell LC, UC, CD and HD. The seed genes are obtained by applying a two-sided Mann–Whitney *U* test on the case/control expression vectors and extracting all genes with Bonferroni-adjusted *P*-values <0.001. Each set of seed genes was projected onto one of the five widely used PPI networks BioGRID (Oughtred *et al.*, 2019), APID (Agile Protein Interactomes DataServer) (Alonso-Lopez *et al.*, 2016, 2019), STRING (Search Tool for the Retrieval of Interacting Genes/Proteins) (Szklarczyk *et al.*, 2019) with high confidence interactions only, HPRD (Human Protein Reference Database) (Keshava Prasad *et al.*, 2009) and IID (Kotlyar *et al.*, 2019). Functional relevance was evaluated via gene set enrichment *P*-values w.r.t. KEGG pathways (Kanehisa *et al.*, 2016) corresponding to the diseases and via overlap coefficients with the disease-associated DisGeNET (Piñero *et al.*, 2020) gene sets. For more details, we refer to Lazareva *et al.* (2021).

3.3.2 Functional relevance in comparison to competitors

Figure 4 shows the distributions of the functional relevance scores for ROBUST, MuST, DIAMOnD and DOMINO run on the five disease networks. The baseline implementation R-MuST was excluded from the tests due to high runtime requirements and poor robustness results. Overall, ROBUST outperformed the other three DMMMs w.r.t. both functional relevance scores. The CD dataset is the only case where DIAMOnD clearly yielded better results than ROBUST.

The test suite introduced by Lazareva *et al.* (2021) also supports permutation tests to assess to which extent DMMMs are potentially biased toward hub nodes in the PPI networks (for details, we refer to the original publication). This was also tested for ROBUST, DIAMOnD and DOMINO, the results are shown in Supplementary Figure S2 (MuST was excluded because of its high runtime and the large number of runs required for the permutation tests). In this dimension, ROBUST performed similarly to DIAMOnD but was outperformed by DOMINO. To reduce the risk of including false positives into the solution, it might hence be advisable to run ROBUST on context-specific networks, e.g. by keeping edges only for PPIs experimentally validated in tissue relevant for the disease of interest. We followed this approach for our case study in MS (Section 3.5).

3.4 Scalability

3.4.1 Protocol and data used for scalability tests

As for the robustness tests, we used a human PPI network obtained from IID, filtered based on experimental validation. We randomly generated seed sets of sizes $k = 25, 50, \ldots, 400$, ran all compared DMMMs on all of them and measured the runtimes. For ROBUST, MuST and R-MuST, we additionally varied the number of trees $n \in \{5, 10, \ldots, 30\}$. For all DMMMs except MuST and R-MuST and each seed set size k, runtimes were measured on 10 random seed sets of size k. MuST and R-MuST were evaluated only on one seed set for each k, because of their high runtime requirements.

3.4. 2 Scalability in comparison to competitors

Figure 5 shows the results of the scalability tests. The most important observations are the following:

- MuST and R-MuST are around 2 orders of magnitude slower than DIAMOnD, DOMINO and ROBUST.
- While, for ROBUST and DOMINO, the runtime increases sublinearly with the number of seeds, we observe a linear increase in runtime for DIAMOnD.
- Increasing the number of trees affects ROBUST's runtime only very moderately.

In sum, ROBUST hence exhibits the best runtime behavior even if the number of trees is set to $n = 30$ as suggested above: For small seed sets, ROBUST is approximately as fast as DIAMOnD and around five times faster than DOMINO. For large seed sets, it is around twice as fast as DIAMOnD and between three and four times faster than DOMINO.

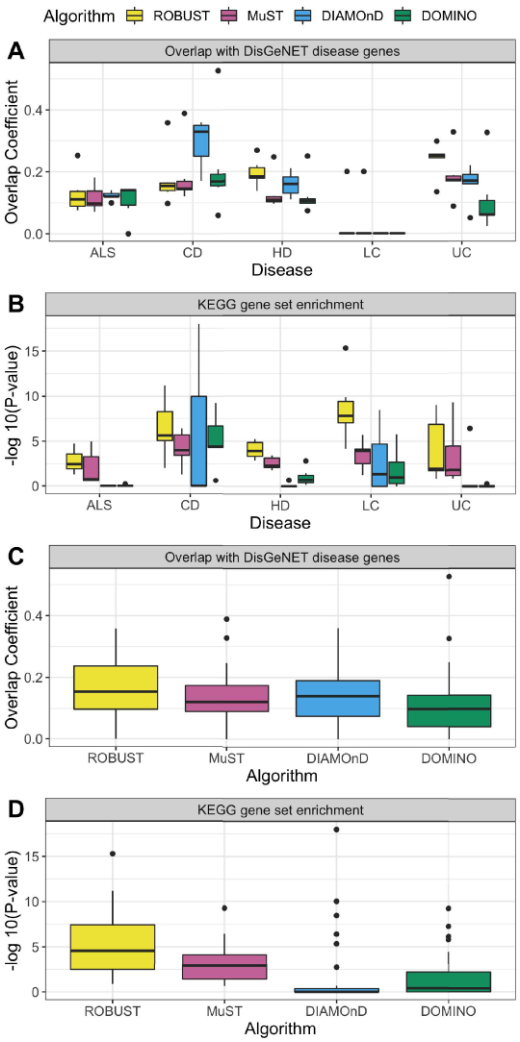

Fig. 4. Distribution of the functional relevance scores for ROBUST, MuST, DIAMOnD and DOMINO. (**A**) Overlap coefficients of the disease modules and the DisGeNET disease genes, split by disease. (**B**) KEGG gene set enrichment *P*-values split by disease. (**C**) Overlap coefficient distributions over all networks and disease seed sets. (**D**) KEGG gene set enrichment distributions over all networks and disease seed sets

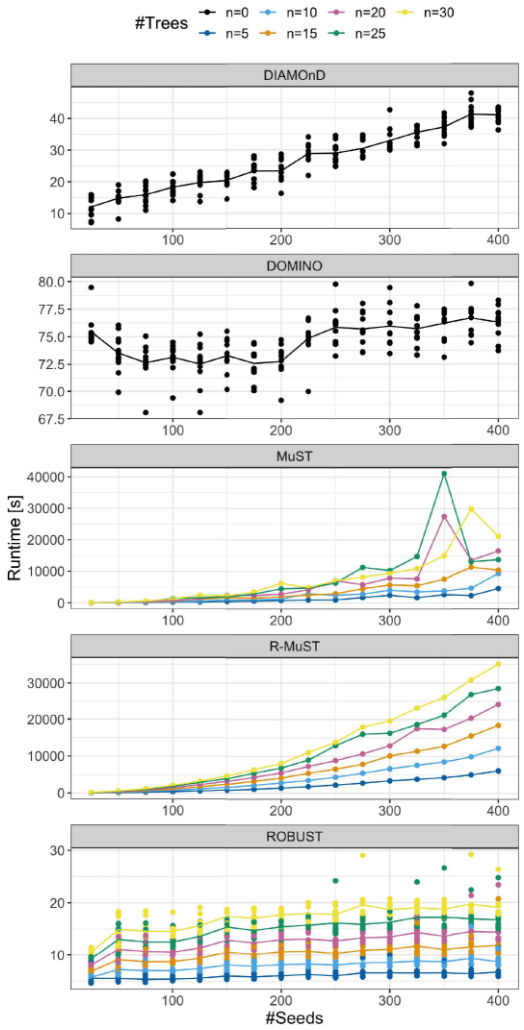

Fig. 5. Runtime of the DMMMs versus number of seeds (all DMMMs) and number of seeds and trees (MuST, R-MuST and ROBUST). The line plots visualize the mean runtimes

3.5 Case study in MS

In addition to the quantitative evaluation reported in the previous sections, we performed a case study in MS to showcase how to use ROBUST for hypothesis generation. First, we constructed a context-specific PPI network from IID by filtering for the interactions experimentally validated in brain tissue. Then, proteins associated with MS were obtained by merging DisGeNet and OMIM annotations. This yielded 42 seeds, 26 of which were contained in the context-specific network.

Running ROBUST on these 26 seeds yielded a disease module with 90 additional proteins (Supplementary Fig. S3), including galectin-1, which was found in each of the 30 trees. It has been shown that galectin-1 plays an important regulatory role in MS patients (Starossom *et al.*, 2012). We then took a closer look at the 2-hop neighborhood of galectin-1 within the computed diseases module (Supplementary Fig. S4). In this submodule, we found thioredoxin (TXN), peroxiredoxin-2 (PRDX2), the mitochondrial thioredoxin-dependent peroxide reductase (PRDX3) and DJ-1 (extended findings in the Supplementary Material).

TXN, PRDX2, PRDX3 and DJ-1 are antioxidant molecules related to oxidative stress, a sign of various neurological disorders including MS (Liu *et al.*, 2020). TXN has been found to be significantly upregulated in MS patients compared to healthy controls (Mahmoudian *et al.*, 2017). PRDX2 and PRDX3 are enzymes which reduce H_2O_2 and hydroperoxides using TXN as substrate (Cao *et al.*, 2007; Kamariah *et al.*, 2016). PRDX2 was shown to be upregulated in white matter MS lesions (Voigt *et al.*, 2017). While DJ-1 is not directly linked to TXN, the two molecules share downstream targets and it has been suggested that there is some cross-talk between these two systems (Raninga *et al.*, 2014) and various studies have linked DJ-1 to MS (Hirotani *et al.*, 2008; van Horssen *et al.*, 2010). These findings show how ROBUST can identify a submodule related to oxidative stress in MS whose participants share common pathways.

4 Conclusions, limitations and outlook

In this article, we have presented a novel DMMM called ROBUST which, unlike existing approaches, computes almost perfectly robust disease modules when run multiple times on equivalent input. ROBUST is also faster than its competitors and, in most settings, outperforms them in terms of functional relevance of the computed modules.

We conclude this article by pointing out three limitations of ROBUST which constitute challenges for future work. First, ROBUST supports only binary input and so future work is needed to overcome the robustness deficit in disease module detection with

continuous input. Second, ROBUST is outperformed by DOMINO w.r.t. resistance to hub node bias and it hence remains an open algorithmic challenge to design a DMMM which is both immune to hub node bias and robust w.r.t. random storage order. Third, it would be interesting from a theoretical point of view to investigate whether the Steiner tree enumeration problem underlying ROBUST can be formalized such that it allows for approximation algorithms with provable approximation guarantees.

Funding

This work received funding from the European Union's Horizon 2020 research and innovation program under Grant Agreements 826078 (J.B.) and 777111 (S.S., T.K., J.B.). This publication reflects only the authors' view, and the European Commission is not responsible for any use that may be made of the information it contains. This work was supported by the German Federal Ministry of Education and Research (BMBF) under Grant 01ZX1908A (J.B., T.K., M.L.), Grant 01ZX1910D (J.B.) and Grant 031L0214A (J.B.).

Data availability

All data required to reproduce the results reported in this article are available on GitHub: https://github.com/bionetslab/robust-eval.

Conflict of Interest: none declared.

References

Alonso-Lopez,D. *et al.* (2016) APID interactomes: providing proteome-based interactomes with controlled quality for multiple species and derived networks. *Nucleic Acids Res.*, **44**, W529–W535.

Alonso-López,D. *et al.* (2019) APID database: redefining protein–protein interaction experimental evidences and binary interactomes. *Database*, **2019**, baz005.

Amberger,J.S. *et al.* (2019) OMIM.org: leveraging knowledge across phenotype-gene relationships. *Nucleic Acids Res.*, **47**, D1038–D1043.

Barabási,A.-L. *et al.* (2011) Network medicine: a network-based approach to human disease. *Nat. Rev. Genet.*, **12**, 56–68.

Barel,G. and Herwig,R. (2020) Netcore: a network propagation approach using node coreness. *Nucleic Acids Res.*, **48**, e98.

Batra,R. *et al.* (2017) On the performance of de novo pathway enrichment. *NPJ Syst. Biol. Appl.*, **3**, 6.

Breitling,R. *et al.* (2004) Graph-based iterative group analysis enhances microarray interpretation. *BMC Bioinform.*, **5**, 1–10.

Byrka,J. *et al.* (2013) Steiner tree approximation via iterative randomized rounding. *J. ACM*, **60**, 1–33.

Cao,Z. *et al.* (2007) Reconstitution of the mitochondrial prxiii antioxidant defence pathway: general properties and factors affecting prxiii activity and oligomeric state. *J. Mol. Biol.*, **372**, 1022–1033.

Ding,Z. *et al.* (2018) ClustEx2: gene module identification using density-based network hierarchical clustering. In: ClustEx2: Gene Module Identification using Density-Based Network Hierarchical Clustering. 2018 Chinese Automation Congress (CAC). pp. 2407–2412.

Fernández-Tajes,J. *et al.* (2019) Developing a network view of type 2 diabetes risk pathways through integration of genetic, genomic and functional data. *Genome Med.*, **11**, 19–14.

Ghiassian,S.D. *et al.* (2015) A DIseAse MOdule Detection (DIAMOnD) algorithm derived from a systematic analysis of connectivity patterns of disease proteins in the human interactome. *PLoS Comput. Biol.*, **11**, e1004120.

Goemans,M.X. and Williamson,D.P. (1995) A general approximation technique for constrained forest problems. *SIAM J. Comput.*, **24**, 296–317.

Hegde,C. *et al.* (2014) A fast, adaptive variant of the Goemans-Williamson scheme for the prize-collecting Steiner tree problem. In: 11th DIMACS Implementation Challenge. https://dimacs11.zib.de/workshop.html.

Hegde,C. *et al.* (2015) A Nearly-Linear Time Framework for Graph-Structured Sparsity. In: Proceedings of the 32nd International Conference on Machine Learning, (ICML) 2015, Lille, France, 6-11 July 2015, Vol. 37, JMLR, pp. 928—937. http://proceedings.mlr.press/v37/hegde15.html.

Hirotani,M. *et al.* (2008) Correlation between DJ-1 levels in the cerebrospinal fluid and the progression of disabilities in multiple sclerosis patients. *Mult. Scler. J.*, **14**, 1056–1060.

Ideker,T. *et al.* (2002) Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, **18**, S233–S240.

Kamariah,N. *et al.* (2016) Transition steps in peroxide reduction and a molecular switch for peroxide robustness of prokaryotic peroxiredoxins. *Sci. Rep.*, **6**, 1–15.

Kanehisa,M. *et al.* (2016) KEGG as a reference resource for gene and protein annotation. *Nucleic Acids Res.*, **44**, D457–D462.

Keshava Prasad,T.S. *et al.* (2009) Human protein reference database—2009 update. *Nucleic Acids Res.*, **37**, D767–D772.

Kotlyar,M. *et al.* (2019) IID 2018 update: context-specific physical protein–-protein interactions in human, model organisms and domesticated species. *Nucleic Acids Res.*, **47**, D581–D589.

Kou,L.T. *et al.* (1981) A fast algorithm for Steiner trees. *Acta Inf.*, **15**, 141–145.

Larsen,S.J. *et al.* (2020) De novo and supervised endophenotyping using network-guided ensemble learning. *Syst. Med.*, **3**, 8–21.

Lazareva,O. *et al.* (2021) On the limits of active module identification. *Brief. Bioinform.*, **22**, bbab066.

Levi,H. *et al.* (2021) DOMINO: a network-based active module identification algorithm with reduced rate of false calls. *Mol. Syst. Biol.*, **17**, e9593.

List,M. *et al.* (2016) KeyPathwayMinerWeb: online multi-omics network enrichment. *Nucleic Acids Res.*, **44**, W98–W104.

Liu,J. *et al.* (2020) Effects of peroxiredoxin 2 in neurological disorders: a review of its molecular mechanisms. *Neurochem. Res.*, **45**, 720–730.

Ma,H. *et al.* (2011) COSINE: condition-specific sub-network identification using a global optimization method. *Bioinformatics*, **27**, 1290–1298.

Mahmoudian,E. *et al.* (2017) Thioredoxin-1, redox factor-1 and thioredoxin-interacting protein, mRNAs are differentially expressed in multiple sclerosis patients exposed and non-exposed to interferon and immunosuppressive treatments. *Gene*, **634**, 29–36.

Matschinske,J. *et al.* (2021) The AIMe registry for artificial intelligence in biomedical research. *Nat. Methods*, **18**, 1128–1131.

Menche,J. *et al.* (2015) Disease networks. uncovering disease-disease relationships through the incomplete interactome. *Science*, **347**, 1257601.

Nacu,Ş. *et al.* (2007) Gene expression network analysis and applications to immunology. *Bioinformatics*, **23**, 850–858.

Oughtred,R. *et al.* (2019) The BioGRID interaction database: 2019 update. *Nucleic Acids Res.*, **47**, D529–D541.

Piñero,J. *et al.* (2020) The disgenet knowledge platform for disease genomics: 2019 update. *Nucleic Acids Res.*, **48**, D845–D855.

Raninga,P.V. *et al.* (2014) Cross talk between two antioxidant systems, thioredoxin and DJ-1: consequences for cancer. *Oncoscience*, **1**, 95–110.

Reyna,M.A. *et al.* (2018) Hierarchical HotNet: identifying hierarchies of altered subnetworks. *Bioinformatics*, **34**, i972–i980.

Roy,J. *et al.* (2014) Network information improves cancer outcome prediction. *Brief. Bioinform.*, **15**, 612–625.

Sadegh,S. *et al.* (2020) Exploring the SARS-CoV-2 virus-host-drug interactome for drug repurposing. *Nat. Commun.*, **11**, 1–9.

Samokhin,A.O. *et al.* (2018) NEDD9 targets COL3A1 to promote endothelial fibrosis and pulmonary arterial hypertension. *Sci. Transl. Med.*, **10**, eaap7294.

Sharma,A. *et al.* (2015) A disease module in the interactome explains disease heterogeneity, drug response and captures novel pathways and genes in asthma. *Hum. Mol. Genet.*, **24**, 3005–3020.

Sharma,A. *et al.* (2018) Controllability in an islet specific regulatory network identifies the transcriptional factor NFATC4, which regulates type 2 diabetes associated genes. *NPJ Syst. Biol. Appl.*, **4**, 1–11.

Starossom,S.C. *et al.* (2012) Galectin-1 deactivates classically activated microglia and protects from inflammation-induced neurodegeneration. *Immunity*, **37**, 249–263.

Szklarczyk,D. *et al.* (2019) STRING v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic Acids Res.*, **47**, D607–D613.

van Horssen,J. *et al.* (2010) Nrf2 and DJ1 are consistently upregulated in inflammatory multiple sclerosis lesions. *Free Radic. Biol. Med.*, **49**, 1283–1289.

Voigt,D. *et al.* (2017) Expression of the antioxidative enzyme peroxiredoxin 2 in multiple sclerosis lesions in relation to inflammation. *Int. J. Mol. Sci.*, **18**, 760.

Wang,R.-S. and Loscalzo,J. (2018) Network-based disease module discovery by a novel seed connector algorithm with pathobiological implications. *J. Mol. Biol.*, **430**, 2939–2950.