Tel-Aviv University

Raymond and Beverly Sackler Faculty of Exact Sciences

The Blavatnik School of Computer Science

Analysis of network-based module discovery algorithms from the perspective of biological enrichment

Thesis submitted in partial fulfillment of graduate requirements for

The degree "Master of Sciences" in Tel-Aviv University

School of Computer Science

By

Hagai Levi

Prepared under the supervision of

Prof. Ron Shamir

Dr. Ran Elkon

November 2019

Acknowledgements

I would like to express here how grateful I am to the people who accompanied me in this journey. First, my supervisors: Prof. Ron Shamir is a textbook example for a scientist: a combination of a researcher, who strains for convincing results while holding a constant critical point of view to his own work, and a distinguished teacher. Dr. Ran Elkon guided me how to evolve our work, one step at a time, until creating a considerable body of research. But even more important, he gave me the confidence and the solid ground I needed in challenging times. The impact of having them both as co-advisers was invaluable. I hope our joint effort will serve the scientific community.

Many thanks are reserved to Ron’s lab members – Lianrong, Dvir, Ron Z., David P., Tom, Nimrod, Idan, Dan, Neta, Aviv, Yael, Roi, Naama K., Hadar, Omer – as well as to Ran’s lab members - Eldad, Avinoam, Michal, David G., Zohar, Naama M. and Amit – for being there, listening and giving good advises for in research and day to day matters. My experience would not have been the same without them.

Special thanks reserved for Gilit Zohar-Oren for all her administrative support.

I would like to thank the funding agencies that supported my MSc studies: The Israel Science Foundation (grants 2193/15 and 1339/18), German-Israeli Project (DFG grant RE 4193/1-1) and the Edmond J. Safra Center for Bioinformatics at Tel Aviv University (MSc fellowship). This funding gave me the conditions I needed to put my focus on research.

The last sentence is dedicated to thanking my family for all their support. It is an honor to mention them here.

Table of Contents

i. Abstract

Network-based module detection (NBMD) algorithms have been used to functionally interpret omics data for almost two decades. These algorithms receive as an input a biological network and nodes' (genes) activity scores and report sub-networks that are putatively biologically meaningful. In this study we aimed to systematically compare the performance of NBMD algorithms on data recorded by transcriptome profiling and genome-wide association studies (GWASs). We focused on six of the most popular NBMD tools: jActiveModules (using either simulated annealing or a greedy search), NetBox, Bionet, HotNet2 and KeyPathwayMiner, and based our evaluation of the NBMD algorithms' solution on module enrichment for Gene Ontology (GO) terms.

We first observed that many GO terms reported by each algorithm were also reported when the same algorithm was applied to permuted data, which we hypothesized to stem from algorithm behavior and network structure. Therefore, to remove from the solutions GO terms that are recurrently called in permuted data, we developed the EMpirical Pipeline (EMP), a method that quantifies the significance of GO enrichment scores of a solution by comparing each term’s scores on the real and permuted data. We then designed novel criteria for evaluating NBMD solutions based on the output of the EMP procedure, and used them to compare the performance of the six NBMD tools on different gene expression and genome wide association study (GWAS) datasets. Notably, NetBox consistently outperformed the other NBMD algorithms. Finally, we designed a novel NBMD algorithm called Domino (Discovery of Modules In Networks using Omics), and demonstrated that it outperformed the algorithms we benchmarked. Importantly, the mean EMP validation rate for enriched GO terms detected for Domino's solutions was above 90%, markedly higher than for the other tools. Running the EMP procedure for several thousand permutations is computationally heavy. Given Domino’s high performance and validation rate, it can be used in biological studies on a standard desktop, without the need for the EMP filtering procedure.

1. Introduction

Network-based analyses have a central role in bioinformatics[1]. These analyses utilize biological networks - graphs that represent intracellular biological behavior, by describing a specific cellular unit (e.g. gene, protein, compound etc.) as a node and a behavior that involves two components (physical interaction, co-expression, regulation etc.) as an edge.

Many types of biological networks exist, including protein-protein interaction networks (PPI)[2], metabolic networks[3], regulatory networks[4] and more. They are usually generated based on many assays that examine intracellular behaviors and are integrated together into a single graph structure. Examples of well-known biological network include STRING[5] – the largest multilayer network, which includes many types of edges such as co-expression and physical interaction, and ReactomeFI[6] – a network that is built from multiple pathway databases and additional high-throughput sources. These networks vary in size and can span from a few thousand nodes and edges up to many thousands of nodes and millions of edges.

One of the fundamental challenge in bioinformatics is extracting the biological signal from an assay: naive selection of gene by some threshold (e.g. fold change or p-value) might miss some real biological signals: Sometimes the assay's measurements are incomplete: Some gene values are missing in some of the samples. Additionally, sometimes the observed signals are too weak when looking at the sum of individual gene measurements[7]. This is especially true for high-throughput assays that apply stringent threshold (to account for multiple testing) and end up filtering many relatively high scoring genes (e.g. GWAS). In order to deal with this challenge, many methods use biological networks as an integrative resource by which the proper context is given to assay measurement, i.e., amplifying biological signals that otherwise - just by looking on the assay - would be overlooked.

Network-based module detection (NBMD) is an approach for joint analysis of an experimental assay and a biological network that has been in broad use for almost two decades. NBMD algorithms are used to infer important gene-sets (modules) by projecting the assay measurements over a biological network and identifying information-rich sub-networks. The input for such algorithms is a biological network and gene activity scores for its nodes. Gene activity scores are derived from a specific assay and represent the biological information in it. These gene activity scores can be binary (e.g. perturbed or not-perturbed) or continuous (e.g. p-values), and can come from various types of omic methods, including DNA mutation[8,9] and gene expression[10,11,12]. In many cases they can be executed with a different assay type from the one they originally were developed for[13], usually by applying a minor preprocessing step on the assay dataset. These algorithms output a solution consisting of several modules, which are usually connected subnetworks. Typically, candidate modules are selected by a certain strategy (e.g. greedy search), are evaluated by some objective function (e.g. average gene activity scores in the set) and the best-performing modules are output in the solution.

Given a set of modules, produced by such an algorithm, a common downstream analysis is functional analysis, aiming to characterize the modules biologically. The most common functional analysis is the hypergeometric (HG) enrichment analysis. Such analysis detects whether a module contains significantly larger proportion of genes with a specific function (annotation) than expected just by chance. The most popular annotation resource against which such enrichments are measured is the Gene Ontology (GO) [14] - a hierarchically-annotated gene-set database that comprises many annotation. These annotations are organized in three directed acyclic graph structures: “Biological Process”, “Molecular Function” and “Cellular Component”, where “Biological Process” is the most comprehensive one.

Since GO functional analysis is very common, and a variety of solutions were proposed to implement it, the need for good criteria for comparing solutions of different NBMD algorithms is evident. Manual analysis of the relevant terms in a solution by domain experts is very common in practice, but it is slow and may be biased. Consequently, functional-based evaluation of NBMD algorithms remains a challenging task.

In this study we performed a systematic evaluation of prominent module-discovery algorithms across several datasets from two different omic data types. Functionally enriched terms in each module and each solution were identified by the standard HG test, without manual intervention. Our analysis revealed that some algorithms recurrently report many terms as enriched, but some of those terms also show up when applying the same algorithm on a random permutation of the same dataset. This suggests that such solutions report terms that are likely false positives. To address this shortcoming, we developed a procedure that we call the EMpirical Pipeline (EMP). It "cleans" the set of enriched terms and their scores by comparing them to the enrichment scores of the same terms on permuted datasets. We used this approach along with evaluation metrics that we developed to evaluate six popular NBMD algorithms on different gene expression (GE) and GWAS datasets. Of those, the NetBox algorithm consistently outperformed the other tested algorithms.

Finally, we designed a novel NBMD algorithm called Domino (Discovery of Modules In Networks using Omics), and demonstrated that it outperformed the algorithms we benchmarked. Importantly, the mean EMP validation rate for enriched GO terms detected for Domino's solutions was above 90%, markedly higher than for the other tools. Running the EMP procedure is computationally heavy, as it requires running the algorithm repeatedly on several thousands of permutations of the gene scores. Domino’s high validation rate suggest that it can be run without the need for the EMP filtering procedure, and thus can be used in biological studies on a standard desktop.

2. Biological Background

2 A. Biological Networks

The cell is a complex system. The subunits composing it include proteins, genes and metabolic compounds – all of which interact together to create its internal ecosystem. Traditionally, researchers tend to examine how a single subunit affects a condition (e.g. diseases). While this approach is proper for some cases, e.g., Mendelian diseases, for many others it gives narrow perspective of few parameters while ignoring many others.

The last two decades were characterized by a rapid rise of the field of bioinformatics. Key drivers of this rise are high-throughput omics technologies – from microarrays to Next-Generation Sequencing (NGS). The abundance of assays served as a good basis for creating and using gene and protein networks – an integrative data structure representing many assays as a whole and providing a graphical and mathematical representation. As knowledge grew, the size of the networks increased, and larger computational resources were required for network-related applications.

A biological network summarizes knowledge on a biological system. Generally, nodes represent units in the network, while edges represent the interactions between the units. Typically, nodes represent cell subunits (e.g. proteins, genes, metabolic compounds) and edges represent relationships between these subunits (e.g. physical interaction, co-expression, regulation). Many network properties can be quantified either on the whole network (e.g. global modularity), sub-graphs (e.g. density) or on a single subunit (e.g. degree of node). All these scores can be used in order to derive biological insights and will be explained in following sections.

Many types of biological networks exist. The most common ones are:

1. Protein–protein interaction networks (PPIs): proteins are nodes and their physical interactions are edges. These are the most commonly used biological networks

2. Protein-DNA interaction \ Regulatory networks: The expression of genes is regulated by transcription factors (TFs), proteins that bind to DNA regions (primarily promoters and enhancers) and control the gene's transcription. In regulatory networks, nodes represent proteins or genes and edges represent regulatory interaction. The edges in such networks are directed, representing the regulatory direction (e.g. *from* a TF *to* a promoter)

3. Co-expression networks: genes (nodes) are linked by edges if their expression is highly correlated over a large number of samples.

4. Metabolic networks: These networks represent metabolites, chemical compounds that participate in biochemical reactions, and directed edges represent transformation of one

compound into another in a reaction. Additionally, enzymes that catalyze these reactions are also present in these networks and point to the reactions they catalyze.

5. Signaling networks: Signals are transduced between different subunits of the cell and thus form complex signaling networks. Typically signaling transduction composed from protein–protein interactions, phosphorylation reactions, and metabolic reactions – all of which are combined into one signaling network, with some of these edges directed.

Sometimes the terms ‘pathway’ and ‘network’ are used interchangeably[15]. Although both are similar concepts, they have certain distinctions. Both comprise systems of cell subunits that carry out biological functions, that giving more biological holistic views. Pathways are small-scale systems of well-studied processes where interactions comprise biochemical reactions and events of regulation, complex formation and signaling. They represent high-quality knowledge based on decades of research and can be visualized in detailed flow diagrams. In contrast, networks summarize interactions of many subunits that are unspecific to a certain biological process. While pathways describe thoroughly a specific biological process, biological network are simplified abstractions of global complex cellular logic. While each interaction in a pathway is well-studied and of high confidence, networks are noisier. However, they likely contain uncovered information that does not exist in pathways. Finally, while pathways are easier to visualize and to interpret by visual inspection, a full biological network usually looks like a giant ‘hairball’, leaving us with little insights as a raw data structure.

2 B. Gene Ontology

The Gene Ontology (GO)[14] is a major bioinformatics resource that unifies the representation of gene and gene product attributes in a cross-species language. GO terms are genes’ biological annotations in this language. These annotations implicitly define gene-sets: Each GO term corresponds to the set of genes that are annotated by this term. In addition, GO provides hierarchical relationships between GO terms. The GO resource covers three domains (definitions are taken from http://geneontology.org/docs/ontology-documentation/):

1. Cellular Component (CC): The locations relative to cellular structures in which a gene product performs a function, either cellular compartments (e.g., mitochondrion), or stable macromolecular complexes of which they are parts (e.g., the ribosome). Unlike the other aspects of GO, cellular component classes refer not to processes but rather a cellular anatomy (~5K terms).
2. Molecular Function (MF): Molecular-level activities performed by gene products. Molecular function terms describe activities that occur at the molecular level, such as “catalysis” or

“transport”. GO molecular function terms represent activities rather than the entities (molecules or complexes) that perform the actions, and do not specify where, when, or in what context the action takes place. Molecular functions generally correspond to activities that can be performed by individual gene products (i.e. a protein or RNA), but some activities are performed by molecular complexes composed of multiple gene products. Examples of broad functional terms are “catalytic activity” and “transporter activity”; examples of narrower functional terms are “adenylate cyclase activity” or “Toll-like receptor binding”. To avoid confusion between gene product names and their molecular functions, GO molecular functions are often appended with the word “activity” (e. g., a protein kinase would have the GO molecular function protein kinase activity). (~12K terms).

3. Biological Process (BP): The larger processes, or ‘biological programs’ accomplished by multiple molecular activities. Examples of broad biological process terms are “DNA repair” or “signal transduction”. Examples of more specific terms are “pyrimidine nucleobase biosynthetic process” or “glucose transmembrane transport”. Note that a biological process is not equivalent to a pathway. At present, the GO does not try to represent the dynamics or dependencies that would be required to fully describe a pathway (~20K terms).

GO is structured as a directed acyclic graph of terms (nodes in the graph), each with defined relationships (represented as directed edges) to one or more other terms in the same domain and sometimes to other domains. There are several types of relationship: “is a”, “part of”, “regulates”. Each GO term has many attributes, such as id, name and description. See Figure 1. GO is a key reference resource for enrichment analysis: a statistical approach by which one can assign a gene-set with putative biological functions (see Section 3 C. Enrichment Analysis: Hypergeometric Test).

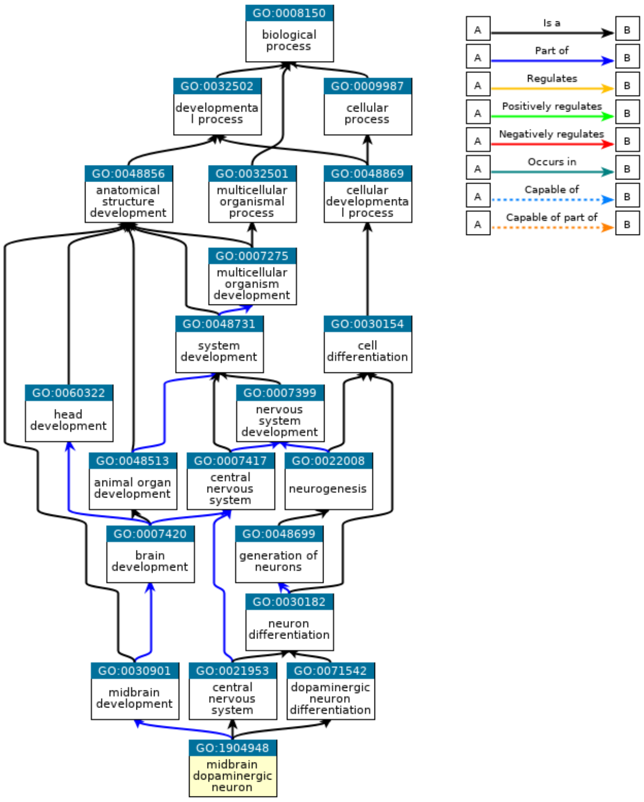
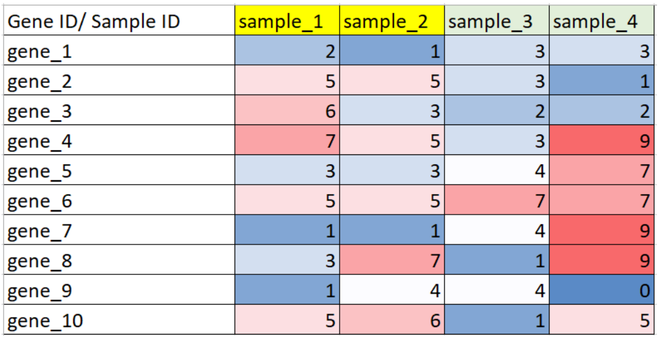


Figure 1: A sub-graph from the domain "biological process" containing all terms that are ancestors of GO term "GO:1904948 - midbrain dopaminergic neuron". Term A is an ancestor of term B if A is a generalization of , i.e., every gene annotated with term B is also annotated with term A (Source: QuickGO website[16]).

2 C. Gene Expression Profiling

Gene expression studies compare expression levels of genes on genomic scale across different samples. The levels of all genes in a sample are called its expression profile. Often, the samples have properties (e.g., control, treatment, condition or cell type) according to which they can be divided into biologically meaningful groups for downstream analyses (e.g., differential expression analysis). The most common technologies to measure expression profiles are microarray and RNA-Seq.

A gene expression matrix summarizes expression levels of genes across samples. Conventionally, each row represents a gene and each column represents a sample. The exact size of these matrices depends on the number of measured genes and the number of samples (Figure 2).



Gene ID/ Sample ID	sample_1	sample_2	sample_3	sample_4
gene_1	2	1	3	3
gene_2	5	5	3	1
gene_3	6	3	2	2
gene_4	7	5	3	9
gene_5	3	3	4	7
gene_6	5	5	7	7
gene_7	1	1	4	9
gene_8	3	7	1	9
gene_9	1	4	4	0
gene_10	5	6	1	5

Figure 2: Illustration of a gene expression matrix. The yellow columns represent the control samples and the green ones are the treated samples. The blue-red colors represent the expression levels.

2 D. Genome-Wide Association Studies

A genome-wide association study (GWAS) scans genetic markers across a very large case-control cohort – typically, many thousands of individuals - and finds genetic variants - single nucleotide polymorphism (SNP) - associated with a particular disease. Such studies are particularly useful in characterizing polygenetic complex diseases. To carry out a GWAS, researchers genotype samples from two groups: individuals with the disease (cases) and individuals without the disease (controls).For each SNP, it is tested if one allele is significantly more frequent in the cases, and therefore putatively associated with the disease[17] (Figure 3). In most cases (due to genetic correlation between SNPs that are closed to each other), the disease-associated SNPs are not the direct cause of the disease, but serve in various downstream analyses as pointers to genomic regions, such as genes, promoters, and enhancers in order to find which contain the causal variants[18,19].

For each individual, several million SNPs are genotyped and examined. After Bonferroni correction for multiple testing, only a few turn out to be significant. Yet, recent studies showed that there is additional information in SNPs that failed reaching the strict cutoff induced by Bonferroni correction (p-value < 10^{-8}). In a typical GWAS, there are thousands of informative SNPs with very weak effect. As a result, functional interpretation of GWAS results is still a major challenge.

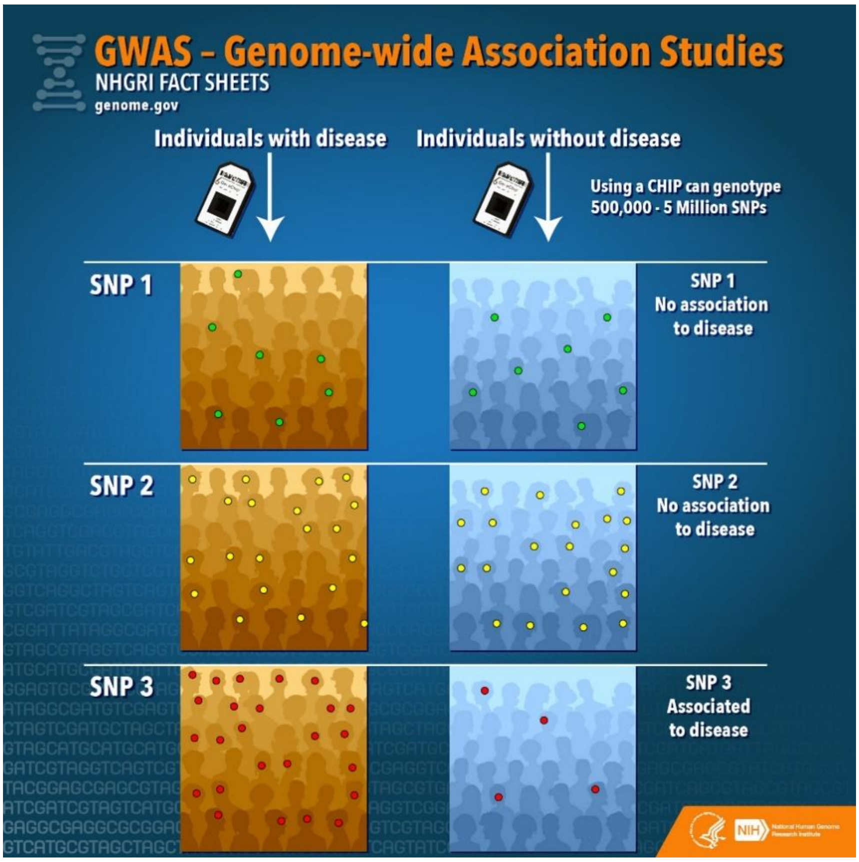

GWAS – Genome-wide Association Studies
NHGRI FACT SHEETS
genome.gov
Individuals with disease
Individuals without disease
Using a CHIP can genotype 500,000 - 5 Million SNPs
SNP 1
SNP 1 No association to disease
SNP 2
SNP 2 No association to disease
SNP 3
SNP 3 Associated to disease
NIH National Human Genome Research Institute

Figure 3: Schematic illustration of GWAS assay. Colored individuals are those who carry the SNP's minor allele (Source: NIH website[20])

3. Computational background

3 A. Module Identification in Networks

High-throughput biological assays can reveal genes/proteins that are perturbed in the analyzed biological condition[15,21]. In a gene-expression assay these can be the differentially expressed genes between control and treatment samples; In a genotyping assay these can be the mutated genes. In GWAS these can be the genes that contain SNPs that are significantly associated with the disease phenotype.

Each perturbed gene by itself may be hard to interpret, but by studying the set of perturbed genes together using a network model, better understanding can be obtained, thanks to the intracellular context the network adds. The analysis can pinpoint biological processes that are altered due to gene perturbations, compare the effect of perturbation of different genes and so on. In this way, the biological network puts the measurements in the context of known gene interactions. Furthermore, it opens the door to examine new properties such as relevance of unmeasured genes.

One way to put the perturbed gene set in a network context is to formulate an objective function that scores subgraphs in order to identify those with high signal of perturbation. The objective function, together with constraints on the allowed subgraph structure, are then used to find best-scoring subgraphs. Most formulations developed for the problem were proven NP-hard. This opens the door for many heuristic approaches for NBMD that give “good” but not provably optimal solutions. A solution of such algorithm is typically one or several modules, where each module is a set of genes corresponding to a connected subnetwork. Often, these modules are disjoint.

3 B. Evaluating Solutions

A common approach to evaluate the quality of an NBMD solution is by examination of its modules' biological signal. It is typically assessed by comparing the genes in each of the modules against biological databases, such as trait-to-gene-set databases (e.g. DisGenNet[22]), pathways (e.g. KEGG[23], Reactome[24]) or ontologies (e.g. GO[14]). One approach measures the recovery of an expected gene-set. This is mostly done for trait-related datasets against gold-standard gene-set. The drawback of this approach is the need to have a gold standard gene-set. An alternative approach is to compute enrichment scores of every gene-set in the database, followed by examining the relation between the (top) significantly enriched sets and the condition tested in the datasets. Both of these approaches are inevitably exposed to researcher bias, e.g., the researcher determines the relevance of gene-sets.

Recently, a benchmark[25] was conducted on 11 NBMD algorithms: The algorithms' input were gene expression datasets and the network was HPRD[26] – a manually curated experimentally verified PPI network (~5000 nodes, ~18000 edges). The first criterion in the benchmark was recovery of ground-truth genes out of simulated datasets. The recovery was measured by precision, recall and F1 scores, which are defined as follows:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

Where TP is the number of actual positives that are correctly identified as such, FP the number of actual negatives that are correctly identified as such, and FN are the number of positives that are incorrectly identified as negative. These criteria did not identify a clear best-performing algorithm.

Another criterion used by the same study measured the recovery of known prostate cancer genes from analysis of gene expression profiles of prostate-cancer patients and healthy individuals. This criterion measured the fold enrichment in the genes included in the solutions to known prostate cancer (PC) genes:

$$\frac{(\#\text{of recovered PC genes}) \cdot (\#\text{ total genes in network})}{(\#\text{ PC related genes}) \cdot (\#\text{ of genes in the solution})}$$

Where the selected genes are those identified by the algorithm as part of modules, and recovered genes are selected prostate cancer genes. Here the authors reported PinnacleZ as the best performing method (fold enrichment=2.494). However, only a small difference separated it from the second best method (WMAXC, fold enrichment=2.35). Both of these methods got relatively high F1 scores.

3 C. Enrichment Analysis: Hypergeometric Test

One of the most popular analyses in bioinformatics is enrichment analysis – a family of statistical tests that aim to determine whether a selected set of genes has unusually large proportion of genes with a particular annotation. Enrichment tests use popular gene-set databases such as MSigDB, GO and KEGG. The most classic enrichment test is the hypergeometric test. It can be defined as follows: Suppose you have N balls, out of which K < N are red ones. Assuming you draw n balls at random without repetition, the chances that exactly k of them are red ones is:

$$hg(N, K, n, k) = \Pr(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

The probability of drawing at least k red balls (or the p-value of the test) is:

$$HG(N, K, n, k) = \Pr(X \geq k) = \sum_{i=k}^{\min(k,n)} hg(N, K, n, i)$$

In this study we refer define the *enrichment score* as: $-\log_{10}(HG(N, K, n, k))$

3 D. Semantic Similarity

Given its hierarchical structure, many GO terms may capture similar biological information. Term similarities are reflected by their topological relationships (e.g. parent-children relationship: “GO:0048839 Inner Ear Development” and “GO:0090102:Cochlea Development”, or siblings relationship: “GO:0090102:Cochlea Development and “GO:0042472:Inner-Ear Morphogenesis”). Moreover, a single gene can be annotated with multiple GO terms, which can contribute to the similarity. Sematic similarity aims to measure how close any two terms are. Intuitively, semantic similarity increases when more properties are in common between the terms and when they are "closer" on the GO DAG (see Section 2 B. Gene Ontology). One powerful semantic similarity application is FastSemSim, which implements 16 different semantic similarity metrics. In our work analysis we used Resnik semantic similarity metric, which is described in the following section.

3 D i. Resnik's Semantic Similarity

This measure, developed by Phillip Resnik in 1999,[27] is one of the earliest semantic similarity methods and is still in use today. We demonstrate how it works over semantic similarity of concepts. We define concepts as objects in hierarchical taxonomy that maintain "is a" relationships (Figure 4). Each concept c has a known probability $p(c)$ to occur, and the *information content* of c, denoted $IC(c)$, is defined as $-\log\big(p(c)\big)$. The information content of a concept measures how common it is, with less frequent concepts having higher information content.

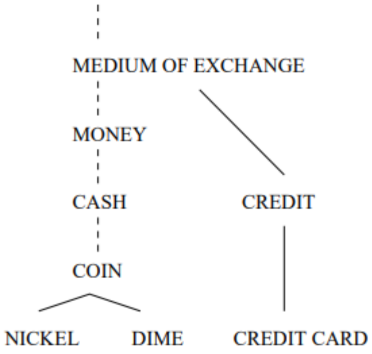


Figure 4: An illustration for concept taxonomy (Source: Resnik[27])

The mutual information of two concepts is determined by their *most informative common ancestor (MICA)*. Formally the similarity of concepts $c1, c2$ is defined as:

$$sim(c_1, c_2) = \max_{c \in S(c_1, c_2)} -\log p(c) = -\log p\big(MICA(c_1, c_2)\big)$$

Where $S(c_1, c_2)$ is the set of c_1's and c_2's common ancestors. Typically, $sim(c_1, c_2)$ values are determined by its least common ancestor, i.e., $MICA(c_1, c_2) = LCA(c_1, c_2)$.

To apply similarity to the GO hierarchy, we simply use terms instead of concepts. The probability of a terms is the fraction of gene-products annotated to it. Thus, the Resnik similarity of terms GO_1 and GO_2 is $IC(MICA(GO_1, GO_2))$. Notably, the higher the similarity of a term pair is, the higher the chance to find larger set of genes that are mutual to both terms. For instance, “GO: 0048839: Inner Ear Development” and “GO:0090102: Cochlea Development” has 6.27 Resnik score, with 4 overlapping genes, while “GO:0060348: Bone Development” and “GO:0090102:Cochlea Development” has only 2.63 score with zero overlapping genes (Figure 5).

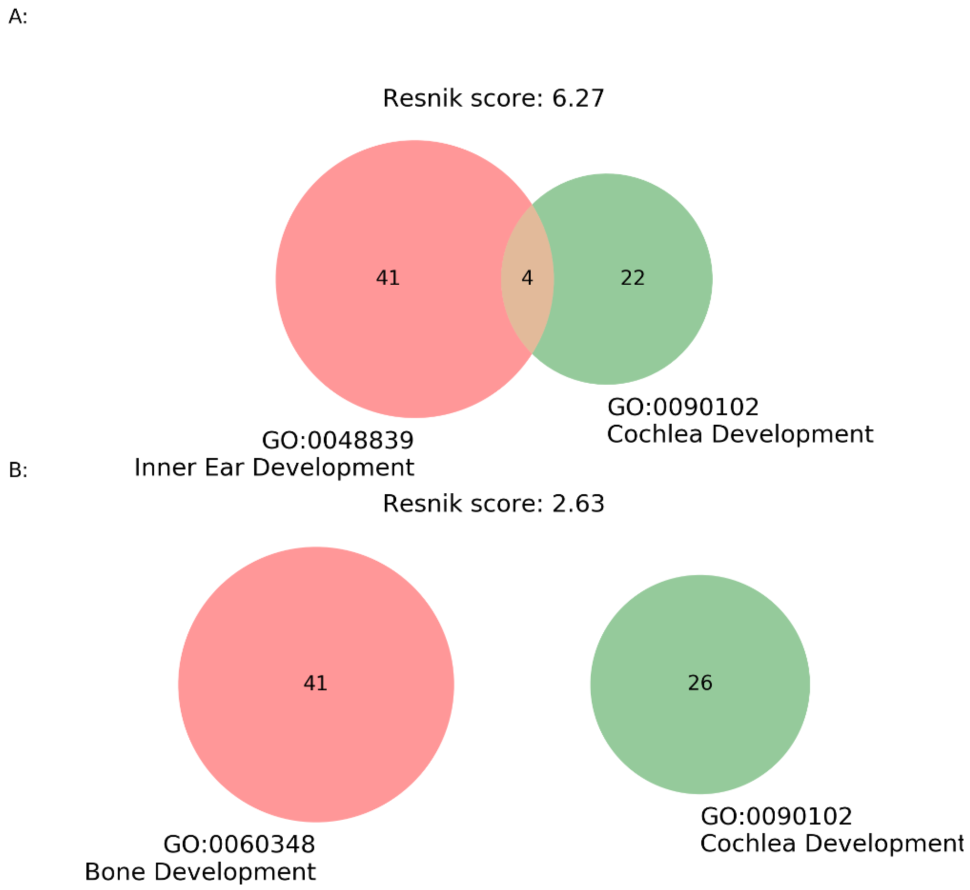


Figure 5: Resnik similarity scores and gene overlap between different GO terms. Gene annotations were acquired from GO resource files and GOATOOLS.

3 D ii. REVIGO

Frequently, GO enrichment analysis yields a solution with a long list of GO terms, many of which are closely related to each other. In order to get less redundant biological signal, one might want to remove highly similar terms from the list and leave only the most suitable representatives. REVIGO[28] groups highly similar GO terms and chooses a representative for each group. Representatives are determined based on their specificity, i.e., avoiding too general terms, and enrichment scores, i.e., preferring terms with higher scores. If the difference between the enrichment scores is below a certain threshold, and one term is a child of the other, REVIGO will choose the parent (more general) term. However, when at least 75% of the genes in the parent term belong also to child term, the child term is chosen instead, The REVIGO procedure is outlined in Figure 6.

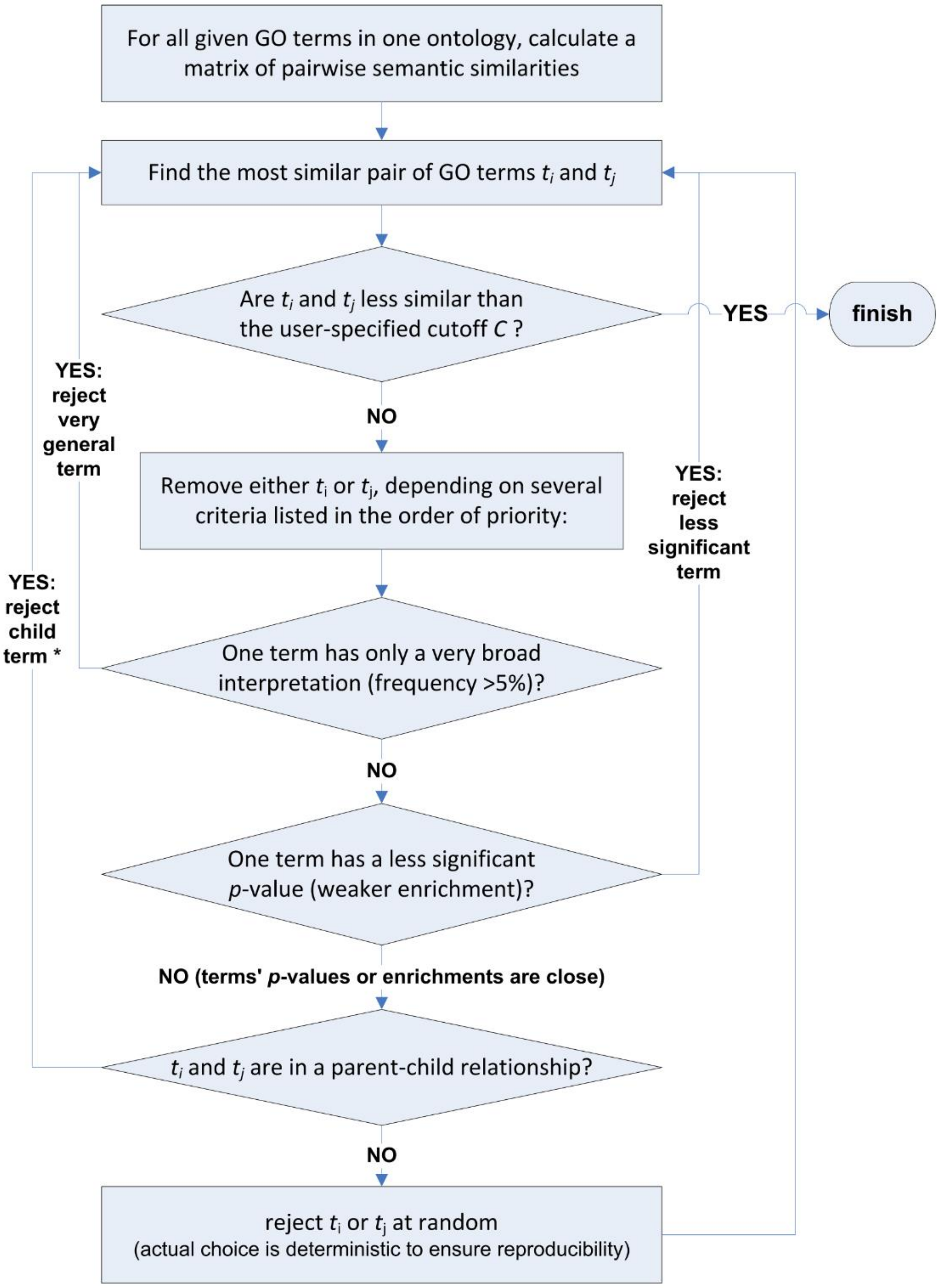

For all given GO terms in one ontology, calculate a matrix of pairwise semantic similarities
Find the most similar pair of GO terms t_i and t_j
Are t_i and t_j less similar than the user-specified cutoff C ?
YES
finish
NO
Remove either t_i or t_j, depending on several criteria listed in the order of priority:
YES: reject very general term
YES: reject child term *
YES: reject less significant term
One term has only a very broad interpretation (frequency >5%)?
NO
One term has a less significant p-value (weaker enrichment)?
NO (terms' p-values or enrichments are close)
t_i and t_j are in a parent-child relationship?
NO
reject t_i or t_j at random
(actual choice is deterministic to ensure reproducibility)

Figure 6: Schematic flow of the original REVIGO procedure. The flow receives a list of GO terms and their enrichment scores as an input and outputs a non-redundant list of GO terms (Source: Fran Supek & Tomislav Šmuc[28]*)*

4. Selected NBMD Algorithms

In the following section, we describe some popular network-based module detection (NBMD) algorithms that we used in our study (Table 1). The algorithms were chosen based on their popularity, diversity of purpose and computational ideas. Since we wished to test these algorithms extensively, we focused on those that had a working tool/codebase that can be executed as a stand-alone tool (some tools were isolated from larger packages), reasonable runtime and could be applied to different data types. We used these algorithms for the benchmark in Section Results.

Method name	**Published on**	**Designed for**	**Algorithmic Approach**	**Code language**	**# citations (updated to 11/2019)**
jActive-Modules[10]	2002	GE	Seek high scoring subnetworks by simulated annealing	Java	1207
NetBox[9]	2010	Mutation	Enrichment of Perturbed neighbors, Newman-Girvan (NG) modularity score	Java, Python	304
Bionet[11]	2010	GE	Prize collecting Steiner tree	R	218
HotNet2[8]	2015	Mutation	Heat diffusion	Python	460
KeyPathway Miner[12]	2012	GE	Choose modules with at most K non-perturbed genes	Java	41

Table 1: The NBMD algorithms used in our study. GE: Gene expression.

4 A. Overview: Approaches to Module Detection

NBMD algorithms typically perform two main steps: apply gene scores on the network (e.g., the vertices), and then detect modules according to these scores. In these steps the main variants are: (1) the way by which genes are scored (e.g., raw p-values or binary scores); (2) the biological network; (3) the module detection method (heat-diffusion, maximize gene scores' objective function etc.); and (4) the constraints of the solution, such as the number of reported modules (e.g. single, defined by an input, or determined by algorithm, extent of overlap allowed between modules etc.) Typically, users of such algorithms has some flexibility in (1) (2) and (4), but the module detection algorithm remains as is used as is.

4 B. Detailed Description of Selected Algorithms

In the following subsections we describe each algorithm and overview some of the results reported by the original papers

4 B i. jActiveModules

jActiveModules[10] was one of the pioneering algorithms for detecting "active gene modules" in a biological network given a gene expression dataset. It is integrated as a plugin in the Cytoscape software, and is still widely used today. It comprises two main concepts – a scoring system (for an individual gene and gene set), and module discovery strategy:

Scoring System

The scoring system consists of the following steps:

1. Compute for each gene i a p-value p_i for node i being differentially expressed between the control and treatment samples.

2. Assign Z-scores to nodes. These Z-score are calculated by the inversed normal CDF for each gene:

$$z_i=\Phi^{-1}(1-p_i)$$

3. Calculate Z-scores for subnetworks: The Z-score (Z_A) of a subnetwork with gene set A of size k is defined as:

$$z_A=\frac{1}{\sqrt{k}}\sum_{i\in A} z_i$$

Notably, the distribution of Z_A can be different for different values of k. The authors calibrated Z_A under normal distribution assumption (S_A).

$$s_A=\frac{(z_A-\mu_k)}{\sigma_k}$$

The parameters for the distribution (μ_k and σ_k) were extracted from background distribution of subnetworks of the same size by random sampling of gene sets of size k. This correction guarantees standard normal distribution function across all modules in all sizes and therefore, a comparable scoring criterion.

Module Discovery

Module identification utilizes simulated annealing[29] in order to explore relatively high-scoring modules.

For an input graph $G=(V,E)$ the score of each gene is set according to the GE measurement as described above. The algorithm progresses in iterations, and each iteration i has a designated parameter value T_i called temperature. Initially, each gene $v \in V$ is set to a state active/inactive with probability 0.5 independently, virtually creating the subnetwork G_w consisting of the active nodes. In each iteration i a random gene in the network is chosen, its state is toggled, while the temperature T_i is decreased, and the score of the gene set in the new active subnetwork is calculated. If the new gene set score s_i is higher than the current one s_{i-1}, we keep it. Otherwise we un-toggle the gene with probability $e^{(s_i - s_{i-1})/T_i}$ – introducing the stochastic element to the process. This process is repeated for a fixed number of step N when $T_N \approx 0$. Finally, we take from the active gene set the M highest-scoring connected components – the putative modules (M is predefined by the user). The score of each module is calculated by S_A.

Additionally, in order to address bias towards large degree nodes ("hubs")– each time a node with degree higher than a specific threshold is toggled, simultaneously all its neighbors that are not in the highest scoring connected component are removed. The particular threshold is defined by the user. Figure 7 gives an example of a solution provided by the algorithm.

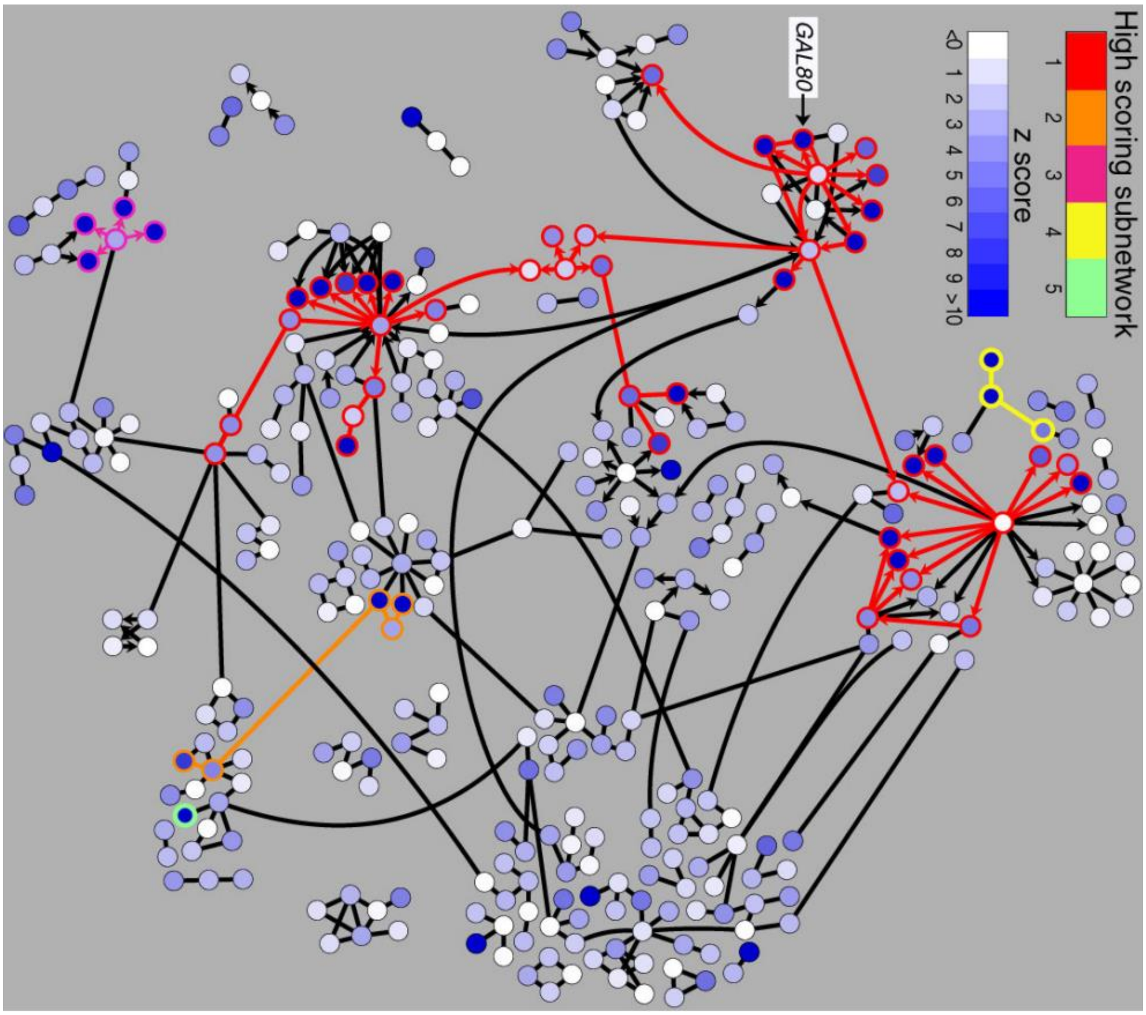


Figure 7:Performance of jActiveModules on a small molecular interaction network. Nodes represent genes, an edge directed from one node to another signifies that the protein encoded by the first gene can influence the transcription of the second by DNA binding (protein→DNA), and an undirected edge between nodes signifies that the corresponding

proteins can physically interact. Z-scores (blue scale) indicate the likelihood of differential expression of each gene in a GAL80 knockout experiment. Z-scores were used to search for active subnetworks using simulated annealing method; the five top-scoring subnets are shown. (Source: Ideker et al.[10]).

4 B ii. NetBox

Netbox[9] was originally developed for analysis of somatic mutations, in order to identify biological subnetworks that are recurrently affected by mutated cancer driver genes.

Netbox works as follows:

1. Mark perturbed nodes: these are the nodes that represent mutated genes.
2. Mark linker nodes. A linker node connects at least two mutated genes, and must be significantly enriched in terms of its perturbed neighbors, that is, each such linker has to have more perturbed neighbors than expected by chance. This is quantified using hypergeometric-test and adjusted for multiple-testing using FDR.
3. Define G', a subgraph of G by filtering out each node that is neither perturbed nor linker.
4. Apply Newman-Girvan algorithm for modularity-detection on G', resulting in a partitioning - or modules - and modularity score of the network.

The Newman-Girvan modularity detection method iteratively removes an edge from the current graph, creating a series of graphs $G' = G_0, G_1, \dots G_k$. The final graph G_k contains no edges. The edge to be removed is chosen by computing the *betweenness-centrality* scores for each edge, defined as its frequency in shortest paths of all gene pairs in the current graph. For each gene pair, a shortest path is calculated and a score of 1 is added to each edge that appears in the path. For gene pairs with multiple shortest paths the score is split evenly among the different paths (e.g. a gene pair with two shortest paths will add 0.5 to the score of an edge for each appearance of the edge in any of the paths). The highest scoring edge is thereafter removed from the graph and the process repeats. In some iterations the process breaks connected components into smaller ones. The graph G_i is assigned a *modularity score* M_i as follows:

$$M = \sum_{s=1}^{N_M} \left[\frac{l_s}{L} - \left(\frac{d_s}{2L} \right)^2 \right]$$

Where N_M is the number of modules (connected components in the current graph), l_s is the number of edges within module s, L is the total number of edges in the network, and d_s is the sum of the degrees of all nodes within module s. Netbox reports the set of modules in the subnetwork G_i that had maximum modularity score.

In order to test NetBox the authors used Glioblastoma's mutation dataset from The Cancer Genomic Atlas (TCGA[30]) - a project comprising multi-OMIC datasets across 33 cancer types - and a literature-curated network they constructed themselves.

The authors validated their results in the following ways:

1. Statistical validation: The authors performed permutations over the network, while retaining its basic structure such as degree distribution etc., and measured the modularity score – creating by that a background distribution. This way they evaluated whether the real modularity score was significantly higher than what is expected by chance. Such significance might indicate that the identified modules have a structural meaning.
2. Biological validation by discovered cancer-driver genes: Out of eight genes identified by TCGA frequency-based approach, seven were contained in the modules detected by Netbox. In addition, many of the genes in the solution are targets of high-level focal amplification or homozygous deletions.
3. Biological validation by overlapping of discovered modules with known ones: the two largest modules are in close agreement to the three critical signaling pathways identified in the original TCGA pathway analysis.
4. Biological validation by GO term enrichment analysis: many modules were enriched with cancer-relevant GO terms.

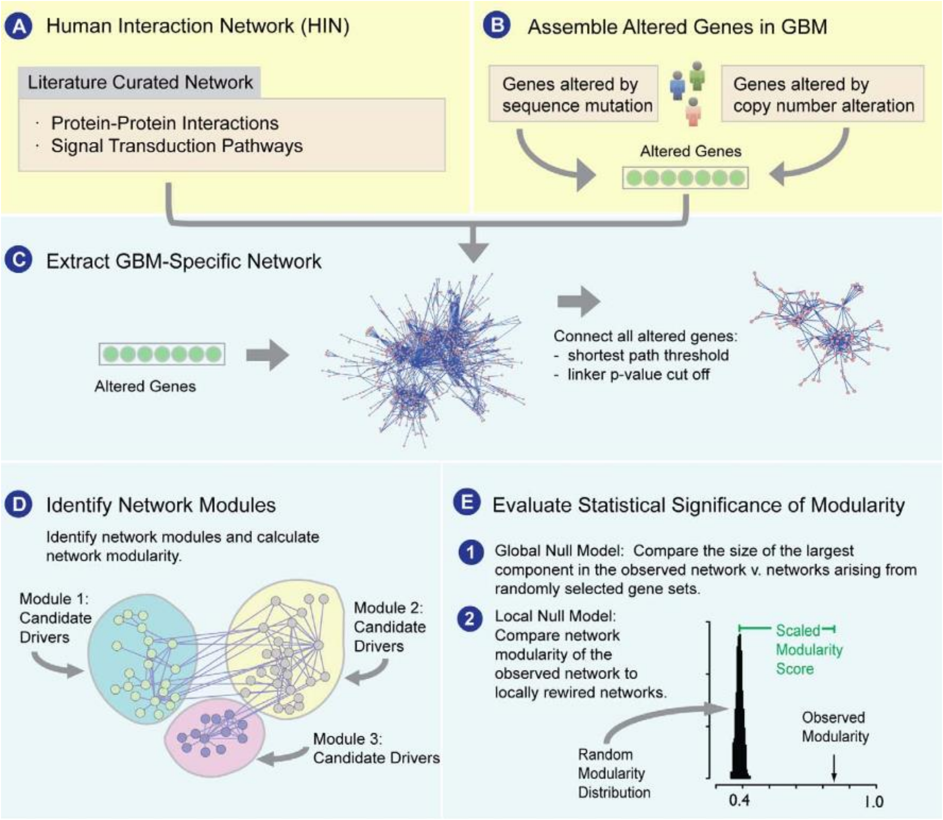

A Human Interaction Network (HIN)
Literature Curated Network
· Protein-Protein Interactions
· Signal Transduction Pathways
B Assemble Altered Genes in GBM
Genes altered by sequence mutation
Genes altered by copy number alteration
Altered Genes
C Extract GBM-Specific Network
Altered Genes
Connect all altered genes:
- shortest path threshold
- linker p-value cut off
D Identify Network Modules
Identify network modules and calculate network modularity.
Module 1: Candidate Drivers
Module 2: Candidate Drivers
Module 3: Candidate Drivers
E Evaluate Statistical Significance of Modularity
1 Global Null Model: Compare the size of the largest component in the observed network v. networks arising from randomly selected gene sets.
2 Local Null Model: Compare network modularity of the observed network to locally rewired networks.
Scaled Modularity Score
Observed Modularity
Random Modularity Distribution
0.4
1.0

Figure 8: Overview of Netbox approach for identifying oncogenic processes and candidate driver genes in GBM. The authors constructed the network from protein-protein interactions and signaling pathways curated from literature (A), and assembling genomic alterations in GBM (B). We then extracted a GBM-specific network of altered genes (C), which was then partitioned into network modules (D). The level of connectivity of the network was assessed by using (E1) a global null model to compare the size of the largest component in the observed network v. networks arising from randomly selected gene sets; and (E2) a local null model to compare network modularity of the observed network to locally rewired networks (Source: Chris Sander et al.[9]).

4 B iii Bionet

The Bionet[11] tool is a combination of a module discovery method with statistical gene and module scoring method.

Scoring system

The scoring system comprises the following steps:

1. Calculate differential expression p-value (using student's t-test) for each gene.

2. Model p-value distribution according to Beta-Uniform Mixture (BUM) model: Assuming no signal exists in our data, p-values should be distributed uniformly. A real signal on the other hand would lead to concentration of p-values in the left side of the distribution histogram (i.e. higher concentration of p-values in the significant region of the histogram compared to the non-significant region). This pattern of distribution can be modeled by BUM, where α and β are the parameters of the beta distribution. β was fixed to 1, and α was estimated by maximizing the likelihood. α variable is thereafter used to calculate the adjusted score for the nodes:

$$s^{FDR}(x) = \log\left(\frac{\alpha x^{\alpha-1}}{\alpha \tau^{\alpha-1}}\right) = (\alpha - 1)(\log(x) - \log\left(\tau(FDR)\right))$$

where x is the p-value and τ is the unadjusted p-value threshold for a given FDR. Finally, let w' be the minimal adjusted gene score. The score of node p is $p(v) = gene\_score(v) - w'$. Additionally, each edge score is set to $-w'$.

Module Discovery

The authors use the Heinz algorithm, which they previously developed[31]. This algorithm aims to find a single module, by solving the prize-collecting Steiner tree problem (PCST). PCST is defined as follows: Given a connected undirected vertex- and edge-weighted graph G=(V,E,c,p) with vertex profits $p: V \rightarrow \mathbb{R}_{\geq 0}$ and edge costs $c: E \rightarrow \mathbb{R}_{\geq 0}$, find a connected subgraph $T = V_T , E_T)$ of $G, V_T \subseteq V, E_T \subseteq E$, that maximizes the profit:

$$p(T) = \sum_{v \in \mathrm{V_T}} p(v) - \sum_{e \in \mathrm{E_T}} c(e)$$

The authors formulated PCST as Integer Linear Programming (ILP) and solved it heuristically.

4 B iv. HotNet2

HotNet2[8] was originally developed for network-based analysis of mutations across cancer types, but since then was put into use for analysis of different omic and GWAS data. It comprises two main concepts - scoring system and module discovery method:

Scoring system

The authors used MutSig[32] in order to score each gene for its mutation load.

Module Discovery

HotNet2 harnesses insulated heat diffusion process in order to identify candidate modules. This process can also be described as a random walk with restart. Denote parameter $0 < \beta < 1$ as the restart coefficient. The process is a walk in the graph, where at each iteration the process resides in one vertex. We start a random walk from some root node. In each step we can either move to a neighbor of the current vertex with probability $1 - \beta$ or jump back to the root with probability β. In the former case, the neighbor is chosen uniformly at random. This process can be described by the transition matrix W:

$$W_{ij} = \begin{cases} \frac{1}{\deg(j)} & \text{if node } i \text{ interacts with node } j, \\ 0 & \text{otherwise.} \end{cases}$$

where $deg(i)$ is the degree of node i.

The Ergodic Theorem guarantees that if the graph is connected such random walk starting from node i reaches stationary distribution described by the vector s_i:

$$\vec{s_i} = \beta(I - (1-\beta)W)^{-1}\vec{e_i}$$

where e_i is a vector with 1 at the i'th element and 0 at the rest.

s_{ij} describes the stationary walk probabilities that random walk starting in node i is at node j. This induces the diffusion matrix F:

$$F = \beta(I - (1-\beta)W)^{-1}$$

Note that F defines an asymmetric relation between nodes: The probability of reaching i from j might differ from the probability of reaching from j to i. Note that F is captured only from topology of the network, regardless of the scores derived from a specific assay.

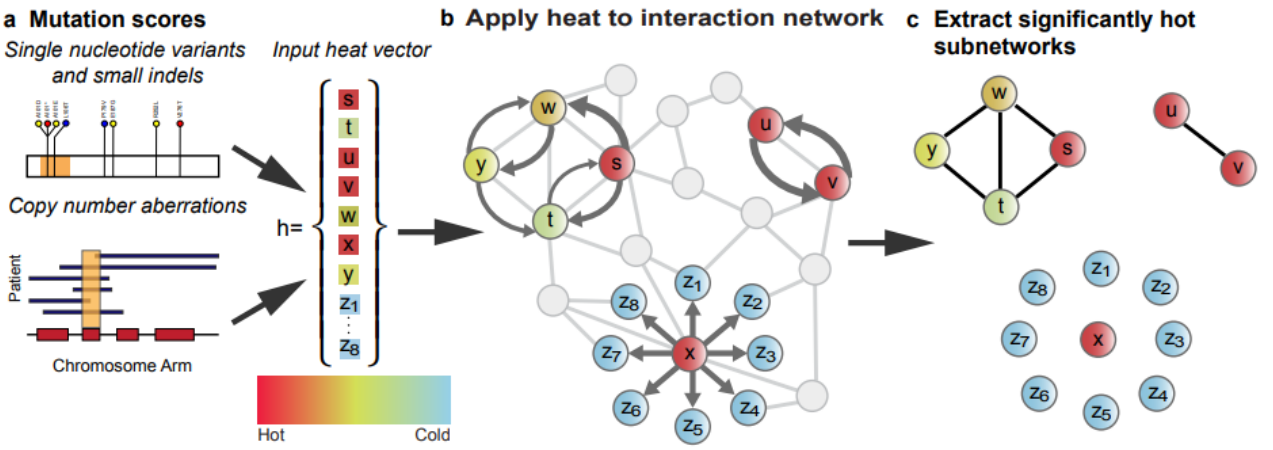
In order to model scores of a specific assay define the scoring vector h where $h(j)$ is the score of gene j, and E the exchanged heat matrix:

$$\overrightarrow{E}(i,j) = F(i,j)\overrightarrow{h(j)}$$

$E(i,j)$ is the amount of heat a specific node absorbs from all the network structure (the heat it retains plus the heat induced by other nodes).

The authors applied a threshold d over the matrix so an edge from node i to j exists only if $E_{ij} \geq d$. Importantly, E is not symmetric (as F is not symmetric) and therefore E induces a directed graph that is different from the original undirected network. Thereafter, strongly connected components in this new graph are identified and reported as modules.

Finally, the significance of the modules was assessed by comparing it to modules of the same size generated by applying the same process to datasets in which gene scores and network edges were randomly permuted, thus computing empirical p-values. These empirical p-values were corrected by BH-FDR. See Figure 9 for an overview.



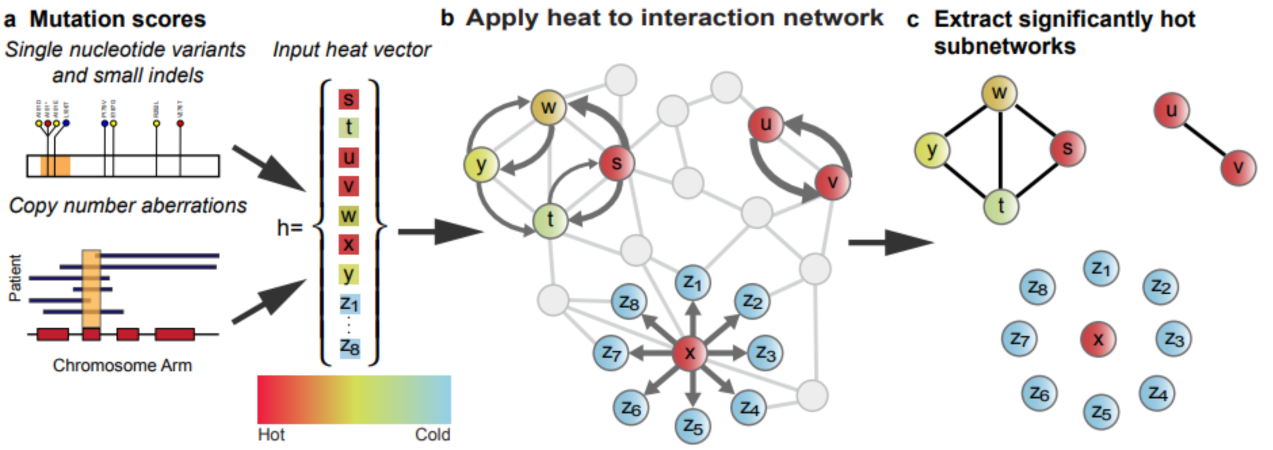


Figure 9: Overview of HotNet2 algorithm and Pan-Cancer analysis. HotNet2 assigns heat to each gene (node) in an interaction network according to a gene score encoding the frequency and/or predicted functional impact of mutations in the gene. This heat spreads to neighboring nodes using an insulated heat diffusion process. At the equilibrium heat distribution, the network is partitioned into subnetworks according to the amount and direction of heat exchange between pairs of nodes. Thus, the partition depends on both the individual genes scores and the local topology of protein interactions. The statistical significance (p-value and FDR) for the resulting subnetworks is computed using the same procedure on random data. In the TCGA Pan-Cancer analysis, gene scores are computed according to single nucleotide variants, small indels, and splice site mutations (from exome sequencing data), copy number aberrations (from SNP array data), and gene expression (from RNA-seq data) (Source: Raphael et al.[8]).

4 B v. KeyPathwayMiner

KeyPathwayMiner[12] was originally developed for gene expression data. The algorithm has two parameters, k and l. It first identifies dysregulated genes in each sample and determines that a gene is perturbed if it is dysregulated in at least l samples. Thereafter, it seeks connected modules of maximum size with at most k non-perturbed genes.

Given an input graph G, the authors create a new graph G' as follows:

1. For each non-perturbed node v_i in G create a corresponding non-perturbed node in G' : u_i
2. Create an edge between each node pair $(u_a, u_b) \in G'$ if and only if there exists a path between v_a to v_b in G that does not pass through any non-perturbed node.

The authors evaluate $U_j = \{u_1, u_2 \dots u_m\}$, a connected component in G', by choosing its corresponding vertices in G, $V_j = \{v_1, v_2 \dots v_m\}$, and assigning a score $S(U_j)$ to U_j which is the number of perturbed nodes that are reachable to any node in V_j through a path that does not use any non-perturbed nodes. The objective function is to find a maximal scoring connected component in G': $\max_{U_j \subseteq G'} S(U_j)$.

To find maximal scoring modules, the authors suggested several heuristic strategies: Greedy algorithm, ant colony optimization or branch and bound. The web implementation offers only the greedy strategy, which works as follows:

For every node u, iteratively construct a set G'_u, starting with $G'_u = \{u\}$. In every iteration i we add to G'_u a node $u_i \in G'$ that is adjacent to G'_u and maximizes the objective function $S(G'_u \cup \{u_i\})$. We stop the iterations when $|G'_u| = k$ or $|G'_u| = |G'|$. Finally, return the maximal scoring G'_u.

Figure 10 gives an example of a module provided by the algorithm.

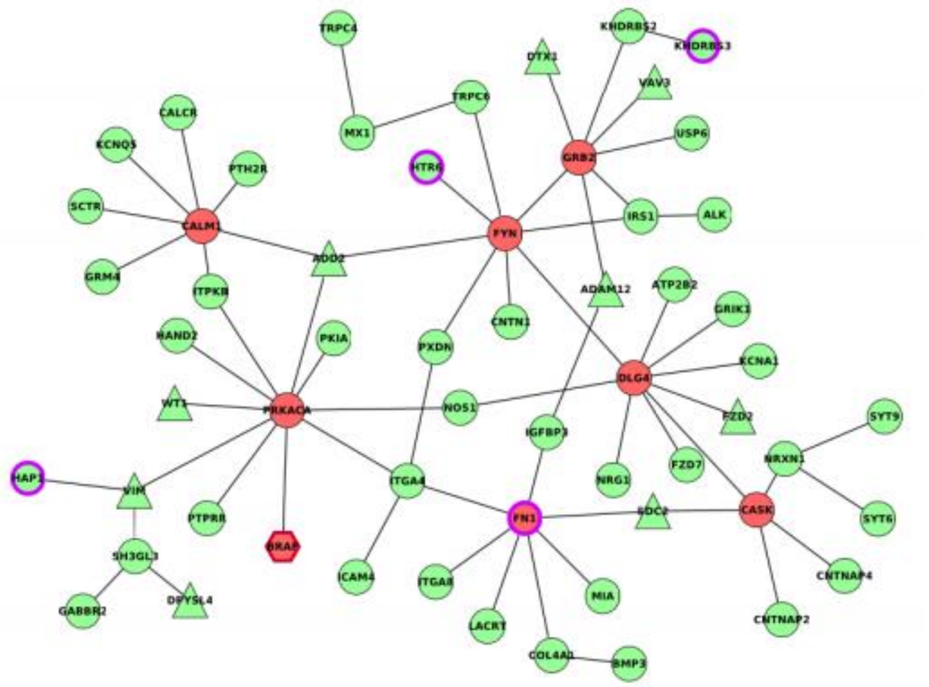


Figure 10: A module discovered by KeyPathwayMiner. The dataset analyzed included 125 colorectal cancer patients whose promoter CpG islands were tested for methylation levels. Dysregulated genes in a patient's sample were defined as those with differential methylation of their promoter's CpG island in comparison to normal samples. Six modules of sizes 56-62 were found when running the greedy algorithm for $k = 8$ and $l = 25$. The largest subnetwork found containing the BRAF gene is shown. Red nodes represent un-perturbed nodes, triangle nodes are hypermethylated genes that also show significant decrease in expression levels, and nodes with purple border are genes with promoters classified as CpG island methylator phenyotype (CIMP), an established factor of colorectal cancer (Source: Alcaraz et al.[12])

5. Methods and Materials

NBMD algorithms can be compared by many technical criteria: running time, memory consumption, number of identified modules, module sizes, etc. More important criteria are those that reflect the biological information the solutions capture. A common evaluation of such algorithms is by the extent they identify expected biological signals. Such evaluation usually relies on “cherry picking” of salient gene sets that are believed to be relevant to the condition studied. This method evidently suffers from bias and not scalable. We therefore turned to fashion a benchmark in which we used GO gene sets and the hypergeometric test in order to systematically evaluate algorithms across several datasets, without manually selecting ground truth gene-sets (e.g., expected GO terms). In the following sections we discuss the main challenges we encountered and the criteria we designed based on the insights we gained.

5 A. The Functional Analysis

The standard functional analysis based on an NBMD algorithm works as follows (Figure 11): We start from a dataset of biological measurement and a biological network. We calculate gene activity scores on the dataset. The vector of scores is provided, along with the biological network, as an input for the algorithm. We then execute the algorithm and get a solution: the identified modules. Last, we perform functional analysis on each module, test functional enrichment of each module against a set of functional gene groups, and report the highly enriched gene groups. This flow has seven main components: (1) the gene activity scores; (2) the gene activity scores' distribution; (3) the NBMD algorithm; (4) the biological network (e.g. STRING[5], DIP[2] etc.); (5) the collection of functionally annotated gene sets (e.g. GO[14], MSigDB[33]); (6) the enrichment analysis method (e.g. Hypergeometric test, GSEA[34] etc.); (7) the criterion for concluding strong biological signals. Below we describe the specific choices we made for each component.

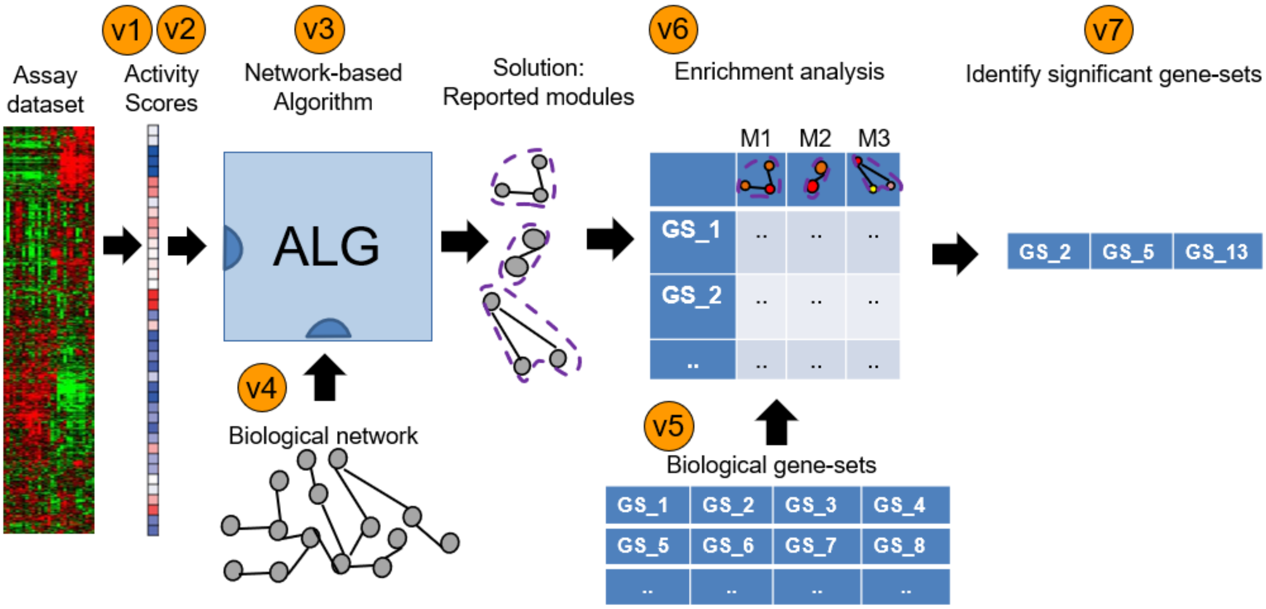


Figure 11: Flow of analysis. The seven components described in the text are marked as orange circles V1-V7.

5 B. Biological Datasets

We included in the benchmark gene expression and GWAS datasets, and computed FDR-corrected q-values per gene as described below. The method by which we derived gene activity scores from q-values differs across algorithms, and is elaborated on Section 5 G. Execution details. The datasets are available in https://github.com/hag007/bnetworks/tree/master/datasets.

5 B i. Gene Expression Datasets

We analyzed seven gene expression datasets that span different physiological processes (Table 2). For each dataset we calculated differential expression p-values using edgeR[35] for RNAseq and student t-test for microarray assays. We computed q-values using Benjamini-Hochberg FDR method[36].

Datasets name (acronym)	access to data	Technology	General description	Reference
TNFa_2	GSE64233	RNA-seq	TNFa which induced immune responses	37
HC12	GSE67478	RNA-seq	hair cell's cochlea and vestibular cells, compared to non hair cell ones (i.e. non-hair cell's cochlea and vestibular cells)	38
SHERA	GSE108693	RNA-seq	Luminal lncRNAs regulation by ERα-controlled enhancers in a ligand-independent manner in breast cancer cells. Comparison was made between ER siRNA to control siRNA	39
SHEZH	GSE109064	RNA-seq	Downregulation of EZH2 Leads to Cellular Senescence with Features of SASP. Comparison between control to 4d samples	40
ERS_1	GSE106847	RNA-seq	ATF6 encodes a transcription factor that is activated during the Unfolded Protein Response to protect cells from ER stress. Comparison was made between ATF-6 pathway activated cell to baseline	41
IEM	---	Microarray	Comparison between 2 different cell types in cochlea: blood cells and mesenchymal cells.	42
ROR_1	GSE74383	RNA-seq	RNA-Seq profiling of estrogen-receptor-positive MCF-7 cell lines with different perturbations of non-canonical WNT signaling. Comparison was made between empty vector transfected cells to ROR2 overexpression construct transfected cells	43

Table 2: Description of the seven GE datasets used in the benchmark.

5 B ii. GWAS Datasets

We used results of five GWAS studies of different traits. The GWAS summary statistics provide a p-value for each SNP, which represents its association with trait. The traits we examined were breast cancer, Crohn's disease, schizophrenia, triglycerides and type 2 diabetes. To derive a p-value per gene, we used PASCAL[18] with flanks of 50kbps around genes. We computed q-values using Benjamini-Hochberg FDR method[36]

5 C. Algorithm

This is the central component in our benchmark. We compared five different algorithms corresponding to six different algorithmic options: (1) jActiveModules[10] with the greedy strategy (denoted jAM_greedy), (2) jActiveModules with simulated-annealing strategy (denoted jAM_sa), (3) Bionet[11], (4) HotNet2[8], (5) NetBox[9], and (6) KeyPathwayMiner with INES GREEDY strategy (denoted KPM)[12] (see Section 4. Selected NBMD Algorithms for more details). We executed each algorithm in a way that it can produce multiple modules in a solution, such that all modules are mutually exclusive, and of size of at least four genes each. Reported modules of 3 genes or less were ignored. Each algorithm's specific execution details in this benchmark are given in Section 5 G. Execution details.

5 D. Biological Network: DIP

The running time of NBMD algorithms can grow very quickly with the network's size. In order to allow systematic executions of numerous algorithm and dataset combinations in a reasonable time, we had to choose a relatively small network. Following a recent benchmark[44], we chose DIP[2] as a proper biological network for our analysis: DIP got the best normalized performance score in that study, and is a relatively small network (~3000 nodes and ~5000 edges). Moreover, as DIP is a global network, namely, not designed to describe any specific biological process, we could use it for various biological datasets regardless of the specific biological conditions examined.

5 E. Functional Enrichment Analysis: GO

We performed enrichment analysis over GO terms. We used GO terms' resource files from Gene-Ontology Consortium and GOATOOLS[45] in order to import GO terms, computed their hypergeometric p-values (i.e. enrichment scores) and corrected for multiple tests using Benjamini-Hochberg FDR method[36]. We considered only Biological-Process terms that contain between 5 to 500 genes).

5 F. REVIGO Implementation

The REVIGO original implementation (see Section 3 D ii. REVIGO) is not applicable for solution with more than 350 enriched terms. Therefore, in our study, we re-implemented it using Resnik similarity implementation of Fastsemsim (https://pypi.org/project/fastsemsim/). We also added some variants to the original implementation and determined some hyper-parameter values:

1. We skipped the omission of broadly interpretable terms. Instead, we considered only terms with 5-500 genes.
2. We extended the usage of the GO hierarchical structure in filtering similar terms: When two terms had similarity above the threshold and one was an ancestor of the other, we removed the former. This modification gives more importance to the hierarchy in determining the non-redundant set of GO terms.
3. The threshold of the enrichment score difference was set to be 1.

5 G. Execution details

The NBDM algorithms that we tested differ in preprocessing, input and output. We describe the specific execution details for each algorithm below.

5 G i. jActiveModules

jActiveModules was written as a plugin for Cytoscape[46], a user interface tool for network analysis. We modified the codebase of jActiveModules so we could run it independently of Cytoscape. jActiveModules expects a list of genes and their p-values as the gene activity scores. We increased the default number of requested modules in order to retrieve more modules and also required that reported modules are mutually exclusive. Notably, the algorithm usually produced no more than 10 modules with more than 3 genes.

5 G ii. NetBox

We modify NetBox codebase so we can choose the networks it uses. In addition, NetBox gets as an input a list of mutated genes, that is, binary gene activity scores. We used the genes' q-values and set the gene score to 1 if its q-value was < 0.05, and 0 otherwise.

5 G iii. Bionet

Bionet is designed to retrieve only one module. In order to retrieve multiple mutually exclusive modules we executed Bionet iteratively, removing the genes in the identified module in each iteration. We stopped these iterations after retrieving modules smaller than four genes five times in a row.

5 G iv. HotNet2

HotNet2 expects gene activity scores that are calculated by mutation p-values (e.g., using MutSig). We transformed the q-values calculated from our datasets into $-log10(q\_value)$ scale and used them as the input activity scores. We took all the reported modules, ignoring their scores reported by HotNet2.

5 G v. KeyPathwayMiner

We used the version of KPM with the greedy strategy. It expects binary gene activity scores: 1 marks a gene as perturbed and 0 otherwise. We used the genes' q-values and scored a gene with 1 if its q-value was < 0.05, and 0 otherwise. As the reported modules considerably overlap each other, we executed the algorithm iteratively, removing in each iteration the genes in the identified module.

6. Results

Our benchmark analysis compared six algorithms (Table 1) - jActiveModules[1] in two strategies: greedy and simulated annealing (abbreviated JAM_greedy and jAM_sa, respectively), bionet[2], hotnet2[3], NetBox[4], and KeyPathwayMiner[5] (abbreviated KPM). Each algorithm was tested on seven gene expression (GE) datasets (Figure 12Table 2) and five GWAS datasets (**Error! Reference source not found.**).

General description of the solutions, such as average module size, number of modules, is provided in **Error! Reference source not found.** and **Error! Reference source not found.** for GE and GWAS datasets, respectively.

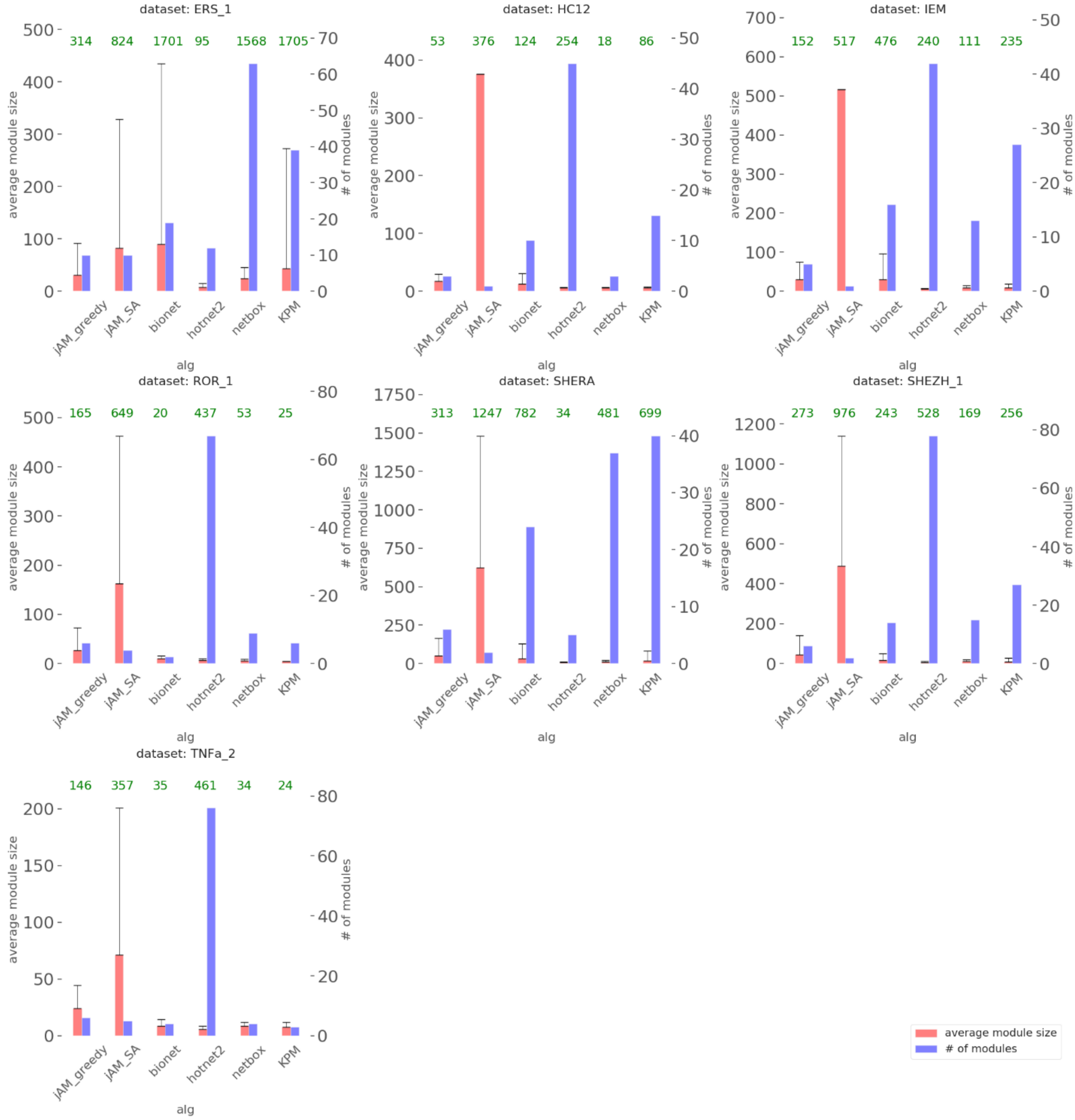


Figure 12: Technical description of the solutions over GE datasets. Green numbers are the total number of genes in the solution.

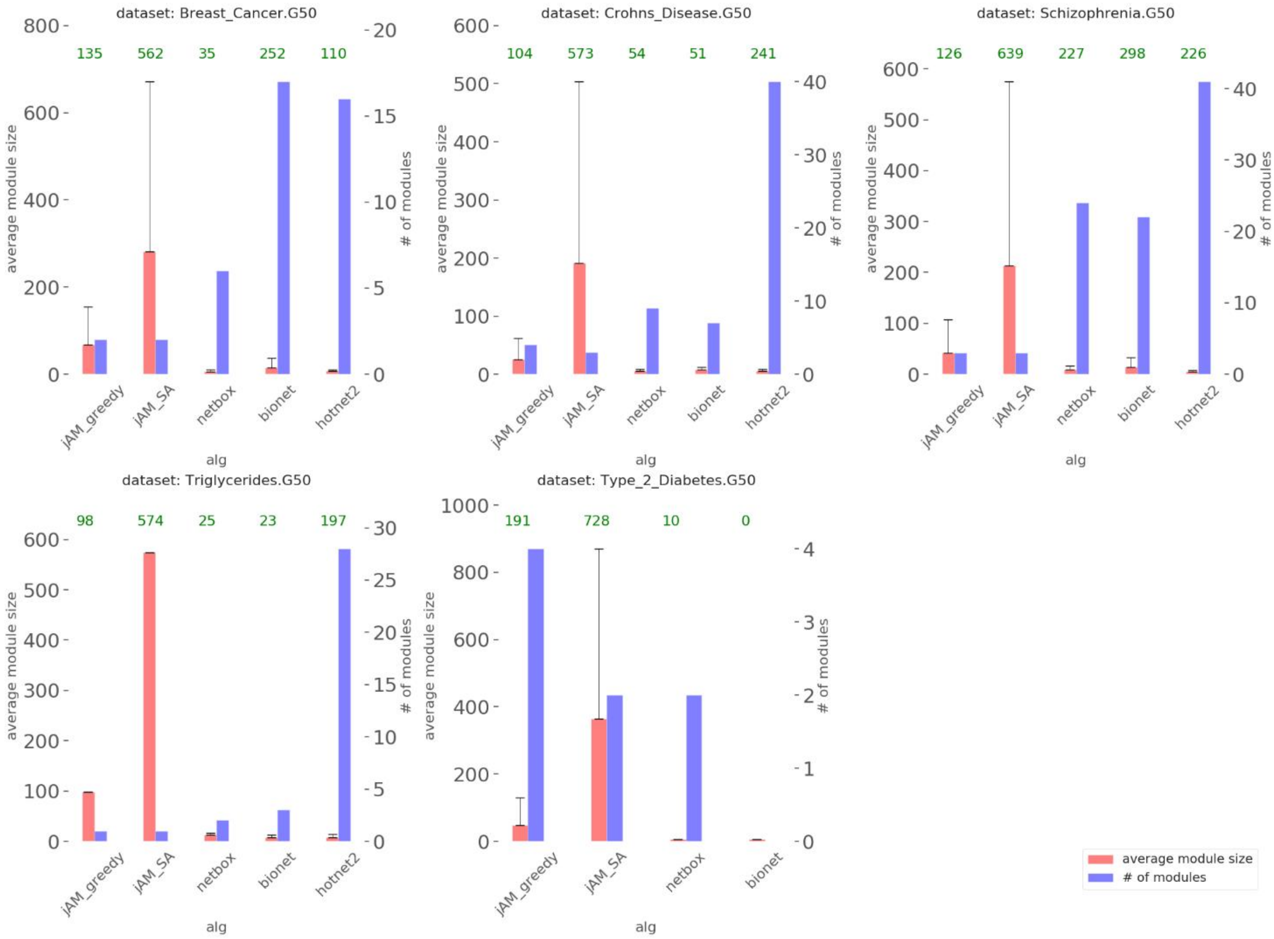


Figure 13: Technical description of the solutions over GWAS datasets. We excluded KeyPathwayMiner, as its solutions were empty. Green numbers are the total number of genes in the solution.

6 A. NBMD Algorithms Suffer from High Rate of Recurrent False Positive GO Terms

NBMD algorithms differ in many ways: The number of modules they produce, the number of genes per module, and the number of enriched terms per module. When we started the benchmark, we observed that some algorithms tended to consistently report more enriched terms with higher enrichment scores. In order to evaluate whether the reported terms are indeed biologically meaningful, we chose to use randomization: For each algorithm, we randomly permuted gene scores, reran the algorithm, and compared the results on the true and permuted input. We observed that in many cases the distribution was not as different as one would expect (Figure 14. Full results in supplementary Tables Table S8Table S9). In fact, some algorithms reported extremely significantly enriched terms over permuted data. We also found some disturbing overlap between the enriched terms detected in the original and the permuted datasets (Figure 14). These findings imply that in some cases, part of the reported signal was not due to the specific biological assay, but might arise from some bias. This bias could stem from the specific biological network, algorithm and score distribution. In addition, some reported term overlap also stems from dependencies between gene sets for which we compute enrichment scores. Of course, the overall bias was due to a combination of these factors.

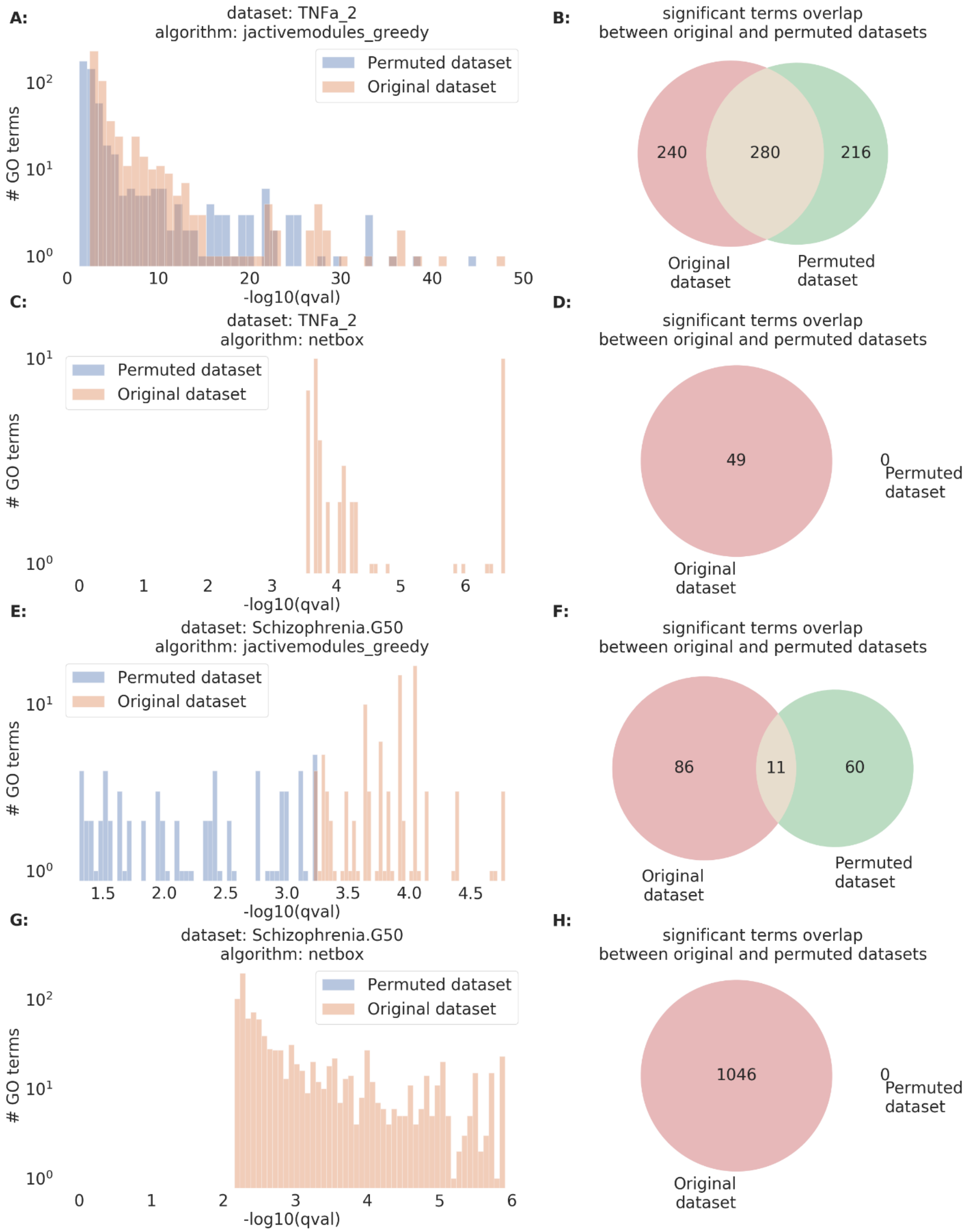


Figure 14: Comparison of GO enrichment scores obtained on original and permuted datasets. Left column: Histograms of the enrichment scores. Right: Venn diagrams of the overlap between reported terms (after FDR correction). (A) (B) Results on the TNFa gene expression dataset using jActiveModules with the greedy strategy. (C) (D) Results on the TNFa gene expression dataset using NetBox. (E) (F) Results on the Schizophrenia GWAS dataset using jActiveModules with the greedy strategy. (G) (H) Results on the Schizophrenia GWAS dataset using NetBox. A higher fraction of the terms reported by jActiveModules is also reported when the algorithm is applied on permuted data.

6 B. Solution: Empirical Cleaning of NBMD Solutions

We chose to deal with the problems that we identified in the reported terms by empirical correction. We developed a method that we call the EMpirical Pipeline (EMP) for identifying and filtering out terms that also show up frequently when analyzing permuted data. For each dataset, algorithm and GO term, the method computes a background distribution of the term's enrichment scores and uses it to filter terms with seemingly high enrichment scores obtained on the original dataset that are also frequently reported in the permuted data.

Specifically, EMP works as follows: Given an algorithm and an assay dataset as an input, it permutes the genes in the dataset, executes the algorithm and performs enrichment analysis, yielding an enrichment score for each GO term (Figure 15-A). The process is repeated many times (typically, in our analysis, 5000) and generates a background distribution per GO term (Figure 15-B). Denote the CDF obtained for term t by F_t. It then executes the algorithm and the enrichment analysis once more using the real (i.e. non-permuted) dataset (Figure 15-C). The empirical significance of GO term t with enrichment score s is $e(t) = 1 - F_t(s)$ (Figure 15-C, Figure 15-D). EMP reports only terms t that passed both the HG test (q-value ≤ 0.05) and the empirical test at $e(t) \leq 0.05$. We call such terms *empirically validated terms* (EV-terms).

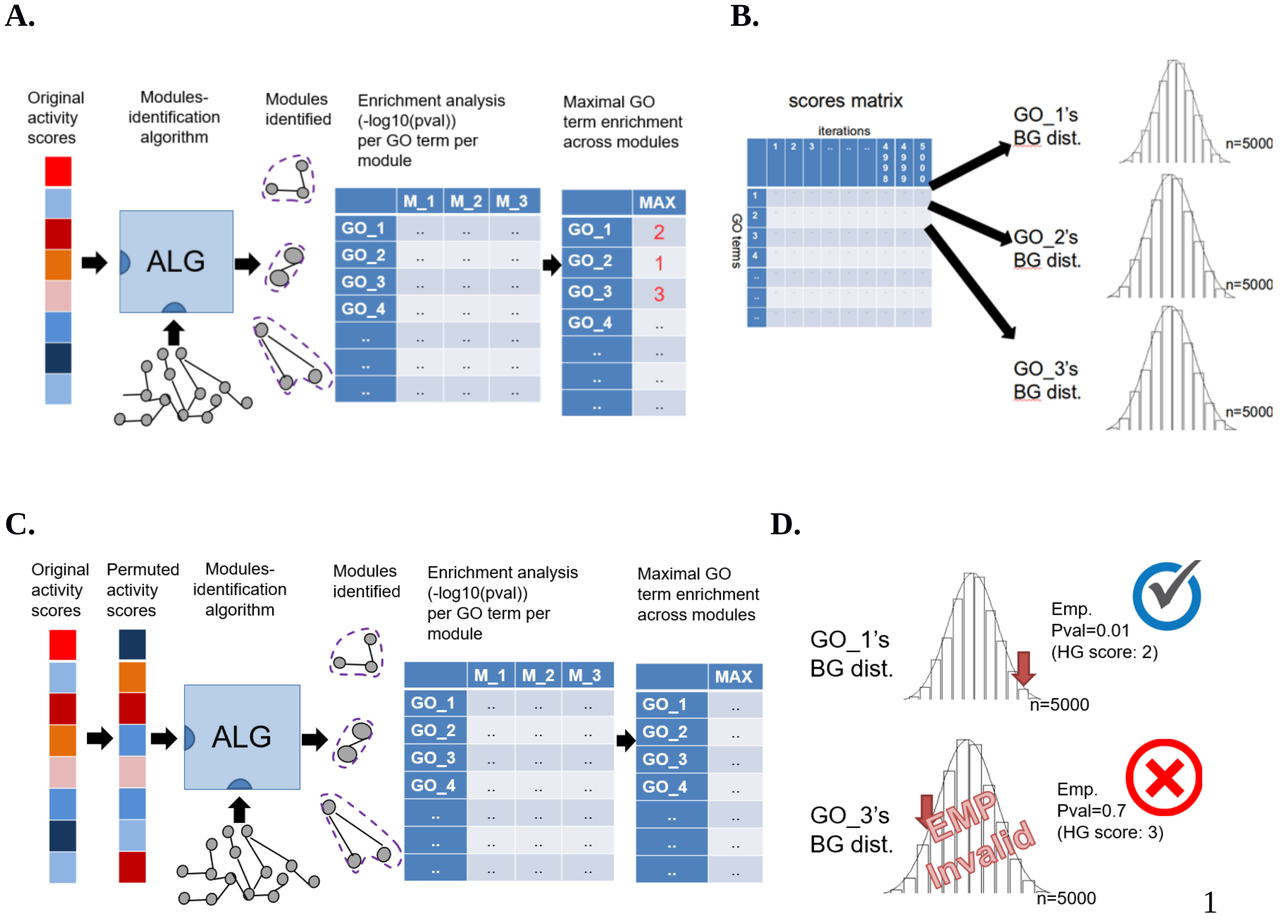


Figure 15: Overview of the EMpirical Pipeline steps. (A) EMP uses the NBMD algorithm and the enrichment analysis pipeline on many instances of permuted activity scores in order to calculate a background distribution of enrichment scores per GO term. (B) A background distribution is produced per GO term. (C) The algorithm is applied on the original un-permuted activity scores, in order to calculate the real enrichment scores. (D) For each GO term, EMP places the real enrichment scores on its corresponding empirical distribution, yielding an empirical p-value, and reports only terms that passed both empirical and the hypergeometric tests' significance threshold. In this example GO_3 passed the HG test. However, it does not pass the empirical test and thus labelled as "EMP invalid"

We then compared the score histograms of the terms that passed the HG test and the EV-terms (See Figure 15. Full results in Figures Figure S4Figure S5). Different algorithms show very different empirical validation rate, which is highly affected by the dataset too.

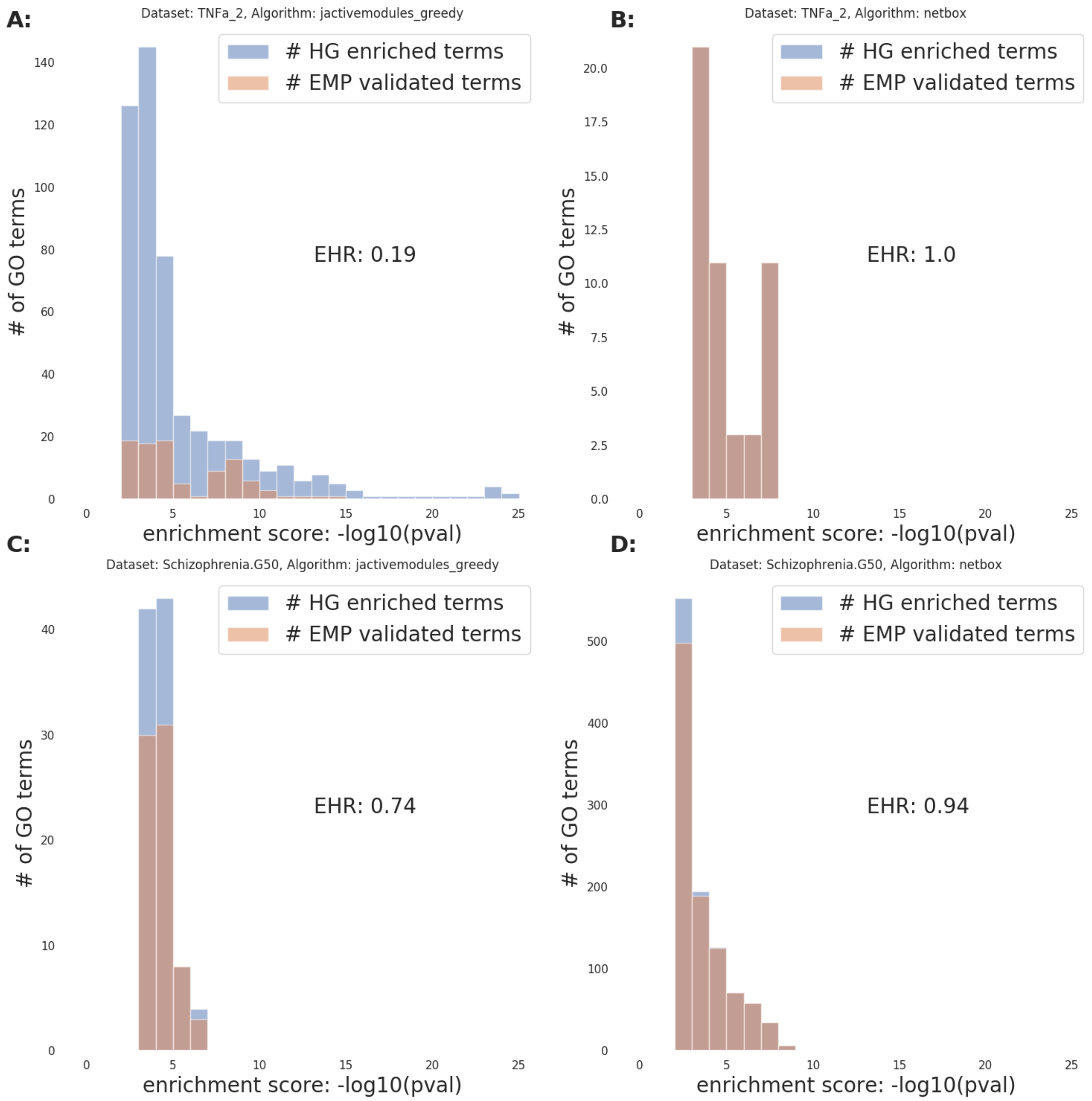


Figure 15: GO term enrichment score distributions for HG-enriched and EV terms. (A) dataset – TNFa_2 (gene expression), algorithm – jActiveModules with greedy strategy. (B) dataset – TNFa_2 (gene expression), algorithm – NetBox. (C) dataset – Schizophrenia (GWAS), algorithm – jActiveModules with greedy strategy. (D) dataset – Schizophrenia (GWAS), algorithm – NetBox. The EHR value is reported above the title in each case. The EHR metric is as defined in Section 7 C i (1). Empirical to Hypergeometric Ratio (EHR).

6 C. Criteria for Evaluating NBMD Solutions

In order to compare the six benchmarked algorithms in light of our new insights, we designed several criteria that measure an algorithm performance. We consider two evaluation perspectives: (1) ignoring the biological information of the terms and measuring solution statistically, i.e., quantitative criteria. (2) Taking into account the biological information each term captures, i.e., qualitative criteria. The EMP process removes certain GO terms from the union of of all terms reported in all the solution's modules. Nevertheless, the identity of the modules comprising the solution is important. Therefore, we evaluated solutions both on the solution level (i.e., in a module agnostic manner; considering union of all modules) and on the module level, i.e., in a module-aware fashion

Our criteria are described in the following subsections. A breakdown of the criteria is in Figure 16:

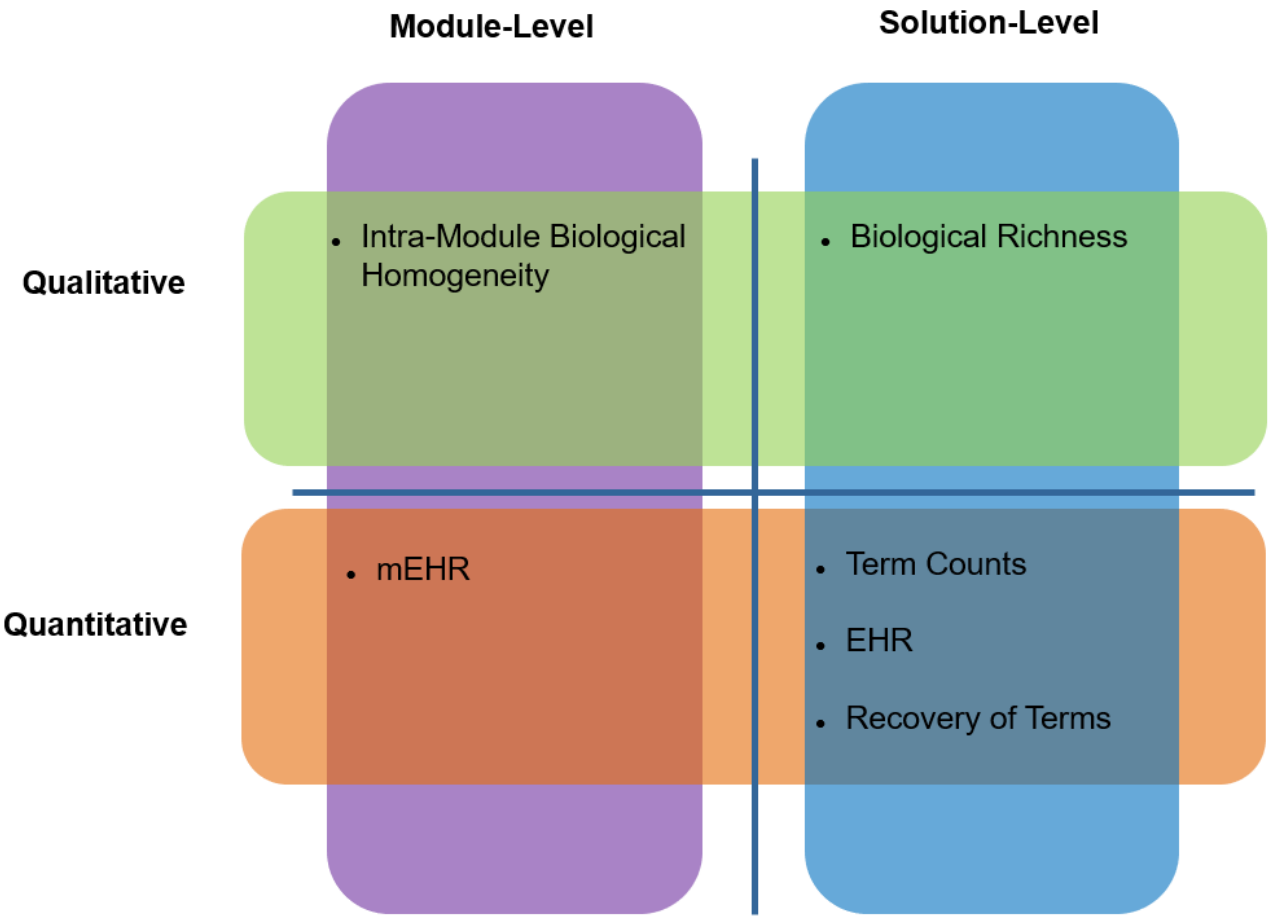


Figure 16: The evaluation criteria that we used in our benchmark analysis and their properties.

6 C i. Solution-Level Criteria

A solution is composed of the modules detected by the algorithm when applied to a dataset. For a specific solution, after applying EMP, we have its list of reported GO terms that passed the HG enrichment test, their enrichment scores and the EV terms. The criteria in the following subsections examine the relations between the reported terms and the EV-terms (See Figure 16).

(1). Empirical to Hypergeometric Ratio (EHR)

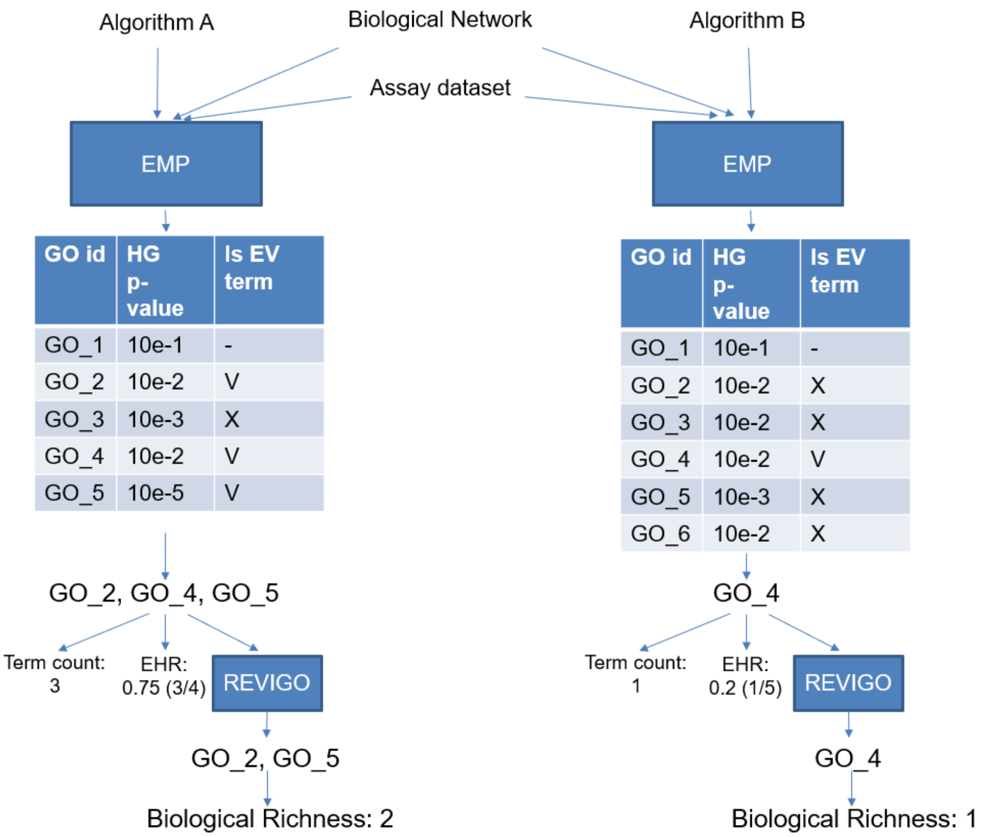
We define *Empirical-to-Hypergeometric Ratio* (EHR) as the ratio between the number of reported terms and EV-terms (Figure 17). EHR summarizes the tendency of an algorithm to over-report terms, with ratio of 1 being best possible and lower values are less desirable. It reflects the precision (true positive rate) of a solution. If an algorithm has EHR values near 1 across many datasets, this suggest that we can trust most terms produced by the algorithm on a new dataset even without running EMP.

(2). Term Count

This criterion counts the number of EV-terms obtained by the algorithm.

(3). Biological Richness

This criterion quantifies the biological information captured collectively by the EV-terms. When using enrichment scores of GO terms, it is important to keep in mind that there is often high redundancy in their biological meaning: multiple high scoring gene sets may stem from nearby terms in the hierarchical structure of GO, common genes that different sets share, and more. In order to evaluate the diversity of the signals that stem from the EV-terms, we produced the similarity matrix using Resnik similarity score[27] between every two EV-terms. We then applied REVIGO[28] to filter out redundant terms, and defined the *biological richness score* as the number of non-redundant terms in a solution. The process was repeated when using similarity cutoffs 1 to 4 in REVIGO. We also compared the median number of terms in the resulting non-redundant gene set list across all datasets. This criterion implies how biologically-diverse the resulted EV-terms are. See Figure 17



GO id	HG p-value	Is EV term
GO_1	10e-1	-
GO_2	10e-2	V
GO_3	10e-3	X
GO_4	10e-2	V
GO_5	10e-5	V

GO id	HG p-value	Is EV term
GO_1	10e-1	-
GO_2	10e-2	X
GO_3	10e-2	X
GO_4	10e-2	V
GO_5	10e-3	X
GO_6	10e-2	X

Figure 17: Term count, EHR and biological richness. In our evaluation, for each algorithm, we average the scores of each criterion across datasets.

(4). Solution Robustness

This criterion evaluates the robustness of a solution to incomplete gene activity data. It compares the EV-terms obtained on the original full dataset with those obtained on randomly subsampled datasets, where non-sampled gene levels are treated as missing. We repeated this procedure for subsampling fractions 0.6, 0.7, 0.8, and 0.9, iterating each fraction 100 times. Using the EV terms of the full dataset as a “gold-standard”, we then computed average precision, recall and F1 scores across these iterations. Another perspective is provided by the examination of the frequency by which terms come up in the subsampled datasets: higher frequency for a specific EV-term implies higher robustness. The most natural way to measure this robustness is using PR-AUC, in which terms are ranked according to their frequency across iterations. As an algorithm could often result in many empty solutions and only few accurate ones – and thus yield a high but misleading PR-AUC score, we also report the fraction of non-empty solutions as an indicator for the “sparseness” of the solutions obtained by an algorithm. See Figure 18.

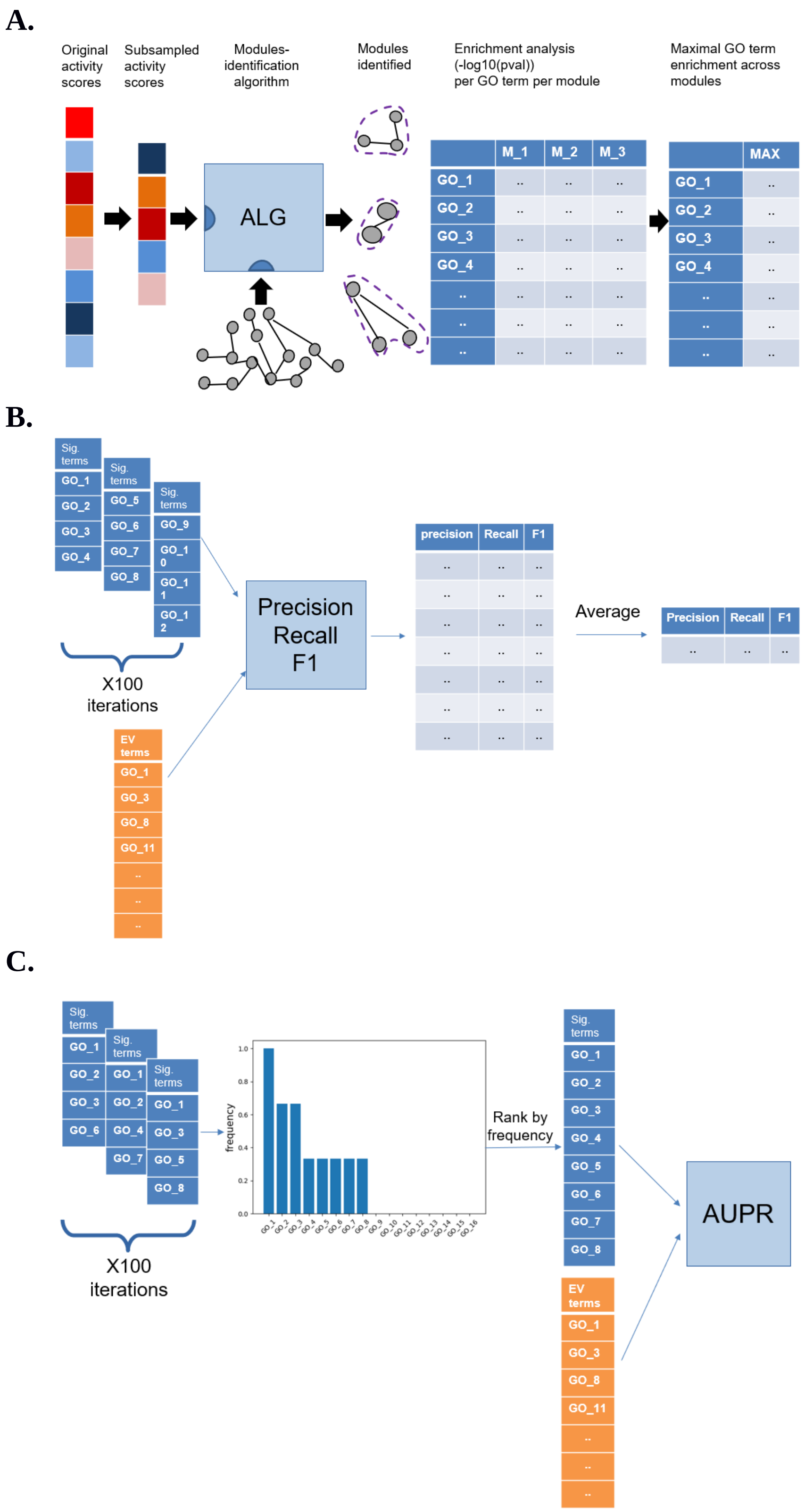


Figure 18: Computing robustness criteria. (A) We subsample datasets and run the functional analysis flow over them. (B) For each GO term list obtained from a subsampled dataset (blue list) we calculate precision recall and F1 against the "gold-standard" EV terms (orange list) obtained from the full dataset, and compute average over 100 random down-sampling runs. (C) We rank GO term by their frequency in the subsampled datasets and compute AUPR. These procedures are repeated for each subsampling fraction.

6 C ii. Module-Level Criteria

All the criteria above evaluate the solution monolithically, while ignoring the quality of individual modules. Evaluation at the solution level is more natural in our EMP procedure, as it examines directly EV-terms, which are computed for a solution, and not per module. However, it is also important to evaluate modules individually: different modules may capture different biological signals with varied quality. The criteria in the following subsections examine a module's reported terms with respect to their solution’s EV-terms.

(1). Module-Level EHR (mEHR)

This criterion calculates a single module’s EHR. As mentioned above (Section 6 B. Solution: Empirical Cleaning of NBMD), we cannot create background distributions at the module level. We therefore calculate module-level EHR, abbreviated *mEHR*, by computing the fraction of terms that are reported in the module and are also included in the EV-terms (Figure 19A). We rank solution modules by their mEHR score, take the *k* top-ranked modules and aggregate their scores. We consider *k* values of 1-20. This criterion enables us to understand the signal-to-noise variability across modules and refines a solution by indicating its “cleanest” modules.

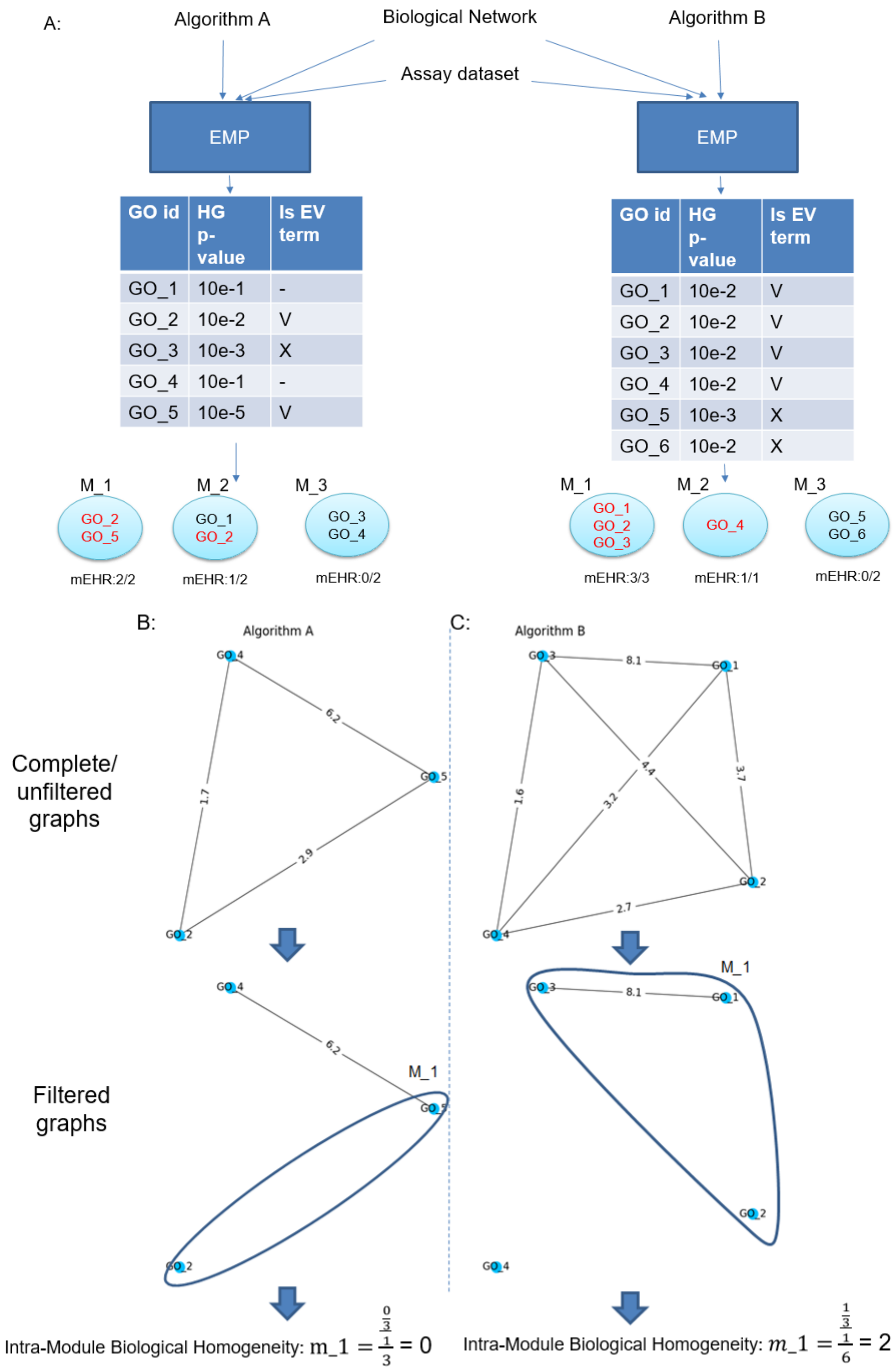
(2). Intra-Module Biological Homogeneity

In Section 7 C I (3). Biological Richness we described a criterion for the biological richness of a solution. Ideally, we prefer solutions that have high richness but where each module shows high homogeneity: A typical dataset reflects an experiment wherein several biological processes are reflected

. A good solution would distinguish these processes as distinct modules, where in each module one or few related processes are identified and reported. This is in contrast, for example, to providing one large module representing all the processes. To capture this notion we build a graph for the solution. Each vertex represents an EV-term, and two vertices are connected by an edge whose weight is their Resnik similarity score. Edges of weight below a cutoff are removed. We calculate *intra-module homogeneity* as the relative module density: the fraction of the edges inside of it compared to fraction of edges in the whole graph.

$$:\frac{\left(\frac{\#\ of\ edges\ in\ module}{\#\ of\ edges\ in\ a\ complete\ module\ of\ the\ size}\right)}{\left(\frac{\#\ of\ edges\ in\ graph}{\#\ of\ edges\ in\ a\ complete\ graph\ of\ the\ same\ size}\right)}$$

Intra-module homogeneity score for a solution is then calculated by averaging its module scores (Figure 19B). We repeat this test for a range of cutoffs – from 1 to 4. This criterion goes one level deeper than solution-level richness, providing complementary view on top of the general biological information in a solution, by measuring how topologically coherent it is.



GO id	HG p-value	Is EV term
GO_1	10e-1	-
GO_2	10e-2	V
GO_3	10e-3	X
GO_4	10e-1	-
GO_5	10e-5	V

GO id	HG p-value	Is EV term
GO_1	10e-2	V
GO_2	10e-2	V
GO_3	10e-2	V
GO_4	10e-2	V
GO_5	10e-3	X
GO_6	10e-2	X

Figure 19: Computing module-level criteria. (A) Enriched GO terms in each module are examined for being EV terms (marked in red) and mEHR is computed. (B),(C) Similarity graph is built using Resnik similarity scores between GO terms, a certain cutoff is applied (5 here) for filtering low scoring edges, and intra-module homogeneity score is calculated.

6 D. Benchmark results

We summarized performance on GE and GWAS datasets separately. The evaluation was based on the criteria described above.

6 D i. Solution-level results

The results for term count and EHR are presented in Figure 20. In both GE and GWAS datasets, NetBox obtained the highest mean scores. Although it has outlier scores in some datasets, it still preserved its top rank when considering the median score, which is less sensitive for outliers.

As HotNet2 yielded poor results in both GE and GWAS datasets, we omitted it from subsequent evaluations. As KPM yielded empty solution in all GWAS datasets, we omitted it from evaluations on the GWAS datasets.

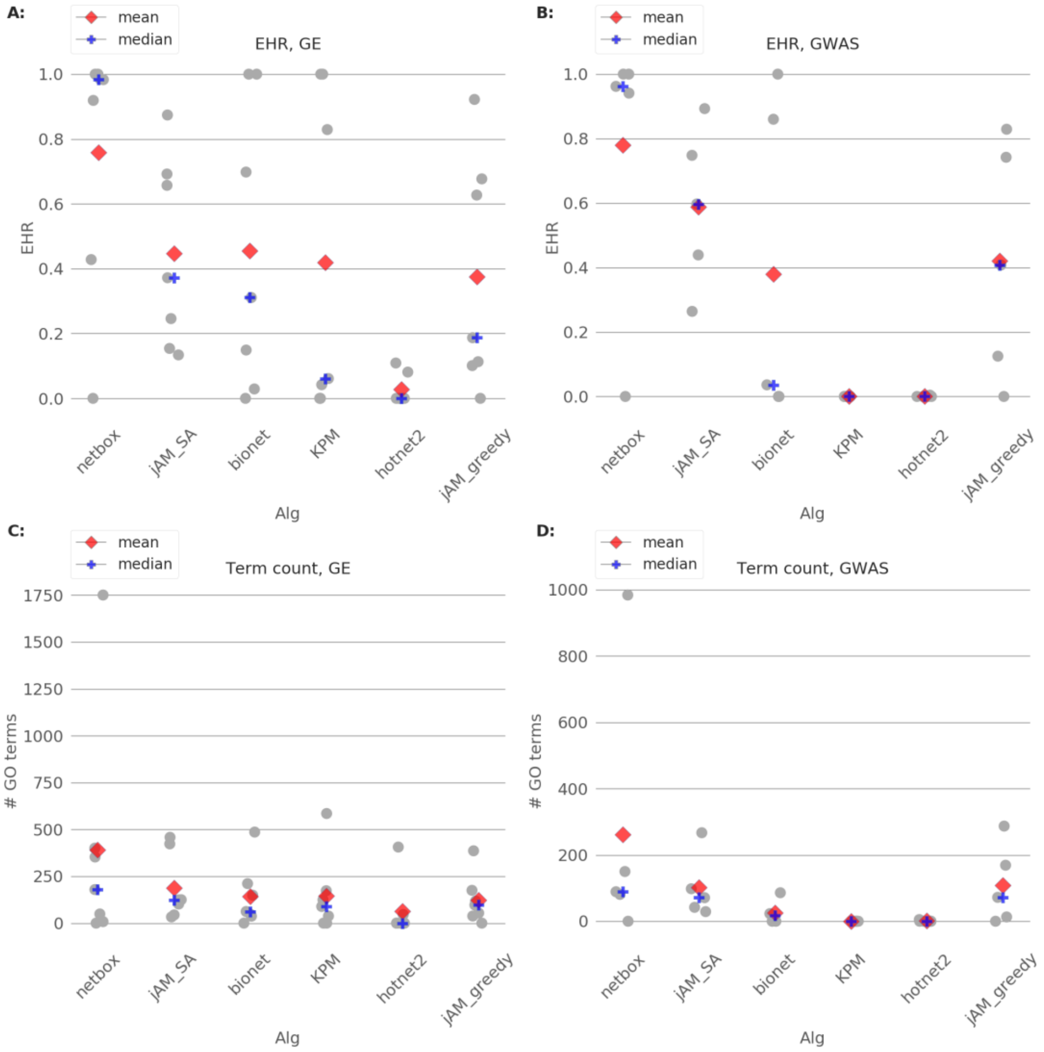


Figure 20: Dot-plot summary for term counts and EHR criteria. (A) EHR for GE. (B) EHR for GWAS. (C) Term count or GE (D) Term counts for GWAS. The dots indicate results for each dataset.

Figure 21 summarizes the results for the biological richness criterion. Here too, NetBox got the highest median score for all similarity cutoffs, implying its solution captures more diverse spectrum of biological processes:

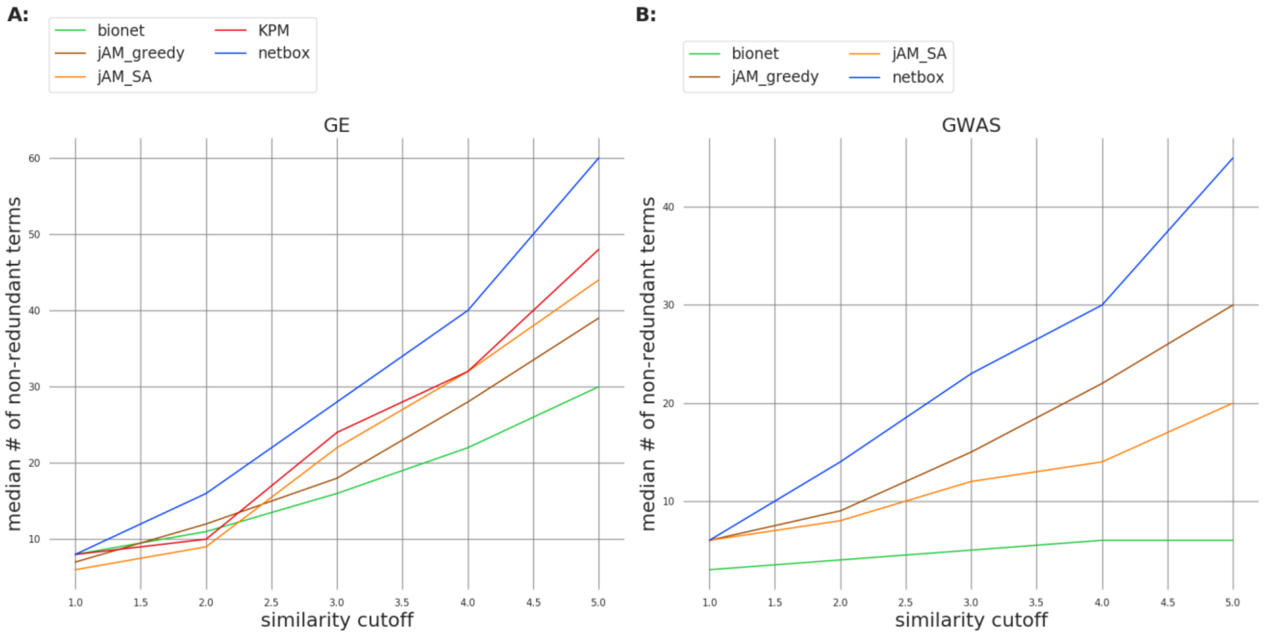


Figure 21: Richness score. The plots show the median number of non-redundant terms (richness score) as a function of Resnik similarity cutoff used. A: GE. B: GWAS datasets.

In order to get a global view of these criteria we presented each solution as a dot in a scatter plot where the axes are richness and EHR scores. Better solutions are located in the upper-right part of the plot. On both GE and GWAS plots, NetBox solutions tend to achieve good results.

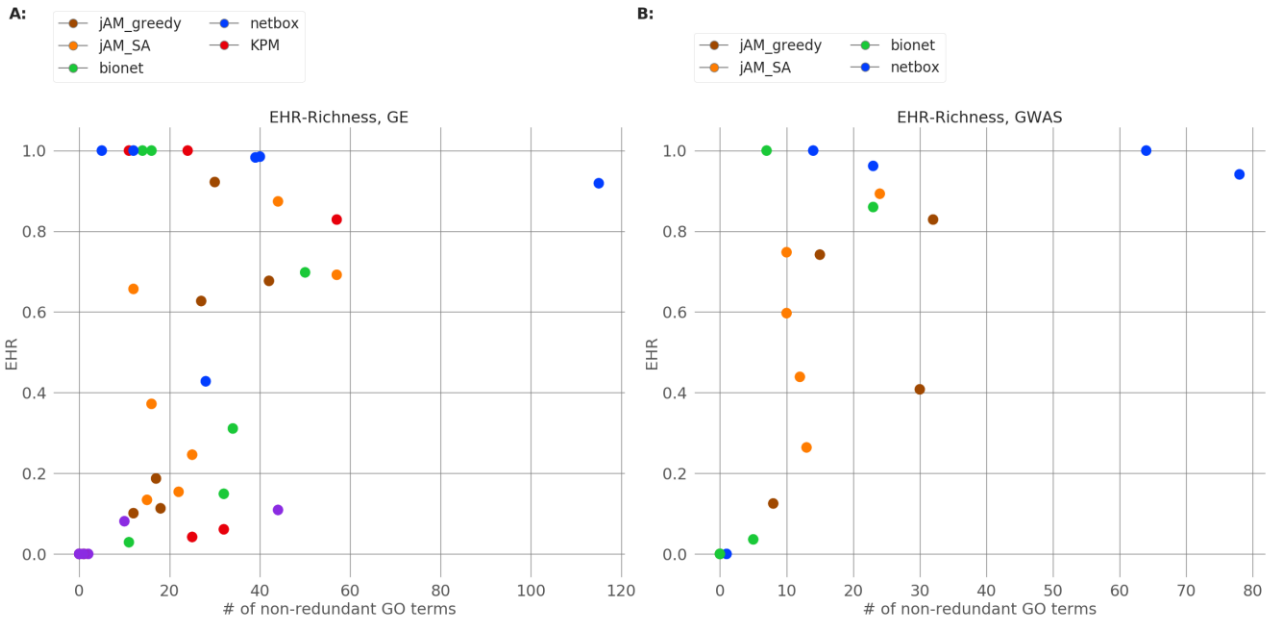


Figure 22: EHR v richness. The figures show the richness (x-axis) and EHR (y-axis) values for each combination of dataset and algorithm. The richness results are for cutoff 3. A: GE. B: GWAS datasets.

Figure 23 shows the robustness of the algorithms. On the GE datasets, NetBox had highest average F1 and PR-AUC scores across all subsampling fractions. Importantly, NetBox also exhibited high rate of non-empty solutions (>0.8). On the GWAS datasets, Bionet showed the highest score in both measurements. NetBox had the poorest AUPR in the GWAS datasets but was second best in F1. Interestingly, in the GWAS analysis we did not observe an improvement trend with growing subsampling fraction, possibly pointing to the high polygenetic nature of complex traits, where each of the numerous affected genes is associated with only very subtle signal.

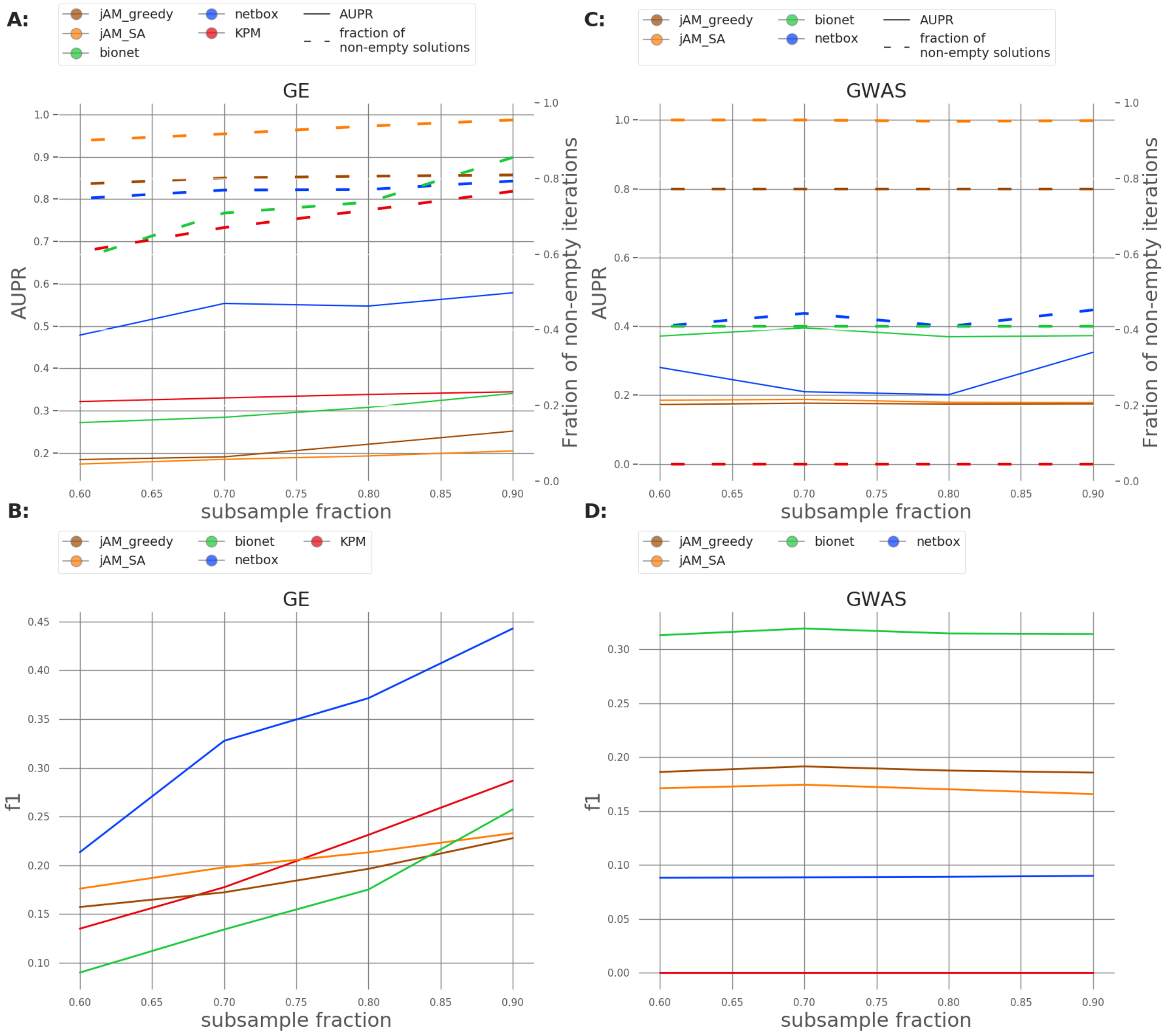
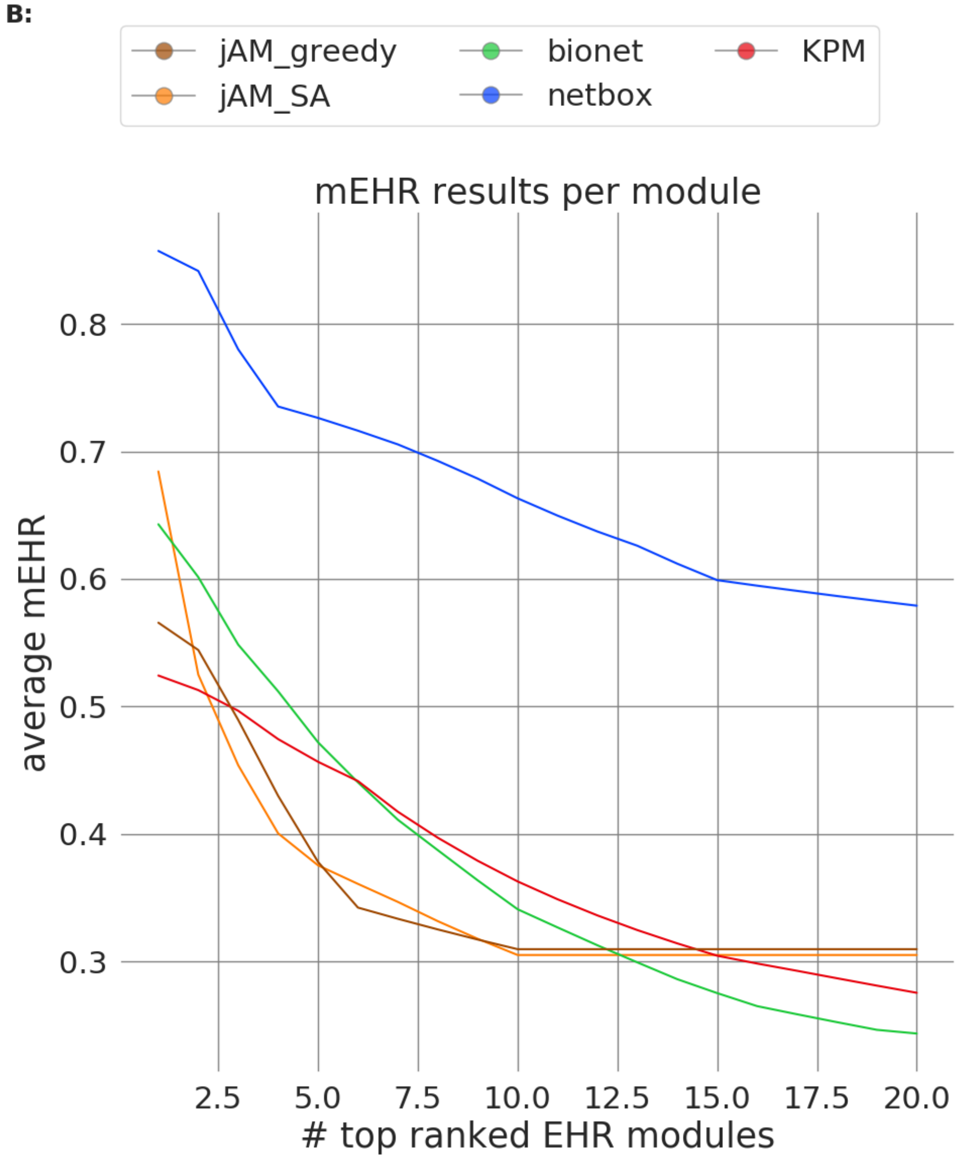


Figure 23: Recovery criteria. PR-AUC score as a function of subsample fraction for A : gene expression and B : GWAS datasets, and average precision, recall and F1 scores as a function of subsample fraction for C - gene expression and D - GWAS datasets. Each subsampling fraction is sampled 100 times

6 D ii. Module-level results

We evaluated the algorithms by mEHR and intra-module biological homogeneity. Figure 24 and Figure 25 summarize the performance in terms of mEHR for the GE datasets. Figures Figure 26 and Figure 27 show the results for the GWAS datasets. Specifically, we calculated mEHR per module and calculated the average mEHR per algorithm for the top-ranked modules. For any number of top-ranked modules, NetBox got the highest scores in both measurements in both GE and GWAS datasets



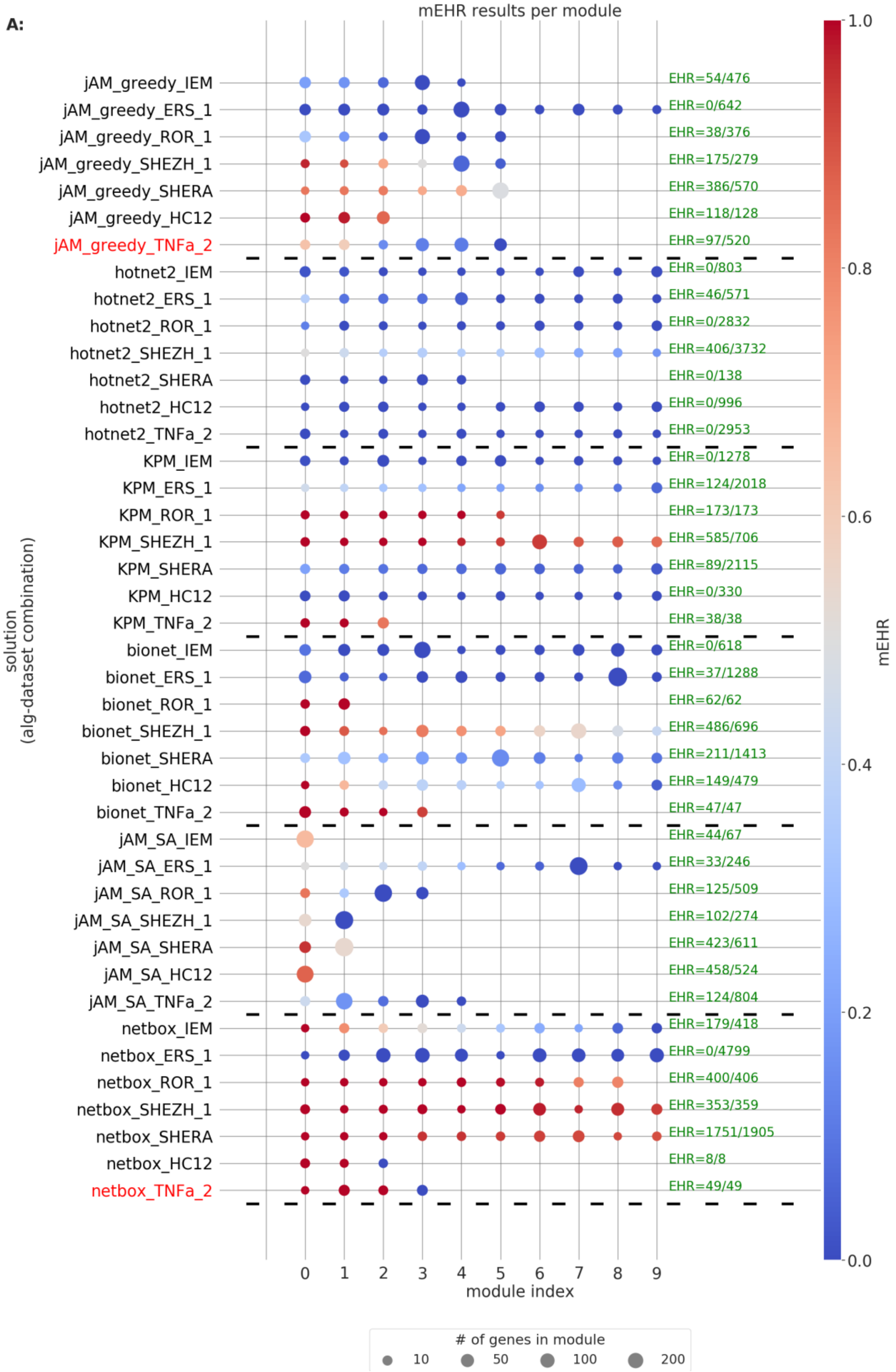


Figure 24: Module-level EHR scores on the GE datasets. mEHR scores for each algorithm and dataset. Note that different solutions have broad range of mEHR score (e.g. the red-marked solutions)

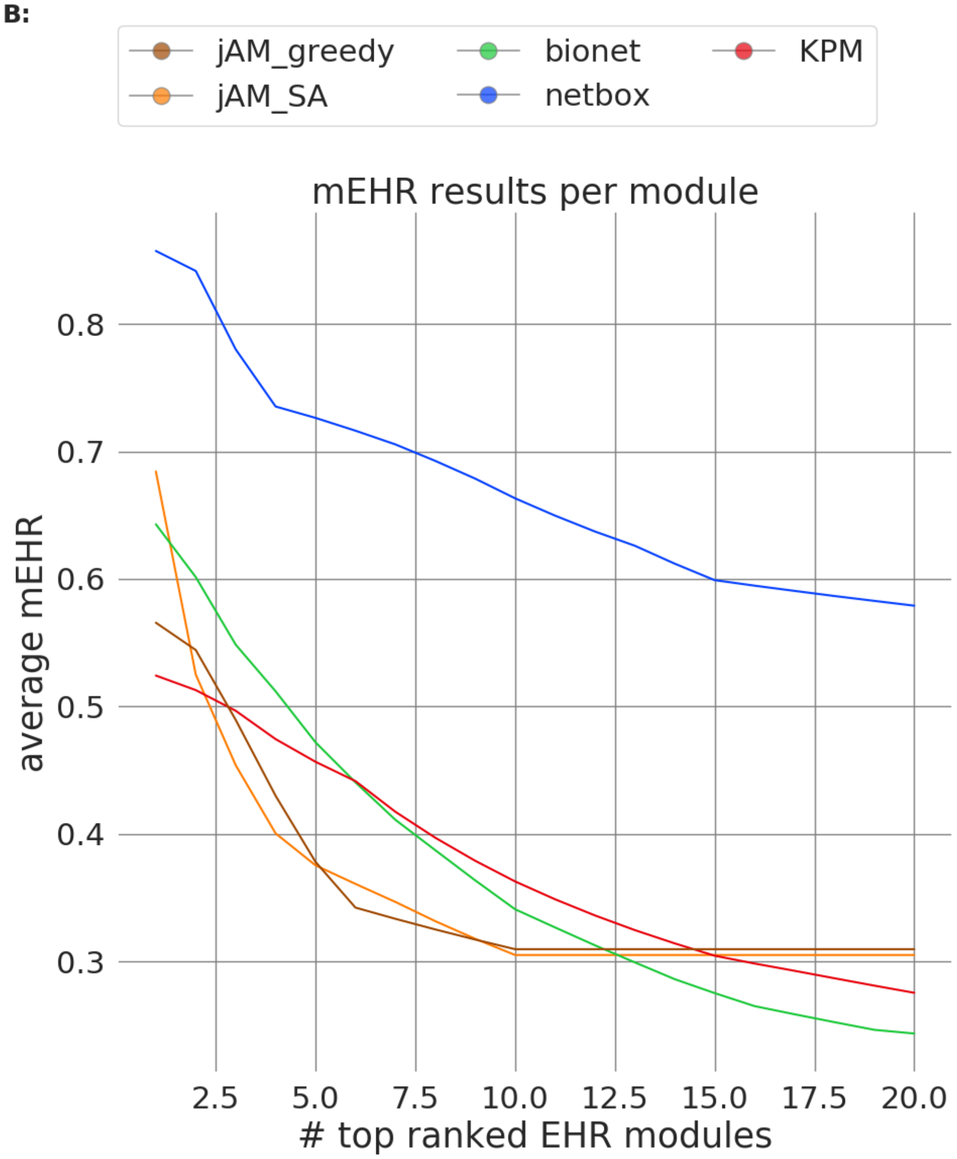


Figure 25: Module-level EHR scores on the GE datasets. Average mEHR score in the k top modules, as a function of k. Modules are ranked by their mEHR scores.

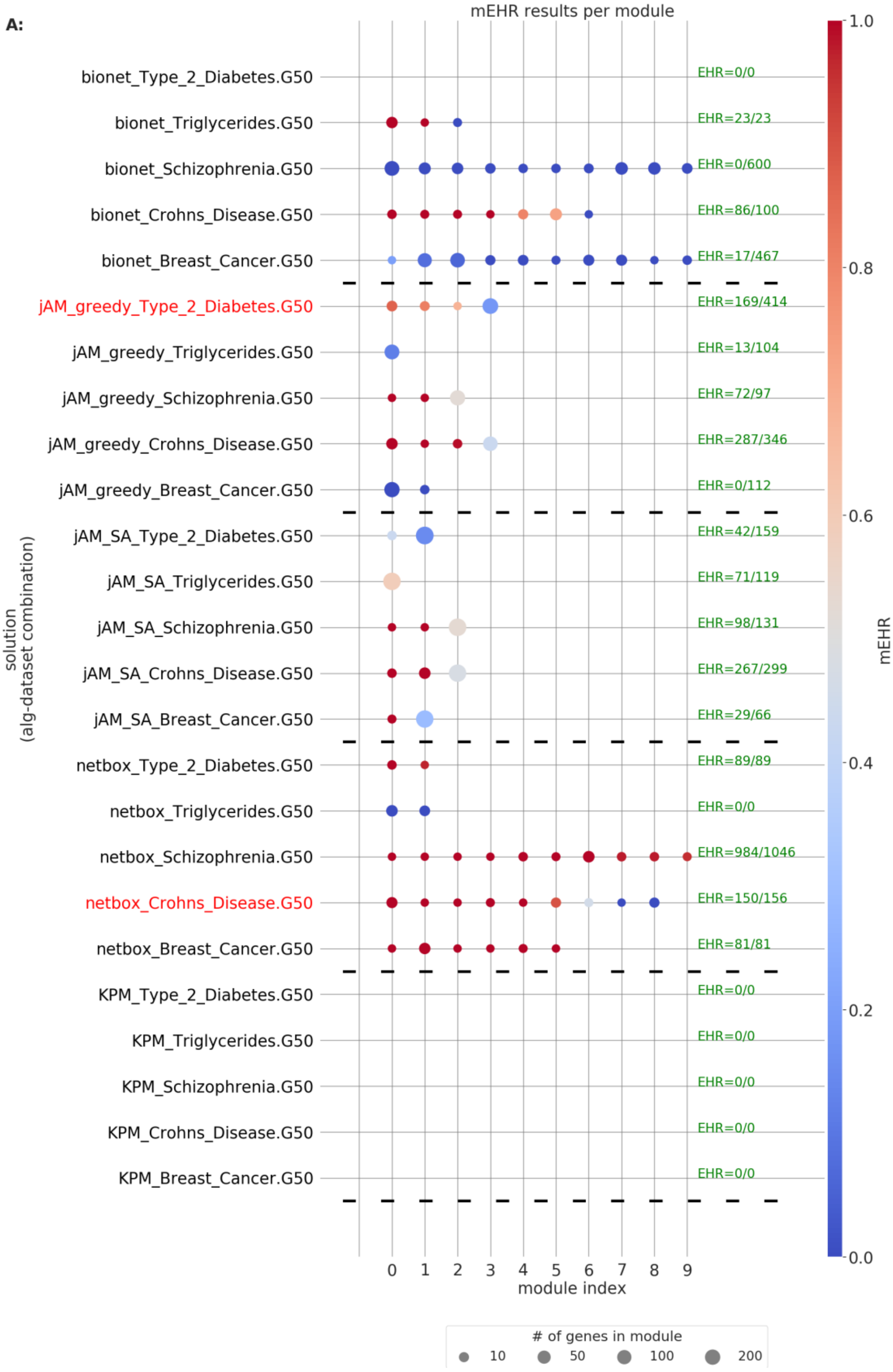


Figure 26: Module-level EHR scores on the GWAS datasets. mEHR scores for each algorithm and dataset. Note that different solutions have broad range of mEHR score (e.g. the red-marked solutions).

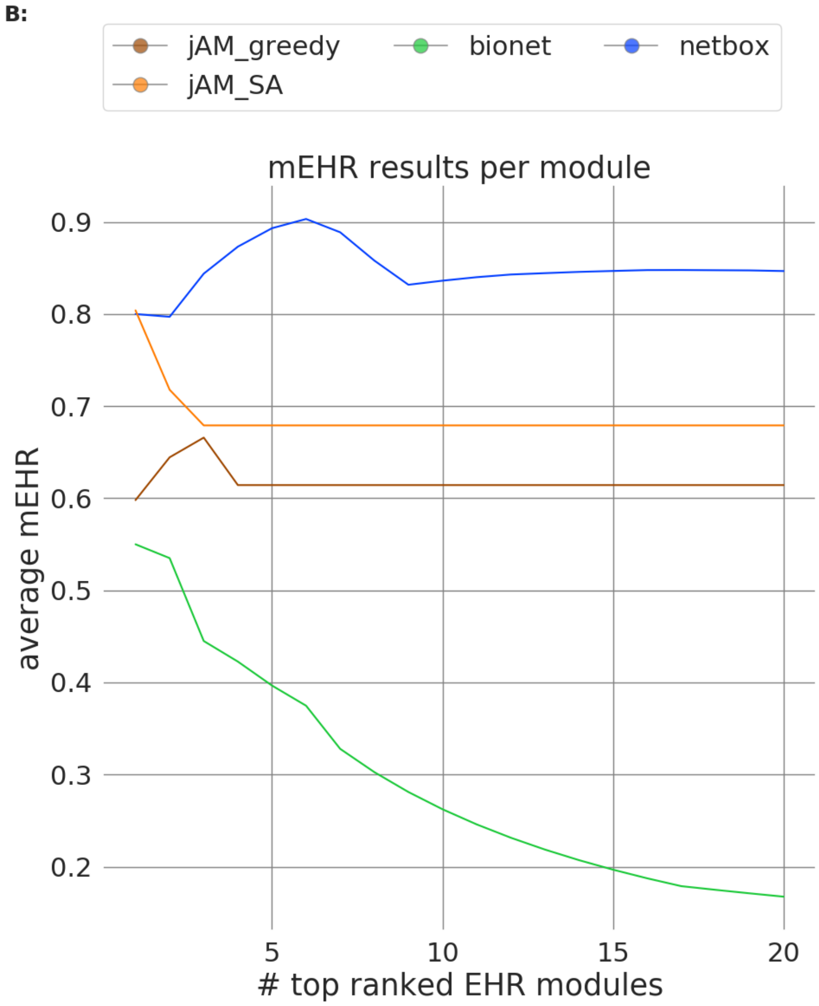


Figure 27: Module-level EHR scores on the GWAS datasets. Average mEHR score in the k top modules, as a function of k. Modules are ranked by their mEHR scores.

Figure 28 shows the results for the median intra-module biological homogeneity across datasets. KPM solutions had highest homogeneity for the GE datasets. In GWAS, NetBox had the highest scores.

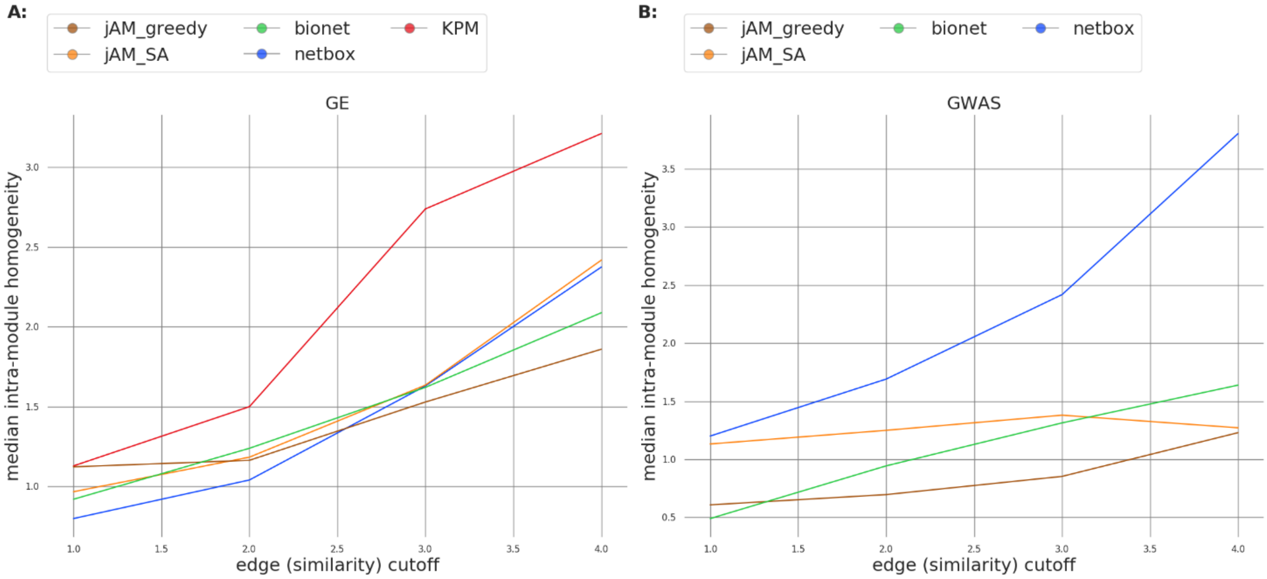


Figure 28: Intra-module homogeneity scores as a function of the edge similarity cutoff. (A) GE (B) GWAS.

As we prefer biologically rich solutions with high biological consistency within each module, we plotted in Figure 29 the intra-module biological homogeneity against biological richness for each dataset and algorithm. Better solutions are located in the upper-right part of the plot. The results for GE are inconclusive, while NetBox tended to give biologically rich and homogeneous solutions in some of the GWAS datasets.

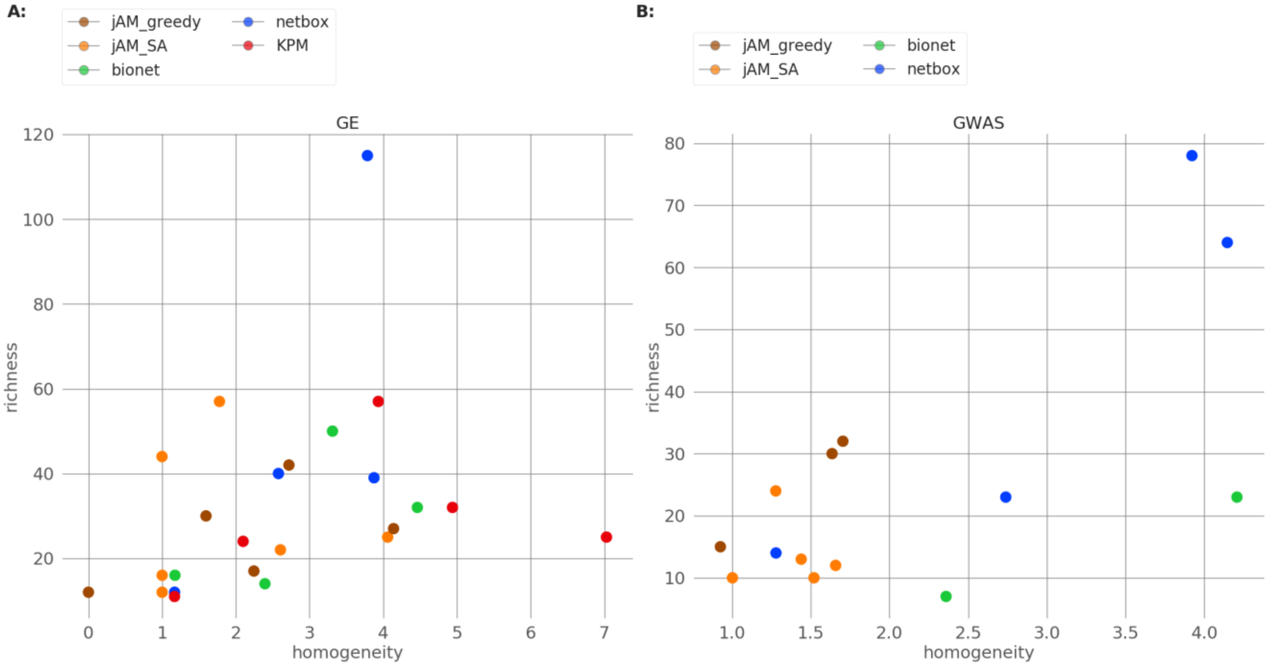


Figure 29: Biological richness vs. intra-module homogeneity. Each plot shows richness (y-axis) and intra-module homogeneity (x-axis) criteria for every algorithm and dataset. The results are for cutoff of 3. (A) GE (B) GWAS.

6 E. Domino – A Novel NBMD Algorithm

After concluding the benchmark, we moved to design a novel NBMD algorithm called Domino (Discovery of Modules In Networks using Omics).

6 E i. Method

Domino receives as input a set of genes called the *active genes* and a network of gene interactions. It aims to find disjoint connected subnetworks in which the active genes are abundant. The algorithm works as follows

0. Dissect the network into disjoint, highly connected subnetworks (*slices)*
1. Detect *relevant slices* where active genes are abundant
2. For each relevant slice S
    a. Refine S to a sub-slice S'
    b. Repartition S' into putative modules
3. Report as final modules those that are rich with active genes.

Step 0 is a pre-processing step that depends only on the network, and is done only once per network. We now describe each step in more detail.

(0) **Dissecting the network into slices.**

This step splits the network into disjoint subnetworks called slices. Splitting is done using a variant of the Newman-Girvan modularity detection algorithm[47]. Briefly, it iteratively removes an edge with the highest betweenness-centrality score and recomputes the modularity score of the resulting network. The process ends after the first iteration where no improvement in the modularity score is achieved. Each connected component in the final network that has more than three nodes is defined as a slice. See Figure 31-A.

(1) Detecting relevant slices.

In this step we test each slice for enrichment in active nodes, using the Hypergeometric (HG) test, and correct the p-values obtained for multiple testing using BH-FDR[36]. Slices with q-values < 0.3 are accepted as relevant slices. See Figure 31-B.

(2a) Refining the relevant slices into sub-slices.

We wish to extract from each slice a single connected component that captures most of the activity signal. This is done by formulating and solving a Prize Collecting Steiner Tree (PCST) problem[48]. In PCST, nodes have values called prizes, and edges have values called penalties. All values are non-negative. The goal is to find a subtree T that maximizes the sum of the prizes of nodes in T minus the sum penalties of the edges in it, i.e., $\sum_{v\in T} p(v) - \sum_{e\in T} c(e)$ where $p(v)$ is the prize of node v, and $c(e)$ is the cost of edge e.

The node prizes are computed by diffusing the activity of the nodes using influence propagation with a linear threshold model[49]. The process is iterative: Initially the set of active nodes is as defined by the input. In each iteration, an inactive node is activated if the sum of the influence of its active neighbors exceeds $\theta = 0.5$. The influence of a node that has k neighbors on each neighbor is $\frac{1}{k}$. Activated nodes remain so in all subsequent iterations. The process ends when no new node is activated. If v became active in iteration l then $p(v) = 0.7^l$. See Figure 30.

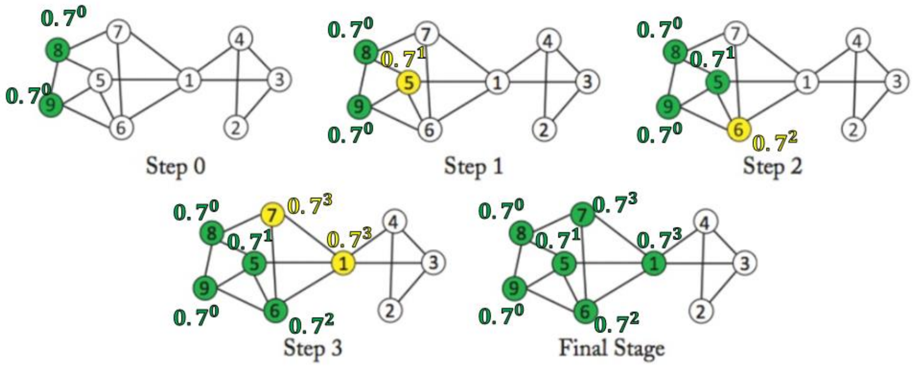


Figure 30: Schematic illustration of the influence propagation process. Each step represents an iteration of the process. Green nodes are active. Nodes marked in yellow in a specific step are activated in this step. $p(v)$ *–the PCST node prize – appears next to each node. Nodes that were not activated ha1ve prize zero.*

To compute the edge penalties, we apply the same influence propagation process on 100 instances with randomly permuted assignment of active nodes, and compute the average score $a(v) = avg(p(v))$ for each node. The normalized permuted score $NPS(u)$ for node u is:

$$NPS(u) = \frac{a(u) - min_{v \in V} a(v)}{max_{v \in V} a(v) - min_{v \in V} a(v)}$$

We define the penalty of edge $e = (u, v)$ as:

$$c(e) = min(NPS(u), NPS(v)).$$

PCST is NP-hard but good heuristics are available. In our algorithm we used FAST-PCST[50]. The resulting subgraph obtained by solving PCST on the slice is called its sub-slice. See Figure 31-C

(2b) Partitioning sub-slices into putative modules

Each sub-slice of >10 nodes is partitioned using the Newman-Girvan algorithm. The algorithm is stopped when the modularity score exceeds $\frac{\log(\#\, of\ nodes\ in\ sub-slice)}{\log(\#\, of\ nodes\ in\ network)}$). The resulting connected components, and the sub-slices of ≤ 10 nodes, are called putative modules. See Figure 31-D.

(3) Identifying the final modules

We test each putative module for enrichment with active nodes using the HG test, and correct for multiple testing using Bonferroni correction. Those with q-value < 0.05 are reported as the final modules. See Figure 31-E.

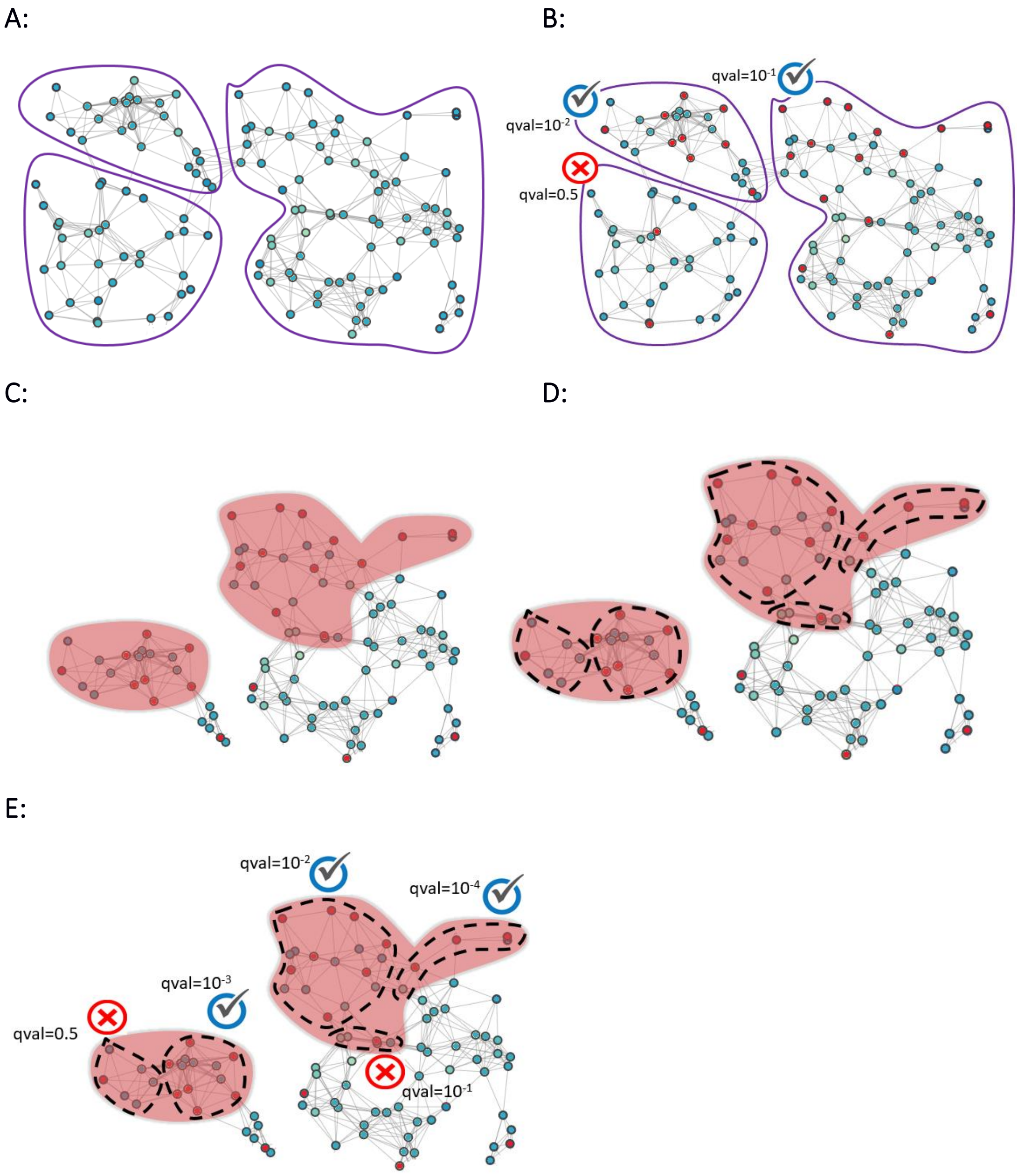


Figure 31: Schematic illustration of Domino: (A) The global network is dissected by the Newman-Girvan (NG) modularity algorithm into three slices (encompassed in purple line). (B) A slice is relevant if it passes a moderate test for enrichment of active nodes ($BH\ qval \leq\ 0.3$). (C) Each relevant slice (red areas) a most active sub-slice is identified using PCST. (D) Sub-slices are dissected further into putative modules using the NG algorithm. (E) Each putative module that passes a strict test enrichment of active nodes ($Bonferroni \leq\ 0.05$) is reported.

6 E ii. Results

We compared Domino to five NBMD methods that we evaluated in the benchmark: jAM_greedy, jAM_SA[10], bionet[11], NetBox[9], and KPM[12]. We excluded hotnet2 due to its lower performance in the benchmark. We used the same GE and GWAS benchmark datasets. We assessed solutions produced by Domino according to the same criteria described above. We report here on EHR, mEHR, biological richness, intra-module biological homogeneity and robustness by F1. Results for terms count and robustness by AUPR are available in Supplementary Tables Table S3, Table S5. For biological richness and intra-module homogeneity, we used Resnik similarity metric with similarity-cutoff of 3. For mEHR we took from each solution up to 10 modules with the highest mEHR score.

Table 3 summarizes the results of each algorithm. For GE, Domino performed best in three of the five criteria, while NetBox had the highest biological richness and KPM the highest intra-module homogeneity. For GWAS, Domino was best or equal best in four criteria and second in intra-module homogeneity, where NetBox was best. A summary of the results in terms of ranking on each criterion are given in Supplementary Tables Table S1Table S2. Fuller results including standard deviations for each criterion are included in Supplementary Tables Table S3, Table S4, Table S5, Table S6.

	NetBox	**JAM_SA**	**Bionet**	**Domino**	**JAM_greedy**	**KPM**
	Gene Expression (GE)					
HER	0.76	0.45	0.46	**0.97**	0.38	0.42
mEHR	0.66	0.31	0.34	**0.92**	0.31	0.36
Robustness	0.37	0.21	0.18	**0.44**	0.20	0.23
Biological Richness	**28**	22	16	22	18	24
Intra-Module Homogeneity	2.85	1.91	2.84	1.79	2.14	**3.83**
	GWAS					
HER	0.78	0.59	0.38	**0.96**	0.42	
mEHR	0.84	0.68	0.26	**0.92**	0.61	
Robustness	0.09	0.17	0.31	**0.76**	0.19	
Biological Richness	**23**	12	5	**23**	15	
Intra-Module Homogeneity	3.02	1.38	**3.28**	3.12	1.42	

Table 3. Summary of the results of each algorithm on each criterion of the benchmarks. Top: Gene expression. Bottom: GWAS. The results are average across the datasets of each type, with the exception of biological richness where the median was used. KPM reported no modules on the GWAS datasets. The F1 score was used in Robustness.

In order to get a global view, we compared pairs of complementary criteria i.e., criteria that are similar in exactly one property.

High EHR can stem from a case in which only a few modules contain most of the EV terms.

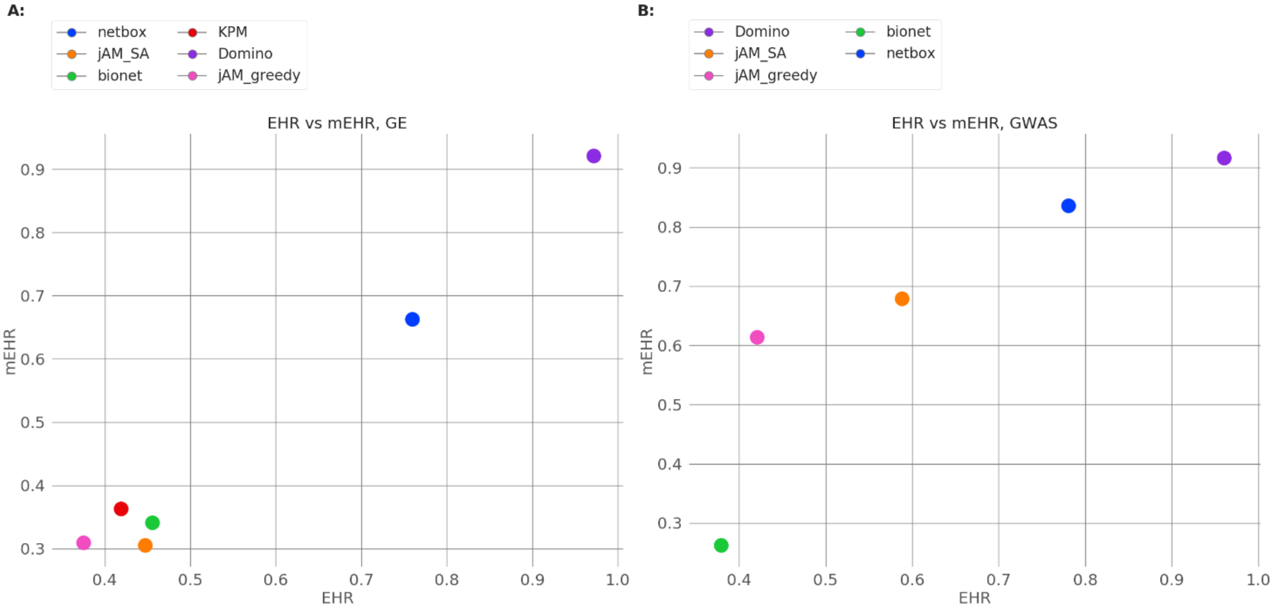


Figure *32* plots EHR vs. mEHR, capturing the full quantitative perspective (Figure 16 in orange). In general, we see agreement between EHR and mEHR, suggesting that EHR levels are not merely affected by a small part of the solution. Notably, on both, GE and GWAS datasets, Domino performs best with near perfect global and per-module scores.

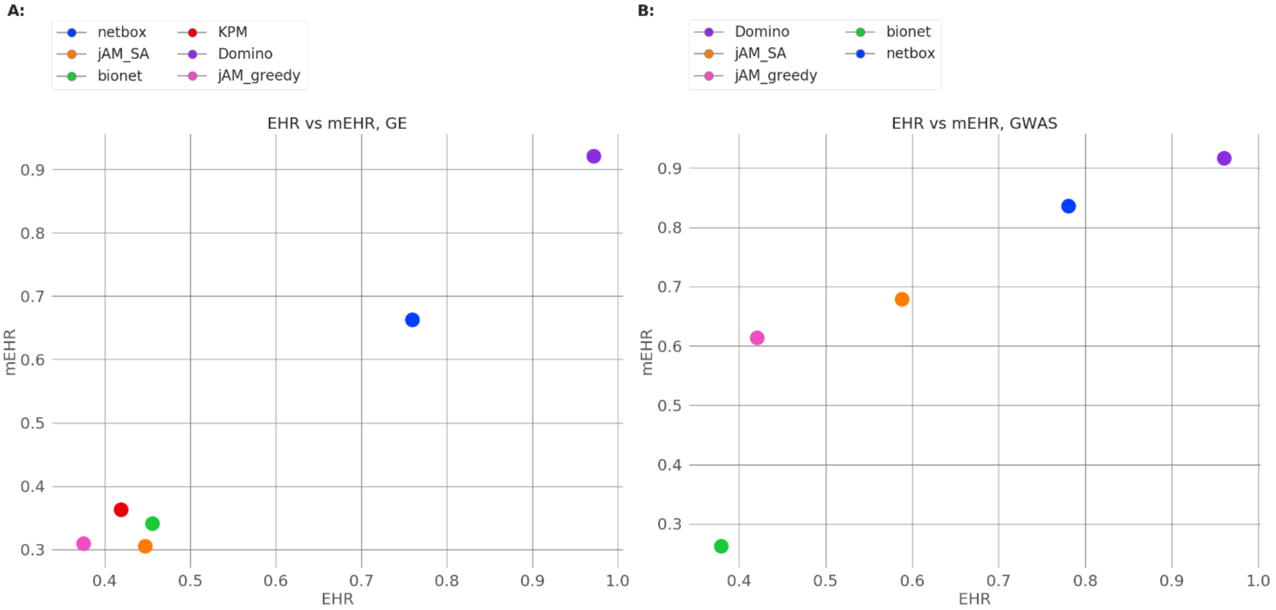


Figure 32: EHR *vs. mEHR. Each plot shows the aggregated scores across datasets for every algorithm. (A) GE (B) GWAS.*

Typically, the downstream analyses of NBMD solutions handle each module separately, looking for the module's unique biological signal. Figure 33 plots mEHR vs. Intra-Module Homogeneity, capturing the full module-level perspective. (Figure 16 in purple). this On the GE datasets, Domino, NetBox and KPM were pareto optimal with KPM highest on Intra-Module Homogeneity and lowest on mEHR, and Domino lowest on Intra-Module Homogeneity and highest in mEHR. On the GWAS datasets, Domino had the best score with NetBox slightly below, and Bionet had slightly higher Intra-Module Homogeneity but the lowest mEHR.

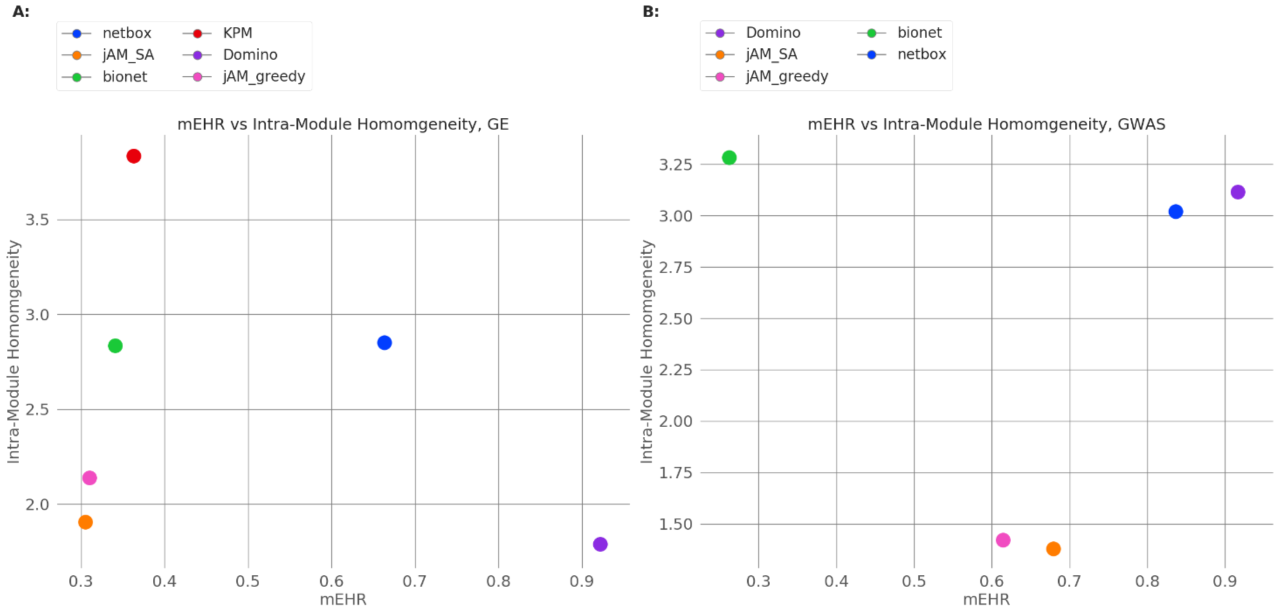
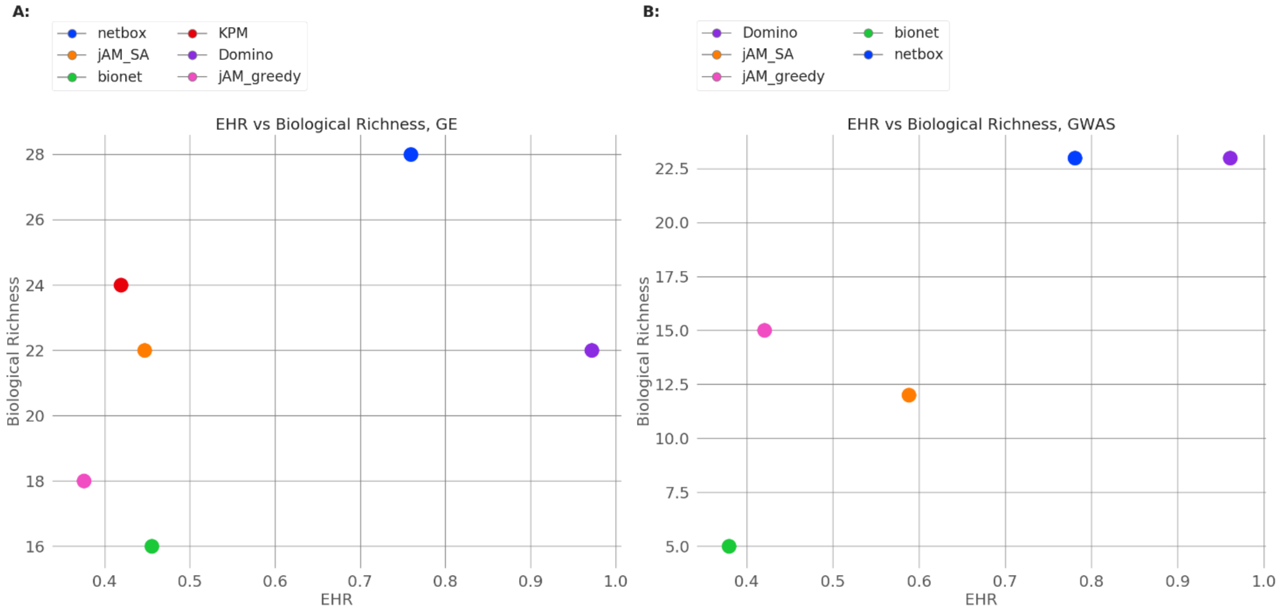


Figure 33: mEHR vs. intra-module homogeneity. Each plot shows the average scores across datasets for every algorithm.(A) GE (B) GWAS.

Additional plots of pairs of criteria are given in supplementary Figures

Figure S1, Figure S2, Figure S3.



7. Discussion

The fundamental challenge of network-based module discovery (NBMD) algorithms is to identify active modules in an underlying network based on genes activity scores. Such scores can be continuous or binary. The comparison of such algorithms is a challenging task, due to the complex nature of the output that such algorithms produce. Algorithms differ dramatically in the number, sizes and the properties of the modules they produce. How can one compare the quality of a solution (set of modules) provided by different algorithms? Although NBMD algorithms have been used for some two decades, there is no accepted community benchmark, and no consensus evaluation criteria have emerged.

To overcome this challenge, since modules are often used to pinpoint functional enrichment, we developed a procedure to filter GO terms that recurrently appear over permuted datasets: We run the algorithm multiple times on randomly permuted gene profiles and obtain a background distribution of the p-values obtained for each GO term. We use this null distribution to empirically correct the results on the real (unpermuted) data accordingly. We call this process EMP (for EMpirical Pipeline) and call the terms that pass it EV (Empirically Validated) terms.

We subsequently used EMP to evaluate how "clean", stable and concise the results of a NBMD algorithm are. Notably, the six algorithms that we tested differed substantially in their EMP validation rates. We observed for some algorithms that many of the GO terms that passed the HG test did not pass the empirical one. Therefore, we recommend users of NBMD algorithms to apply the empirical correction procedure in order to alleviate functional misinterpretation of a network-based algorithm’s solution. We made the EMP code available to the community.

Of the six algorithms that we tested, NetBox[9] ranked consistently high in many criteria. Notably, NetBox also had a low rate of terms that did not pass the empirical evaluation, so our analysis indicates that running it without the time-consuming EMP is relatively safe.

Ideally, one would like to correct the reported terms on the module level. EMP does not consider directly module-based correction for enrichment scores, since each randomized run can produce different sized modules and a different number of them. However, we do provide additional analysis on the module level by marking, for each module in the solution, the enriched GO terms that passed the EMP filter (namely, were reported as EV terms).

Based on our experience in the benchmark, we designed the NBMD algorithm, and demonstrated that it outperformed the algorithms we benchmarked in most criteria. In particular, the mean EMP validation rate for enriched GO terms detected for Domino's solutions was above 90%, making it practical without the need for the slow EMP filtering procedure.

Domino, as NetBox, binarizes the activity scores. Intuitively one may think that such step could lead to a loss of important biological signals. However, the performance of these algorithms suggests that in fact binarizing might help in denoising the process. Further study of this phenomenon is needed.

An obvious drawback of the empirical analysis is running time. Running each algorithm thousands of time for each dataset is slow and expensive, even on a cluster: A correction of a single solution could take several hours up to one week, depending on the algorithm and dataset associated with the solution being analyzed. Understanding better the role of each putative source of bias can lead to runtime improvement of the EMP that would enable its execution on a standard desktop station in a reasonable time. This can also lead to more efficient systematic execution of EMP in a way that will contribute to the selection of the algorithm’s hyper-parameter – another open issue in our analysis. As currently running EMP on a desktop station is infeasible, our results indicate that running Domino can be confidently used even without EMP.

8. References

1. Aittokallio, T. & Schwikowski, B. Graph-based methods for analysing networks in cell biology. *Brief. Bioinform.* **7**, 243–255 (2006).

2. Xenarios, I. *et al.* DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.* **30**, 303–305 (2002).

3. Magger, O., Waldman, Y. Y., Ruppin, E. & Sharan, R. Enhancing the Prioritization of Disease-Causing Genes through Tissue Specific Protein Interaction Networks. *PLoS Comput. Biol.* (2012) doi:10.1371/journal.pcbi.1002690.

4. Margolin, A. A. *et al.* (ARACNE) ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics* (2006) doi:10.1186/1471-2105-7-S1-S7.

5. Szklarczyk, D. *et al.* The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Res.* **45**, D362–D368 (2017).

6. Wu, G., Dawson, E., Duong, A., Haw, R. & Stein, L. ReactomeFIViz: a Cytoscape app for pathway and network-based data analysis. *F1000Research* **3**, 146 (2014).

7. Marbach, D. *et al.* Tissue-specific regulatory circuits reveal variable modular perturbations across complex diseases. *Nat. Methods* **13**, 366 (2016).

8. Leiserson, M. D. M. *et al.* Pan-Cancer Network Analysis Identifies Combinations of Rare Somatic Mutations across Pathways and Protein Complexes. doi:10.1038/ng.3168.

9. Cerami, E., Demir, E., Schultz, N., Taylor, B. S. & Sander, C. (NetBox) Automated network analysis identifies core pathways in glioblastoma. *PLoS One* (2010) doi:10.1371/journal.pone.0008918.

10. Ideker, T., Ozier, O., Schwikowski, B. & Siegel, A. F. *(jActiveModules) Discovering regulatory and signaling circuits in molecular interaction networks*. www.cytoscape.org.

11. Beisser, D., Klau, G. W., Dandekar, T., Müller, T. & Dittrich, M. T. (BioNet) BioNet: An R-Package for the functional analysis of biological networks. *Bioinformatics* (2010) doi:10.1093/bioinformatics/btq089.

12. Baumbach, J. *et al.* (KeyPathwayMiner) Efficient algorithms for extracting biological key pathways with global constraints. in *Proceedings of the fourteenth international conference on Genetic and evolutionary computation conference - GECCO '12* 169 (ACM Press, 2012). doi:10.1145/2330163.2330188.

13. Nakka, P., Raphael, B. J. & Ramachandran, S. Gene and Network Analysis of Common Variants Reveals Novel Associations in Multiple Complex Diseases. *Genetics* **204**, 783–798 (2016).

14. The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Res.* **47**, D330–D338 (2019).

15. Consortium, the M. C. and P. A. working group of the I. C. G. *et al.* Pathway and network analysis of cancer genomes. *Nat. Methods* **12**, 615–621 (2015).

16. Available at https://www.ebi.ac.uk/QuickGO/term/GO:1904948. https://www.ebi.ac.uk/QuickGO/.

17. Khera, A. V. *et al.* Genome-wide polygenic scores for common diseases identify individuals with risk equivalent to monogenic mutations. *Nat. Genet.* **50**, 1219–1224 (2018).

18. Lamparter, D., Marbach, D., Rueedi, R., Kutalik, Z. & Bergmann, S. Fast and Rigorous Computation of Gene and Pathway Scores from SNP-Based Summary Statistics. *PLOS Comput. Biol.* **12**, e1004714 (2016).

19. de Leeuw, C. A., Mooij, J. M., Heskes, T. & Posthuma, D. MAGMA: Generalized Gene-Set Analysis of GWAS Data. *PLOS Comput. Biol.* **11**, e1004219 (2015).

20. Available at https://www.genome.gov/about-genomics/fact-sheets/Genome-Wide-Association-Studies-Fact-Sheet. https://www.genome.gov/about-genomics/fact-sheets/Genome-Wide-Association-Studies-Fact-Sheet.

21. Mitra, K., Carvunis, A.-R., Ramesh, S. K. & Ideker, T. Integrative approaches for finding modular structure in biological networks. *Nat. Rev. Genet.* (2013) doi:10.1038/nrg3552.

22. Piñero, J. *et al.* DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Res.* **45**, D833–D839 (2017).

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

24. Fabregat, A. *et al.* The Reactome Pathway Knowledgebase. *Nucleic Acids Res.* **46**, D649–D655 (2018).

25. He, H., Lin, D., Zhang, J., Wang, Y. & Deng, H. Comparison of statistical methods for subnetwork detection in the integration of gene expression and protein interaction network. *BMC Bioinformatics* (2017) doi:10.1186/s12859-017-1567-2.

26. Peri, S. Development of Human Protein Reference Database as an Initial Platform for Approaching Systems Biology in Humans. *Genome Res.* **13**, 2363–2371 (2003).

27. Resnik, P. *Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language*. *Journal of Artiicial Intelligence Research* vol. 11 https://arxiv.org/pdf/1105.5444.pdf (1999).

28. Supek, F., Bošnjak, M., Škunca, N. & Šmuc, T. REVIGO Summarizes and Visualizes Long Lists of Gene Ontology Terms. *PLoS One* **6**, e21800 (2011).

29. Kirkpatrick, S., Gelatt, C. D. & Vecchi, M. P. Optimization by simulated annealing. *Science* **220**, 671–80 (1983).

30. Network, T. C. G. A. R. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature* **455**, 1061–1068 (2008).

31. Dittrich, M. T., Klau, G. W., Rosenwald, A., Dandekar, T. & Müller, T. (Heinz) Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics* **24**, i223-31 (2008).

32. Lawrence, M. S. *et al.* Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature* **499**, 214–218 (2013).

33. Liberzon, A. *et al.* Molecular signatures database (MSigDB) 3.0. *Bioinformatics* **27**, 1739–40 (2011).

34. Subramanian, A. *et al.* Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. U. S. A.* **102**, 15545–50 (2005).

35. Robinson, M. D., McCarthy, D. J. & Smyth, G. K. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**, 139–40 (2010).

36. Benjamini, Y. & Hochberg, Y. Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing. *Journal of the Royal Statistical Society* Series B: Methodological 57:289-300 DOI: 10.2307 (1995).

37. Schmidt, S. F. *et al.* Acute TNF-induced repression of cell identity genes is mediated by NFκB-directed redistribution of cofactors from super-enhancers. *Genome Res.* **25**, 1281–1294 (2015).

38. Elkon, R. *et al.* RFX transcription factors are essential for hearing in mice. *Nat. Commun.* **6**, (2015).

39. Miano, V. *et al.* Luminal lncRNAs regulation by ERα-controlled enhancers in a ligand-independent manner in breast cancer cells. *Int. J. Mol. Sci.* **19**, (2018).

40. Ito, T., Teo, Y. V., Evans, S. A., Neretti, N. & Sedivy, J. M. Regulation of Cellular Senescence by Polycomb Chromatin Modifiers through Distinct DNA Damage- and Histone Methylation-Dependent Pathways. *Cell Rep.* **22**, 3480–3492 (2018).

41. Kroeger, H. *et al.* The unfolded protein response regulator ATF6 promotes mesodermal differentiation. *Sci. Signal.* **11**, (2018).

42. Hertzano, R. *et al.* Cell type-specific transcriptome analysis reveals a major role for Zeb1 and miR-200b in mouse inner ear morphogenesis. *PLoS Genet.* **7**, (2011).

43. Bayerlová, M. *et al.* Ror2 signaling and its relevance in breast cancer progression. *Front. Oncol.* **7**, (2017).

44. Huang, J. K. *et al.* Systematic Evaluation of Molecular Networks for Discovery of Disease Genes. *Cell Syst.* (2018) doi:10.1016/j.cels.2018.03.001.

45. Klopfenstein, D. V. *et al.* GOATOOLS: A Python library for Gene Ontology analyses. *Sci. Rep.* **8**, 10872 (2018).

46. Shannon, P. *et al.* Cytoscape: A Software Environment for Integrated Models of Biomolecular

Interaction Networks. *Genome Res.* **13**, 2498–2504 (2003).

47. Girvan, M. & Newman, M. E. J. Community structure in social and biological networks. *Proc. Natl. Acad. Sci. U. S. A.* **99**, 7821–7826 (2002).

48. Ljubić, I. *et al.* An Algorithmic Framework for the Exact Solution of the Prize-Collecting Steiner Tree Problem. *Math. Program.* **105**, 427–449 (2006).

49. Kempe, D., Kleinberg, J. & Tardos, E. *Maximizing the Spread of Influence through a Social Network*. (2003).

50. Hegde, C., Indyk, P. & Schmidt, L. *A Fast, Adaptive Variant of the Goemans-Williamson Scheme for the Prize-Collecting Steiner Tree Problem*.

9. Supplementary

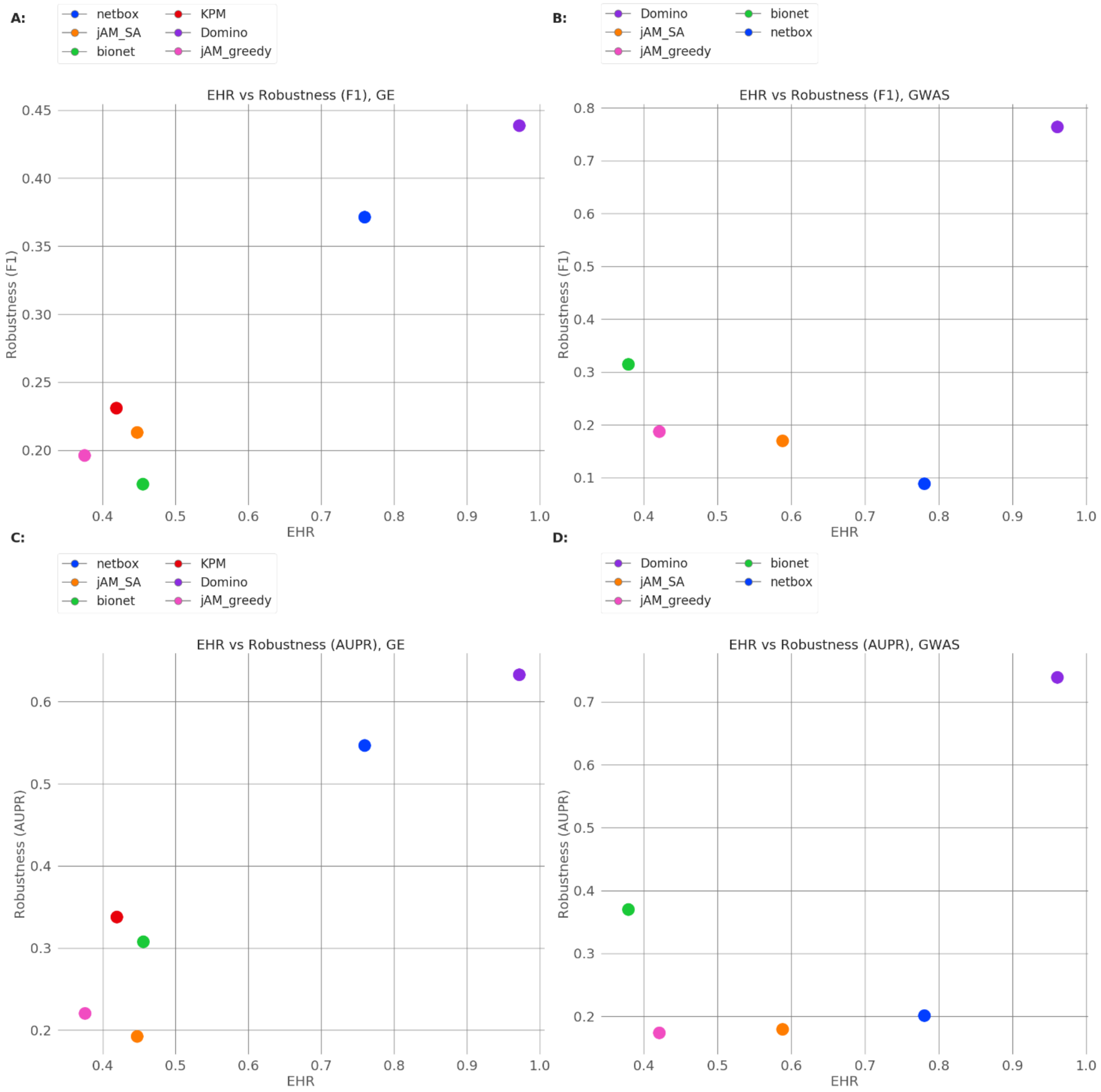


Figure S1: Robustness measured by F1 vs. robustness measured by AUPR. Each plot shows the average scores across datasets for every algorithm.(A) GE (B) GWAS.

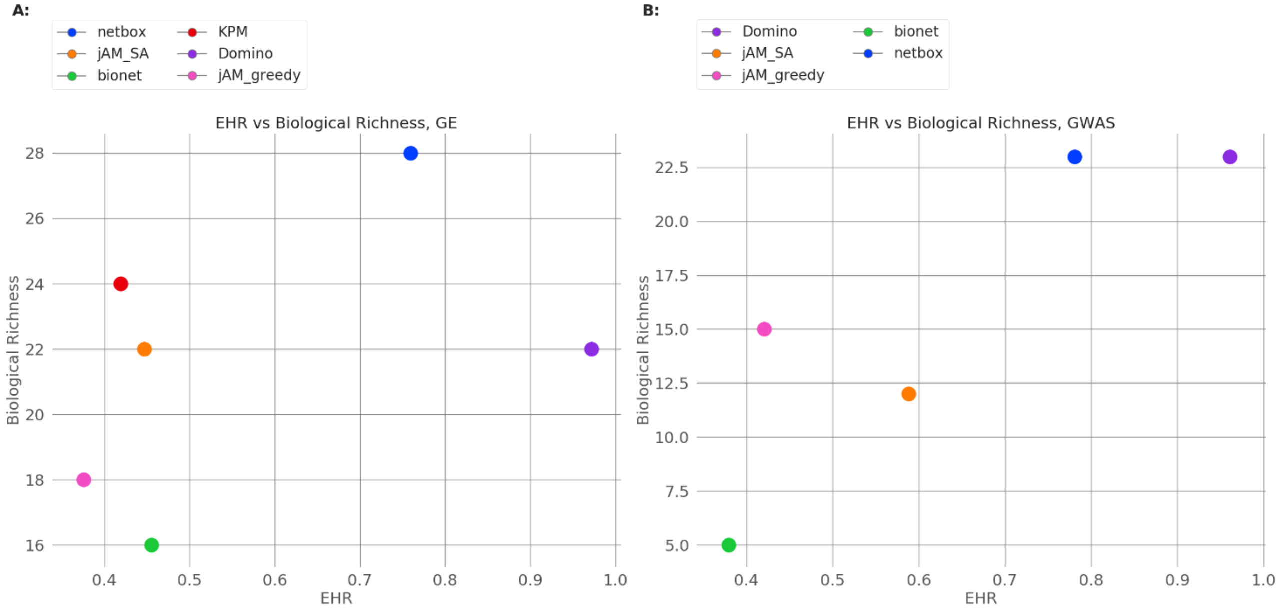


Figure S2: EHR vs. biological richness. Each plot shows the average scores across datasets for every algorithm.(A) GE (B) GWAS.

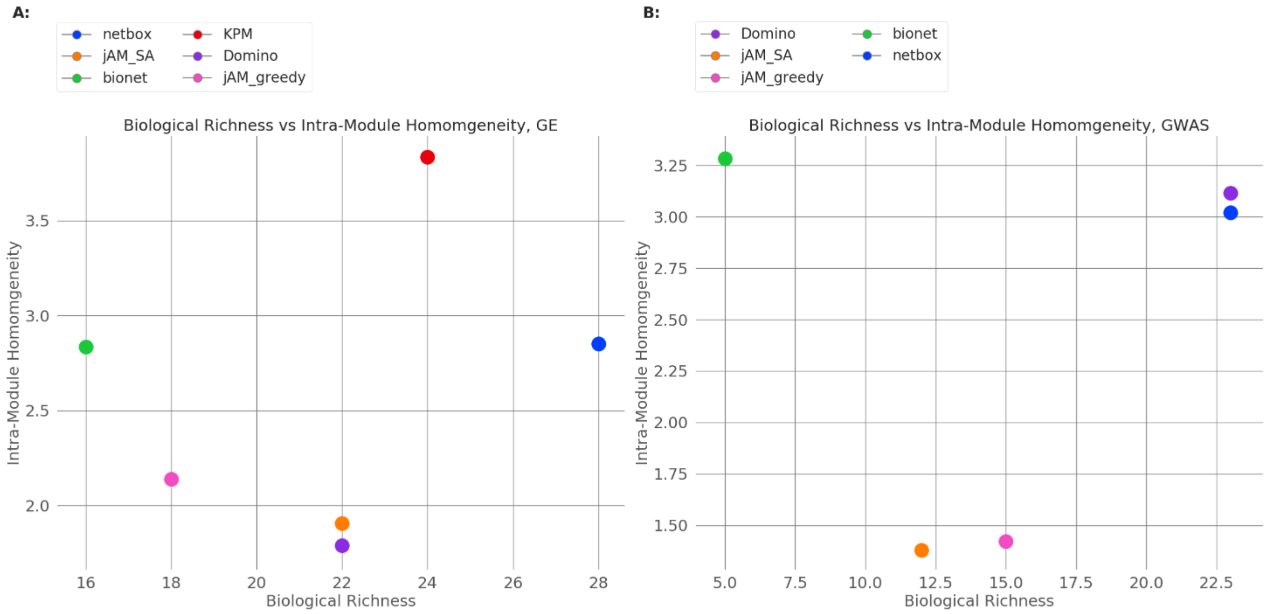


Figure S3: Biological richness vs. intra-module homogeneity. Each plot shows the average scores across datasets for every algorithm. (A) GE (B) GWAS.

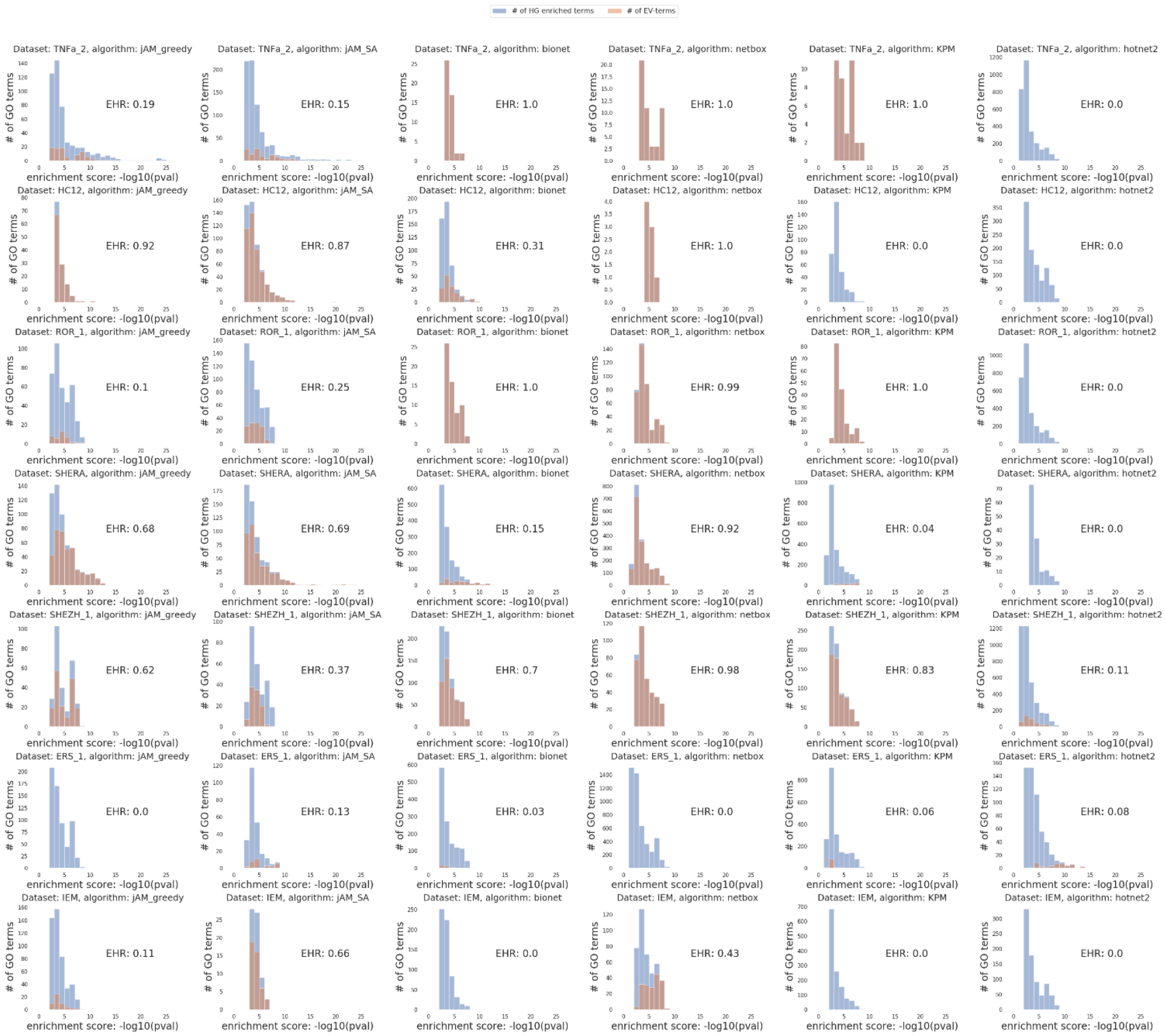


Figure S4: GO term enrichment score distributions for HG-enriched and EV terms for each algorithm and *gene expression* datasets*. The EHR value is reported in each case. The EHR metric is as defined in Section 7 C i (1). Empirical to Hypergeometric Ratio (EHR).*

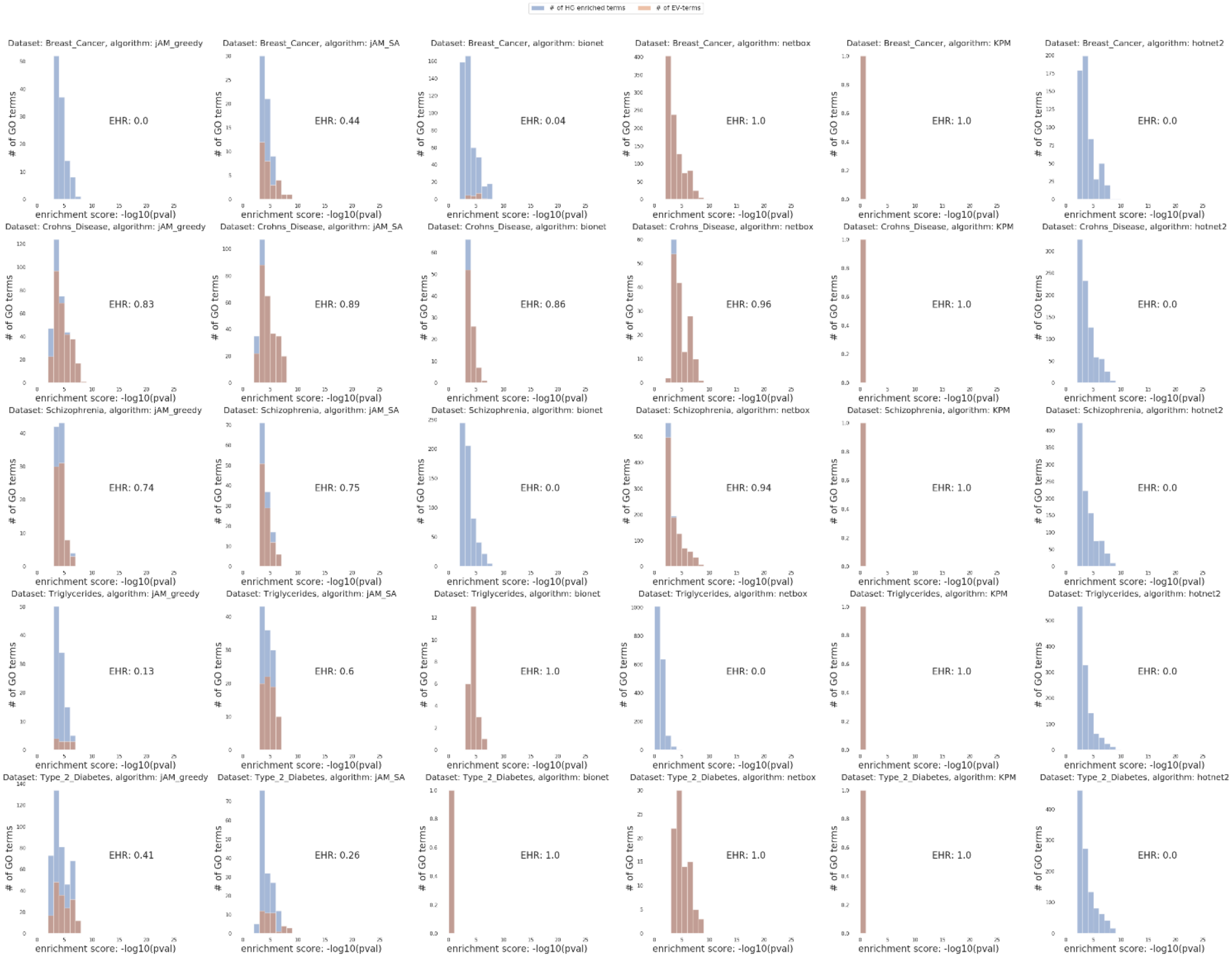


Figure S5: *GO term enrichment score distributions for HG-enriched and EV terms for each algorithm and GWAS datasets. The EHR value is reported in each case. The EHR metric is as defined in Section 7 C i (1). Empirical to Hypergeometric Ratio (EHR).*

	EHR	Biological richness	mEHR	Intra module homogeneity	Recovery: F1	Recovery: AUPR	Average rank	# top rank
Domino	1	3.5	1	6	1	1	2.25	4
NetBox	2	1	2	2.5	2	2	1.92	1
JAM_SA	4	3.5	4.5	5	6	6	4.83	0
Bionet	3	6	6	2.5	5	4	4.42	0
KPM	5	2	3	1	3	3	2.83	1
JAM_Greedy	6	5	4.5	4	4	5	4.75	0

Table S1: Ranking table for GE. For each criterion, ranks are from 1 (best performer) to 6, with fractional numbers for ties. We ranked the scores from 1 (top) to 6 (bottom), and averaged tied ranks. In terms of number of criteria ranked first Domino got the highest score. In terms of the average rank, Domino was second-best after Netbox.

	EHR	Biological richness	mEHR	Intra module homogeneity	Recovery: F1	Recovery: AUPR	Average rank	# top rank
Domino	1	1.5	1	2	1	1	1.25	5
NetBox	2	1.5	2	3	5	5	3.08	1
JAM_SA	3	4	3	5	4	4	3.83	0
Bionet	5	5	5	1	2	2	3.33	1
JAM_Greedy	4	3	4	4	3	3	3.50	0

Table S2: Ranking table for GWAS. For each criterion, ranks are from 1 (best performer) to 5, with fractional numbers for ties. We ranked the scores from 1 (top) to 6 (bottom), and averaged tied ranks. In terms of number of criteria ranked first and the average rank, Domino got the highest score.

	Term count	EHR	mEHR	Robustness (F1)	Robustness (AUPR)	Biological richness	Intra-module homogeneity
NetBox	391.43	0.76	0.66	0.37	0.55	28	2.85
JAM_SA	187.00	0.45	0.31	0.21	0.19	22	1.91
Bionet	141.71	0.46	0.34	0.18	0.31	16	2.84
KPM	144.14	0.42	0.36	0.23	0.34	24	3.83
Domino	267.57	0.97	0.92	0.44	0.63	22	1.79
JAM_greedy	124.00	0.38	0.31	0.20	0.22	18	2.14

Table S3: Aggregated results of each algorithm on each criterion of the benchmarks on gene expression datasets. Aggregation was done using mean across datasets for each criterion except for "biological richness", where median was applied.

	Term count	EHR	mEHR	Robustness (F1)	Robustness (AUPR)	Biological richness	Intra-module homogeneity
NetBox	620.83	0.39	0.43	0.21	0.30	39.03	1.27
JAM_SA	177.17	0.29	0.31	0.18	0.20	16.87	1.23
Bionet	168.21	0.44	0.37	0.14	0.32	16.77	1.39
KPM	204.72	0.49	0.43	0.26	0.41	19.92	2.32
Domino	221.28	0.04	0.11	0.13	0.07	19.02	0.39
JAM_greedy	128.89	0.36	0.37	0.19	0.24	13.57	1.52

Table S4: standard deviation of results of each algorithm on each criterion of the benchmarks on gene expression datasets.

	Term count	EHR	mEHR	Robustness (F1)	Robustness (AUPR)	Biological Richness	Intra-module homogeneity
NetBox	260.8	0.78	0.84	0.09	0.20	23	3.02
JAM_SA	101.4	0.59	0.68	0.17	0.18	12	1.38
Bionet	25.2	0.38	0.26	0.31	0.37	5	3.28
Domino	173.4	0.96	0.92	0.76	0.74	23	3.12
JAM_greedy	108.2	0.42	0.61	0.19	0.17	15	1.42

Table S5: Aggregated results of each algorithm on each criterion of the benchmarks on GWAS datasets. Aggregation was done using mean across datasets for each criterion except for "biological richness", where median was applied.

	Term count	EHR	mEHR	Robustness (F1)	Robustness (AUPR)	Biological richness	Intra-module homogeneity
NetBox	407.79	0.44	0.35	0.18	0.28	33.26	1.32
JAM_SA	96.34	0.25	0.33	0.17	0.23	5.85	0.25
Bionet	35.49	0.51	0.42	0.45	0.48	9.46	1.31
Domino	142.08	0.04	0.10	0.15	0.18	18.55	1.15
JAM_greedy	120.15	0.37	0.40	0.21	0.19	13.86	0.43

Table S6: standard deviation of each algorithm on each criterion of the benchmarks on GWAS datasets.

	HC12	SHERA	ROR_1	TNFa_2	SHEZH_1	ERS_1	IEM
hotnet2	0.03	0.13	0	0	0.7	0.1	0
KPM	0.28	0.08	0	0	0.01	0.56	0.28
netbox	0.16	0	0	0	0	0.48	0.09
JAM_greedy	0	0.15	0.27	0.54	0.13	0.28	0.36
bionet	0.2	0.07	0	0	0.03	0.02	0.19
JAM_SA	0.04	0.12	0.2	0.56	0.44	0.19	0.12

Table S7: Ratio between the number of overlapping terms of one permuted dataset and the unpermuted dataset, and the number of terms in the unpermuted dataset, for each algorithm and gene expression dataset.

	HC12	SHERA	ROR_1	TNFa_2	SHEZH_1	ERS_1	IEM
hotnet2	0.03	0.13	0	0	0.7	0.1	0
KPM	0.28	0.08	0	0	0.01	0.56	0.28
netbox	0.16	0	0	0	0	0.48	0.09
JAM_greedy	0	0.15	0.27	0.54	0.13	0.28	0.36
bionet	0.2	0.07	0	0	0.03	0.02	0.19
JAM_SA	0.04	0.12	0.2	0.56	0.44	0.19	0.12

Table S8: Ratio between the number of overlapping terms of one permuted dataset and the unpermuted dataset, and the number of terms in the unpermuted dataset, for each algorithm and gene expression dataset.

	Triglycerides. G50	Crohns_Disease .G50	Breast_Cancer. G50	Schizophrenia. G50	Type_2_Diabetes .G50
hotnet2	0.02	0.17	0.02	0	0.1
netbox	0	0	0	0	0
JAM_greedy	0.29	0.04	0.26	0.12	0.03
JAM_SA	0.09	0	0.15	0	0.17
bionet	0	0	0.12	0.26	0

Table S9: Ratio between the number of overlapping terms of one permuted dataset and the unpermuted dataset, and the number of terms in the unpermuted dataset, for each algorithm and GWAS datasets.

תקציר:

אלגוריתמים מבוססי רשת למציאת מודולים נמצאים בשימוש לצרכי אפיון ביולוגי של פרופיליים גנומיים במשך כשני עשורים. אלגורתמים אלו מקבלים כקלט רשת ביולוגית וניקוד לגנים המבוסס על הפרופיל הגנומי ומדווחים תתי רשתות (מודולים) המתיימרים להיות בעלי חשיבות ביולוגית.

במחקר ערכנו השוואה שיטתית בין אלגוריתמים מבוססי רשת למציאת מודולים. ההשוואה נערכה על בסיס מידע גנומי מפרופיל ביטויי גנים ומניקוד מבוסס GWAS. במחקר התמקדנו בחמש מהשיטות המובילות: jActiveModules, NetBox, Hotnet2, Keypathwayminer, bionet. ההשוואה בין האלגוריתמים התבססה על ההעשרה הביולוגית של קבוצות Gene Ontology (GO) שנמצאה במודולים אותם כל שיטה דיווחה.

במהלך המחקר הבחנו כי מספר גדול של העשרות ביולוגיות חוזרות ומופיעות גם בפתרונות בהן ערבבנו את קלט ניקוד הגנים. בעקבות כך העלינו השערה כי העשרות אלו עולות בעקבות הטיה שמקורה באלגוריתם ומבנה הרשת. על מנת לנקות העשרות שעולות בעקבות הטיות שכאלו פיתחנו כלי בשם EMP לניקוי העשרות ביולוגיות המדווחות מהמודולים. הכלי מבוסס על השוואת ההעשרות של המודולים המתקבלים מניקוד הגנים המקורי, לאלו המתקבלות לאחר ערבוב ניקוד הגנים. לאחר מכן פנינו לתכנון קריטריוני השוואה בין האלגוריתמים השונים המבוססים על תוצאות פרוצדורת ה-EMP ועל פיהם הערכנו את האלגוריתמים השונים, ומצאנו כי NetBox עולה על מתחריו באופן עקבי ברוב הקריטריונים.

לסיום, פיתחנו שיטה חדשה בשם Domino (Discovery of Modules In Networks using Omics) והראינו כי היא עולה על השיטות האחרות על פי הקריטריונים שפיתחנו. חשוב לציין כי בפתרונות שהתקבלו מ-Domino מעל 90% מההעשרות הביולוגיות שרדו את פרוצדורת ה-EMP, שיעור גבוה משמעותית מהכלים האחרים.

הרצת פרוצדורת EMP לצורך ניקוי השערות מוטות היא תהליך כבד מבחינה חישובית. בהתחשב בשיעור הנמוך של השערות מוטות באלגוריתם Domino, הכלי יכול לשמש במחקרים ביולוגיים על גבי עמדת קצה סטנדרטית ללא ניקוי.

אוניברסיטת תל אביב

הפקולטה למדעים מדויקים ע"ש ריימונד ובברלי סאקלר

בית הספר למדעי המחשב ע"ש בלווטניק

ניתוח אלגוריתמים מבוססי רשת למציאת מודולים
מנקודת מבט של העשרות ביולוגיות

חיבור זה הוגש כחלק מהדרישות לקבלת התואר

"מוסמך אוניברסיטה" – M.Sc. באוניברסיטת תל-אביב

ביה"ס למדעי המחשב

על ידי

חגי לוי

בהנחיית

פרופ' רון שמיר

דר' רן אלקון

חשוון תש"ף