DOMINO: a novel algorithm for network-based identification of active modules with reduced rate of false calls

Hagai Levi[1], Ran Elkon[2,3,*] and Ron Shamir[1,*]

[1]The Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. [2]Department of Human Molecular Genetics and Biochemistry, Sackler School of Medicine, Tel Aviv University 69978, Tel Aviv, Israel. [3]Sagol School of Neuroscience, Tel Aviv University, Tel Aviv 69978, Israel. *equal contribution.

Abstract

Algorithms for active module identification (AMI) are central to analysis of omics data. Such algorithms receive a gene network and nodes' activity scores as input and report sub-networks that show significant over-representation of accrued activity signal ('active modules'), thus representing biological processes that presumably play key roles in the analyzed biological conditions. Although such methods exist for almost two decades, only a handful of studies attempted to compare the biological signals captured by different methods. Here, we systematically evaluated six popular AMI methods on gene expression (GE) and GWAS data. Notably, we observed that GO terms enriched in modules detected by these methods on the real data were often also enriched on modules found on randomly permuted input data. This indicated that AMI methods frequently report modules that are not specific to the biological context measured by the analyzed omics dataset. To tackle this bias, we designed a permutation-based method that evaluates the empirical significance of GO terms reported as enriched in modules. We used the method to fashion five novel performance criteria for evaluating AMI methods. Last, we developed DOMINO, a novel AMI algorithm, that outperformed the other six algorithms in extensive testing on GE and GWAS data. Software is available at https://github.com/Shamir-Lab.

Introduction

The maturation of high-throughput technologies has led to an unprecedented abundance of omics studies. With the ever-increasing volume of publicly available genomic, transcriptomic and proteomic data (Perez-Riverol *et al*, 2019), it remains a challenge to uncover biological and biomedical insights out of it. As data accumulated over the last two decades strongly indicate that the functional organization of the cell is fundamentally modular, a leading approach to this challenge relies on biological networks, simplified yet solid mathematical abstractions of complex intra-cellular systems. In these networks, each node represents a cellular subunit (e.g. a gene or its protein product) and each edge represents a relationship between two subunits (e.g. a physical interaction between two proteins) (reviewed in (McGillivray *et al*, 2018)). A biological module is described as a connected subnetwork of - molecules that take part in a common biological process. As such, modules are regarded as functional building blocks of the cell (Hartwell *et al*, 1999; Alon, 2003; Barabási & Oltvai, 2004).

The challenge of identifying modules in biological networks, frequently referred to as *network-based module identification* or *community detection*, has yielded many computational methods (for a recent comparative study see (Choobdar *et al*, 2019)), and successfully identified molecular machineries that perform basic biological functions and underlie pathological phenotypes (Ideker & Sharan, 2008; Barabási *et al*, 2011). However, such analysis is limited as it is based on a static snapshot of an abstract universal cell provided by the network, while the state of the cell greatly varies under different physiological conditions. One very powerful way to overcome this limitation is by integrating the analysis of omics data and biological networks. This approach overlays molecular profiles (e.g., transcriptomic, genomic, proteomic or epigenomic profiles) on the network, by scoring nodes or weighting edges. This additional layer of condition-specific information is then used to detect modules that are relevant to the analyzed molecular profile (Mitra *et al*, 2013). A prominent class of such algorithms seek subnetworks that show a marked over-representation of accrued node scores (Ideker *et al*, 2002; Mitra *et al*, 2013; Reyna *et al*, 2020). Modules detected by such methods are often called '*active modules'*, and following this terminology we refer to nodes' scores as '*activity scores'* and to the task of detecting active modules using such scores as *Active Module Identification* (AMI). (The task is sometimes called community detection with node attributes (Yang *et al*, 2014) Hereafter, for brevity, where clear from the context, we refer to active modules reported by AMI methods simply as modules.

Modules detected by AMI algorithms are expected to capture context-specific molecular processes that correlate with the specific cellular state or phenotype that is probed by the analyzed omics profile (Mitra *et al*, 2013). Different AMI methods use different scoring metrics, objective functions and constraints. For example, activity scores may be binary or continuous, the objective function could

penalize for including low-scoring nodes, and constraints can limit the number of 'non-active' nodes in a module. While the metrics by which modules are scored may differ from one method to another, the activity scores are always derived from the data (e.g. $\log_2(fold - change\ of\ expression)$ for transcriptomic data). As the AMI problem has been proven to be NP-hard (Ideker *et al*, 2002), many heuristics were suggested for solving it (Mitra *et al*, 2013; Creixell *et al*, 2015).

Solutions reported by AMI methods comprise a set of active modules. A common downstream analysis is to ascribe each module some biological annotations that will point to the biological processes that it affects (Leiserson *et al*, 2015; Cerami *et al*, 2010; Barel & Herwig, 2018). This is most commonly done by testing enrichment of the modules for GO terms (The Gene Ontology Consortium, 2019). AMI solutions would ideally break down complex biological states into distinct functional modules, each mediating one or several highly related biological processes. For example, biological responses to genotoxic stress often comprises the concurrent activation and repression of multiple biological processes (e.g., DNA repair, cell-cycle arrest, apoptosis), each mediated by a single or a few dedicated signaling pathways (Kyriakis & Avruch, 2012; Ashcroft *et al*, 2000).

Another key advantage of AMI methods is the amplification of weak signals, where a reported active module comprises multiple nodes that individually have only marginal scores, but when considered in aggregate score significantly higher. This merit of AMI methods is especially critical for the functional interpretation of Genome-Wide Association Studies (GWASs) (Cowen *et al*, 2017; Carter *et al*, 2013). Numerous GWASs conducted over the last decade have demonstrated that the genetic component of complex diseases is highly polygenic (Khera *et al*, 2018; Musunuru & Kathiresan, 2019; Sullivan & Geschwind, 2019), affected by hundreds or thousands of genetic variants, the vast majority of which have only a very subtle effect. Therefore, most "risk SNPs" do not pass statistical significance when tested individually after correcting for multiple testing (Stringer *et al*, 2011; Boyle *et al*, 2017). This stresses the need for computational methods that consider multiple genetic elements together, to allow detection of biological pathways that carry high association signal. As a first step in this challenge, gene-level scores are inferred from the scores of the genetic variants that map to the same gene (Lamparter *et al*, 2016; de Leeuw *et al*, 2015). These gene scores then serve as activity scores by AMI methods for integrated analysis of GWAS data and biological networks. Recently, such analyses successfully elucidated novel process that are implicated in the pathogenesis of inflammatory bowel disease, Schizophrenia and Type-2 diabetes (Nakka *et al*, 2016; Chang *et al*, 2015; Fernández-Tajes *et al*, 2019).

In this study, we first aimed to systematically evaluate popular AMI algorithms across multiple gene expression (GE) and GWAS datasets based on enrichment of the called modules for GO terms. Remarkably, our analysis revealed that AMI algorithms often reported modules that showed

enrichment for a high number of GO terms even when run on permuted datasets. Moreover, some of the GO terms that were often enriched on permuted datasets were also enriched on the original dataset, indicating that AMI solutions frequently include modules that are not specific to the biological context measured by the analyzed omics dataset. To tackle this bias, we designed a procedure for validating the functional analysis of AMI solutions by comparing them to null distributions obtained on permuted datasets. We used the empirically validated set of GO terms to define novel metrics for evaluation of AMI algorithm results. Finally, we developed DOMINO (Discovery of active Modules In Networks using Omics) – a novel AMI method, and showed its advantage in comparison it to the previously developed algorithms.

Results

AMI algorithms suffer from a high rate of non-specific GO term enrichments

We set out to evaluate the performance of leading AMI algorithms. Our analysis included six algorithms – jActiveModules (Ideker *et al*, 2002) in two strategies: greedy and simulated annealing (abbreviated jAM_greedy and jAM_SA, respectively), BioNet (Beisser *et al*, 2010), HotNet2 (Leiserson *et al*, 2015), NetBox (Cerami *et al*, 2010) and KeyPathwayMiner (Baumbach *et al*, 2012) (abbreviated KPM). These algorithms were chosen based on their popularity, computational methodology and diversity of original application (e.g., gene expression data, somatic mutations) (**Table S1**). As we wished to test these algorithms extensively, we focused on those that had a working tool/codebase that can be executed in a stand-alone manner, have reasonable runtime and could be applied to different omics data types. Details on the execution procedure of each algorithm are available in the Appendix. We applied these algorithms to two types of data: (1) a set of ten gene-expression (GE) datasets of diverse biological physiologies (**Table S2**) where gene activity scores correspond to differential expression between test and control conditions, and (2) a set of ten GWAS datasets of diverse pathological conditions (**Table S3**) where gene activity scores correspond to genetic association with the trait (Methods). Note that for uniformity, we use the term activity also for the GWAS scores. In our analyses, we mainly used the Database of Interacting Proteins (DIP; (Xenarios *et al*, 2002)) as the underlying global network. Although the DIP network is relatively small - comprising about 3000 nodes and 5000 edges, in a recent benchmark analysis (Huang *et al*, 2018) it got the best normalized score on recovering literature-curated disease gene sets, making it ideal for multiple systematic executions.

First, applying the algorithms to the GE and GWAS datasets we observed that their solutions showed high variability in the number and size of active modules they detected (**Figure S1** and

Figure S2). On the GE datasets, jAM_SA tended to report a small number of very large modules while HotNet2 usually reported a high number of small modules (**Figure S1**). jAM_SA showed the same tendency for reporting large modules also on the GWAS datasets (**Figure S2**). Next, to functionally characterize the solutions obtained by the algorithms, we tested the modules for enriched GO terms using the hypergeometric (HG) test with the genes in the entire network as the background set. Specifically, we used GO terms from the Biological Process (BP) ontology, using only terms with 5-500 genes. To avoid potential bias caused by the underlying network and datasets, we excluded from each GO class genes that were included in it based on physical interaction, expression pattern, genetic interaction or mutant phenotype (GO evidence codes: IPI, IEP, IGI, IMP, HMP, HGI and HEP). Next, as part of our evaluation analysis, we applied the algorithms also on random datasets generated by permuting the original gene activity scores uniformly at random. Notably, we observed that modules detected on the permuted datasets, too, were frequently enriched for GO terms (**Figure 1A**) Moreover, different algorithms showed varying degree of overlap between the enriched terms obtained on real and permuted datasets (**Figure 1B**). These findings imply that many terms reported by AMI algorithms do not stem from the specific biological condition that was assayed in each dataset, but rather from other non-specific factors that bias the solution, such as the structure of the network, the methodology of the algorithm and the distribution of the activity scores.

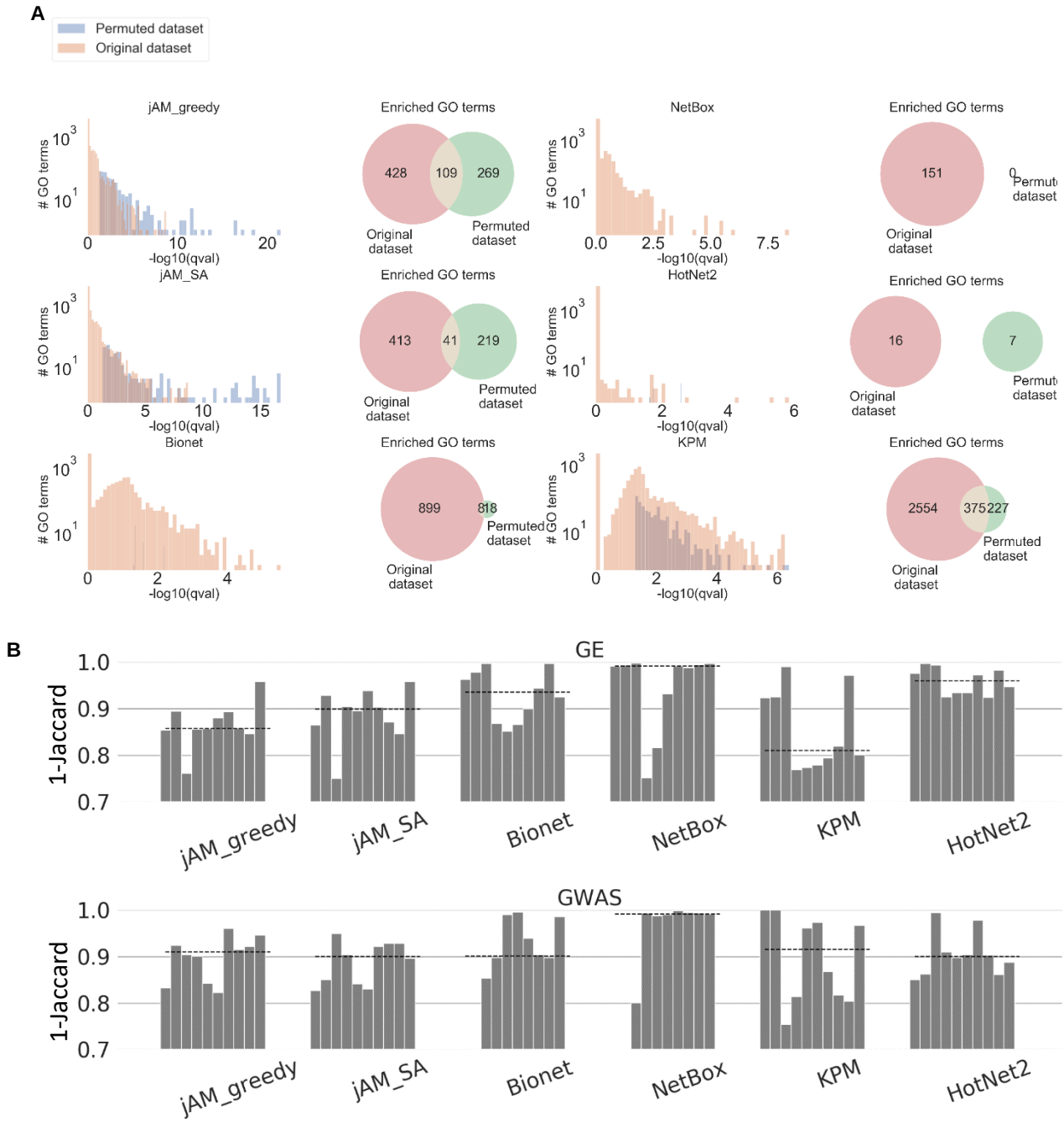

Figure 1. A. *Comparison of GO enrichment results obtained on the original CBX GE dataset and on one random permutation of the original gene activity scores of this dataset. The histograms show the distributions of GO enrichment scores obtained for the modules detected on the original and permuted datasets. The Venn diagrams show the overlap between the GO terms detected in the two solutions.* ***B****. Comparison of GO terms reported on the original and permuted datasets. We used 1-Jaccard score to measure the dissimilarity between the GO terms detected on the two datasets. Values closer to 1 indicate low similarity (that is, lower bias). Each bar shows, per algorithm, this measure on the ten datasets, averaged over 100 random permutations. Datasets are ordered from left to right as in Tables S2 and S3. Dashed lines show the median score.*

A permutation-based method for filtering false GO terms

The high overlap between sets of enriched GO terms obtained on real and permuted datasets indicates that the results of most AMI algorithms tested are highly susceptible to false calls that might lead to functional misinterpretation of the analyzed omics data. We looked for a way to filter out such non-specific terms while preserving the ones that are biologically meaningful in the context of the analyzed dataset. For this purpose, we developed a procedure called the EMpirical Pipeline (EMP). It works as follows: Given an AMI algorithm and a dataset, EMP permutes genes' activity scores in the dataset and executes the algorithm. For each module reported by the algorithm, it performs GO enrichment analysis. The overall reported enrichment score for each GO term is its maximal score over all the solution's modules (**Figure 2A**). The process is repeated many times (typically, in our analysis, 5,000 times), generating a background distribution per GO term (**Figure 2B**). Next, the algorithm and the enrichment analysis are run on the real (i.e. non-permuted) dataset (**Figure 2C**). Denoting the background CDF obtained for GO term t by F_t, the empirical significance of t with enrichment score s is $e(t) = 1 - F_t(s)$. EMP reports only terms t that passed the HG test (q-value ≤ 0.05 on the original data) and had empirical significance $e(t) \leq 0.05$ (**Figure 2D**). We call such terms *empirically validated GO terms (EV terms)*. In addition, for each AMI algorithm solution, we define the *Empirical-to-Hypergeometric Ratio* (EHR) as the fraction of EV terms out of the GO terms that passed the HG test (**Figure 2E-F**).

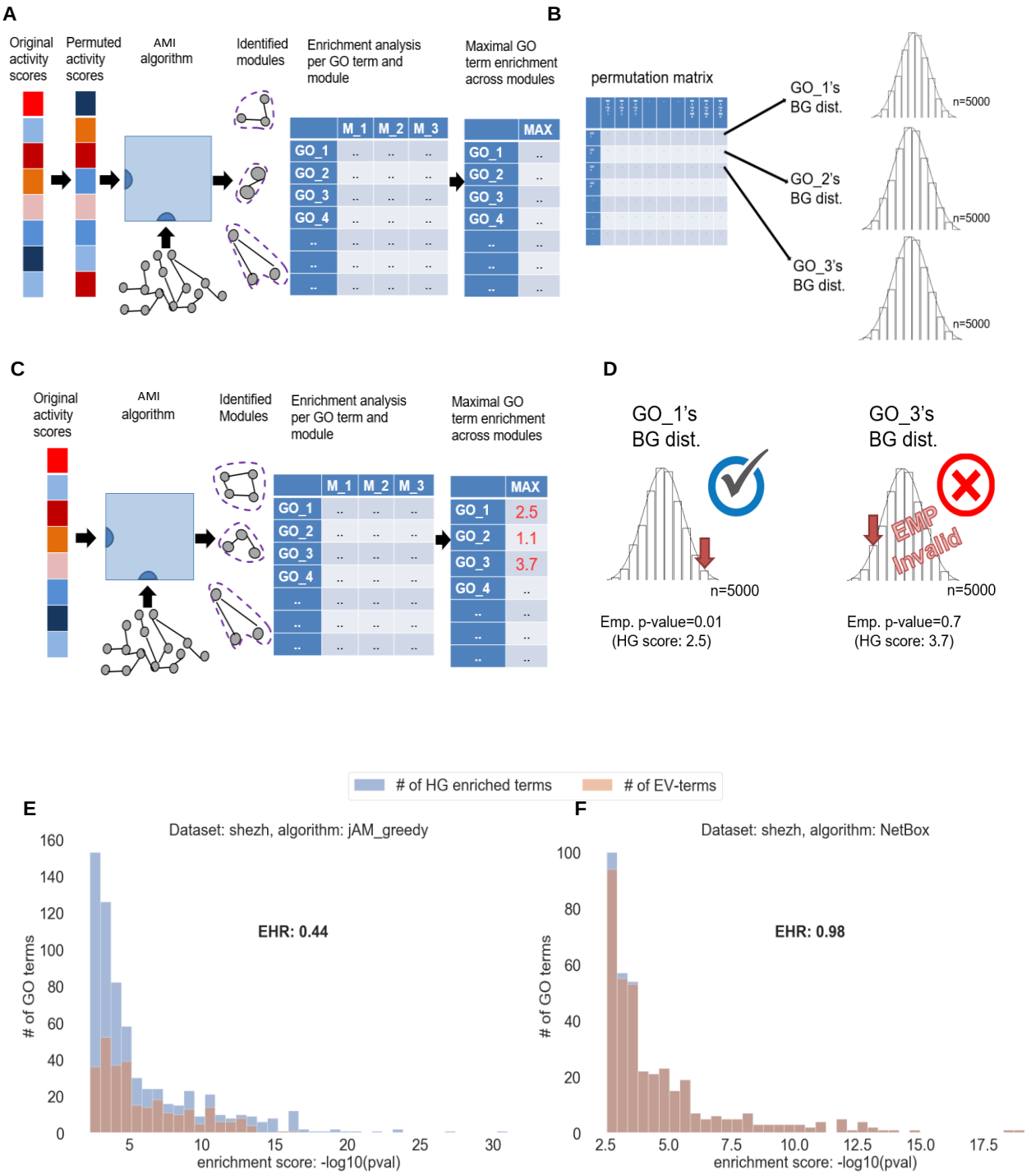

Figure 2. *Overview of the EMpirical Pipeline (EMP) procedure.* ***A.*** *The AMI algorithm and the GO enrichment analysis are applied on multiple (typically, n=5000) permuted activity scores.* ***B.*** *A null distribution of enrichment scores (-log$_{10}$Pval) is produced per GO term.* ***C.*** *The AMI algorithm is applied to the original (un-permuted) activity scores, to calculate the real GO enrichment scores.* ***D.*** *For each GO term, the real enrichment score is compared to its corresponding empirical null distribution to derive an empirical score. In this example, GO_3 passed the HG test, but failed the empirical test and thus was filtered out.* ***E, F.*** *Distributions of HG enrichment scores for all the GO terms that passed the HG test and for the subset of the EV terms obtained on the shEZH expression dataset by jActiveModules with greedy strategy (E) and NetBox (F). EHR measures the ratio between the number of EV terms and the number of GO terms that passed the HG test. The high EHR score obtained by NetBox (close to 1.0) demonstrates the advantage of this alg. in avoiding false terms.*

The DOMINO algorithm

Our results demonstrated that popular AMI algorithms often suffer from high rates of false GO terms. While the EMP method is a potent way for filtering out non-specific GO term calls from AMI solutions, this procedure is computationally demanding, as it requires several thousands of permutation runs. In our analyses, using a 44-cores server, EMP runs typically took several days to complete, depending on the algorithm and the dataset. Seeking a more frugal alternative that can be used on a desktop computer, we developed a novel AMI algorithm called DOMINO (*Discovery of active Modules In Networks using Omics*), with the goal of producing highly confident active modules characterized by high validation rate (that is, high EHR values).

DOMINO receives as input a set of genes flagged as the *active genes* in a dataset (e.g., the set of genes that in the analyzed transcriptomic dataset passed a test for differential expression) and a network of gene interactions, aiming to find disjoint connected subnetworks in which the active genes are over-represented. DOMINO has four main steps:

0. Partition the network into disjoint, highly connected subnetworks (*slices)*.
1. Detect *relevant slices* where active genes are over-represented
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 over-represented by active genes.

Step 0 - Partitioning the network into slices. This time-consuming preprocessing step is done once per network (and reused for any analyzed dataset). In this step, the network is split into disjoint subnetworks called slices. Splitting is done using a variant of the Louvain modularity algorithm (Blondel *et al*, 2008) (Methods). Each connected component in the final network that has more than three nodes is defined as a *slice* (**Figure 3A**).

Step 1 - Detecting relevant slices. Each slice that contains more active nodes than a certain threshold (see Methods) is tested for active nodes over-representation using the Hypergeometric (HG) test, correcting the p-values for multiple testing using FDR (Benjamini & Hochberg, 1995). In this initial step we use a lenient threshold of q-values < 0.3 to accept a slice as a *relevant* one (**Figure 3B**).

Step 2a - Refining the relevant slices into sub-slices. From each slice**,** the algorithm extracts a single connected component that captures most of the activity signal. The single component is obtained by solving the Prize Collecting Steiner Tree (PCST) problem (Johnson *et al*, 2000) (Methods). The resulting subgraph is called a *sub-slice* (**Figure 3C**).

Step 2b - Partitioning sub-slices into putative active modules. Each sub-slice that is not over-represented by active nodes and has more than 10 nodes is partitioned using the Newman-Girvan algorithm (Methods). The resulting parts, as well as all the sub-slices from step 2a of ≤ 10 nodes, are called *putative active modules* (**Figure 3D**).

Step 3 - Identifying the final set of active modules. Each putative active module is tested for over-representation of active nodes using the HG test. In this step, we correct for multiple testing using the more stringent Bonferroni correction. Those with q-value < 0.05 are reported as the final active modules (**Figure 3E**).

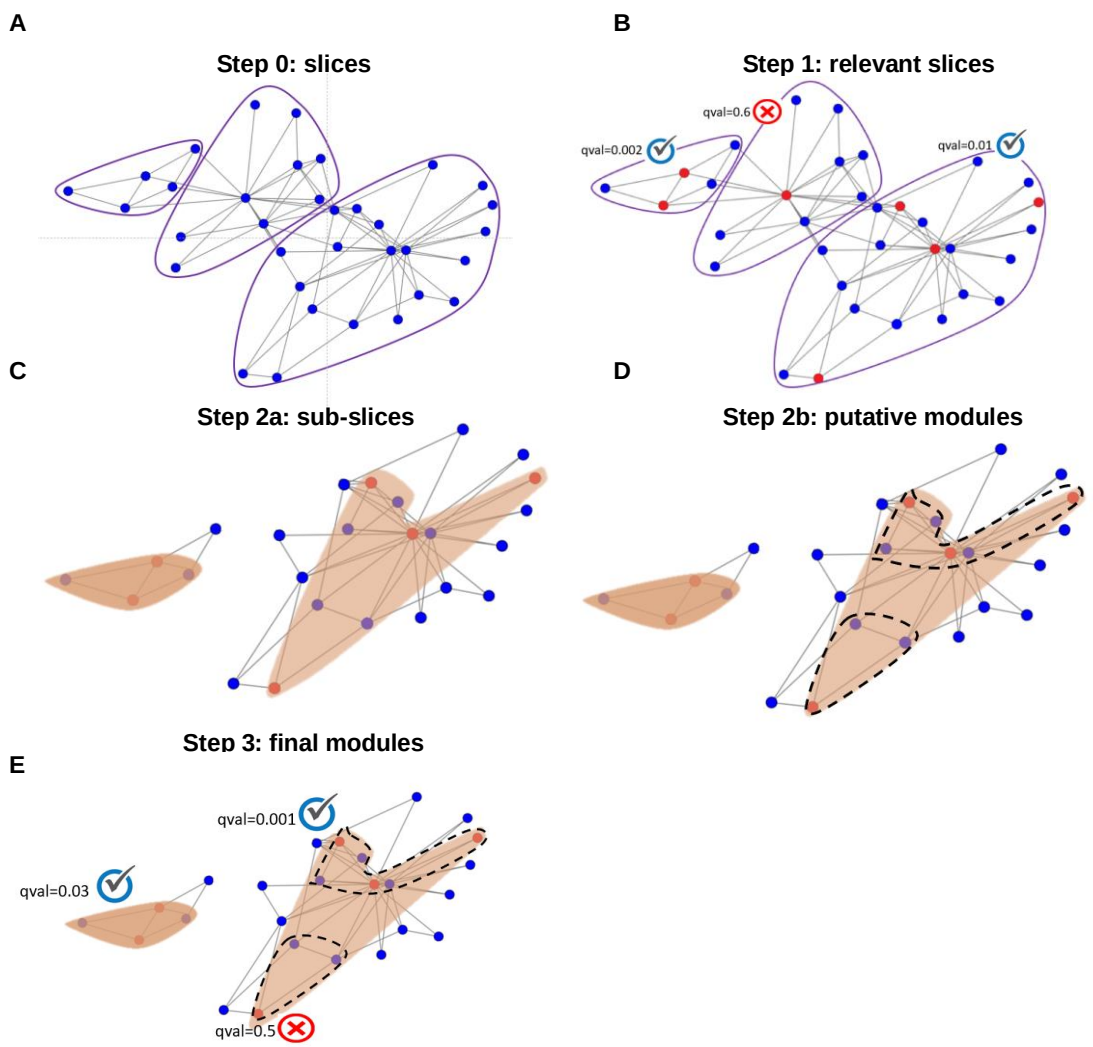

Figure 3. *Schematic illustration of DOMINO.* ***A.*** *The global network is partitioned by the Louvain modularity algorithm into slices (encompassed in purple line).* ***B.*** *A slice is considered relevant if it passes a moderate HG test for enrichment for active nodes (FDR $q \leq 0.3$).* ***C.*** *For each relevant slice the most active sub-slice is identified using PCST (red areas).* ***D.*** *Sub-slices are further partitioned into putative active modules using the Newman-Girvan (NG) modularity algorithm.* ***E.*** *Each putative active module that passes a strict over-representation test for active nodes (Bonferroni $qval \leq 0.05$) is included in the final solution.*

Systematic evaluation of AMI algorithms on gene-expression and GWAS datasets

We next carried out a comparative evaluation of DOMINO and the six AMI algorithms described above over the same ten GE and ten GWAS datasets. This evaluation task is challenging as there are no "gold-standard" solutions to benchmark against. To address this difficulty, we introduce five novel scores as evaluation criteria of AMI algorithms. These scores are based on the EMP method and the GO terms that pass this empirical validation procedure. The scores are described in Methods and the results on all algorithms are summarized in **Figures 4-6.**

(a) EHR (*Empirical-to-Hypergeometric Ratio*). EHR summarizes the tendency of an AMI algorithm to capture biological signals that are specific to the analyzed omics dataset, i.e. GO terms that are enriched in modules found on the real but not on permuted data. EHR has values between 0 to 1, with higher values indicating better performance. In our evaluation, DOMINO and NetBox scored highest on EHR. In both GE and GWAS datasets, DOMINO performed best with an average above 0.8. (**Figure 4A-B**). Importantly, these high EHR levels were not a result of reporting low number of terms: DOMINO reported on average more enriched GO terms than the other algorithms, except NetBox on GE datasets (**Figure 4C-D**).

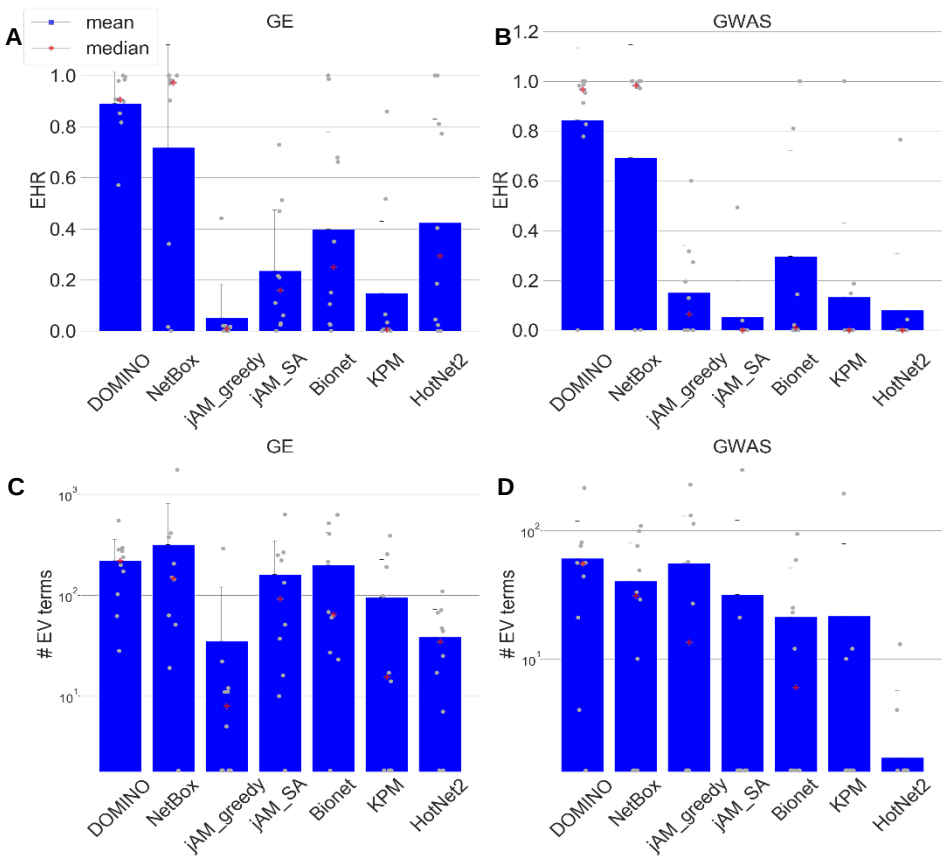

***Figure 4**. EHR and number of reported terms. **A**. Average EHR for the GE datasets. **B.** Average EHR for the GWAS datasets. **C**. The number of EV terms reported for the GE datasets. **D**. The number of EV terms reported for the GWAS datasets. The grey dots indicate results for each dataset. Error bars indicate SD across datasets.*

(b) **Module-level EHR (mEHR).** While the EHR characterizes a solution as a whole by considering the union of GO terms enriched on any module, biological insights are often obtained by functionally characterizing each module individually. We therefore next evaluated the EHR of each module separately. Specifically, for each module, we calculated the fraction of its EV terms out of the HG terms detected on it (Methods). Modules with high mEHR score are the biologically most relevant ones, in the context of the analyzed omics dataset, while modules with low mEHR mostly capture non-specific signals. The comparison between mEHR scores obtained by the different AMI algorithms is summarized in **Figure 5A**. Notably, solutions can have a broad range of mEHR scores (for example, in NetBox solution on the IEM dataset, the best module has mEHR=0.78 while the poorest has mEHR=0). To summarize the results over multiple modules, we averaged the k top scoring modules (from k=1 to 20; **Figure 6A**). In this criterion, DOMINO scored highest, followed by NetBox. The results for GWAS datasets are shown in **Figure S3** and **Figure S4A**.
Furthermore, the EMP procedure enhances the functional interpretation of each module by distinguishing between its enriched GO terms that are specific to the real data (i.e., the EV terms) and those that are recurrently enriched also on permuted ones. This utility of EMP is demonstrated, as one example, on a module detected by NetBox on the ROR GE dataset (**Figure 5B**). This study examined roles of the ROR2 receptor in breast cancer progression, and the GO terms that passed EMP validation are highly relevant for this process (e.g., GO terms related to steroid hormone mediated signaling pathways). In contrast, GO terms that failed passing this validation procedure represent less specific processes (e.g., "DNA-templated transcription, initiation").

(c) **Biological richness.** This criterion aims to measure the diversity of biological processes captured by a solution. Our underlying assumption here is that biological systems are complex and their responses to triggers typically involve the concurrent modulation of multiple biological processes. For example, genotoxic stress concurrently activates DNA damage repair mechanisms and apoptotic pathways and suppresses cell-cycle progression. However, merely counting the number of EV terms of a solution would not faithfully reflect its biological richness because of the high redundancy between GO terms. This redundancy stems from overlaps between sets of genes assigned to different GO terms, mainly due to the hierarchical structure of the ontology. We therefore used REVIGO (Supek *et al*, 2011) to derive a non-redundant set of GO terms based on semantic similarity scores (Lord *et al*, 2003; Resnik, 1999). We defined the *biological richness score* of a solution as the number of its non-redundant EV terms (Methods). The results in **Figure 6B** show that on the GE datasets, DOMINO and NetBox performed best. On the GWAS datasets, DOMINO performed best (**Figure S4B**). Note that the interpretation of this criterion is condition dependent. High biological richness can be revealing or an indication of spurious results.

(d) **Intra-module homogeneity**. While high biological diversity (richness) is desirable at the solution level, each individual module should ideally capture only a few related biological processes. Solutions in which the entire response is partitioned into separate modules where each represents a distinct biological process are easier to interpret biologically and are preferred over solutions with larger modules that represent several composite processes. To reflect this preference, we introduced the *intra-module homogeneity score,* which quantifies how functionally homogeneous the EV terms captured by each module are (Methods; **Figure S5**). For each solution, we take the average score of its modules. On the GE datasets, NetBox performed best (**Figure 6C**). On the GWAS datasets, DOMINO scored highest (**Figure S4C**).

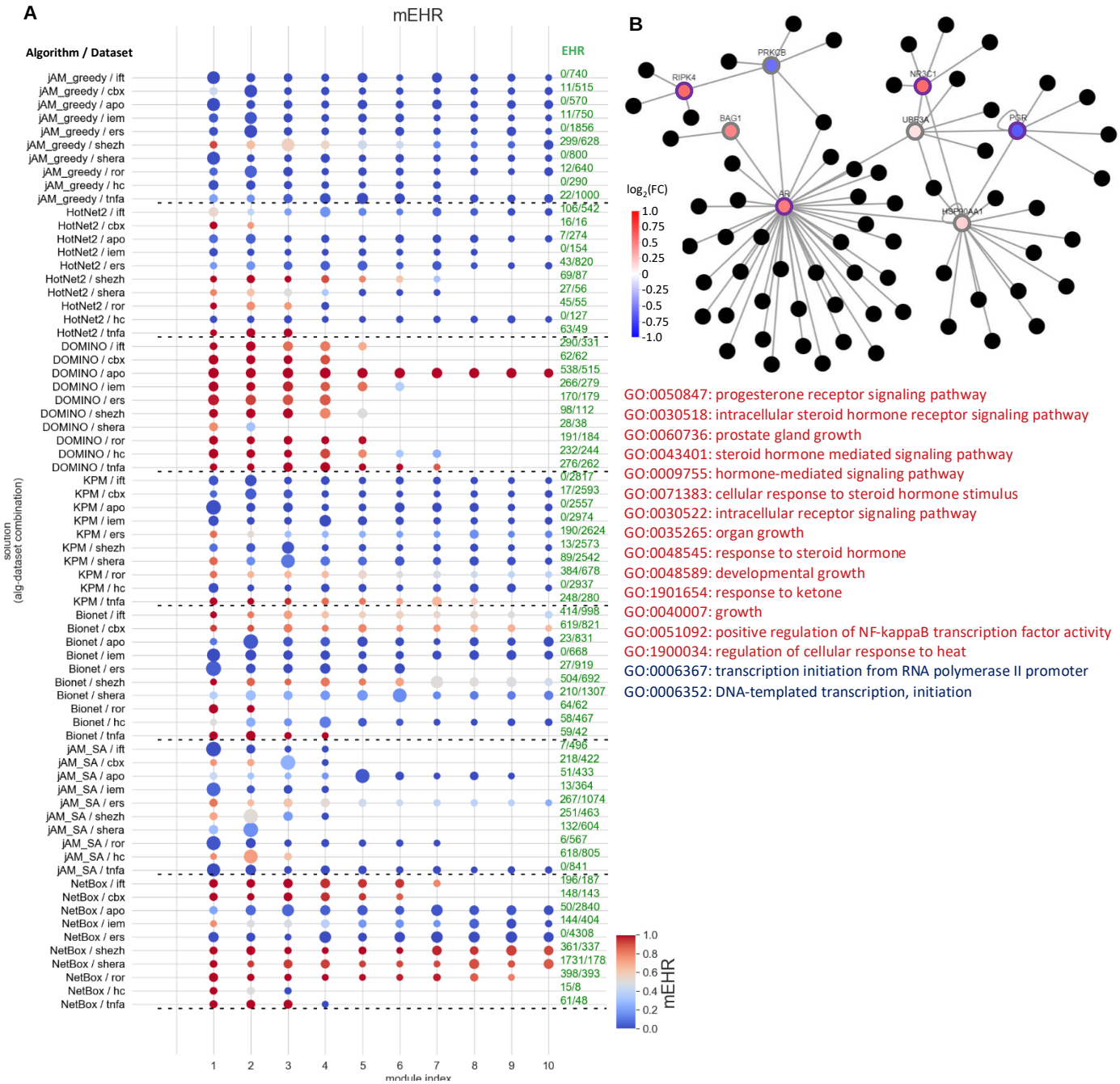

Figure 5*. AMI algorithms evaluated by the module-level EHR (mEHR) criterion on GE datasets.* ***A****. mEHR scores for each algorithm and dataset. Up to ten top k modules are shown per datasets, ranked by their mEHR. Dot size represents module's size.* ***B.*** *An example of a module from the solution reported by NetBox on the ROR dataset (mEHR=0.88). The nodes' color indicates expression fold change (log scale) in the dataset. The black nodes are the network neighbors of the module's nodes. Nodes with purple border have significant activity scores (that is, significant differential expression;* $qval < 0.05$*). The EV terms for this module are shown in red and those that did not pass the empirical validation in blue.*

(e) **Robustness**. This criterion measures how robust an algorithm's results are to subsampling of the data. It compares the EV-terms obtained on the original dataset with those obtained on randomly subsampled datasets. Running 100 subsampling iterations and using the EV terms found on the original dataset as the gold-standard GO terms, we compute AUPR and average F1 scores for each solution (Methods). On the GE datasets, solutions produced by DOMINO and NetBox showed the

highest robustness over a wide range of subsampling fractions (**Figure 6D-E**). On the GWAS datasets, DOMINO's solutions scored highest (**Figure S4D-E**).

A breakdown of the evaluation criteria by their properties is shown in **Figure 6F.**

Table 1 summarizes the benchmark results. DOMINO performed best on the GE datasets in five of the six criteria, and in all six criteria on the GWAS datasets. NetBox came second, performed best or timed for best in two criteria and second in the rest.

In addition, DOMINO ran much faster than the other algorithms, taking 1-3 orders of magnitude less time (**Tables S4-S6**). This speed allows to run DOMINO and the EMP procedure in reasonable time on a desktop machine. We also noticed that runtimes were markedly shorter on permuted datasets, probably since after permutation activity scores are spread more uniformly across the network, producing smaller modules.

Analysis of large-scale networks

Our benchmark used the highly informative but relatively small DIP network (~3k nodes and ~5k edges) in order to allow systematic evaluation of multiple AMI methods on many datasets. Yet, much larger networks are currently available. To examine how DOMINO performs on larger network, we applied it on two state-of-the-art human networks: the HuRI network (8,272 nodes and 52,549 edges) (Luck *et al*, 2020) and STRING (with >18K nodes and >11M edges) (Szklarczyk *et al*, 2017). We also tested NetBox, the second-best performer in our benchmark, on these larger networks. The edges of the STRING network are weighted with a confidence score, ranging from 0 to 1000, based on the strength of their supporting evidence. To make the execution of the EMP feasible, we kept only edges with score > 900. The resulting network had 11,972 nodes and 243,385 edges. Setting a running time limit of 5 hrs, DOMINO completed all runs on both the HuRI and STRING networks, while NetBox did so on these two network for 8/10 and 2/10 of the GE datasets and for 9/10 and 9/10 of the GWAS datasets, respectively. Notably, DOMINO consistently outperformed NetBox on 23 of the 24 criteria on both networks and both types of datasets (**Table 2).** DOMINO also performed overall better when using different HG q-value thresholds (**External Table 1)**. Taken together, these results demonstrate that DOMINO maintains high performance when applied to large networks as well.

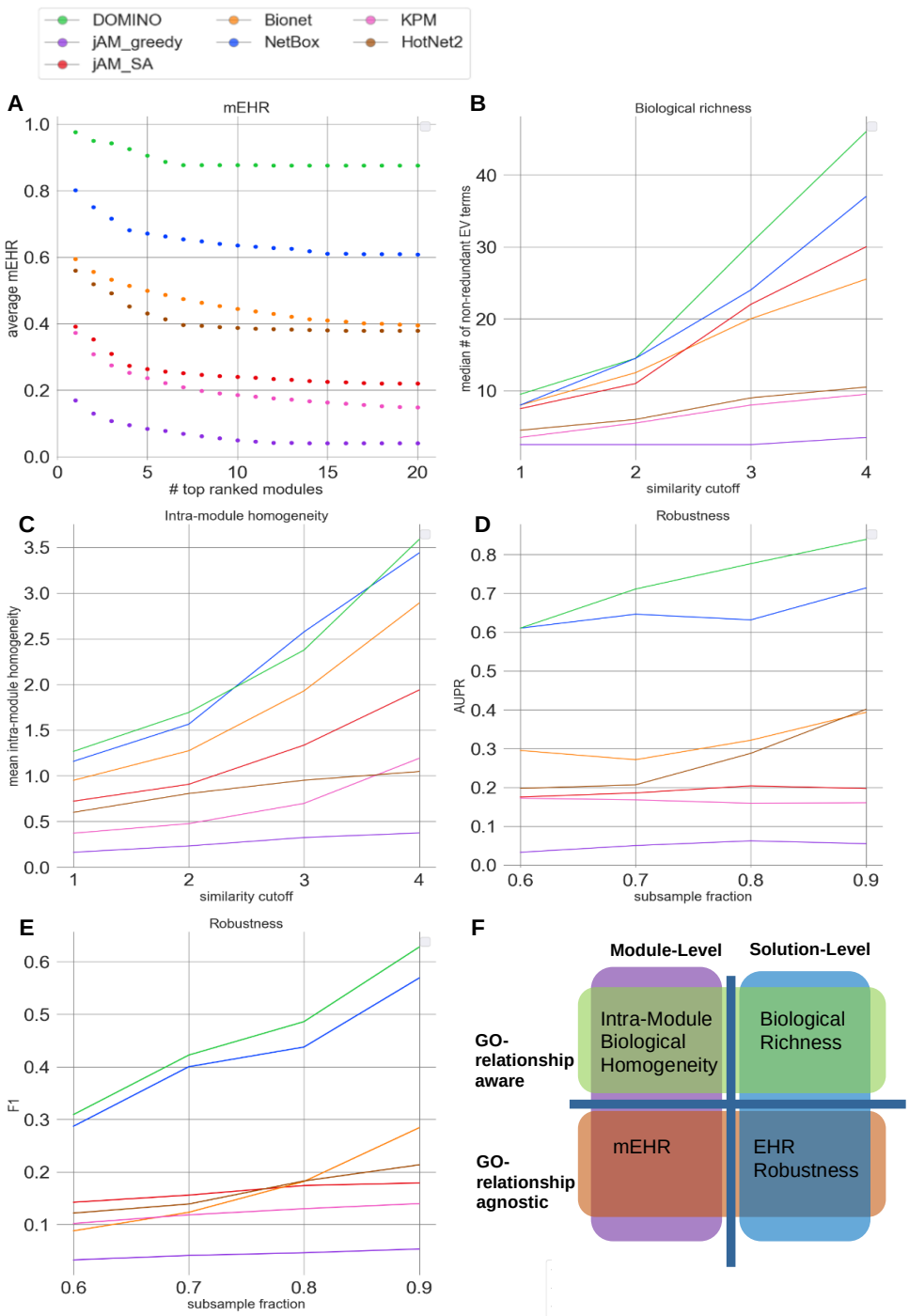

Figure 6. *Evaluation results for the GE datasets.* ***A.*** *Module-level EHR scores. The plot shows the average mEHR score of the* k *top modules, as a function of* k *in each solution. Modules were ranked, for each solution, by their mEHR score. Then, for each solution with* n *modules we calculate the average mEHR of the top* $\min(k, n)$ *modules. Finally, we averaged the results and got the average mEHR of an algorithm.* ***B.*** *Biological richness. The plot shows the median number of non-redundant terms (richness score) as a function of the Resnik similarity cutoff (Methods).* ***C.*** *Intra-module homogeneity scores as a function of the similarity cutoff.* ***D.*** *Robustness measured by the average AUPR over the datasets, shown as a function of the subsampling fraction.* ***E.*** *Robustness measured by the average F1 over the datasets shown as a function of the subsample fraction. In D and E, 100 samples were drawn and averaged for each dataset and subsampling fraction.* ***F.*** *A breakdown of the evaluation criteria by their properties. Richness, EHR and robustness score solutions based only on the whole set of the reported GO terms, without taking into account the results for individual modules. In contrast, mEHR and intra-module homogeneity score solutions in a module-aware fashion. From another perspective, biological richness and intra-module homogeneity consider the relations among the reported GO terms, while EHR, mEHR, and robustness do not.*

Analyzing the network contribution to non-specific GO enrichment bias

Understanding the causes for over-reporting of enriched GO terms is a key question that arises from our study. One prominent potential cause is the network topology, as the modules sought are connected subnetworks, and connectivity also reflects functional similarity. To explore the contribution of the network to the GO enrichment bias, we next detected modules in the underlying DIP network without use of any condition-specific activity profile, and identified the GO terms these modules were enriched for. Overall, 2,450 out of 6,573 (37%) BP GO terms were detected by this analysis, and we refer to them as *net-terms*. Notably, while net-terms were in general highly over-represented among the GO terms reported by AMI solutions (**Figure 7A-B**), these terms did not show higher rejection rate by the EMP procedure than the other BP GO terms (**Figure 7C-D**) (see **Appendix** for full details of this analysis). These results show that simple exclusion of GO net-terms from AMI analyses cannot replace the empirical validation to lessen over-reporting of non-specific GO terms. Better understanding of the bias origin is required.

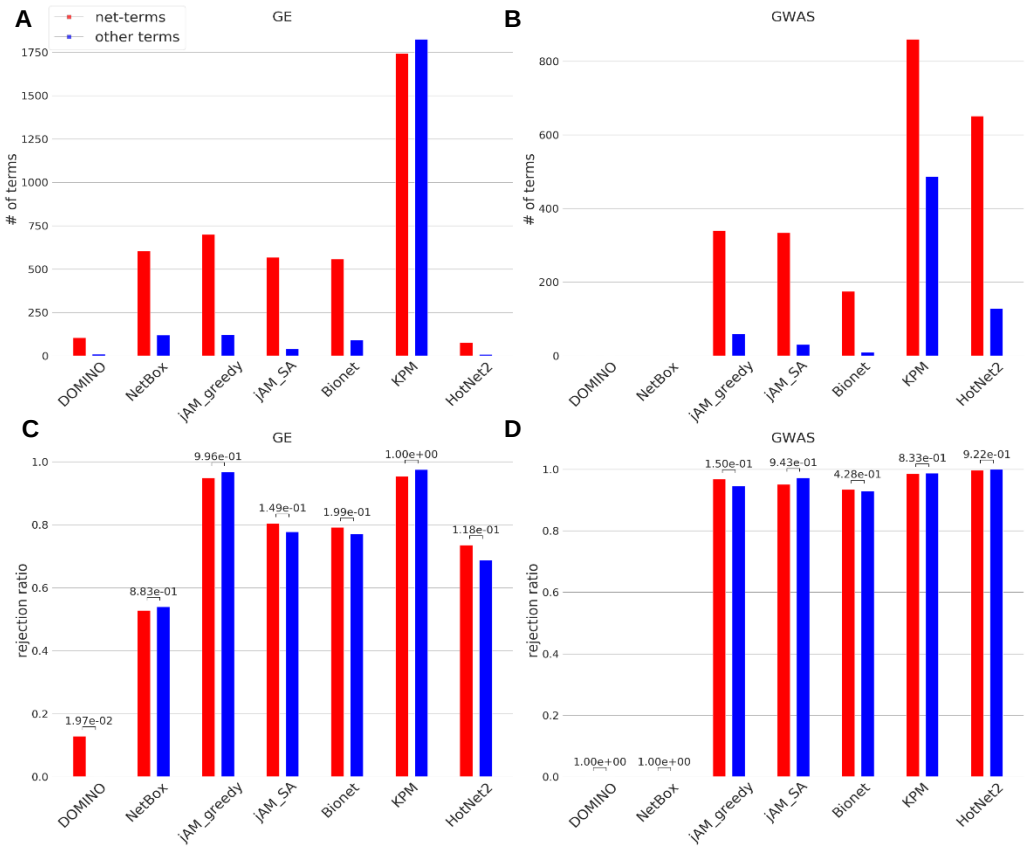

Figure 7. Comparison of the GO terms identified by each benchmarked algorithm to the terms identified by using the network only (net-terms). **A-B.** Average number of net-terms and other terms. Only terms reported in four datasets or more were included. Note that no terms were reported in more than four GWAS datasets by DOMINO and NetBox, which obtained the best overall results (Table 1). **A**. GE; **B**. GWAS. **C-D**. Average rejection ratio of net-terms and other terms. The rejection ratio of a GO term in an algorithm is the fraction of datasets in which the term appeared as significant but was not empirically validated (see Appendix). **C**. GE; **D**. GWAS. P-values were calculated by comparing the rejection ratios between net-terms and other terms using Mann-Whitney U one-sided test.

Discussion

The fundamental task of active module identification (AMI) algorithms is to identify active modules in an underlying network based on context-specific gene activity profiles. The comparison of AMI algorithms is challenging due to the complex nature of the solutions they produce. Algorithms differ markedly in the number, size, and properties of the modules they detect. Although AMI algorithms have been extensively used for almost two decades (Ideker *et al*, 2002), there is no accepted community benchmark for this task and no consensus on evaluation criteria. As active modules are often used to characterize the biological processes that are most relevant in the context of the profiled activity, we analyzed the solutions produced by AMI algorithms from the perspective of enrichment for GO terms annotating biological processes.

Previous works reported that the scheme used by the popular jActiveModule algorithm to score active modules is biased towards large modules, and suggested ways to alleviate this bias (Nikolayeva *et al*, 2018) (Reyna *et al*, 2020). Our study reports on a different bias that is prevalent in AMI solutions: their tendency to report non-specific GO terms. Early on in our analysis, we observed that many enriched GO terms also appear on permuted datasets, suggesting that such enrichment stems from some proprieties of the network, algorithm or the data that bias the results. To overcome this bias, we developed the EMP procedure, which empirically calibrates the enrichment scores and filters out non-specific terms. This procedure can be applied to any AMI algorithm.

To exemplify their merits, studies that present a novel AMI method usually report a collection of enriched gene sets (e.g., GO terms or pathways) obtained on the algorithm's solution and are biologically relevant to the analyzed condition. While this approach is valid for demonstrating capabilities of an algorithm, it is problematic for a systematic evaluation of algorithms, due lack of gold-standard bias-free set of biologically relevant GO terms for a given condition. An additional difficulty is the hierarchical structure of GO ontology. A previous benchmark of AMI algorithms used as an evaluation criterion the fold enrichment of the output genes using a single set of biologically relevant genes (He *et al*, 2017). In our work, we defined five novel evaluation criteria based on the GO terms enriched in a solution, each emphasizing a different aspect of the solution (**Figure 6F**).

We used these criteria to benchmark six popular AMI algorithms and DOMINO, a novel algorithm we developed, on ten GE and ten GWAS datasets, which collectively cover a very wide spectrum of biological conditions. Overall, DOMINO performed best, indicating its ability to produce "clean", stable and concise modules. NetBox also scored high in our evaluation. Interestingly, both DOMINO and NetBox use binary gene activity scores. One may expect that binarizing measured activity scores could degrade relevant biological signals. However, at least on our benchmark, binarizing the data

helped in reducing noise and detecting modules that are specifically relevant for the analyzed conditions. Further study of this observation is needed.

Notably, the algorithms that we tested substantially differ in their empirical validation rates. Some algorithms produced solutions with very low EHR (<0.5), and therefore running the EMP on them was critical. While empirical correction is desirable and adds confidence to the reported results, it is computationally highly demanding even with a relatively small network such as DIP. Naturally, using larger networks makes this procedure even slower (**Tables S4-S6**). A notable advantage of DOMINO is the high validation rates: on our benchmark its average EHR and mEHR were above 0.84, suggesting that DOMINO can be confidently run without empirical validation when computational resources are limited.

A common caveat in any report comparing a novel method to extant ones is that the new method may be better tuned to the data than the other methods. This may introduce a bias in the reported results. In our case, we could not tune each of the other AMI methods due to the long running time of EMP. Community efforts like the DREAM challenges (Choobdar *et al*, 2019) help reduce potential bias by allowing authors to calibrate their own methods on a common set of test datasets. To enable additional testing, the code of DOMINO, EMP and the evaluation criteria is freely available at https://github.com/Shamir-Lab/.

In summary, in this study we (1) reported on a highly prevalent bias in popular AMI algorithms, which leads to non-specific calls of enriched GO terms, (2) developed a procedure to allow for the correction of this bias, (3) introduced novel criteria for evaluation of AMI solutions, and (4) developed DOMINO – a novel AMI algorithm with low rate of non-specific calls and better performance across most of the criteria.

***Table 1**. Summary of the benchmark analysis.*

Alg.	EHR	mEHR*	Robustness (F1)†	Robustness (AUPR)†	Biological Richness#	Intra-Module Homogeneity#
	GE datasets					
jAM_greedy	0.052±0.137	0.049	0.046±0.117	0.062±0.123	2.5±9.837	0.325±0.748
jAM_SA	0.236±0.25	0.24	0.174±0.188	0.204±0.229	22±18.02	1.335±1.097
Bionet	0.398±0.4	0.444	0.182±0.15	0.321±0.322	20±23.819	1.929±1.302
NetBox	0.719±0.425	0.635	0.438±0.266	0.632±0.417	24±32.301	**2.575±1.453**
KPM	0.149±0.296	0.185	0.13±0.241	0.159±0.297	8±15.621	0.698±0.983
DOMINO	**0.891±0.129**	**0.876**	**0.486±0.192**	**0.776±0.174**	**30.5+12.69**	2.376±0.754
HotNet2	0.424±0.429	0.387	0.183±0.158	0.288±0.23	9±6.004	0.951±0.791
	GWAS datasets					
jAM_greedy	0.151±0.2	0.131	0.105±0.138	0.125±0.171	3.5±15.481	1.165±1.287
jAM_SA	0.053±0.155	0.033	0.046±0.134	0.058±0.173	0±11.83	0.327±0.736
Bionet	0.298±0.466	0.323	0.138±0.21	0.267±0.407	2±8.634	0.857±1.211
NetBox	0.694±0.497	0.782	0.335±0.298	0.617±0.456	**11±8.485**	1.172±1.338
KPM	0.134±0.312	0.155	0.144±28	0.16±0.324	0±9.073	0.438±0.719
DOMINO	**0.844±0.307**	**0.884**	**0.452±0.268**	**0.673±0.291**	**11±9.55**	**2.085±1.61**
HotNet2	0.081±0.241	0.033	0.016±0.037	0.061±0.183	0±0.843	0.036±0.115

Per algorithm, average score over the ten datasets is shown. Best score in each criterion is in bold.
*Results are average over the top 10 modules
†Results shown for subsampling fraction=0.8
Results shown for Resnik cutoff=3

***Table 2.** Performance of DOMINO and NetBox on the larger networks.*

Network	Alg.	EHR	mEHR*	Robustness (F1)†	Robustness (AUPR)†	Biological Richness#	Intra-Module Homogeneity#
		GE datasets					
HURI	**NetBox**	0.505±0.482	0.435	0.223±0.228	0.458±0.416	**10±25.738**	1.084±1.694
	DOMINO	**0.881±0.313**	**0.536**	**0.3±0.289**	**0.642±0.38**	6.5±17.515	**1.354±1.48**
STRING	**NetBox**	0.18±0.38	0.185	0.144±0.305	0.177±0.374	0±39.314	0.477±1.019
	DOMINO	**0.939±0.046**	**0.892**	**0.547±0.282**	**0.788±0.285**	**43±43.818**	**2.326±0.687**
		GWAS datasets					
HURI	**NetBox**	0.3±0.483	0.233	0.08±0.144	0.234±0.399	0±3.9	0.165±0.371
	DOMINO	**0.939±0.585**	**0.356**	**0.547±0.251**	**0.506±0.447**	**4.5±9.878**	**2.445±0.533**
STRING	**NetBox**	0.425±0.438	0.372	0.328±0.342	0.422±0.438	12±27.683	1.392±1.371
	DOMINO	**0.692±0.371**	**0.796**	**0.389±0.231**	**0.532±0.346**	**13±18.093**	**1.986±1.629**

Per algorithm, average score over the ten datasets is shown. Best score in each criterion is in bold.
*Results are average over the top 10 modules
†Results shown for subsampling fraction=0.8
Results shown for Resnik cutoff=3

Methods:

1. The Louvain algorithm in DOMINO

The Louvain algorithm is a fast community detection method for large network (Blondel *et al*, 2008) . This method aims to optimize an objective function by iteratively moving nodes between community to improve the objective function and fusing together the nodes of each community. In our benchmark we used a variant (Lambiotte *et al*, 2008) that incorporates a resolution parameter denoted r, which we set to 0.15.

2. Threshold for testing relevant slices

Slices that contain only a few active nodes are unlikely to be relevant. Testing multiple such slices would diminish the significance of genuine relevant slices. Therefore, we test for relevance only slices that satisfy either

$$\frac{\#active\ nodes\ in\ slice}{\#active\ nodes\ in\ network} \geq 0.1\ .$$

or

$$\frac{\#active\ nodes\ in\ slice}{\#\ nodes\ in\ slice} \geq \alpha.$$

where

$$\alpha = min(0.7, \frac{\#\ active\ nodes\ in\ network}{\#\ nodes\ in\ network} * (1 + \frac{100}{\sqrt{\#\ nodes\ in\ network}})$$

3. The PCST application in DOMINO

In PCST (Johnson *et al*, 2000), 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 the linear threshold model (Kempe *et al*, 2015). 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 $p(v) = \beta^l$, where $\beta = max(0, 1 - 3 * \frac{\#\ active\ nodes\ in\ network}{\#\ of\ active\ nodes\ in\ network}$. We define the penalty of edge e as $c(e) = 0$ if it is connected to an active node, and $c(e) = 1 - \epsilon$ otherwise (we used $\epsilon = 10^{-4}$).

PCST is NP-hard but good heuristics are available. In DOMINO we used FAST-PCST (Hegde *et al*, 2014). The resulting subgraph obtained by solving PCST on each slice is called its sub-slice. See **Figure 3C**.

4. The Newman-Girvan algorithm in DOMINO

The Newman-Girvan (NG) algorithm is a community detection method (Girvan & Newman, 2002). This method iteratively removes edges using the Betweenness-centrality metric for edges and recomputes the modularity score for each intermediate graph. Let M_i be the modularity score for the graph in iteration i. The process continues until a stopping criterion is met. The stopping criterion we used in step (2b) is $\frac{log\ (\#\ of\ nodes\ in\ sub-slice)}{log\ (\#\ of\ nodes\ in\ network)} \leq M_i$..

5. Derivation of p-values and q-values for the GE and GWAS datasets

For the GE datasets, we calculated p-values for differential expression between test and control conditions using edgeR (Robinson *et al*, 2010) for RNA-seq and student t-test for microarray datasets. We computed q-values using Benjamini-Hochberg FDR method (Benjamini & Hochberg, 1995). For GWAS, we used SNP-level p-values for association with the analyzed trait to derive gene-level association p-values using PASCAL (Lamparter *et al*, 2016), using the sum chi-square option and flanks of 50k bps around genes. We computed q-values using Benjamini-Hochberg FDR method (Benjamini & Hochberg, 1995).

6. Criteria for evaluating AMI solutions

We defined five novel criteria to allow systematic evaluation of solutions provided by AMI algorithms. For a specific solution we considered the list of BP GO terms that passed the HG enrichment test (HG terms) and the terms that passed the EMP validation procedure (EV terms).

Solution-Level Criteria

a) **Empirical to Hypergeometric Ratio (EHR)**. We define the *Empirical-to-Hypergeometric Ratio* (EHR) as the ratio between the number of EV-terms and reported HG terms. EHR summarizes the tendency of an algorithm to report non-specific GO terms, with values close to 1.0 indicating good solutions while values close to 0 indicating poor ones. EHR reflects the precision (true positive rate) of a solution.

b) **Biological Richness.** This criterion quantifies the biological information collectively captured by the solution's EV-terms. As there is high redundancy among GO terms - mainly due to the hierarchical structure of the GO ontology - we use the method implemented in REVIGO (Supek

et al, 2011) to derive a non-redundant set of EV terms. This method is based on a similarity matrix of GO terms, which is generated using Resnik similarity score (Resnik, 1999). The *biological richness score* is defined as the number of non-redundant EV terms in a solution. We calculated this measure using different similarity cutoffs (1.0 to 4.0 in REVIGO).

c) **Solution Robustness.** This criterion evaluates the robustness of a solution to incomplete gene activity data. It compares the EV terms obtained on the original 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 the positive set, we computed average precision, recall and F1 scores across these iterations. Another perspective is provided by the examination of the frequency by which GO terms are detected in the subsampled datasets: higher frequency for a specific EV term implies higher robustness. We measured this robustness aspect of a solution using AUPR, in which EV terms are ranked according to their frequency across iterations (again, using EV terms detected on the full dataset as the positive instances). Note that cases in which an algorithm results in many empty solutions (that is, solutions with no enriched GO terms) and a few non-empty ones that are enriched for true EV terms can yield a high but misleading AUPR score. Therefore we validated that the fraction of non-empty solutions obtained by the algorithms on the subsampled runs is high: all the algorithms achieved around 60% or more non-empty solutions on GE data (**Figure S6**).

Module-Level Criteria

a) **Module-Level EHR (mEHR).** This criterion calculates a single module's EHR. We define the module-level EHR (*mEHR*), as the ratio between the number of a module's EV terms and HG terms (**Figure S5**). We score each solution by averaging the mEHR of its k top-ranked modules (k values ranging from 1-20).

b) **Intra-Module Homogeneity.** This index measures the homogeneity of the biological signal that is captured by each module compared to the biological signal in the entire solution. For its calculation, we build a (complete) graph for the solution's EV terms (GO graph) in which nodes represent the EV terms and the weights on the edges are the pairwise Resnik similarity score (**Figure S5B**). Next, edges whose weight is below a cutoff are removed. The *intra-module homogeneity* is defined as the module's relative edge-density in this GO graph:

$$\frac{\left(\frac{\#\ of\ edges\ in\ module's\ GO\ graph}{\#\ of\ edges\ in\ a\ complete\ graph\ of\ that\ size}\right)}{\left(\frac{\#\ of\ edges\ in\ the\ solution's\ GO\ graph}{\#\ of\ edges\ in\ a\ complete\ graph\ of\ the\ same\ size}\right)}$$

We calculate the intra-module homogeneity score for a solution by averaging its modules' scores **(Figure S5B)**. We repeat this test for a range of similarity cutoffs – from 1.0 to 4.0. This criterion provides a complementary view on top of the one captured by the biological richness criterion, by characterizing the biological coherence of the reported modules.

Acknowledgements

We thank the reviewers for insightful comments which helped us to substantially improve the paper. This study was supported in part by German-Israeli Project DFG RE 4193/1-1 (to RS and RE), by the Israel Science Foundation grants No. 1339/18 (to RS) and No. 2118/19 (to RE), by Len Blavatnik and the Blavatnik Family foundation (to RS) and the Koret-UC Berkeley-Tel Aviv University Initiative in Computational Biology and Bioinformatics (to R.E.). HL was supported in part by a fellowship from the Edmond J. Safra Center for Bioinformatics at Tel-Aviv University. R.E. is a Faculty Fellow of the Edmond J. Safra Center for Bioinformatics at Tel Aviv University.

Author contribution

RE and RS conceived the project. HL designed and developed the algorithms, implemented them and performed the analysis under the supervision of RE and RS. All authors wrote the manuscript and approved the final manuscript.

Conflict of Interest

None.

Data Availability

The datasets used in this study are listed in Supplementary Information Tables S2 and S3. The GE datasets and the gene scores of both GE and GWAS datasets are also available at https://github.com/Shamir-Lab/EMP under the "datasets" folder.

References

Alon U (2003) Biological networks: The tinkerer as an engineer. *Science (80-.).* **301:** 1866–1867

Ashcroft M, Taya Y & Vousden KH (2000) Stress signals utilize multiple pathways to stabilize p53. *Mol. Cell. Biol.* **20:** 3224–3233

Barabási AL, Gulbahce N & Loscalzo J (2011) Network medicine: A network-based approach to human disease. *Nat. Rev. Genet.* **12:** 56–68

Barabási AL & Oltvai ZN (2004) Network biology: Understanding the cell's functional organization. *Nat. Rev. Genet.* **5:** 101–113

Barel G & Herwig R (2018) Network and pathway analysis of toxicogenomics data. *Front. Genet.* **9:** 484

Baumbach J, Friedrich T, Kötzing T, Krohmer A, Müller J & Pauling J (2012) Efficient algorithms for extracting biological key pathways with global constraints. In *Proceedings of the genetic and evolutionary computation conference, GECCO 2012* pp 169–176.

Beisser D, Klau GW, Dandekar T, Müller T & Dittrich MT (2010) BioNet: An R-Package for the functional analysis of biological networks. *Bioinformatics* **26:** 1129–1130

Benjamini Y & Hochberg Y (1995) Controlling the false discovery rate: A practical and powerful approach to multiple testing. *J. R. Stat. Soc.* **57:** 289–300

Blondel VD, Guillaume JL, Lambiotte R & Lefebvre E (2008) Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008:** P10008

Boyle EA, Li YI & Pritchard JK (2017) An expanded view of complex traits: from polygenic to omnigenic. *Cell* **169:** 1177–1186

Carter H, Hofree M & Ideker T (2013) Genotype to phenotype via network analysis. *Curr. Opin. Genet. Dev.* **23:** 611–621

Cerami E, Demir E, Schultz N, Taylor BS & Sander C (2010) Automated network analysis identifies core pathways in glioblastoma. *PLoS One* **5:** e8918

Chang S, Fang K, Zhang K & Wang J (2015) Network-based analysis of schizophrenia genome-wide association data to detect the joint functional association signals. *PLoS One* **10:** e0133404

Choobdar S, Ahsen ME, Crawford J, Tomasoni M, Fang T, Lamparter D, Lin J, Hescott B, Hu X, Mercer J, Natoli T, Narayan R, Aicheler F, Amoroso N, Arenas A, Azhagesan K, Baker A, Banf M, Batzoglou S, Baudot A, et al (2019) Assessment of network module identification across complex diseases. *Nat. Methods* **16:** 843–852

Cowen L, Ideker T, Raphael BJ & Sharan R (2017) Network propagation: A universal amplifier of genetic associations. *Nat. Rev. Genet.*: 551–562

Creixell P, Reimand J, Haider S, Wu G, Shibata T, Vazquez M, Mustonen V, Gonzalez-Perez A,

Pearson J, Sander C, Raphael BJ, Marks DS, Ouellette BFFF, Valencia A, Bader GD, Boutros PC, Stuart JM, Linding R, Lopez-Bigas N, Stein LD, et al (2015) Pathway and network analysis of cancer genomes. *Nat. Methods* **12:** 615–621

Fernández-Tajes J, Gaulton KJ, Van De Bunt M, Torres J, Thurner M, Mahajan A, Gloyn AL, Lage K & McCarthy MI (2019) Developing a network view of type 2 diabetes risk pathways through integration of genetic, genomic and functional data. *Genome Med.* **11:**

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

Hartwell LH, Hopfield JJ, Leibler S & Murray AW (1999) From molecular to modular cell biology. *Nature* **402:** C47-52

He H, Lin D, Zhang J, Wang Y & Deng H (2017) Comparison of statistical methods for subnetwork detection in the integration of gene expression and protein interaction network. *BMC Bioinformatics* **18:** 149

Hegde C, Indyk P & Schmidt L (2014) A fast, adaptive variant of the Goemans-Williamson scheme for the prize-collecting Steiner tree problem. *Work. 11th DIMACS Implement. Chall.*

Huang JK, Carlin DE, Yu MK, Zhang W, Kreisberg JF, Tamayo P & Ideker T (2018) Systematic evaluation of molecular networks for discovery of disease genes. *Cell Syst.* **6:** 484-495.e5

Ideker T, Ozier O, Schwikowski B & Siegel AF (2002) Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* **18:** S233–S240

Ideker T & Sharan R (2008) Protein networks in disease. *Genome Res.* **18:** 644–652

Johnson DS, Minkoo M & Phillips S (2000) The prize collecting Steiner tree problem: Theory and practice. *SODA '00 Proc. Elev. Annu. ACM-SIAM Symp. Discret. algorithms***:** 760–769

Kempe D, Kleinberg J, Tardos É & Tardos E (2015) Maximizing the spread of influence through a social network. *Theory Comput.* **11:** 105–147

Khera A V., Chaffin M, Aragam KG, Haas ME, Roselli C, Choi SH, Natarajan P, Lander ES, Lubitz SA, Ellinor PT & Kathiresan S (2018) Genome-wide polygenic scores for common diseases identify individuals with risk equivalent to monogenic mutations. *Nat. Genet.* **50:** 1219–1224

Kyriakis JM & Avruch J (2012) Mammalian MAPK signal transduction pathways activated by stress and inflammation: A 10-year update. *Physiol. Rev.* **92:** 689–737

Lambiotte R, Delvenne J-C & Barahona M (2008) Laplacian dynamics and multiscale modular structure in networks. *IEEE Trans. Netw. Sci. Eng.* **1:** 76–90

Lamparter D, Marbach D, Rueedi R, Kutalik Z & Bergmann S (2016) Fast and rigorous computation of gene and pathway scores from SNP-based summary statistics. *PLOS Comput. Biol.* **12:** e1004714

de Leeuw CA, Mooij JM, Heskes T & Posthuma D (2015) MAGMA: Generalized gene-set analysis

of GWAS data. *PLOS Comput. Biol.* **11:** e1004219

Leiserson MDM, Vandin F, Wu H-T, Dobson JR, Eldridge J V, Thomas JL, Papoutsaki A, Kim Y, Niu B, McLellan M, Lawrence MS, Gonzalez-Perez A, Tamborero D, Cheng Y, Ryslik GA, Lopez-Bigas N, Getz G, Ding L & Raphael BJ (2015) Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nat. Genet.* **47:** 106–14

Lord PW, Stevens RD, Brass A & Goble CA (2003) Semantic similarity measures as tools for exploring the gene ontology. *Pac. Symp. Biocomput.***:** 601–612

Luck K, Kim DK, Lambourne L, Spirohn K, Begg BE, Bian W, Brignall R, Cafarelli T, Campos-Laborie FJ, Charloteaux B, Choi D, Coté AG, Daley M, Deimling S, Desbuleux A, Dricot A, Gebbia M, Hardy MF, Kishore N, Knapp JJ, et al (2020) A reference map of the human binary protein interactome. *Nature* **580:** 402–408

McGillivray P, Clarke D, Meyerson W, Zhang J, Lee D, Gu M, Kumar S, Zhou H & Gerstein M (2018) Network analysis as a grand unifier in biomedical data science. *Annu. Rev. Biomed. Data Sci.* **1:** 153–180

Mitra K, Carvunis A-RR, Ramesh SK & Ideker T (2013) Integrative approaches for finding modular structure in biological networks. *Nat. Rev. Genet.* **14:** 719–732

Musunuru K & Kathiresan S (2019) Genetics of common, complex coronary artery disease. *Cell* **177:** 132–145

Nakka P, Raphael BJ & Ramachandran S (2016) Gene and network analysis of common variants reveals novel associations in multiple complex diseases. *Genetics* **204:** 783–798

Nikolayeva I, Guitart Pla O & Schwikowski B (2018) Network module identification—A widespread theoretical bias and best practices. *Methods* **132:** 19–25

Perez-Riverol Y, Zorin A, Dass G, Vu MT, Xu P, Glont M, Vizcaíno JA, Jarnuczak AF, Petryszak R, Ping P & Hermjakob H (2019) Quantifying the impact of public omics data. *Nat. Commun.* **10:** 3512

Resnik P (1999) Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artiicial Intell. Res.* **11:** 95–130

Reyna M, Chitra U, Elyanow R & Raphael B (2020) NetMix: A network-structured mixture model for reduced-bias estimation of altered subnetworks. *bioRxiv***:** 2020.01.18.911438

Robinson MD, McCarthy DJ & Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26:** 139–40

Stringer S, Wray NR, Kahn RS & Derks EM (2011) Underestimated effect sizes in GWAS: Fundamental limitations of single SNP analysis for dichotomous phenotypes. *PLoS One* **6:** e27964

Sullivan PF & Geschwind DH (2019) Defining the genetic, genomic, cellular, and diagnostic architectures of psychiatric disorders. *Cell* **177:** 162–183

Supek F, Bošnjak M, Škunca N & Šmuc T (2011) REVIGO summarizes and visualizes long lists of gene ontology terms. *PLoS One* **6:** e21800

Szklarczyk D, Morris JH, Cook H, Kuhn M, Wyder S, Simonovic M, Santos A, Doncheva NT, Roth A, Bork P, Jensen LJ & von Mering C (2017) The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Res.* **45:** D362–D368

The Gene Ontology Consortium (2019) The gene ontology resource: 20 years and still GOing strong. *Nucleic Acids Res.* **47:** D330–D338

Xenarios I, Salwínski L, Duan XJ, Higney P, Kim S-M & Eisenberg D (2002) DIP, the Database of Interacting Proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.* **30:** 303–305

Yang J, McAuley J & Leskovec J (2014) Community detection in networks with node attributes. *Proc. - IEEE Int. Conf. Data Mining, ICDM***:** 1151–1156

Appendix

AMI tools - execution details

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

jActiveModules (Ideker *et al*, 2002). jActiveModules was written as a plugin for Cytoscape (Shannon *et al*, 2003), a powerful platform for network analysis of biological data. 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 (from n=5 to n=50) to retrieve more modules and required that reported modules would be mutually exclusive. The algorithm typically produced no more than 10 modules with more than 3 genes.

NetBox (Cerami *et al*, 2010). We modified NetBox codebase so we can choose the networks it uses. 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.

BioNet (Beisser *et al*, 2010)**.** BioNet is designed to retrieve only one module. 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 in five consecutive runs.

HotNet2 (Leiserson *et al*, 2015)**.** HotNet2 expects gene activity scores that are calculated by mutation rate 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. As HotNet2 execution time was longer than the other algorithms we generated only 1000 permutations for its background distribution.

KeyPathwayMiner (Baumbach *et al*, 2012)**.** We used the version of KPM with the greedy strategy. It expects binary gene activity scores: 1 marks a gene as active 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.

DOMINO. DOMINO gets as an input 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.

Analyzing the network contribution to non-specific GO enrichment bias

Since the structure of the input network can potentially be a source for the bias we detected in AMI solutions, we sought to identify GO terms that are enriched on modules detected on the network only, without considering any gene scores based on specific omics data. We wished to examine if excluding such terms could serve as an effective alternative to EMP in removing non-specific terms reported by AMI methods.

To this goal we identified modules in the DIP network using a network-based module identification algorithm (that is, an algorithm that finds modules based on network structure only, without considering activity scores). We chose the algorithm denoted as 'M1' in the module detection DREAM challenge (Choobdar *et al*, 2019). M1 was a leading performer in the challenge and is also implemented as a convenient software.

We extracted GO terms that were enriched ($q - value < 10^{-3}$) on these network-based modules (here too, using the entire set of network genes as the background set). Overall, 2,450 out of 6,573 (37%) BP GO terms were detected by this analysis, and we refer to them as *net-terms*. We found that the net-terms were significantly over-represented among GO terms reported by most AMI algorithms on the GE and GWAS datasets (**Figure 7 A, B**). This is expected, since connectivity is a feature sought by both community detection and AMI methods. Over-representation increased with the number of different datasets on which the GO term was detected (**Figure S7 A, B).** For example, GO terms that were called on all 10 datasets were almost exclusively net-terms.

Next, we examined whether net-terms were over-represented among the terms that failed the empirical validation. We first computed, for each algorithm and term τ, the fraction of datasets in which the term appeared as significant but was not empirically validated. Let $\# EV\ (\tau)$ be the number of datasets in which τ appeared as empirically validated. Let $\# non - EV\ (\tau)$ be the number of datasets in which τ appeared as significant but was rejected, i.e., not empirically validated. Then we compute the following *rejection ratio* $\rho(\tau)$:

$$\rho(\tau) = \frac{\# non{-}EV\ (\tau)}{\# non{-}EV\ \ (\tau) + \ \#EV\ \ (\tau)}.$$

Terms with high rejection ratio are those that frequently appear as significant and fail the empirical test, possibly due to the network structure. Of course, this number should be considered alongside the value of the denominator, which counts how frequently τ appeared across datasets. In the following we considered only terms that were enriched in at least four datasets.

Figure S8 A and **B** show for each algorithm the distribution of rejection ratios across GO terms. As can be seen, DOMINO has substantially lower rejection ratios. The same, albeit to a lesser extent, is true for NetBox.

We then examined whether network structure is a key source for rejected terms (**Figure 7 C, D**). For each algorithm, we compared the rejection ratio of net-terms and other terms. Notably, net-terms did not show significantly high rejection ratios, except for DOMINO on GE, where it was marginally significant (p=0.02).

Last, we examined the association between net-terms and rejected terms in each solution. We summarized the results of each algorithm on each dataset in a contingency table, computed the enrichment factor (EF) for rejected terms among net-terms, and calculated p-value using the Fisher's exact test, and corrected the results of each algorithm for multiple testing using FDR (**External Tables 2-3).** Most solutions had EF>1, and about half the results showed significant association between net-terms and rejected terms. The overall magnitude of the association was rather mild (mean q-value = 0.26 for GE and 0.38 for GWAS), and many false calls involved non net-GO terms.

In summary, GO terms detected by AMI methods are enriched for net-terms. This enrichment increases with the number of datasets on which the GO term was detected. In addition, a fraction of the rejected calls were driven by network topology. However, this association was overall mild and cannot be used to effectively distinguish between GO terms that reflect true biological signals (EV terms) and non-specific ones (rejected terms). Therefore, filtering out net-terms cannot serve as an alternative to our EMP procedure.

Table S1. *AMI algorithms included in our analysis.*

Method name	**Published on**	**Designed for**	**Algorithmic Approach**	**Code language**	**# citations (updated to 11/2019)**
jActiveModules (Ideker *et al*, 2002)	2002	GE	Seek high scoring subnetworks either by simulated annealing (jAM_SA) or by a greedy search (jAM_greedy)	Java	1207
NetBox (Cerami *et al*, 2010)	2010	Somatic Mutations	Enrichment of Perturbed neighbors, Newman-Girvan (NG) modularity score	Java, Python	304
BioNet (Beisser *et al*, 2010)	2010	GE	Prize collecting Steiner tree	R	218
HotNet2 (Leiserson *et al*, 2015)	2015	Somatic Mutations	Heat diffusion	Python	460
KeyPathwayMiner (Baumbach *et al*, 2012)	2012	GE	Choose modules with at most K non-perturbed genes	Java	41

***Table S2**. The ten gene expression datasets used in our benchmark analysis.*

Datasets name (acronym)	access to data	Technology	General description
TNFa (Schmidt *et al*, 2015)	GSE64233	RNA-seq	TNFa, a potent inducer of immune responses
HC (Elkon *et al*, 2015)	GSE67478	RNA-seq	Hair cell from the cochlea and vestibular system, compared to non-hair cell from these inner-ear organs.
SHERA (Miano *et al*, 2018)	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
SHEZH (Ito *et al*, 2018)	GSE109064	RNA-seq	Downregulation of EZH2 leads to cellular senescence with features of SASP. Comparison between control to 4d samples.
ERS (Kroeger *et al*, 2018)	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 ATF6-activated and control cells.
IEM (Hertzano *et al*, 2011)	---	Microarray	Comparison between 2 different cell types in the inner-ear: blood cells and mesenchymal cells.
ROR (Bayerlová *et al*, 2017)	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 ROR2-overexpression and control conditions.
APO (Pulikkan *et al*, 2018)	GSE101788	RNA-seq	Comparison between ME-1 cells (a human leukemia cell line) treated with either the AI-10-49 drug (which induces apoptosis) or DMSO (control).
CBX (Connelly *et al*, 2019)	GSE123689	RNA-seq	CBX8 is a subunit of the polycomb repressive complex 1 (PRC1). This RNA-seq experiment compared CBX8-KO and control cells.
IFT (Forbes *et al*, 2018)	GSE107230	RNA-seq	IFT140 is involved in the formation and maintenance of cilia. This RNA-seq experiment compared uncorrected (IFT140 compound heterozygous) and gene-corrected (IFT140 heterozygous) epithelial cells isolated from patient's iPSC that were derived from kidney organoids.

Table S3. *The ten GWAS datasets used in our benchmark analysis.*

Datasets name (acronym)	Trait	Cohort size
BC (Michailidou *et al*, 2017)	Breast Cancer	228,951
CHD (De Lange *et al*, 2017)	Crohn's Disease	59,957
SCZ (Ripke *et al*, 2014)	Schizophrenia	150,064
TRI (Teslovich *et al*, 2010)	Triglycerides	~100,000
T2D (Mahajan *et al*, 2018)	Type 2 Diabetes	898,130
CAD (Nelson *et al*, 2017)	Coronary Artery Disease	155,197
BMD (Kemp *et al*, 2017)	Bone Mineral Density	142,487
Height (Allen, 2010)	Height	183,727
AF (Nielsen *et al*, 2018)	Arterial Fibrillation	1,030,836
AMD (Fritsche *et al*, 2016)	Age-related Macular Degeneration	33,976

Table S4. Runtimes on the original GE datasets (in seconds)

Network	algorithm	tnfa	hc	ror	shera	shezh	ers	iem	apo	cbx	ift
DIP	NetBox	23.4	24.6	24.9	124.2	32.2	2706.2	27.1	1079.1	24.7	29.9
	DOMINO *	**0.8**	**0.9**	**0.9**	**1.1**	**0.7**	**3.5**	**1.0**	**2.1**	**0.8**	**0.8**
	jAM_Greedy	20.0	25.3	21.9	21.3	24.6	26.9	24.9	22.7	28.4	26.2
	jAM_SA	368.3	452.6	658.5	1361.8	847.8	1010.4	801.8	915.2	1056.9	1189.2
	Bionet	66.3	199.0	85.9	500.6	287.7	193.4	206.0	172.8	369.6	634.1
	HotNet2 **	60.9	68.8	61.9	66.6	62.1	72.2	69.6	71.1	63.0	68.3
	KPM	22.5	78.1	42.0	66.2	76.5	118.1	70.1	75.7	79.5	75.5
HuRI	NetBox	26.0	38.9	26.4	5809	34.6	>5hrs	78.1	>5hrs	30.2	43.5
	DOMINO *	**1.2**	**1.9**	**2.6**	**1.4**	**1.8**	**4.8**	**1.3**	**2.0**	**1.4**	**1.6**
STRING	NetBox	4445.8	>5 hrs	76.6	>5 hrs	>5 hrs	>5 hrs	>5 hrs	>5 hrs	>5 hrs	>5 hrs
	DOMINO *	**4.7**	**5.3**	**4.4**	**5.4**	**8.5**	**30.5**	**5.3**	**9.0**	**3.4**	**4.3**

Performance was measured on a 44-core, 2.2 GHz server with 792 GB of RAM
* DOMINO runtimes are for steps 1-3 of the algorithm only, excluding step 0, which is executed only once per network.
** HotNet2 was the only algorithm running on multiple cores

Table S5. Runtimes on the original GWAS datasets (in seconds)

network	algorithm	brca	crh	scz	tri	t2d	bmd	amd	af	hgt	cad
DIP	**NetBox**	24.7	23.4	24.4	23.4	23.6	179.6	24.0	24.3	26.4	23.4
	DOMINO *	**0.7**	**0.7**	**0.8**	**0.9**	**0.5**	**1.2**	**1.0**	**0.8**	**0.8**	**0.8**
	jAM_Greedy	23.3	21.4	25.0	24.4	22.8	21.9	29.6	20.2	22.5	22.3
	jAM_SA	680.1	545.4	925.8	1183.6	1193.3	896.1	1030.4	758.1	930.6	508.2
	Bionet	311.0	144.6	496.7	68.6	36.6	338.0	60.6	267.7	312.8	39.7
	HotNet2 **	73.4	74.1	73.8	70.1	73.7	73.9	70.5	74.5	73.1	73.3
	KPM	75.4	47.9	76.6	26.0	11.1	50.7	25.8	64.9	75.5	9.6
HuRI	**NetBox**	59.5	26.6	37.7	26.6	26.3	> 5 hrs.	26.4	29.2	37.8	25.9
	DOMINO *	**1.6**	**2.0**	**1.6**	**2.3**	**1.7**	**1.3**	**0.9**	**1.4**	**1.6**	**0.7**
STRING	**NetBox**	940.2	289.5	573.5	76.9	75.1	> 5 hrs.	313.0	171.8	632.0	69.4
	DOMINO *	**2.3**	**4.5**	**2.9**	**4.7**	**1.8**	**2.0**	**4.7**	**2.6**	**2.9**	**6.6**

Performance was measured on a 44-core, 2.2 GHz server with 792 GB of RAM
* DOMINO runtimes are for steps 1-3 of the algorithm only, excluding step 0, which is executed only once per network.
** HotNet2 was the only algorithm running on multiple cores

Table S6. Runtimes on ten permutations of GE and GWAS datasets (in seconds)

network	algorithm	omics	Mean runtime, excluding jobs that did not terminate in 30 min	Number of permutations on which jobs exceeded 30 min	Average runtime on the original (unpermuted) datasets, excluding jobs that did not terminate in 5 hrs	Number of jobs on the original datasets not terminating in 5 hrs
DIP	NetBox	GE	44.31	0	409.63	0
	DOMINO *	GE	0.71	0	1.25	0
	jAM_Greedy	GE	17.53	0	24.21	0
	jAM_SA	GE	17.37	0	866.25	0
	Bionet	GE	2.95	0	271.54	0
	HotNet2 **	GE	64.37	0	66.44	0
	KPM	GE	55.15	0	70.42	0
	NetBox	GWAS	14.71	0	39.71	0
	DOMINO	GWAS	0.60	0	0.82	0
	jAM_Greedy	GWAS	17.45	0	23.33	0
	jAM_SA	GWAS	17.49	0	865.16	0
	Bionet	GWAS	3.00	0	207.62	0
	HotNet2 **	GWAS	76.69	0	73.03	0
	KPM	GWAS	43.61	0	46.36	0
HuRI	NetBox	GE	25.63	20	760.86	2
	DOMINO *	GE	1.37	0	1.99	0
	NetBox	GWAS	13.36	10	32.87	1
	DOMINO *	GWAS	1.37	0	1.5	0
STRING	NetBox	GE	233.62	36	3443.87	8
	DOMINO *	GE	2.97	0	8.07	0
	NetBox	GWAS	167.03	10	349.03	1
	DOMINO *	GWAS	3.13	0	3.5	0

Performance was measured on a 44-core, 2.2 GHz server with 792 GB of RAM
* DOMINO runtimes are for steps 1-3 of the algorithm only, excluding step 0, which is executed only once per network.
** HotNet2 was the only algorithm running on multiple cores

External Table 1 Summary of the benchmark results on different networks and HG thresholds

External Table 2. Association analysis between net-terms and rejected terms in GE datasets.

External Table 3. Association analysis between net-terms and rejected terms in GWAS datasets.

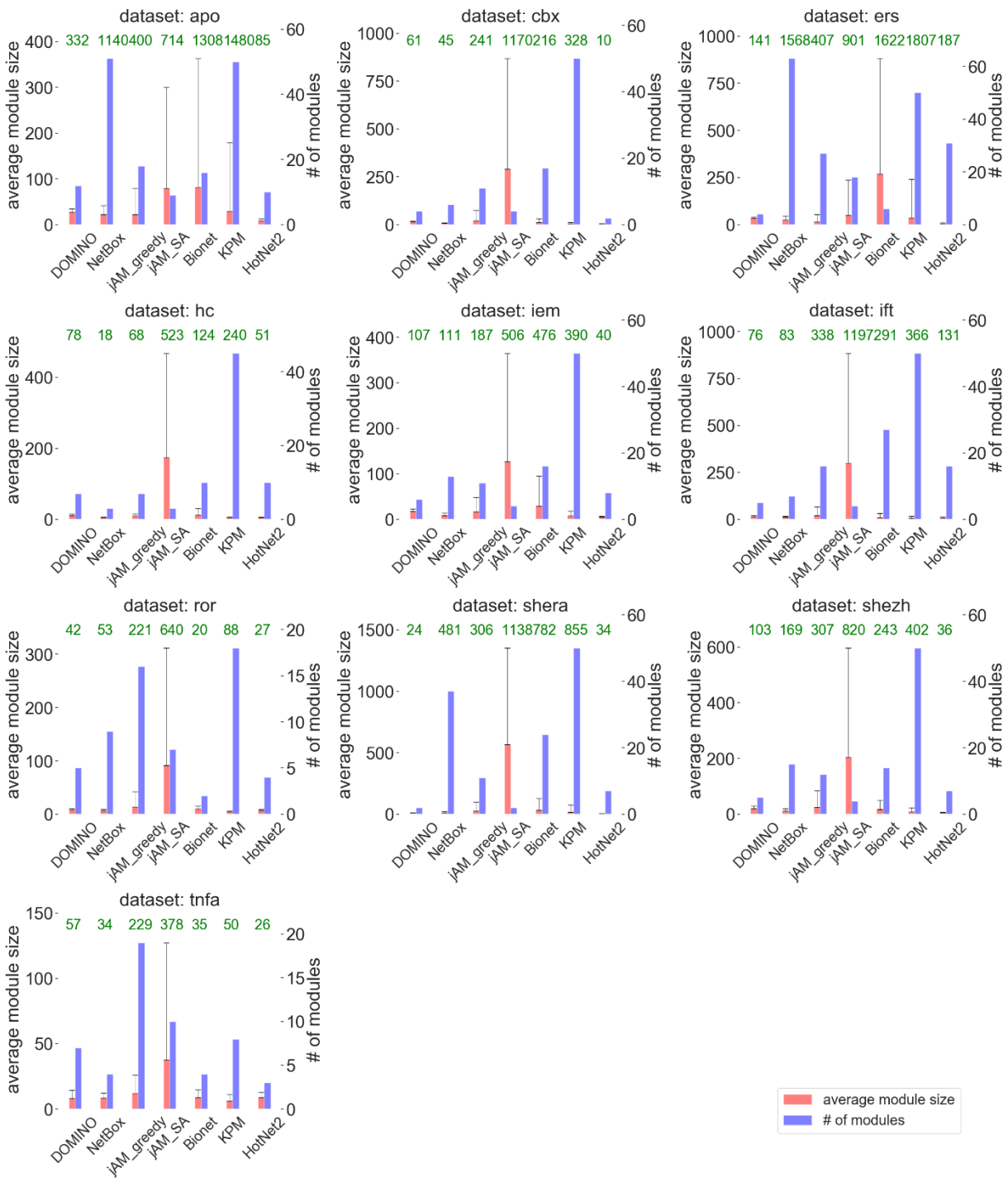

Figure S1. *Summary statistics of the solutions obtained on the GE datasets. For each dataset, the number of modules detected by each AMI algorithm and their sizes are indicated. (Error bars represent 1 SD of the number of genes in modules). The numbers in green are the total number of genes in the union of all modules in the solution.*

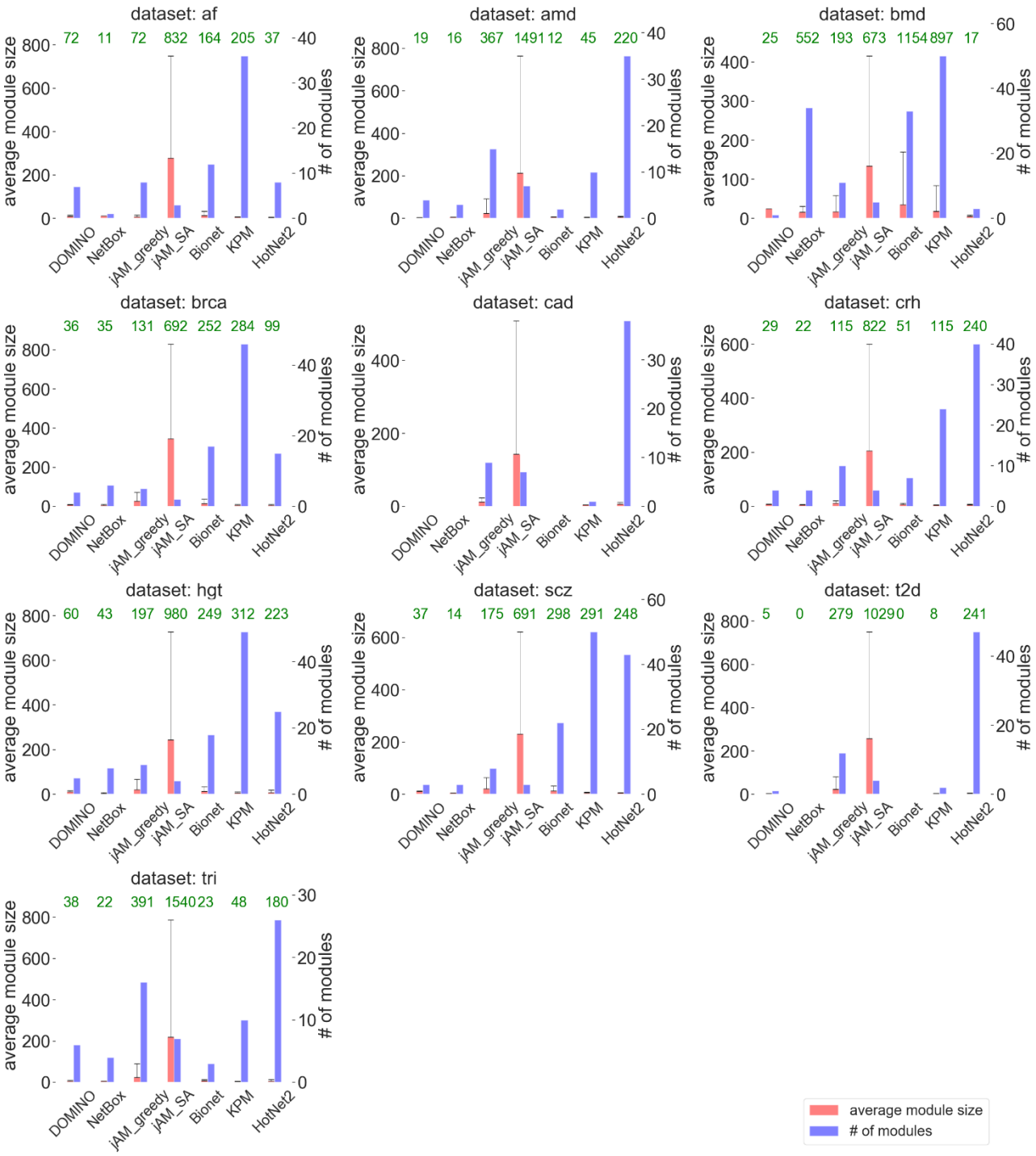

Figure S2. *Summary statistics of the solutions obtained on the GWAS datasets. For each dataset, the number of modules detected by each AMI algorithm and their sizes are indicated. (Error bars represent 1 SD of the number of genes in modules). We excluded empty solutions. The numbers in green are the total number of genes in the union of all modules in the solution.*

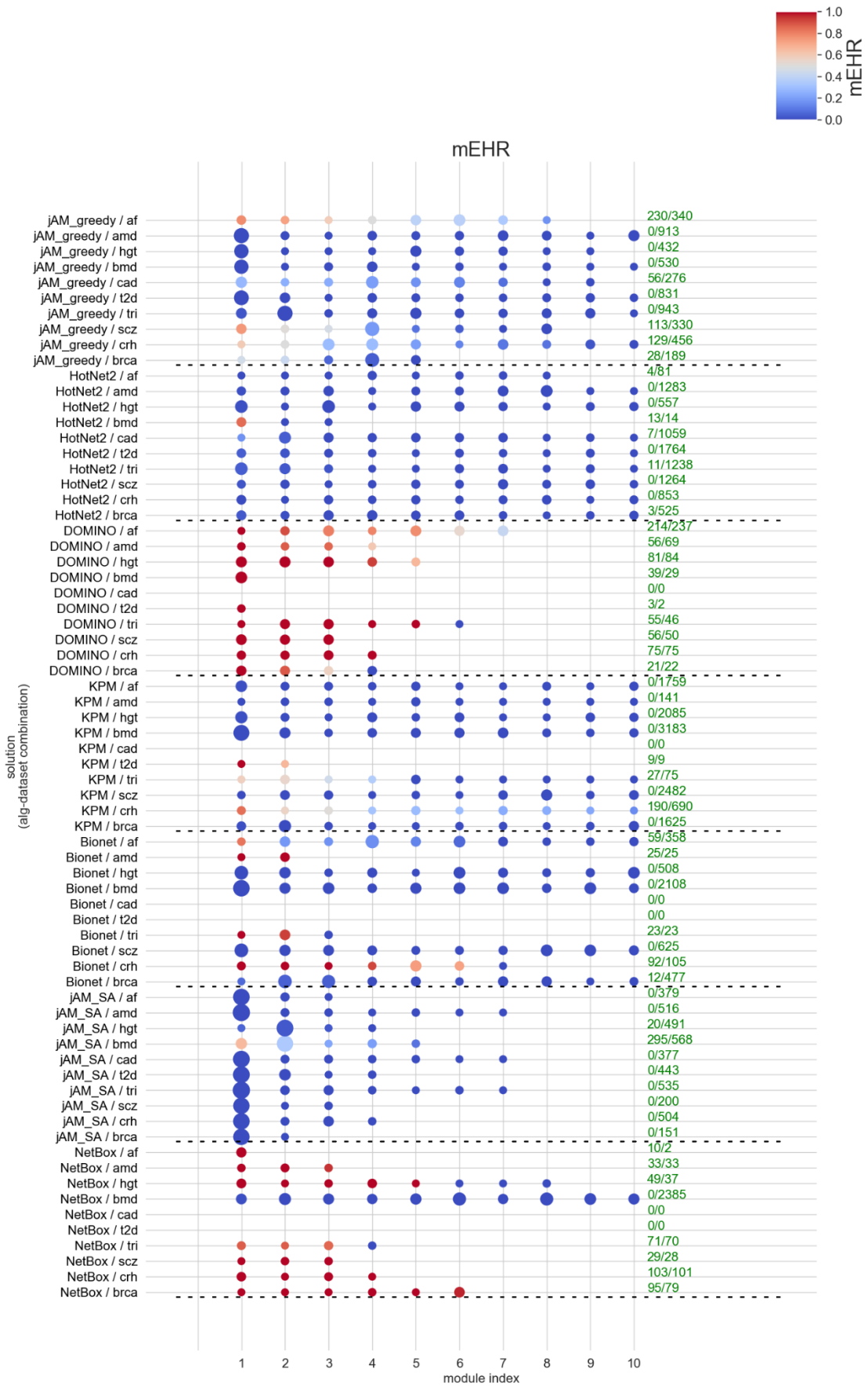

***Figure S3**. Module-level EHR (mEHR) scores on the GWAS datasets for each algorithm and GWAS dataset*

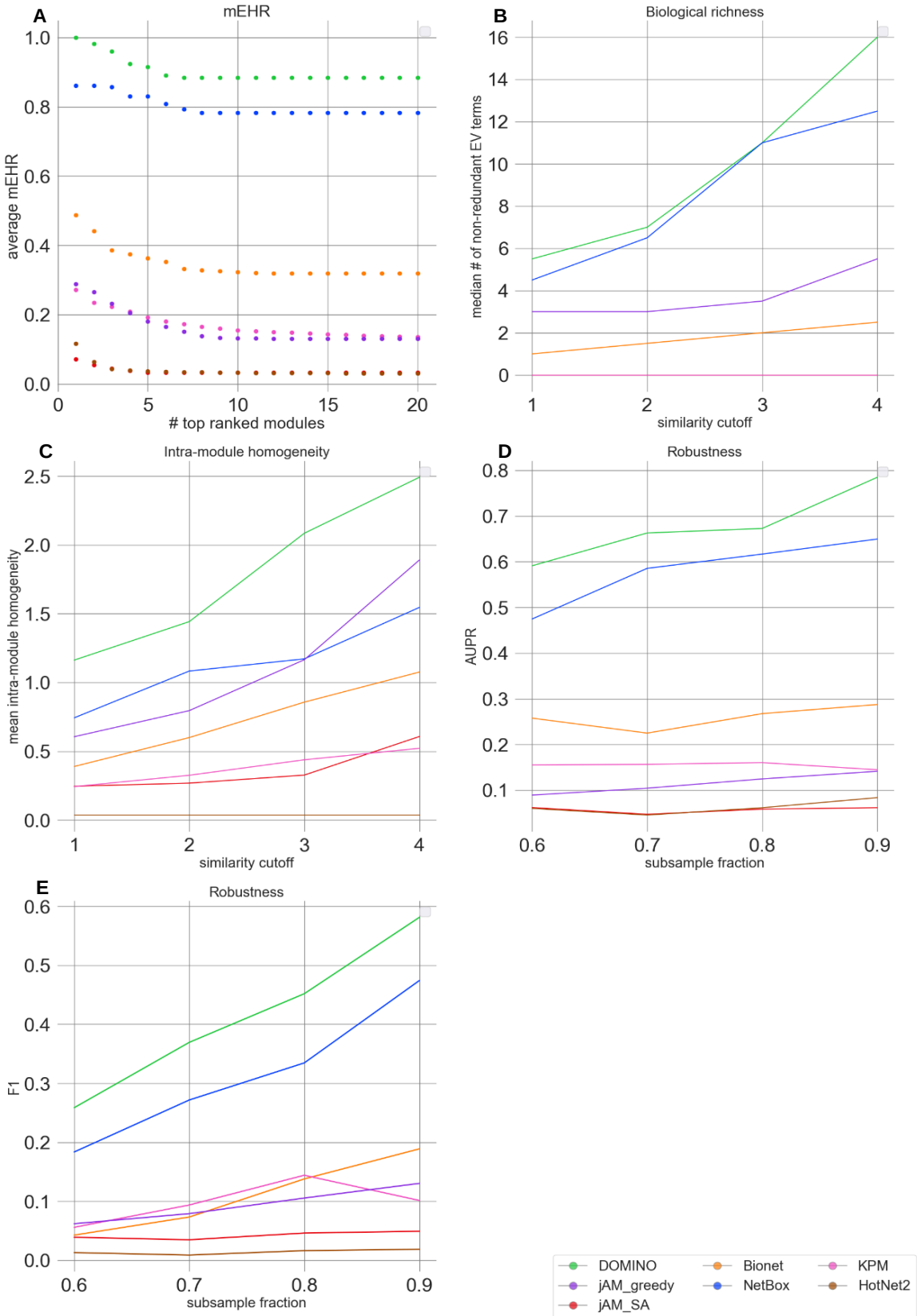

Figure S4. *Evaluation results for the GWAS datasets.* ***A.*** *Module-level EHR scores. The plots show average mEHR score in the k top modules, as a function of k. Modules are ranked by their mEHR scores.* ***B.*** *Biological richness. The plots show the median number of non-redundant terms (richness score) as a function of the Resnik similarity cutoff.* ***C.*** *Intra-module homogeneity scores as a function of Resnik similarity cutoff.* ***D.*** *Robustness measured by the average AUPR over the datasets, shown as a function of the subsampling fraction.* ***E.*** *Robustness measured by the average F1 over the datasets shown as a function of subsample fraction (results for each dataset and fraction were averaged over 100 subsampling).*

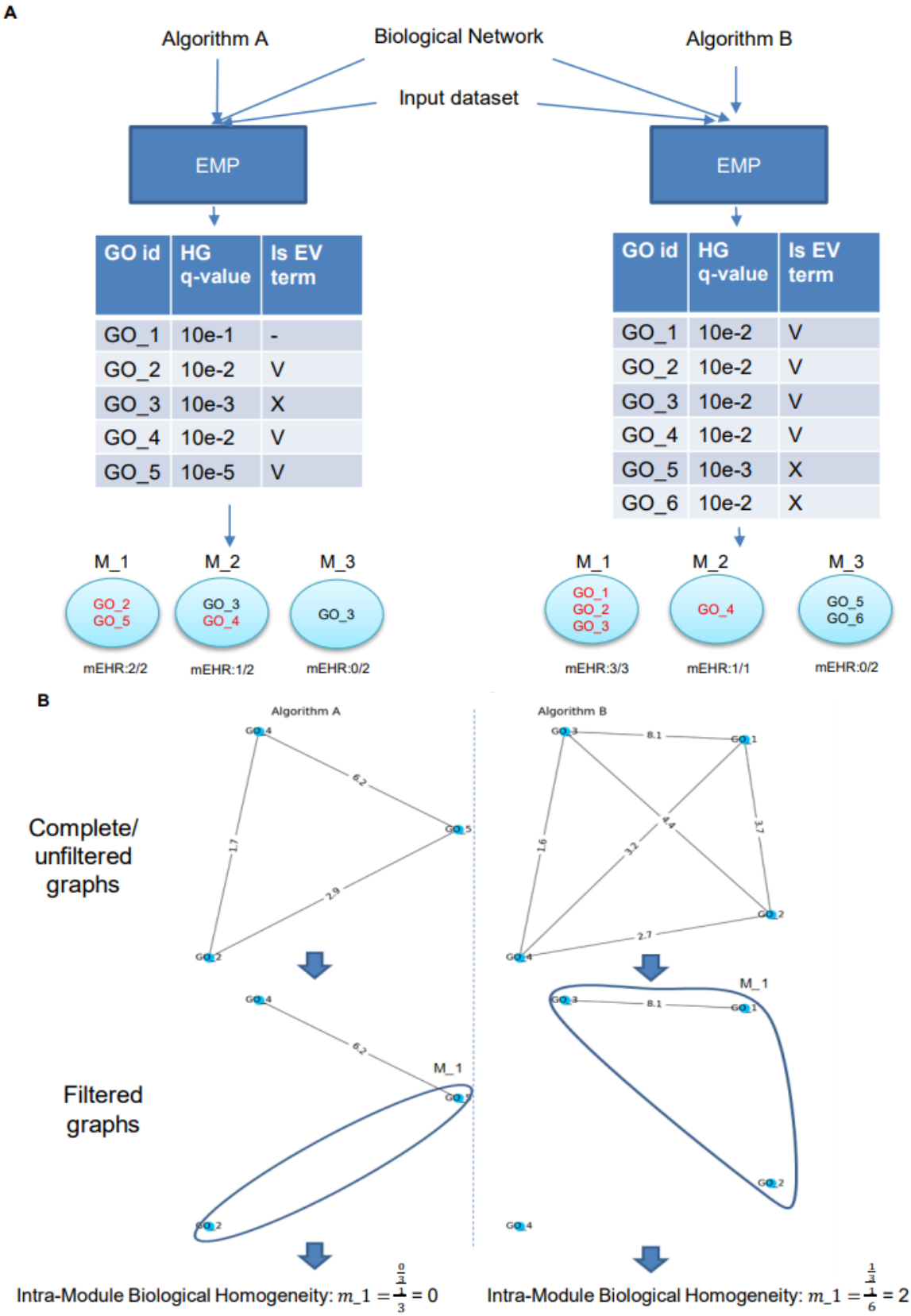

Figure S5. *Module-level evaluation criteria.* ***A. mEHR***. *Enriched GO terms in each module are examined by the EMP procedure (EV terms are colored in red) and mEHR is calculated for each module in the solution.* ***B. Intra-module homogeneity***. *A GO graph is first built for the union of all the EV terms in a solution using Resnik similarity scores. Then, a certain cut-off is applied (here, 4.0) for filtering low scoring edges. Last, the intra-module homogeneity score is calculated as the density ratio between the EV terms that are enriched in the module and the entire GO graph.*

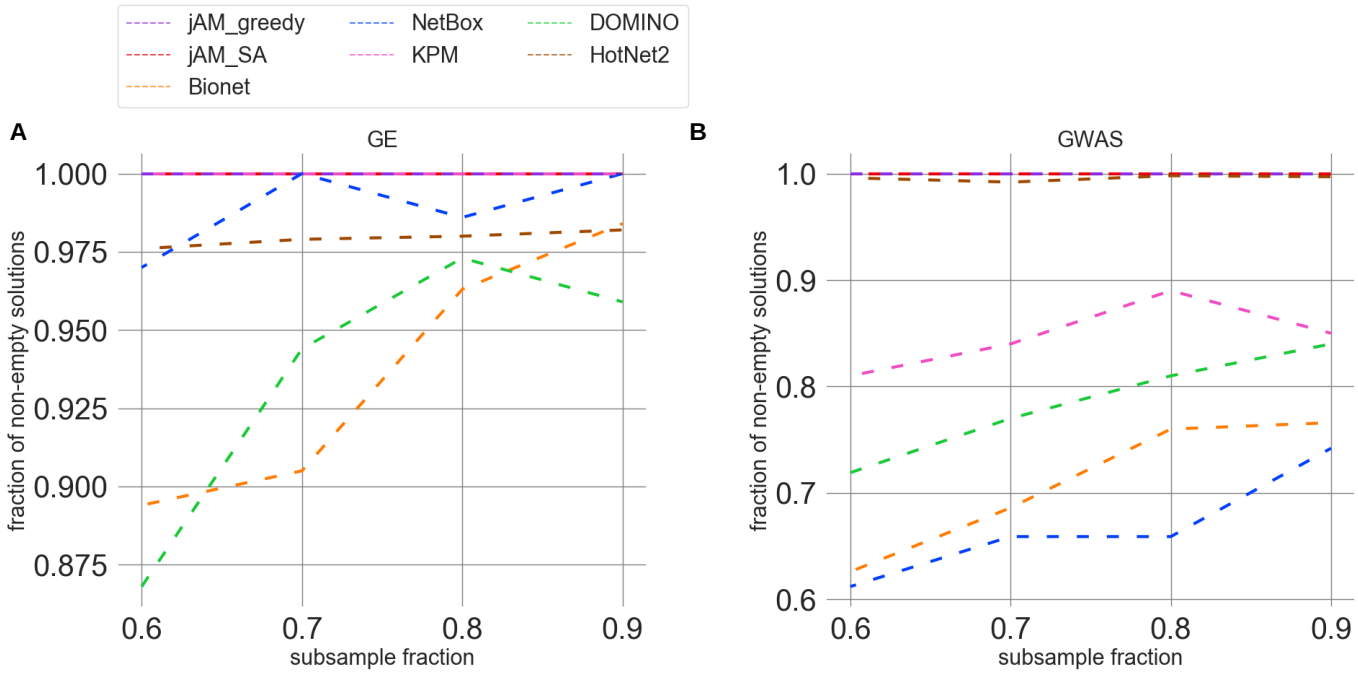

Figure S6*. The fraction of non-empty solutions as a function of the subsampling fraction. For each algorithm and subsampling fraction we report the average over the datasets.*

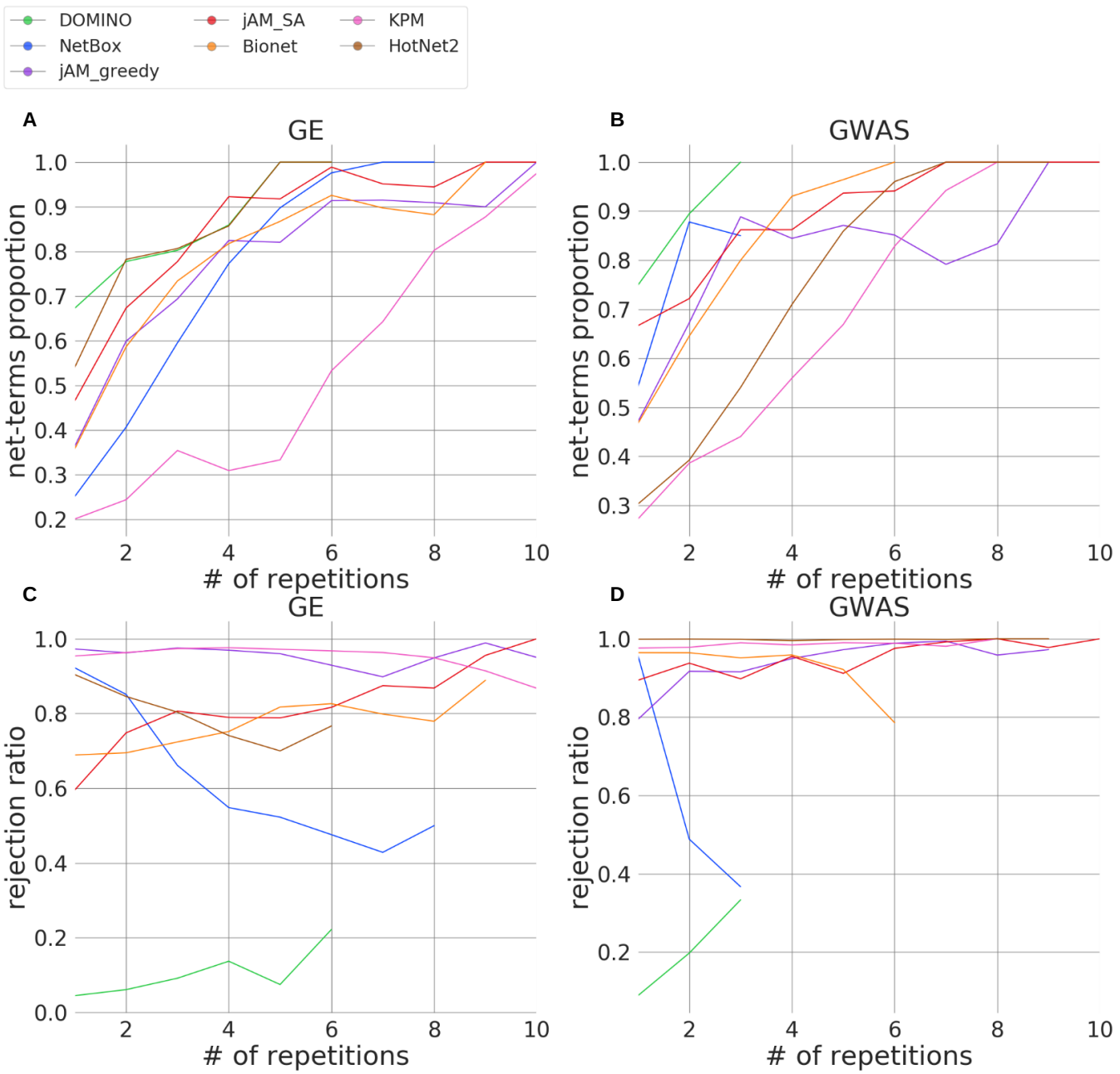

Figure S7. *Comparison of the GO terms identified by each benchmarked algorithm to those identified by using the network only (net-terms).* ***A-B.*** *Proportion of net-terms as a function of the number of times the term was reported. A: GE; B: GWAS.* ***C-D.*** *Rejection ratio as a function of the number of times the term was reported. C: GE; D: GWAS.*

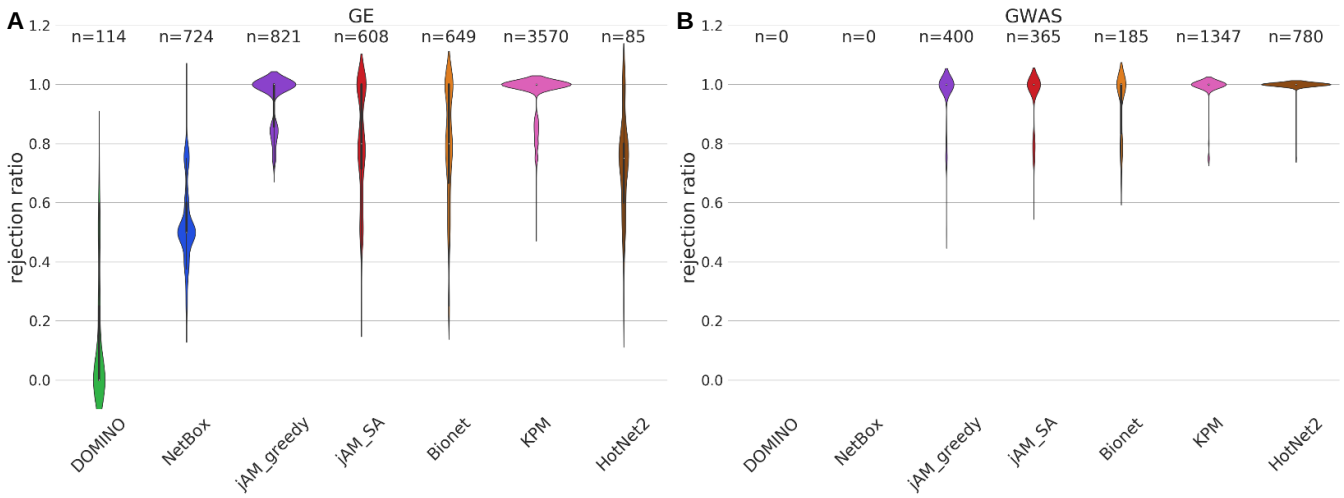

Figure S8*. Comparison of rejection ratios of GO terms across algorithms****. A-B****. Violin plots of rejection ratios of each algorithm in GE(A) and GWAS (B) datasets. Only terms reported in four datasets or more were included. n indicates the number of such terms. Note that no terms were reported in more than four GWAS datasets by DOMINO and NetBox, which obtained the best overall results in our benchmark.*

References

Allen HL (2010) Hundreds of variants clustered in genomic loci and biological pathways affect human height. *Nature* **467:** 832–838

Baumbach J, Friedrich T, Kötzing T, Krohmer A, Müller J & Pauling J (2012) Efficient algorithms for extracting biological key pathways with global constraints. In *Proceedings of the genetic and evolutionary computation conference, GECCO 2012* pp 169–176.

Bayerlová M, Menck K, Klemm F, Wolff A, Pukrop T, Binder C, Beißbarth T & Bleckmann A (2017) Ror2 signaling and its relevance in breast cancer progression. *Front. Oncol.* **7:** 135

Beisser D, Klau GW, Dandekar T, Müller T & Dittrich MT (2010) BioNet: An R-Package for the functional analysis of biological networks. *Bioinformatics* **26:** 1129–1130

Cerami E, Demir E, Schultz N, Taylor BS & Sander C (2010) Automated network analysis identifies core pathways in glioblastoma. *PLoS One* **5:** e8918

Choobdar S, Ahsen ME, Crawford J, Tomasoni M, Fang T, Lamparter D, Lin J, Hescott B, Hu X, Mercer J, Natoli T, Narayan R, Aicheler F, Amoroso N, Arenas A, Azhagesan K, Baker A, Banf M, Batzoglou S, Baudot A, et al (2019) Assessment of network module identification across complex diseases. *Nat. Methods* **16:** 843–852

Connelly KE, Weaver TM, Alpsoy A, Gu BX, Musselman CA & Dykhuizen EC (2019) Engagement of DNA and H3K27me3 by the CBX8 chromodomain drives chromatin association. *Nucleic Acids Res.* **47:** 2289–2305

Elkon R, Milon B, Morrison L, Shah M, Vijayakumar S, Racherla M, Leitch CC, Silipino L, Hadi S, Weiss-Gayet M, Barras E, Schmid CD, Ait-Lounis A, Barnes A, Song Y, Eisenman DJ, Eliyahu E, Frolenkov GI, Strome SE, Durand B, et al (2015) RFX transcription factors are essential for hearing in mice. *Nat. Commun.* **6:** 593

Forbes TA, Howden SE, Lawlor K, Phipson B, Maksimovic J, Hale L, Wilson S, Quinlan C, Ho G, Holman K, Bennetts B, Crawford J, Trnka P, Oshlack A, Patel C, Mallett A, Simons C & Little MH (2018) Patient-iPSC-derived kidney organoids show functional validation of a ciliopathic renal phenotype and reveal underlying pathogenetic mechanisms. *Am. J. Hum. Genet.* **102:** 816–831

Fritsche LG, Igl W, Bailey JNC, Grassmann F, Sengupta S, Bragg-Gresham JL, Burdon KP, Hebbring SJ, Wen C, Gorski M, Kim IK, Cho D, Zack D, Souied E, Scholl HPN, Bala E, ELee K, Hunter DJ, Sardell RJ, Mitchell P, et al (2016) A large genome-wide association study of age-related macular degeneration highlights contributions of rare and common variants. *Nat. Genet.* **48:** 134–143

Hertzano R, Elkon R, Kurima K, Morrisson A, Chan SL, Sallin M, Biedlingmaier A, Darling DS, Griffith AJ, Eisenman DJ & Strome SE (2011) Cell type-specific transcriptome analysis

reveals a major role for Zeb1 and miR-200b in mouse inner ear morphogenesis. *PLoS Genet.* **7:** e1002309

Ideker T, Ozier O, Schwikowski B & Siegel AF (2002) Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* **18:** S233–S240

Ito T, Teo YV, Evans SA, Neretti N & Sedivy JM (2018) Regulation of cellular senescence by polycomb chromatin modifiers through distinct DNA damage- and histone methylation-dependent pathways. *Cell Rep.* **22:** 3480–3492

Kemp JP, Morris JA, Medina-Gomez C, Forgetta V, Warrington NM, Youlten SE, Zheng J, Gregson CL, Grundberg E, Trajanoska K, Logan JG, Pollard AS, Sparkes PC, Ghirardello EJ, Allen R, Leitch VD, Butterfield NC, Komla-Ebri D, Adoum AT, Curry KF, et al (2017) Identification of 153 new loci associated with heel bone mineral density and functional involvement of GPC6 in osteoporosis. *Nat. Genet.* **49:** 1468–1475

Kroeger H, Grimsey N, Paxman R, Chiang WC, Plate L, Jones Y, Shaw PX, Trejo JA, Tsang SH, Powers E, Kelly JW, Luke Wiseman R & Lin JH (2018) The unfolded protein response regulator ATF6 promotes mesodermal differentiation. *Sci. Signal.* **11:** eaan5785

De Lange KM, Moutsianas L, Lee JC, Lamb CA, Luo Y, Kennedy NA, Jostins L, Rice DL, Gutierrez-Achury J, Ji SG, Heap G, Nimmo ER, Edwards C, Henderson P, Mowat C, Sanderson J, Satsangi J, Simmons A, Wilson DC, Tremelling M, et al (2017) Genome-wide association study implicates immune activation of multiple integrin genes in inflammatory bowel disease. *Nat. Genet.* **49:** 256–261

Leiserson MDM, Vandin F, Wu H-T, Dobson JR, Eldridge J V, Thomas JL, Papoutsaki A, Kim Y, Niu B, McLellan M, Lawrence MS, Gonzalez-Perez A, Tamborero D, Cheng Y, Ryslik GA, Lopez-Bigas N, Getz G, Ding L & Raphael BJ (2015) Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nat. Genet.* **47:** 106–14

Mahajan A, Taliun D, Thurner M, Robertson NR, Torres JM, Rayner NW, Payne AJ, Steinthorsdottir V, Scott RA, Grarup N, Cook JP, Schmidt EM, Wuttke M, Sarnowski C, Mägi R, Nano J, Gieger C, Trompet S, Lecoeur C, Preuss MH, et al (2018) Fine-mapping type 2 diabetes loci to single-variant resolution using high-density imputation and islet-specific epigenome maps. *Nat. Genet.* **50:** 1505–1513

Miano V, Ferrero G, Rosti V, Manitta E, Elhasnaoui J, Basile G & De Bortoli M (2018) Luminal lncRNAs regulation by ERα-controlled enhancers in a ligand-independent manner in breast cancer cells. *Int. J. Mol. Sci.* **19:** 593

Michailidou K, Lindström S, Dennis J, Beesley J, Hui S, Kar S, Lemaçon A, Soucy P, Glubb D, Rostamianfar A, Bolla MK, Wang Q, Tyrer J, Dicks E, Lee A, Wang Z, Allen J, Keeman R,

Eilber U, French JD, et al (2017) Association analysis identifies 65 new breast cancer risk loci. *Nature* **551:** 92–94

Nelson CP, Goel A, Butterworth AS, Kanoni S, Webb TR, Marouli E, Zeng L, Ntalla I, Lai FY, Hopewell JC, Giannakopoulou O, Jiang T, Hamby SE, Di Angelantonio E, Assimes TL, Bottinger EP, Chambers JC, Clarke R, Palmer CNA, Cubbon RM, et al (2017) Association analyses based on false discovery rate implicate new loci for coronary artery disease. *Nat. Genet.* **49:** 1385–1391

Nielsen JB, Thorolfsdottir RB, Fritsche LG, Zhou W, Skov MW, Graham SE, Herron TJ, McCarthy S, Schmidt EM, Sveinbjornsson G, Surakka I, Mathis MR, Yamazaki M, Crawford RD, Gabrielsen ME, Skogholt AH, Holmen OL, Lin M, Wolford BN, Dey R, et al (2018) Biobank-driven genomic discovery yields new insight into atrial fibrillation biology. *Nat. Genet.* **50:** 1234–1239

Pulikkan JA, Hegde M, Ahmad HM, Belaghzal H, Illendula A, Yu J, O'Hagan K, Ou J, Muller-Tidow C, Wolfe SA, Zhu LJ, Dekker J, Bushweller JH & Castilla LH (2018) CBFβ-SMMHC inhibition triggers apoptosis by disrupting MYC chromatin dynamics in acute myeloid leukemia. *Cell* **174:** 172-186.e21

Ripke S, Neale BM, Corvin A, Walters JTR, Farh KH, Holmans PA, Lee P, Bulik-Sullivan B, Collier DA, Huang H, Pers TH, Agartz I, Agerbo E, Albus M, Alexander M, Amin F, Bacanu SA, Begemann M, Belliveau RA, Bene J, et al (2014) Biological insights from 108 schizophrenia-associated genetic loci. *Nature* **511:** 421–427

Schmidt SF, Larsen BD, Loft A, Nielsen R, Madsen JGS & Mandrup S (2015) Acute TNF-induced repression of cell identity genes is mediated by NFκB-directed redistribution of cofactors from super-enhancers. *Genome Res.* **25:** 1281–1294

Shannon P, Markiel A, Ozier O, Baliga NS, Wang JT, Ramage D, Amin N, Schwikowski B & Ideker T (2003) Cytoscape: A software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13:** 2498–2504

Teslovich TM, Musunuru K, Smith A V., Edmondson AC, Stylianou IM, Koseki M, Pirruccello JP, Ripatti S, Chasman DI, Willer CJ, Johansen CT, Fouchier SW, Isaacs A, Peloso GM, Barbalic M, Ricketts SL, Bis JC, Aulchenko YS, Thorleifsson G, Feitosa MF, et al (2010) Biological, clinical and population relevance of 95 loci for blood lipids. *Nature* **466:** 707–713