nature methods

ANALYSIS

https://doi.org/10.1038/s41592-019-0509-5

OPEN

Assessment of network module identification across complex diseases

Sarvenaz Choobdar[1,2,20], Mehmet E. Ahsen[3,117], Jake Crawford[4,117], Mattia Tomasoni[1,2], Tao Fang[5], David Lamparter[1,2,6], Junyuan Lin[7], Benjamin Hescott[8], Xiaozhe Hu[7], Johnathan Mercer[9,10], Ted Natoli[11], Rajiv Narayan[11], The DREAM Module Identification Challenge Consortium[12], Aravind Subramanian[11], Jitao D. Zhang[5], Gustavo Stolovitzky[3,13], Zoltán Kutalik[2,14], Kasper Lage[9,10,15], Donna K. Slonim[4,16], Julio Saez-Rodriguez[17,18], Lenore J. Cowen[4,7], Sven Bergmann[1,2,19,21*] and Daniel Marbach[1,2,5,21*]

Many bioinformatics methods have been proposed for reducing the complexity of large gene or protein networks into relevant subnetworks or modules. Yet, how such methods compare to each other in terms of their ability to identify disease-relevant modules in different types of network remains poorly understood. We launched the 'Disease Module Identification DREAM Challenge', an open competition to comprehensively assess module identification methods across diverse protein–protein interaction, signaling, gene co-expression, homology and cancer-gene networks. Predicted network modules were tested for association with complex traits and diseases using a unique collection of 180 genome-wide association studies. Our robust assessment of 75 module identification methods reveals top-performing algorithms, which recover complementary trait-associated modules. We find that most of these modules correspond to core disease-relevant pathways, which often comprise therapeutic targets. This community challenge establishes biologically interpretable benchmarks, tools and guidelines for molecular network analysis to study human disease biology.

Complex diseases involve many genes and molecules that interact within cellular networks[1–3]. Advances in experimental and computational techniques enable both physical interaction networks (for example, protein–protein interaction, signaling and regulatory networks) and functional networks (for example, co-expression, genetic and single-cell interaction networks) to be mapped with increasing accuracy. A key problem in the analysis of these networks is the identification of functional units, called modules or pathways[4]. It is well-known that molecular networks have a high degree of modularity (that is, subsets of nodes are more densely connected than expected by chance), and that individual modules often comprise genes or proteins that are involved in the same biological functions[5]. Moreover, biological networks are typically too large to be examined as a whole. Consequently, module identification is often a crucial step to gain biological insights from network data[6–9].

Module identification, also called community detection or graph clustering, is a classic problem in network science for which a wide range of methods have been proposed[10]. These methods are typically assessed on in silico generated benchmark graphs[11]. However, how well different approaches uncover biologically relevant modules in real molecular networks remains poorly understood. Crowdsourced data competitions (known as challenges) have proved to be an effective means to rigorously assess methods and foster collaborative communities. The Dialogue on Reverse Engineering and Assessment (DREAM) is a community-driven initiative promoting data challenges in biomedicine (http://dreamchallenges.org). DREAM challenges have established robust methodologies for diverse problems, including the inference of molecular networks[12,13]. But, so far there has been no community effort addressing the downstream analysis of reconstructed networks.

Here we present the Disease Module Identification DREAM Challenge, where over 400 participants from all over the world predicted disease-relevant modules in diverse gene and protein networks (Fig. 1; https://synapse.org/modulechallenge). We introduce community-driven benchmarks, dissect top-performing approaches and explore the biology of discovered network modules.

[1]Department of Computational Biology, University of Lausanne, Lausanne, Switzerland. [2]Swiss Institute of Bioinformatics, Lausanne, Switzerland. [3]Icahn Institute for Genomics and Multiscale Biology and Department of Genetics and Genomic Sciences, Icahn School of Medicine at Mount Sinai, New York, NY, USA. [4]Department of Computer Science, Tufts University, Medford, MA, USA. [5]Roche Pharma Research and Early Development, Pharmaceutical Sciences, Roche Innovation Center Basel, F. Hoffmann-La Roche Ltd, Basel, Switzerland. [6]Verge Genomics, San Francisco, CA, USA. [7]Department of Mathematics, Tufts University, Medford, MA, USA. [8]College of Computer and Information Science, Northeastern University, Boston, MA, USA. [9]Department of Surgery, Massachusetts General Hospital, Harvard Medical School, Boston, MA, USA. [10]Stanley Center at the Broad Institute of MIT and Harvard, Cambridge, MA, USA. [11]Broad Institute of MIT and Harvard, Cambridge, MA, USA. [12]Full list of members appears at the end of the paper. [13]IBM T.J. Watson Research Center, Yorktown Heights, NY, USA. [14]University Institute of Primary Care and Public Health, University of Lausanne, Lausanne, Switzerland. [15]Institute for Biological Psychiatry, Mental Health Center Sct. Hans, University of Copenhagen, Roskilde, Denmark. [16]Department of Immunology, Tufts University School of Medicine, Boston, MA, USA. [17]Institute for Computational Biomedicine, Faculty of Medicine, Heidelberg University, Bioquant, Heidelberg, Germany. [18]RWTH Aachen University, Faculty of Medicine, Joint Research Center for Computational Biomedicine, Aachen, Germany. [19]Department of Integrative Biomedical Sciences, University of Cape Town, Cape Town, South Africa. [20]These authors contributed equally: Sarvenaz Choobdar, Mehmet E. Ahsen, Jake Crawford. [21]These authors jointly supervised this work: Sven Bergmann, Daniel Marbach. *e-mail: sven.bergmann@unil.ch; daniel.marbach.dm1@roche.com

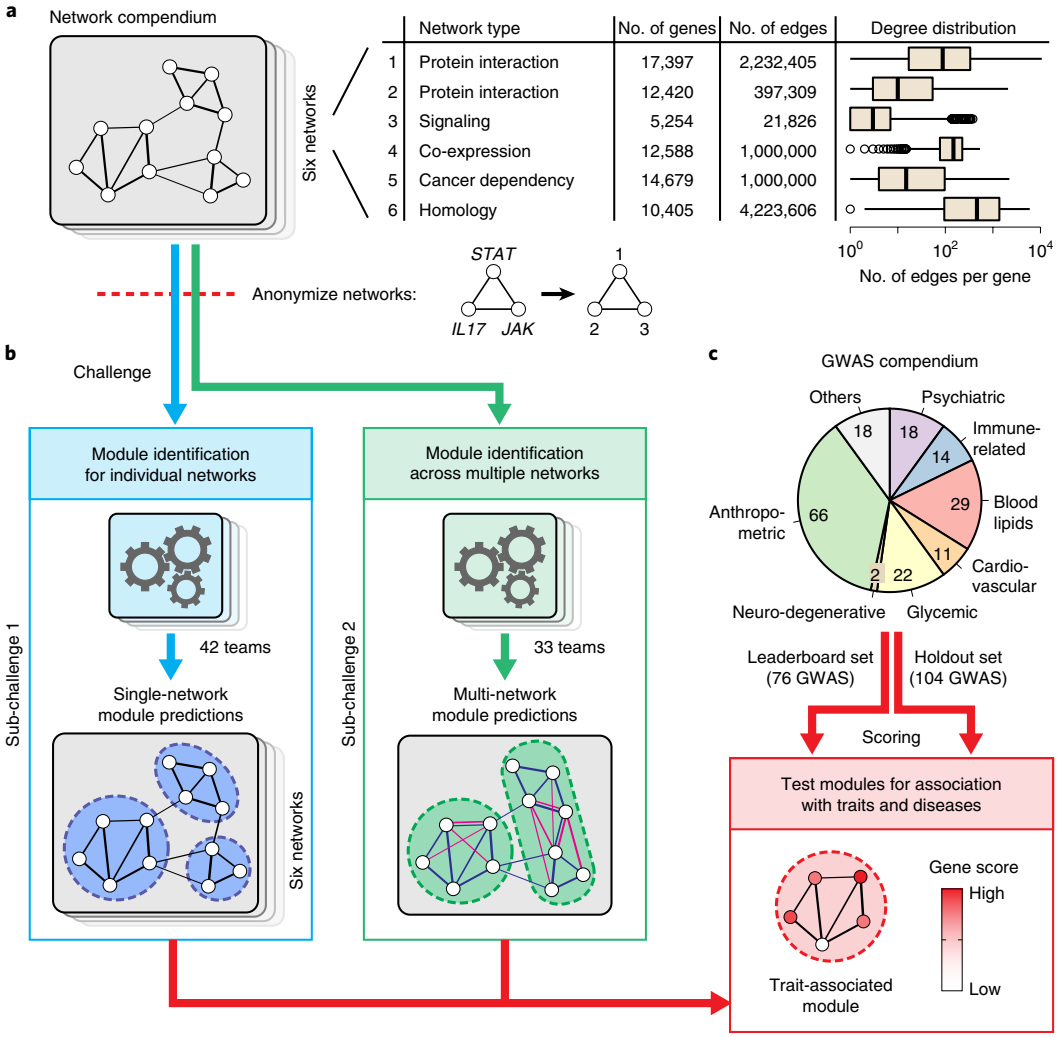

Fig. 1 | The Disease Module Identification DREAM Challenge. a, Network types included in the challenge. Throughout the paper, boxplot center lines show the median, box limits show upper and lower quartiles, whiskers show 1.5× interquartile range and points show outliers. **b**, Outline of the challenge. **c**, Outline of the scoring.

Results

A community challenge to assess network module identification methods. We developed a panel of diverse, human molecular networks for the challenge, including custom versions of two protein–protein interaction and a signaling network extracted from the STRING[14], InWeb[15] and OmniPath[16] databases, a co-expression network inferred from 19,019 tissue samples from the Gene Expression Omnibus (GEO) repository[17], a network of genetic dependencies derived from loss-of-function screens in 216 cancer cell lines[18,19] and a homology-based network built from phylogenetic patterns across 138 eukaryotic species[20,21] (Methods). We included different types of network, which also vary in their size and structural properties, to provide a heterogeneous benchmark resource (Fig. 1a).

Each network was generated specifically for the challenge and released in anonymized form (that is, we did not disclose the gene names and the identity of the networks), thus enabling rigorous 'blinded' assessment. That is, participants could only use unsupervised clustering algorithms, which rely exclusively on the network structure and do not depend on additional biological information such as known disease genes.

We solicited participation in two types of module identification challenge (Fig. 1b). In Sub-challenge 1, solvers were asked to run module identification on each of the provided networks individually (single-network module identification). In Sub-challenge 2, the networks were reanonymized in a way that the same gene identifier represented the same gene across all six networks. Solvers were then asked to identify a single set of non-overlapping modules by sharing information across the six networks (multi-network module identification), which allowed us to assess the potential improvement in performance offered by emerging multi-network methods compared to single-network methods. In both sub-challenges, predicted modules had to be non-overlapping and comprise between 3 and 100 genes.

The challenge was run using the open-science Synapse platform[22]. Over a 2-month period, participants could make a limited number of submissions and see the performance of all teams on a real-time leaderboard. In the final round, teams could make a single submission for each sub-challenge, which had to include method descriptions and code for reproducibility.

Biologically interpretable scoring of modules based on trait associations. Evaluation of predicted modules is challenging because there is no ground truth of 'correct' modules in molecular networks. Here, we introduce a framework to empirically assess modules based on their association with complex traits and diseases using genome-wide association studies (GWAS) data (Fig. 1c).

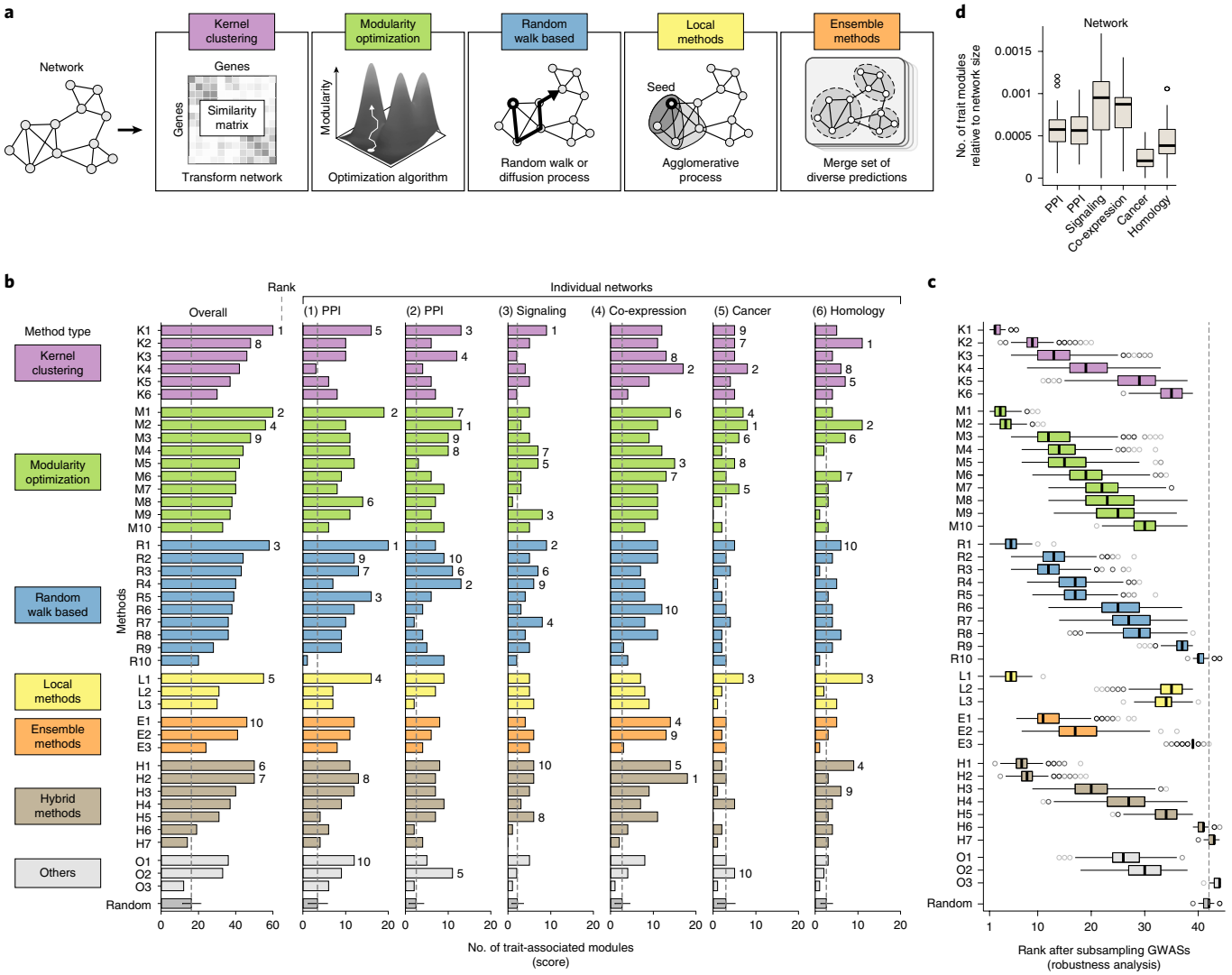

Fig. 2 | Assessment of module identification methods. a, Main types of module identification approach used in the challenge. **b**, Final scores of the 42 module identification methods applied in Sub-challenge 1 for each of the six networks, as well as the overall score summarizing performance across networks (evaluated using the holdout GWAS set at 5% FDR; method IDs are defined in Supplementary Table 2). Ranks are indicated for the top ten methods. The last row shows the mean performance of 17 random modularizations of the networks (error bars show the standard deviation). **c**, Robustness of the overall ranking was evaluated by subsampling the GWAS set used for evaluation 1,000 times. For each method, the resulting distribution of ranks is shown as a boxplot. **d**, Number of trait-associated modules per network. Boxplots show the number of trait-associated modules across the 42 methods, normalized by the size of the respective network.

Since GWAS are based on data completely different from those used to construct the networks, they can provide independent support for biologically relevant modules. To cover diverse molecular processes, we have compiled a large collection of 180 GWAS datasets (Supplementary Table 1). Predicted modules were scored on each GWAS using the Pascal tool[23], which aggregates trait-association *P* values of single nucleotide polymorphisms (SNPs) at the level of genes and modules. Modules that scored significantly for at least one GWAS trait were called trait-associated. Finally, the score of each challenge submission was defined as the total number of its trait-associated modules (at 5% false discovery rate (FDR), see Methods).

To detect potential overfitting, the collection of 180 GWASs was split into a leaderboard set for scoring the leaderboard submissions and a separate holdout set for scoring the single, final submission of each team. Results reported below are from the final evaluation on the holdout set.

Top methods from different categories achieve comparable performance. The community contributed 42 single-network and 33 multi-network module identification methods in the final round of the two sub-challenges. We first discuss the single-network methods (Sub-challenge 1), which we grouped into seven broad categories: (1) kernel clustering, (2) modularity optimization, (3) random-walk-based, (4) local methods, (5) ensemble methods, (6) hybrid methods and (7) other methods (Fig. 2a and Supplementary Table 2). While many teams adapted existing algorithms for community detection, other teams—including the best performers—developed novel approaches. The top five methods achieved comparable performance with scores between 55 and 60, while the remaining methods did not get to scores above 50 (Fig. 2b). Although the scores were close, the top-scoring method K1 (method IDs are defined in Supplementary Table 2) performed more robustly than runner-up methods, achieving the best score: (1) in the leaderboard and final rounds (Supplementary Table 3); (2) at varying FDR

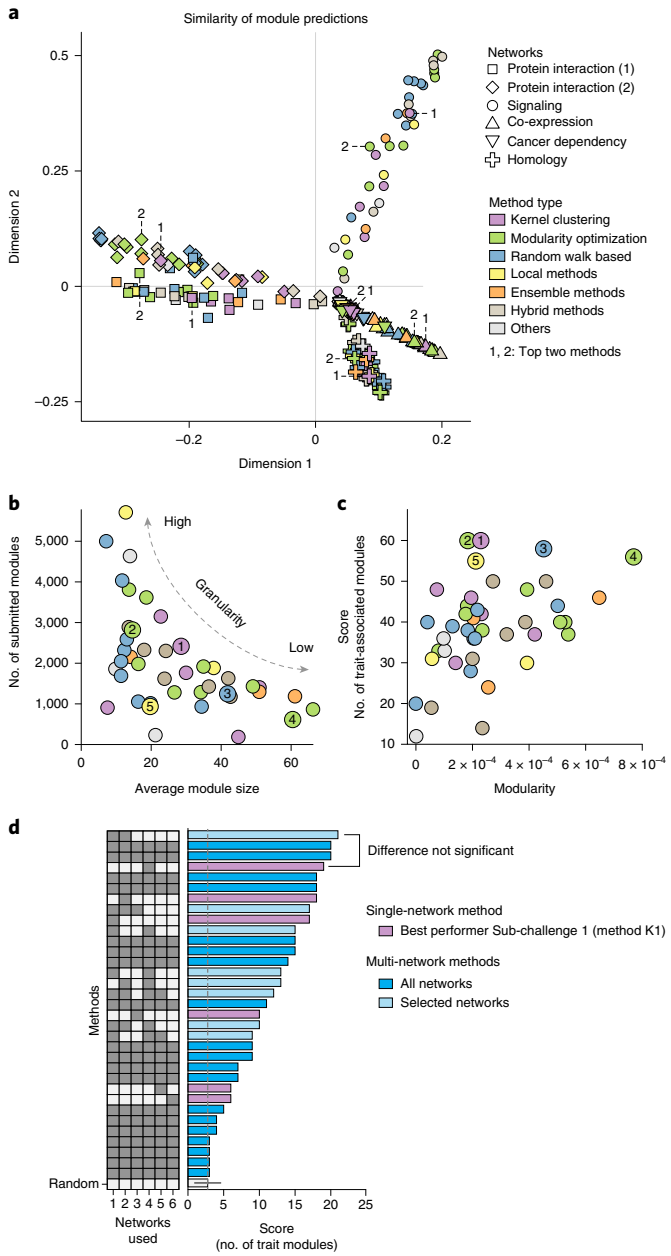

Fig. 3 | Complementarity of module predictions from different methods and networks. a, Similarity of module predictions from different methods (color) and networks (shape). The closer two points are in the plot, the more similar the corresponding module predictions (multidimensional scaling, see Methods). The top two methods are highlighted for each network. **b**, Total number of predicted modules versus average module size for each method (same color scheme as in **a**). The top five methods (numbered) produced modular decompositions of varying granularity. **c**, Challenge score (number of trait-associated modules) versus modularity is shown for each of the 42 methods (same color scheme as in **a**). Modularity is a topological quality metric for modules based on the fraction of within-module edges[43]. **d**, Final scores of multi-network module identification methods in Sub-challenge 2 (evaluated using the holdout GWAS set at 5% FDR). For comparison, the overall best-performing method from Sub-challenge 1 is also shown (method K1, purple). Teams used different combinations of the six challenge networks for their multi-network predictions (shown on the left). The difference between the top single-network module predictions and the top multi-network module predictions is not significant when subsampling the GWASs (Bayes factor < 3, Supplementary Fig. 5). The last row shows the mean performance of 17 random modularizations of the networks (error bars show standard deviation).

cutoffs (Supplementary Fig. 1) and (3) on subsamples of the GWAS holdout set (Fig. 2c).

The top teams used different approaches: the best performers (K1) developed a novel kernel approach leveraging a diffusion-based distance metric[24,25] and spectral clustering[26]; the runner-up team (M1) extended different modularity optimization methods with a resistance parameter that controls the granularity of modules[27] and the third-ranking team (R1) used a random-walk method based on Markov clustering with locally adaptive granularity to balance module sizes[28] (Methods). These teams further collaborated after the challenge to bundle their methods in a user-friendly tool[29].

Four different method categories are represented among the top five performers, suggesting that no single approach is inherently superior for module identification. Rather, performance depends on the specifics of each individual method, including the strategy used to define the resolution (the number and size of modules). Preprocessing steps also affected performance: many of the top teams first sparsified the networks by discarding weak edges. A notable exception is the top method (K1), which performed robustly without any preprocessing of the networks.

The challenge also allows us to explore how informative different types of molecular network are for finding modules underlying complex traits. In absolute numbers, methods recovered the most trait-associated modules in the co-expression and protein–protein interaction networks (Supplementary Fig. 1). However, relative to the network size, the signaling network contained the most trait modules (Fig. 2d). These results are consistent with the importance of signaling pathways for many of the considered traits and diseases. The cancer cell line and homology-based networks, on the other hand, were less relevant for the traits in our GWAS compendium and thus comprised only a few trait modules.

Complementarity of different module identification approaches. To test whether predictions from different methods and networks tend to capture the same or complementary modules, we applied a pairwise similarity metric to all 252 module predictions from Sub-challenge 1 (42 methods × 6 networks, see Methods). We find that similarity of module predictions is primarily driven by the underlying network and top-performing methods do not converge to similar module predictions (Fig. 3a and Supplementary Fig. 2). Indeed, only 46% of trait modules are recovered by multiple methods with good agreement in a given network (high overlap or submodules, Supplementary Fig. 2). Across different networks, the number of recovered modules with substantial overlap is even lower (17%). Thus, the majority of trait modules are method- and network-specific.

The modules produced by different methods also vary in terms of their structural properties. For example, submissions included between 16 and 1,552 modules per network, with an average module size ranging from 7 to 66 genes. Neither the number nor the size of submitted modules correlates with performance (Fig. 3b and Supplementary Figs. 3 and 4). Thus, there is no single optimal granularity for a given network; rather, different methods captured trait-relevant modules at varying levels of granularity. Topological quality metrics of modules such as modularity showed only modest correlation with the challenge score (Pearson's $r = 0.45$, Fig. 3c), highlighting the need for biologically interpretable assessment of module identification methods.

Multi-network module identification methods did not provide added power. In Sub-challenge 2, teams submitted a single modularization of the genes, for which they could leverage information from all six networks together. While some teams developed dedicated multi-network (multi-layer) community detection methods[30,31], the majority of teams first merged the networks and then applied single-network methods (Methods).

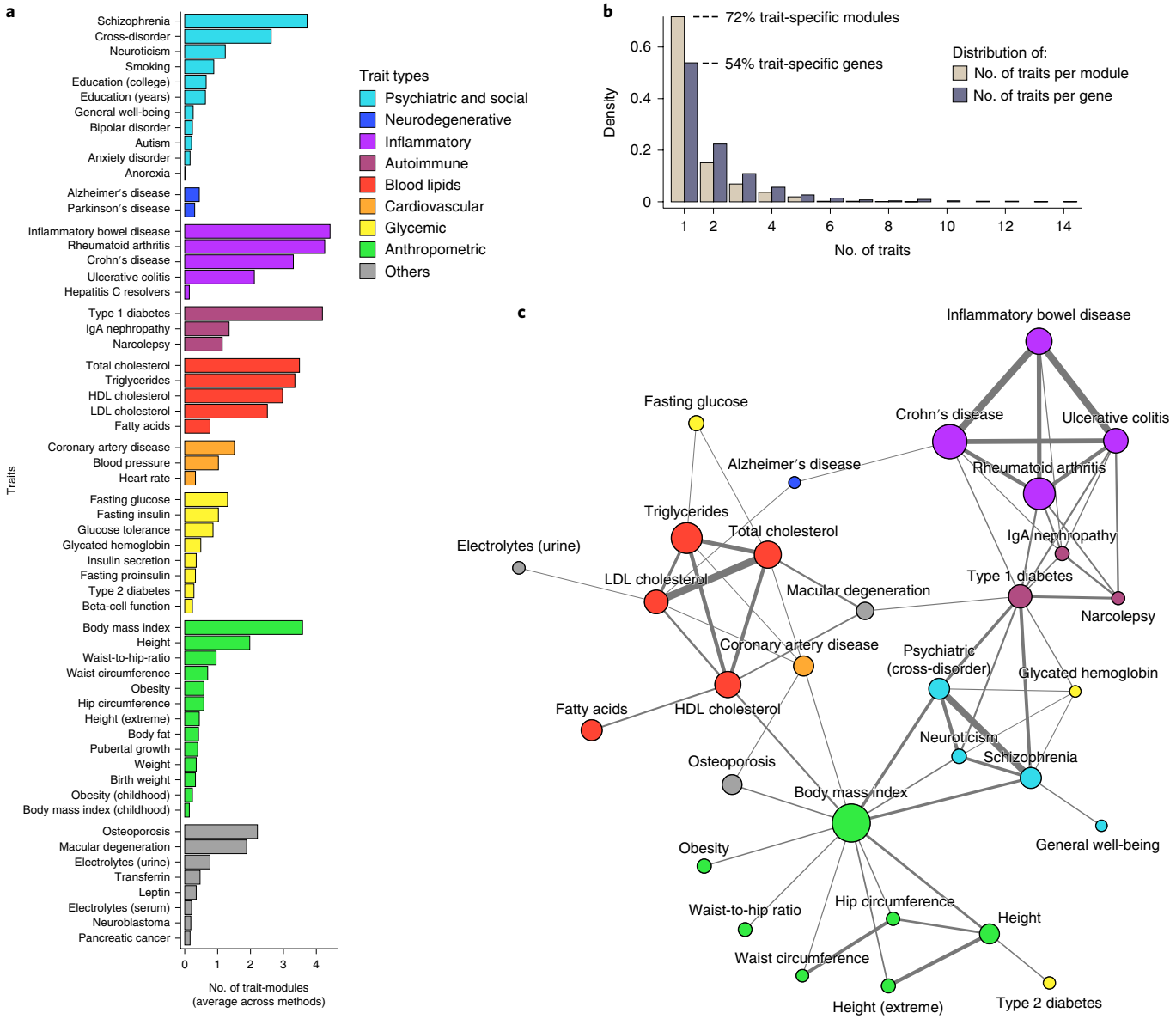

Fig. 4 | Overlap between modules associated with different traits and diseases. a, Average number of trait-associated modules identified by challenge methods for each trait in Sub-challenge 1. For traits where multiple GWASs were available, results for the best-powered study are shown. HDL, high-density lipoprotein; LDL, low-density lipoprotein. **b**, Histograms showing the number of distinct traits per trait-associated module (brown) and gene (gray). **c**, Trait network showing similarity between GWAS traits based on overlap of associated modules (force-directed graph layout). Node size corresponds to the number of genes in trait-associated modules and edge width corresponds to the degree of overlap (Jaccard index, only edges for which the overlap is significant are shown (Bonferroni-corrected hypergeometric $P < 0.05$, see Methods)). Traits without any edges are not shown.

It turned out to be very difficult to effectively leverage complementary networks for module identification. While three teams achieved marginally higher scores than single-network module predictions (Fig. 3d), the difference is not significant when subsampling the GWASs (Bayes factor < 3, Supplementary Fig. 5). Moreover, the best-scoring team simply merged the two protein–protein interaction networks (the two most similar networks, Supplementary Fig. 6), discarding the other types of network. Since no significant improvement over single-network methods was achieved, the winning position of Sub-challenge 2 was declared vacant.

Integration of challenge submissions leads to robust consensus modules. To derive consensus modules from team submissions, we integrated module predictions from different methods in a consensus matrix C, where each element c_{ij} is proportional to the number of methods that put gene i and j together in the same module. The consensus matrix was then clustered using the top-performing module identification method from the challenge (Methods).

We generated consensus modules for each challenge network by applying this approach to the top 21 (50%) of methods from the leaderboard round. The score of the consensus modules outperforms the top individual method predictions in both sub-challenges (Supplementary Figs. 1 and 5). However, when applied to fewer methods, the performance of the consensus drops (Supplementary Fig. 7). We conclude that the consensus approach is only suitable in a challenge context, since applying such a large number of methods is not practical for users. Indeed, we found that the total number of trait-associated modules when considering the modules from top-performing methods individually is higher than the number of modules resulting from their consensus (Supplementary Fig. 8).

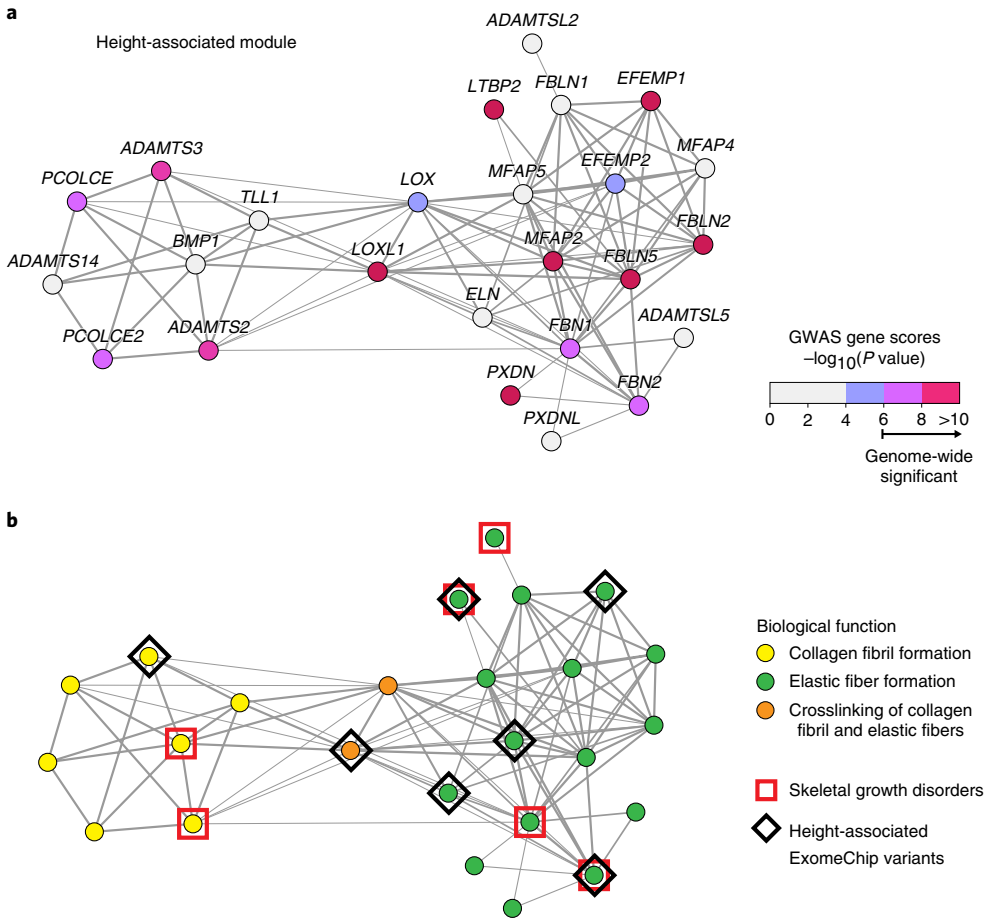

Fig. 5 | Support for trait-module genes in diverse datasets. a, Example module from the consensus analysis in the STRING protein–protein interaction network (force-directed graph layout). The module is associated to height ($n = 25$ genes, FDR-corrected Pascal $P = 0.005$, see Methods). Color indicates Pascal GWAS gene scores (Methods). The module includes genes that are genome-wide significant (magenta and pink) as well as genes that do not reach the genome-wide significance threshold, but are predicted to be involved in height due to their module membership (blue and gray). **b**, Member genes of the height-associated module are supported by independent datasets: 24% of module genes are implicated in monogenic skeletal growth disorders (red squares, enrichment $P = 7.5 \times 10^{-4}$ (one-sided Fisher's exact test)) and 28% of module genes have coding variants associated to height in an ExomeChip study published after the challenge[32] (black diamonds, enrichment $P = 1.9 \times 10^{-6}$). The form of this module follows its function: two submodules comprise proteins involved in collagen fibril (yellow) and elastic fiber formation (green), while the proteins that link these submodules (orange) indeed have the biological function of crosslinking collagen fibril and elastic fibers.

Network modules reveal trait-specific and shared pathways. We next sought to explore biological properties of predicted modules. The most trait-associated modules were found for immune-related, psychiatric, blood cholesterol and anthropometric traits, for which high-powered GWAS are available that are known to show strong pathway enrichment (Fig. 4a). Significant GWAS loci often show association to multiple traits. Across our GWAS compendium, we found that 46% of trait-associated genes but only 28% of trait-associated modules are associated with multiple traits (Fig. 4b). Thus, mapping genes onto network modules may help in disentangling trait-specific pathways at shared loci.

We next asked which traits are similar in terms of the implicated network components. To this end, we considered the union of all genes within network modules associated with a given trait (trait-module genes). We then evaluated the pairwise similarity of traits based on the significance of the overlap between the respective trait-module genes (Methods). Trait relationships thus inferred are consistent with known biology and comorbidities between the considered traits and diseases (Fig. 4c).

Trait-associated modules implicate core disease genes and pathways. Due to linkage disequilibrium (LD), many genes that show association to a trait may not causally influence it. A key question is whether our trait modules, and the corresponding genes, are correctly predicted as being biologically or therapeutically relevant for that trait or disease. We thus sought to evaluate trait modules using additional independent datasets, including ExomeChip data, monogenic disease genes, functional annotations and known therapeutic targets.

We first consider a module from the consensus analysis that shows association to height—a classic polygenic trait—as an example (Fig. 5a). Forty percent of genes in this module either comprise coding variants associated to height in an independent ExomeChip study[32] or are known to be implicated in monogenic skeletal growth disorders, supporting their causal role in the phenotype (Fig. 5b). Gene Ontology (GO) annotations further show that this module consists of two submodules comprising extracellular matrix proteins responsible for, respectively, collagen fibril and elastic fiber formation—pathways that are essential for growth (Fig. 5b). Indeed, mutations of homologous genes in mouse lead to abnormal elastic fiber morphology (Supplementary Fig. 9). Some of the genes supported by these additional datasets did not show signal in the GWAS used to discover the module. For example, the module gene *BMP1* (*Bone Morphogenic Protein 1*)

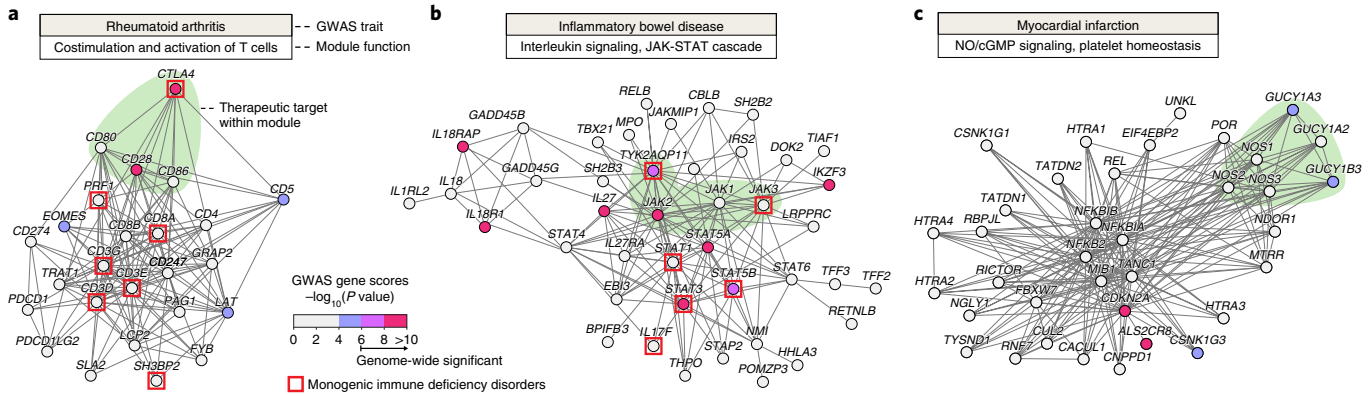

Fig. 6 | Example trait modules comprising therapeutically relevant pathways. a–**c**, The modules are from the STRING protein–protein interaction networks and were generated using the consensus method. Node colors correspond to Pascal gene scores in the respective GWAS (Methods). For the two inflammatory disorders (**a**,**b**), red squares indicate genes causing monogenic immunodeficiency disorders (enrichment *P* values of 4.1×10^{-8} and 1.2×10^{-6}, respectively (one-sided Fisher's exact test)). **a**, Module associated with rheumatoid arthritis ($n=25$ genes, FDR-corrected Pascal $P=0.04$) that is involved in T cell activation. A costimulatory pathway is highlighted green, T cell response is regulated by activating (*CD28*) and inhibitory (*CTLA4*) surface receptors, which bind B7 family ligands (*CD80* and *CD86*) expressed on the surface of activated antigen-presenting cells. The therapeutic agent CTLA4-Ig binds and blocks B7 ligands, thus inhibiting T cell response. **b**, Cytokine signaling module associated with inflammatory bowel disease ($n=42$ genes, FDR-corrected Pascal $P=0.0006$). The module includes the four known Janus kinases (*JAK1-3* and *TYK2*, highlighted green), which are engaged by cytokine receptors to mediate activation of specific transcription factors (*STATs*). Inhibitors of JAK–STAT signaling are being tested in clinical trials for both ulcerative colitis and Crohn's disease[44]. **c**, Module associated with myocardial infarction ($n=36$ genes, FDR-corrected Pascal $P=0.0001$) comprising two main components of the NO/cGMP signaling pathway (endothelial nitric oxide synthases (*NOS1-3*) and soluble guanylate cyclases (*GUCY1A2*, *GUCY1A3* and *GUCY1B3*), highlighted green), a key therapeutic target for cardiovascular disease[45].

causes osteogenesis imperfecta, which is associated with short stature. Yet, *BMP1* does not show association to height in current GWAS and ExomeChip studies, demonstrating how network modules can implicate additional disease-relevant pathway genes (see Supplementary Figs. 10 and 11 for a comprehensive evaluation of prioritized trait-module genes).

To evaluate more generally whether trait-associated modules correspond to generic or disease-specific pathways, we systematically tested modules for functional enrichment of GO annotations, mouse mutant phenotypes and pathway databases. We further selected the most representative annotations for each module using a regression framework[33] (Methods). We find that the majority of trait modules reflect core disease-specific pathways. For example, in the STRING protein–protein interaction network only 33% of trait modules from the consensus analysis have generic functions; the remaining 66% of trait modules correspond to core disease-specific pathways, some of which are therapeutic targets (Supplementary Fig. 12 and Supplementary Table 4). Examples include a module associated with rheumatoid arthritis that comprises the B7:CD28 costimulatory pathway required for T cell activation, which is blocked by an approved drug (Fig. 6a); a module associated with inflammatory bowel disease corresponding to cytokine signaling pathways mediated by Janus kinases (JAKs), which are therapeutic targets (Fig. 6b) and a module associated with myocardial infarction that includes the NO/cGMP signaling cascade, which plays a key role in cardiovascular pathophysiology and therapeutics (Fig. 6c). We further applied our pipeline to a GWAS on IgA nephropathy (IgAN) obtained after the challenge, an autoimmune disorder with poorly understood etiology[34]. We find two IgAN-associated modules, which prioritize novel candidate genes involved in NF-kB signaling, complement and coagulation cascades, demonstrating how our challenge resources can be used for network-based analysis of new GWAS datasets (Supplementary Fig. 13).

Discussion

As large-scale network data become pervasive in many fields, robust tools for detection of network communities are of critical importance. With this challenge we have conducted an impartial and interpretable assessment of module identification methods on biological networks, providing much-needed guidance for users. While it is important to keep in mind that the exact ranking of methods is specific to the task and datasets considered, the resulting collection of top-performing module identification tools and methodological insights will be broadly useful for modular analysis of complex networks in biology and other domains.

Kernel clustering, modularity optimization, random-walk-based and local methods were all represented among the top performers, suggesting that no single type of approach is inherently superior. In contrast, the popular weighted gene co-expression network analysis (WGCNA) method[7] did not perform competitively, likely because it relies on hierarchical clustering, which—unlike the top-performing approaches—was not specifically designed for network clustering. Moreover, while most published studies in network biology rely on a single clustering method, the results of this challenge demonstrate the value of applying multiple methods from different categories to detect complementary types of module.

The challenge further emphasized the importance of the resolution (size and number of modules). Biological networks typically have a hierarchical modular structure, which implies that disease-relevant pathways can be captured at different levels of granularity[35]. Indeed, we found that there is no intrinsic optimal resolution for a given network; rather, it depends on the type of method used (Supplementary Fig. 3). Top-performing challenge methods allowed the resolution to be tuned, enabling users to explore different module granularities.

Our analysis showed that signaling, protein–protein interaction and co-expression networks comprise complementary trait-relevant modules. Considering different types of network is thus clearly advantageous. However, multi-network module identification methods that attempted to reveal integrated modules across these networks failed to significantly improve predictions compared to methods that considered each network individually. These results are contrary to the common assumption that multi-network integration improves module predictions. However, this finding

remains specific to the challenge networks, which may not have been sufficiently related. Indeed, the best result was obtained by the team merging only the two most related networks (the two protein–protein interaction networks), and the runner-up team confirmed in a post hoc analysis that focusing on networks with similar modular structures improved their results[36]. Multi-network methods may thus be better suited to networks that are more closely related, possibly from the same tissue- and disease-context[37].

On the basis of these challenge findings, we make the following practical recommendations for module identification: (1) methods from diverse categories should be applied to identify complementary modules (for example, the top three challenge methods, which are available in a user-friendly tool[29]); (2) the resulting modules from different methods should be used as is, without forming a consensus (consensus modules were only competitive when integrating over 20 methods); (3) whenever possible, diverse networks should be leveraged (for example, co-expression, protein–protein interaction and signaling), as they comprise complementary types of module; (4) module identification methods should first be applied to each network individually, without merging the networks and (5) multi-network methods may be used to reveal modules in layered networks, but performance depends heavily on whether networks are sufficiently related.

There is continuing debate over the value of GWASs for revealing disease mechanisms and therapeutic targets. Indeed, the number of GWAS hits continues to grow as sample sizes increase, but the bulk of these hits does not correspond to core genes with specific roles in disease etiology[38]. While thousands of genes may show association to a given disease, we have demonstrated that much more specific disease modules comprising only dozens of genes can be identified within networks. These modules prioritize novel candidate genes, reveal pathway-level similarity between diseases and correspond to core disease pathways in the majority of cases. This is consistent with the robustness of biological networks: presumably, the many genes that influence disease indirectly are broadly distributed across network modules, while core disease genes cluster in specific pathways underlying pathophysiological processes[39,40].

In this study we used generic networks, not context-specific networks, because the focus was on method assessment across diverse disorders. In the near future, we expect much more detailed maps of tissue- and disease-specific networks, along with diverse high-powered genetic datasets, to become available[2,41,42]. We hope that our challenge resources will provide a foundation to dissect these networks and reveal pathways implicated in human disease.

Online content

Any methods, additional references, Nature Research reporting summaries, source data, statements of code and data availability and associated accession codes are available at https://doi.org/10.1038/s41592-019-0509-5.

Received: 29 November 2018; Accepted: 10 July 2019;
Published online: 30 August 2019

References

1. Schadt, E. E. Molecular networks as sensors and drivers of common human diseases. *Nature* **461**, 218–223 (2009).
2. Marbach, D. et al. Tissue-specific regulatory circuits reveal variable modular perturbations across complex diseases. *Nat. Methods* **13**, 366–370 (2016).
3. Bonder, M. J. et al. Disease variants alter transcription factor levels and methylation of their binding sites. *Nat. Genet.* **49**, 131–138 (2017).
4. Califano, A., Butte, A. J., Friend, S., Ideker, T. & Schadt, E. Leveraging models of cell regulation and GWAS data in integrative network-based association studies. *Nat. Genet.* **44**, 841–847 (2012).
5. Hartwell, L. H., Hopfield, J. J., Leibler, S. & Murray, A. W. From molecular to modular cell biology. *Nature* **402**, C47–C52 (1999).
6. Chen, Y. et al. Variations in DNA elucidate molecular networks that cause disease. *Nature* **452**, 429–435 (2008).
7. Langfelder, P. & Horvath, S. WGCNA: an R package for weighted correlation network analysis. *BMC Bioinforma.* **9**, 559 (2008).
8. Pe'er, D., Regev, A., Elidan, G. & Friedman, N. Inferring subnetworks from perturbed expression profiles. *Bioinformatics* **17**(Suppl 1), S215–S224 (2001).
9. Huttenhower, C. et al. Exploring the human genome with functional maps. *Genome Res.* **19**, 1093–1106 (2009).
10. Fortunato, S. & Hric, D. Community detection in networks: a user guide. *Phys. Rep.* **659**, 1–44 (2016).
11. Girvan, M. & Newman, M. E. J. Community structure in social and biological networks. *Proc. Natl Acad. Sci. USA* **99**, 7821–7826 (2002).
12. Hill, S. M. et al. Inferring causal molecular networks: empirical assessment through a community-based effort. *Nat. Methods* **13**, 310–318 (2016).
13. Marbach, D. et al. Wisdom of crowds for robust gene network inference. *Nat. Methods* **9**, 796–804 (2012).
14. Szklarczyk, D. et al. STRINGv10: protein–protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.* **43**, D447–D452 (2015).
15. Li, T. et al. A scored human protein–protein interaction network to catalyze genomic interpretation. *Nat. Methods* **14**, 61–64 (2017).
16. Türei, D., Korcsmáros, T. & Saez-Rodriguez, J. OmniPath: guidelines and gateway for literature-curated signaling pathway resources. *Nat. Methods* **13**, 966–967 (2016).
17. Subramanian, A. et al. A next generation connectivity map: L1000 platform and the first 1,000,000 profiles. *Cell* **171**, 1437–1452.e17 (2017).
18. Cowley, G. S. et al. Parallel genome-scale loss of function screens in 216 cancer cell lines for the identification of context-specific genetic dependencies. *Sci. Data* **1**, 140035 (2014).
19. Tsherniak, A. et al. Defining a cancer dependency map. *Cell* **170**, 564–576.e16 (2017).
20. Li, T. et al. GeNets: a unified web platform for network-based genomic analyses. *Nat. Methods* **15**, 543–546 (2018).
21. Li, Y., Calvo, S. E., Gutman, R., Liu, J. S. & Mootha, V. K. Expansion of biological pathways based on evolutionary inference. *Cell* **158**, 213–225 (2014).
22. Derry, J. M. J. et al. Developing predictive molecular maps of human disease through community-based modeling. *Nat. Genet.* **44**, 127–130 (2012).
23. Lamparter, D., Marbach, D., Rico, 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).
24. Cao, M. et al. Going the distance for protein function prediction: a new distance metric for protein interaction networks. *PloS ONE* **8**, e76339 (2013).
25. Cao, M. et al. New directions for diffusion-based network prediction of protein function: incorporating pathways with confidence. *Bioinforma.* **30**, i219–i227 (2014).
26. Ng, A. Y., Jordan, M. I. & Weiss, Y. In *Proc. 14th International Conference on Neural Information Processing Systems: Natural and Synthetic* (eds Dietterich, T. G., Becker, S. & Ghahramani, Z.) 849–856 (MIT Press, 2001).
27. Arenas, A., Fernández, A. & Gómez, S. Analysis of the structure of complex networks at different resolution levels. *New J. Phys.* **10**, 053039 (2008).
28. Satuluri, V., Parthasarathy, S. & Ucar, D. In *Proc. First ACM International Conference on Bioinformatics and Computational Biology* 247–256 (ACM, 2010).
29. Tomasoni, M. et al. MONET: a toolbox integrating top-performing methods for network modularisation. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/611418v4 (2019).
30. De Domenico, M., Lancichinetti, A., Arenas, A. & Rosvall, M. Identifying modular flows on multilayer networks reveals highly overlapping organization in interconnected systems. *Phys. Rev. X* **5**, 011027 (2015).
31. Didier, G., Brun, C. & Baudot, A. Identifying communities from multiplex biological networks. *PeerJ* **3**, e1525 (2015).
32. Marouli, E. et al. Rare and low-frequency coding variants alter human adult height. *Nature* **542**, 186–190 (2017).
33. Fang, T., Davydov, I., Marbach, D. & Zhang, J. D. Gene-set enrichment with regularized regression. Preprint at *bioRxiv* https://www.biorxiv.org/content/10.1101/659920v1 (2019).
34. Kiryluk, K. et al. Discovery of new risk loci for IgA nephropathy implicates genes involved in immunity against intestinal pathogens. *Nat. Genet.* **46**, 1187–1196 (2014).
35. Ravasz, E., Somera, A. L., Mongru, D. A., Oltvai, Z. N. & Barabási, A. L. Hierarchical organization of modularity in metabolic networks. *Science* **297**, 1551–1555 (2002).
36. Didier, G., Valdeolivas, A. & Baudot, A. Identifying communities from multiplex biological networks by randomized optimization of modularity. *F1000Research* **7**, 1042 (2018).
37. Krishnan, A., Taroni, J. N. & Greene, C. S. Integrative networks illuminate biological factors underlying gene–disease associations. *Curr. Genet. Med. Rep.* **4**, 155–162 (2016).
38. Boyle, E. A., Li, Y. I. & Pritchard, J. K. An expanded view of complex traits: from polygenic to omnigenic. *Cell* **169**, 1177–1186 (2017).
39. Pers, T. H. et al. Biological interpretation of genome-wide association studies using predicted gene functions. *Nat. Commun.* **6**, 5890 (2015).

40. Sullivan, P. F. & Posthuma, D. Biological pathways and networks implicated in psychiatric disorders. *Curr. Opin. Behav. Sci.* **2**, 58–68 (2015).
41. Greene, C. S. et al. Understanding multicellular function and disease with human tissue-specific networks. *Nat. Genet.* **47**, 569–576 (2015).
42. Delaneau, O. et al. Chromatin three-dimensional interactions mediate genetic effects on gene expression. *Science* **364**, eaat8266 (2019).
43. Newman, M. E. J. & Girvan, M. Finding and evaluating community structure in networks. *Phys. Rev. E* **69**, 026113 (2004).
44. Neurath, M. F. Current and emerging therapeutic targets for IBD. *Nat. Rev. Gastroenterol. Hepatol.* **14**, 269–278 (2017).
45. Kraehling, J. R. & Sessa, W. C. Contemporary approaches to modulating the nitric Oxide-cGMP pathway in cardiovascular disease. *Circ. Res.* **120**, 1174–1182 (2017).

Acknowledgements

The challenge was hosted on Sage Bionetwork's Synapse platform (https://synapse.org/). The computations were performed at the Vital-IT (http://www.vital-it.ch) Center for high-performance computing of the SIB Swiss Institute of Bioinformatics. This work was supported by the Swiss National Science Foundation (grant no. FN 310030_152724/1 to S.B. and grant no. FN 31003A-169929 to Z.K.), SystemsX.ch (grant no. SysGenetiX to S.B. and grant no. AgingX to Z.K.), the Swiss Institute of Bioinformatics (Z.K. and S.B.), the US National Science Foundation (grant no. DMS-1812503 to L.C. and X.H.) and the National Institutes of Health (grant no. R01 HD076140 to D.K.S.).

Author contributions

S.C., D.L., Z.K., G.S., J.M., K.L., J.S.-R., S.B. and D.M. conceived the challenge. S.C., G.S., J.S.-R., S.B. and D.M. organized the challenge. S.C. and D.M. performed team scoring. S.C., M.E.A., J.C., M.T., T.F., J.D.Z., D.K.S., L.J.C. and D.M. analyzed the results. J.M., T.N., R.N., A.S., K.L. and J.S.-R. constructed networks. J.C., J.L., B.H., X.H., D.K.S. and L.J.C. designed the top-performing method. The DREAM Module Identification Consortium provided data and performed module identification. S.B. and D.M. designed the study. D.M. prepared the manuscript. All authors discussed the results and implications and commented on the manuscript at all stages.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information is available for this paper at https://doi.org/10.1038/s41592-019-0509-5.

Reprints and permissions information is available at www.nature.com/reprints.

Correspondence and requests for materials should be addressed to S.B. or D.M.

Peer review information: Nicole Rusk was the primary editor on this article and managed its editorial process and peer review in collaboration with the rest of the editorial team.

Publisher's note: Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

The DREAM Module Identification Challenge Consortium

Fabian Aicheler[22], Nicola Amoroso[23,24], Alex Arenas[25], Karthik Azhagesan[26,27,28], Aaron Baker[29,30,31], Michael Banf[32], Serafim Batzoglou[33], Anaïs Baudot[34], Roberto Bellotti[23,24,35], Sven Bergmann[36,37], Keith A. Boroevich[38], Christine Brun[39,40], Stanley Cai[41,42,43], Michael Caldera[44], Alberto Calderone[45], Gianni Cesareni[45], Weiqi Chen[46], Christine Chichester[47], Sarvenaz Choobdar[36,37], Lenore Cowen[48,49], Jake Crawford[48], Hongzhu Cui[50], Phuong Dao[51], Manlio De Domenico[25,52], Andi Dhroso[50], Gilles Didier[34], Mathew Divine[22], Antonio del Sol[53,54,55], Tao Fang[56], Xuyang Feng[57], Jose C. Flores-Canales[58,59], Santo Fortunato[60], Anthony Gitter[29,30,31], Anna Gorska[61], Yuanfang Guan[62], Alain Guénoche[34], Sergio Gómez[25], Hatem Hamza[47], András Hartmann[53], Shan He[46], Anton Heijs[63], Julian Heinrich[22], Benjamin Hescott[64], Xiaozhe Hu[49], Ying Hu[65], Xiaoqing Huang[51], V. Keith Hughitt[66,67], Minji Jeon[68], Lucas Jeub[60], Nathan T. Johnson[50], Keehyoung Joo[59,69], InSuk Joung[58,59], Sascha Jung[53], Susana G. Kalko[53], Piotr J. Kamola[38], Jaewoo Kang[68,70], Benjapun Kaveelerdpotjana[46], Minjun Kim[71], Yoo-Ah Kim[51], Oliver Kohlbacher[22,72,73,74], Dmitry Korkin[50,75,76], Kiryluk Krzysztof[77], Khalid Kunji[78], Zoltàn Kutalik[37,79], Kasper Lage[80,81,82], David Lamparter[36,37,83], Sean Lang-Brown[84], Thuc Duy Le[85,86], Jooyoung Lee[58,59], Sunwon Lee[68], Juyong Lee[87], Dong Li[46], Jiuyong Li[86], Junyuan Lin[49], Lin Liu[86], Antonis Loizou[88], Zhenhua Luo[89], Artem Lysenko[38], Tianle Ma[90], Raghvendra Mall[78], Daniel Marbach[36,37,56], Tomasoni Mattia[36,37], Mario Medvedovic[91], Jörg Menche[44], Johnathan Mercer[80,82], Elisa Micarelli[45], Alfonso Monaco[24], Felix Müller[44], Rajiv Narayan[92], Oleksandr Narykov[76], Ted Natoli[92], Thea Norman[93], Sungjoon Park[68], Livia Perfetto[45], Dimitri Perrin[94], Stefano Pirrò[45], Teresa M. Przytycka[51], Xiaoning Qian[95], Karthik Raman[26,27,28], Daniele Ramazzotti[33], Emilie Ramsahai[96], Balaraman Ravindran[27,28,97], Philip Rennert[98], Julio Saez-Rodriguez[99,100], Charlotta Schärfe[22], Roded Sharan[101], Ning Shi[46], Wonho Shin[70], Hai Shu[102], Himanshu Sinha[26,27,28], Donna K. Slonim[48], Lionel Spinelli[39], Suhas Srinivasan[75], Aravind Subramanian[92], Christine Suver[103], Damian Szklarczyk[104], Sabina Tangaro[24], Suresh Thiagarajan[105], Laurent Tichit[34], Thorsten Tiede[22],

Beethika Tripathi[27,28,97], Aviad Tsherniak[92], Tatsuhiko Tsunoda[38,106,107], Dénes Türei[100,108], Ehsan Ullah[78], Golnaz Vahedi[41,42,43], Alberto Valdeolivas[34,109], Jayaswal Vivek[110], Christian von Mering[104], Andra Waagmeester[63], Bo Wang[33], Yijie Wang[51], Barbara A. Weir[111,112], Shana White[91], Sebastian Winkler[22], Ke Xu[113], Taosheng Xu[114], Chunhua Yan[65], Liuqing Yang[115], Kaixian Yu[102], Xiangtian Yu[116], Gaia Zaffaroni[53], Mikhail Zaslavskiy[117], Tao Zeng[116], Jitao D. Zhang[56], Lu Zhang[33], Weijia Zhang[86], Lixia Zhang[91], Xinyu Zhang[113], Junpeng Zhang[118], Xin Zhou[33], Jiarui Zhou[46], Hongtu Zhu[81], Junjie Zhu[119] and Guido Zuccon[94]

[22]Applied Bioinformatics, Center for Bioinformatics, University of Tübingen, Tübingen, Germany. [23]Department of Physics 'Michelangelo Merlin', University of Bari 'Aldo Moro', Bari, Italy. [24]INFN, Sezione di Bari, Bari, Italy. [25]Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Tarragona, Spain. [26]Department of Biotechnology, Bhupat and Jyoti Mehta School of Biosciences, Indian Institute of Technology Madras, Chennai, India. [27]Initiative for Biological Systems Engineering, Indian Institute of Technology Madras, Chennai, India. [28]Robert Bosch Centre for Data Science and Artificial Intelligence, Indian Institute of Technology Madras, Chennai, India. [29]Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison, Madison, WI, USA. [30]Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI, USA. [31]Morgridge Institute for Research, Madison, WI, USA. [32]Department of Plant Biology, Carnegie Institution for Science, Stanford, USA. [33]Department of Computer Science, Stanford University, Stanford, USA. [34]Aix Marseille University, CNRS, Centrale Marseille, I2M, Marseille, France. [35]Centro TIRES, Bari, Italy. [36]Department of Computational Biology, University of Lausanne, Lausanne, Switzerland. [37]Swiss Institute of Bioinformatics, Lausanne, Switzerland. [38]RIKEN Center for Integrative Medical Sciences, Yokohama, Japan. [39]Aix Marseille Univ, INSERM, TAGC, Marseille, France. [40]CNRS, Marseille, France. [41]Department of Genetics, Perelman School of Medicine at the University of Pennsylvania, Philadelphia, PA, USA. [42]Institute for Immunology, Perelman School of Medicine at the University of Pennsylvania, Philadelphia, PA, USA. [43]Epigenetics Institute, Perelman School of Medicine at the University of Pennsylvania, Philadelphia, PA, USA. [44]CeMM Research Center for Molecular Medicine of the Austrian Academy of Sciences, Vienna, Austria. [45]Bioinformatics and Computational Biology Unit, Department of Biology, Tor Vergata University, Roma, Italy. [46]School of Computer Science, The University of Birmingham, Birmingham, UK. [47]Nestle Institute of Health Sciences, Lausanne, Switzerland. [48]Department of Computer Science, Tufts University, Medford, MA, USA. [49]Department of Mathematics, Tufts University, Medford, MA, USA. [50]Bioinformatics and Computational Biology Program, Worcester Polytechnic Institute, Worcester, MA, USA. [51]National Center for Biotechnology Information, National Institute of Health (NCBI/NLM/NIH), Bethesda, MD, USA. [52]Fondazione Bruno Kessler, Povo, Italy. [53]LCSB - Luxembourg Centre for Systems Biomedicine, University of Luxembourg, Esch-sur-Alzette, Luxembourg. [54]CIC bioGUNE, Bizkaia Technology Park, Derio, Spain. [55]IKERBASQUE, Basque Foundation for Science, Bilbao, Spain. [56]Roche Pharma Research and Early Development, Pharmaceutical Sciences, Roche Innovation Center Basel, F. Hoffmann-La Roche Ltd, Basel, Switzerland. [57]Department of Cancer Biology, University of Cincinnati, Cincinnati, OH, USA. [58]Center for In Silico Protein Science, Korea Institute for Advanced Study, Seoul, Korea. [59]School of Computational Sciences, Korea Institute for Advanced Study, Seoul, Korea. [60]School of Informatics, Computing and Engineering, Indiana University, Bloomington, USA. [61]Algorithms in Bioinformatics, Center for Bioinformatics, University of Tübingen, Tübingen, Germany. [62]Department of Computational Medicine and Bioinformatics, University of Michigan, Ann Arbor, MI, USA. [63]Micelio, Antwerp, Belgium. [64]College of Computer and Information Science, Northeastern University, Boston, MA, USA. [65]National Cancer Institute, Center for Biomedical Informatics & Information Technology, Bethesda, MD, USA. [66]Center for Bioinformatics and Computational Biology, University of Maryland, College Park, MD, USA. [67]Department of Cell Biology and Molecular Genetics, University of Maryland, College Park, MD, USA. [68]Department of Computer Science and Engineering, Korea University, Seoul, Korea. [69]Center for Advanced Computation, Korea Institute for Advanced Study, Seoul, Korea. [70]Interdisciplinary Graduate Program in Bioinformatics, Korea University, Seoul, Korea. [71]Community High School, Ann Arbor, MI, USA. [72]Biomolecular Interactions, Max Planck Institute for Developmental Biology, Tübingen, Germany. [73]Quantitative Biology Center, University of Tübingen, Tübingen, Germany. [74]Institute for Translational Bioinformatics, University Hospital Tübingen, Tübingen, Germany. [75]Data Science Program, Worcester Polytechnic Institute, Worcester, MA, USA. [76]Department of Computer Science, Worcester Polytechnic Institute, Worcester, MA, USA. [77]Department of Medicine, College of Physicians & Surgeons, Columbia University, New York, NY, USA. [78]Qatar Computing Research Institute, Hamad Bin Khalifa University, Doha, Qatar. [79]Institute of Social and Preventive Medicine (IUMSP), Lausanne University Hospital, Lausanne, Switzerland. [80]Department of Surgery, Massachusetts General Hospital, Harvard Medical School, Boston, Massachusetts, USA. [81]Institute for Biological Psychiatry, Mental Health Center Sct. Hans, University of Copenhagen, Roskilde, Denmark. [82]Stanley Center at the Broad Institute of MIT and Harvard, Cambridge, Massachusetts, USA. [83]Verge Genomics, San Francisco, CA, USA. [84]Division of Geriatrics, Department of Medicine, University of California, San Francisco, USA. [85]Centre for Cancer Biology, University of South Australia, Adelaide, SA, Australia. [86]School of Information Technology and Mathematical Sciences, University of South Australia, Adelaide, SA, Australia. [87]Department of Chemistry, Kangwon National University, Chuncheon, Republic of Korea. [88]BlueSkyIt, Amsterdam, the Netherlands. [89]The Liver Care Center and Divisions of Gastroenterology, Hepatology and Nutrition, Cincinnati Children's Hospital Medical Center, Cincinnati, OH, USA. [90]Department of Computer Science and Engineering, University at Buffalo, Buffalo, NY, USA. [91]Department of Environmental Health, Division of Biostatistics and Bioinformatics, University of Cincinnati, Cincinnati, OH, USA. [92]Broad Institute of Harvard and MIT, Cambridge, MA, USA. [93]Bill and Melinda Gates Foundation, Washington, USA. [94]School of Electrical Engineering and Computer Science, Queensland University of Technology, Brisbane, Australia. [95]Dept. of Electrical & Computer Engineering, Texas A&M University, College Station, USA. [96]Department of Mathematics and Statistics, The University of the West Indies, Saint Augustine, Trinidad and Tobago. [97]Department of Computer Science and Engineering, Indian Institute of Technology Madras, Chennai, India. [98], Rockville, MD, USA. [99]Institute for Computational Biomedicine, Faculty of Medicine, Heidelberg University, Bioquant, Heidelberg, Germany. [100]RWTH Aachen University, Faculty of Medicine, Joint Research Centre for Computational Biomedicine, Aachen, Germany. [101]Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel. [102]Department of Biostatistics, the University of Texas MD Anderson Cancer Center, Houston, TX, USA. [103]Sage Bionetworks, Seattle, Washington, USA. [104]Institute of Molecular Life Sciences and Swiss Institute of Bioinformatics, University of Zürich, Zürich, Switzerland. [105], Memphis, TN, USA. [106]CREST, JST, Tokyo, Japan. [107]Department of Medical Science Mathematics, Medical Research Institute, Tokyo Medical and Dental University, Tokyo, Japan. [108]European Molecular Biology Laboratory, European Bioinformatics Institute, Wellcome Genome Campus, Cambridge, UK. [109]ProGeLife, Marseille, France. [110]Disease Science & Technology, Biocon Bristol-Myers Squibb Research Centre, Bangalore, India. [111]Broad Institute of MIT and Harvard, Cambridge, MA, UK. [112]Janssen Research and Development, Cambridge, MA, USA. [113]Department of Psychiatry, Yale School of Medicine, New Haven, CT, USA. [114]Institute of Intelligent Machines, Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei, Anhui, China. [115]Department of Statistics and Operations Research, University of North Carolina at Chapel Hill, Chapel Hill, NC, USA. [116]Key Laboratory of Systems Biology, Institute of Biochemistry and Cell Biology, Shanghai Institutes for Biological Sciences, Chinese Academy of Sciences, Shanghai, China. [117]Computational Biology Consulting, Paris, France. [118]School of Engineering, Dali University, Dali, Yunnan, China. [119]Department of Electrical Engineering, Stanford University, Stanford, USA.

Methods

Network compendium. A collection of six gene and protein networks for human were provided by different groups for this challenge. The two protein–protein interaction and signaling networks are custom or new versions of existing interaction databases that were not publicly available at the time of the challenge. The remaining networks were yet unpublished at the time of the challenge. This was important to prevent participants from deanonymizing challenge networks by aligning them to the original networks.

Networks were released for the challenge in anonymized form. Anonymization consisted in replacing the gene symbols with randomly assigned ID numbers. In Sub-challenge 1, each network was anonymized individually; that is, node *k* of network A and node *k* of network B are generally not the same genes. In Sub-challenge 2, all networks were anonymized using the same mapping; that is, node *k* of network A and node *k* of network B are the same gene.

All networks are undirected and weighted, except for the signaling network, which is directed and weighted. Below we briefly summarize each of the six networks. Detailed descriptions of networks 4, 5 and 6 are available on GeNets[20], a web platform for network-based analysis of genetic data (http://apps.broadinstitute.org/genets).

The first network was obtained from STRING, a database of known and predicted protein–protein interactions[14]. STRING includes aggregated interactions from primary databases as well as computationally predicted associations. Both physical protein interactions (direct) and functional associations (indirect) are included. The challenge network corresponds to the human protein–protein interactions of STRING v.10.0, where interactions derived from text mining were removed. Edge weights correspond to the STRING association score after removing evidence from text mining.

The second network is the InWeb protein–protein interaction network[15]. InWeb aggregates physical protein–protein interactions from primary databases and the literature. The challenge network corresponds to InWeb v.3. Edge weights correspond to a confidence score that integrates the evidence of the interaction from different sources.

The third network is the OmniPath signaling network[16]. OmniPath integrates literature-curated human signaling pathways from 27 different sources, of which 20 provide causal interaction and seven deliver undirected interactions. These data were integrated to form a directed weighted network. The edge weights correspond to a confidence score that summarizes the strength of evidence from the different sources.

The fourth network is a co-expression network based on Affymetrix HG-U133 Plus 2 arrays extracted from the GEO[46]. To adjust for non-biological variation, data were rescaled by fitting a loess-smoothed power law curve to a collection of 80 reference genes (ten sets of roughly eight genes each, representing different strata of expression) using non-linear least squares regression within each sample. All samples were then quantile normalized together as a cohort[17]. After filtering out samples that did not pass quality control, a gene expression matrix of 22,268 probesets by 19,019 samples was obtained. Probes were mapped to genes by averaging and the pairwise Spearman correlation of genes across samples was computed. The matrix was thresholded to include the top 1 million strongest positive correlations resulting in an undirected, weighted network. The edge weights correspond to the correlation coefficients.

The fifth network is a functional gene network derived from the Project Achilles dataset v.2.4.3 (ref. [18]). Project Achilles performed genome-scale loss-of-function screens in 216 cancer cell lines using massively parallel pooled short-hairpin RNA screens. Cell lines were transduced with a library of 54,000 shRNAs, each targeting one of 11,000 genes for RNA interference knockdown (~5 shRNAs per gene). The proliferation effect of each shRNA in a given cell line could be assessed using Next Generation Sequencing. From these data, the dependency of a cell line on each gene (the gene essentiality) was estimated using the ATARiS method. This led to a gene essentiality matrix of 11,000 genes by 216 cell lines. Pairwise correlations between genes were computed and the resulting codependency network was thresholded to the top 1M strongest positive correlations, analogous to how the co-expression network was constructed.

The sixth network is a functional gene network based on phylogenetic relationships identified using the CLIME (clustering by inferred models of evolution) algorithm[21]. CLIME can be used to expand pathways (gene sets) with additional genes using an evolutionary model. Briefly, given a eukaryotic species tree and homology matrix, the input gene set is partitioned into evolutionarily conserved modules (ECMs), which are then expanded with new genes sharing the same evolutionary history. To this end, each gene is assigned a log-likelihood ratio (LLR) score based on the ECMs inferred model of evolution. CLIME was applied to 1,025 curated human gene sets from GO and the Kyoto Encyclopedia of Genes and Genomes using a 138 eukaryotic species tree, which resulted in 13,307 expanded ECMs. The network was constructed by adding an edge between every pair of genes that co-occurred in at least one ECM. Edge weights correspond to the mean LLR scores of the two genes.

Challenge structure. Participants were challenged to apply network module identification methods to predict functional modules (gene sets) based on network topology. Valid modules had to be non-overlapping (a given gene could be part of either zero or one module, but not multiple modules) and comprise between 3 and 100 genes (modules with over 100 genes are typically less useful to gain specific biological insights). Modules did not have to cover all genes in a network. The number of modules per network was not fixed: teams could submit any number of modules for a given network (the maximum number was limited due to the fact that modules had to be non-overlapping). In Sub-challenge 1, teams were required to submit a separate set of modules for each of the six networks. In Sub-challenge 2, teams were required to submit a single set of modules by integrating information across multiple networks (it was permitted to use only a subset of the six networks).

The challenge consisted of a leaderboard phase and the final evaluation. The leaderboard phase was organized in four rounds, where participants could make repeated submissions and see their score for each network, along with the scores of other teams, on a real-time leaderboard. Due to the high computational cost of scoring the module predictions on a large number of GWAS datasets, a limit for the number of submissions per team was set in each round. The total number of submissions that any given team could make over the four leaderboard rounds was thus limited to only 25 and 41 for the two sub-challenges, respectively. For the final evaluation, a single submission including method descriptions and code was required per team, which was scored on a separate holdout set of GWASs after the challenge closed to determine the top performers.

With this challenge we assessed unsupervised clustering methods that define modules based solely on the topological structure of networks, unbiased by existing biological knowledge. Additional biological information, such as GWAS data and functional annotations, was integrated afterwards to assess and characterize the predicted modules. It is important to benchmark unsupervised methods in this blinded setting, because they are often relied on in regions of the network for which a paucity of biological information is currently available.

The submission format and rules are described in detail on the challenge website (https://www.synapse.org/modulechallenge).

Gene and module scoring using Pascal. We have developed a framework to empirically assess module identification methods on molecular networks using GWAS data. Since we are employing a large collection of 180 GWAS datasets ranging over diverse disease-related human phenotypes, this approach covers a broad spectrum of molecular processes. In contrast to evaluation of module enrichment using existing gene and pathway annotations, where it is sometimes difficult to ascertain that annotations were not derived from similar data types as the networks (for example, gene expression, protein–protein interactions or homology), the GWAS-based approach provides an orthogonal means to assess disease-relevant modules.

SNP trait-association *P* values from a given GWAS were integrated across genes and modules using the Pascal (pathway scoring algorithm) tool[23]. Briefly, Pascal combines analytical and numerical solutions to efficiently compute gene and module scores from SNP *P* values, while properly correcting for LD correlation structure prevalent in GWAS data. To this end, LD information from a reference population is used (here, the European population of the 1,000 Genomes Project was employed as we only included GWASs with predominantly European cohorts). For gene scores we used the sum of chi-squared statistics of all SNPs within a window extending 50 kb up and downstream from the gene of interest. Since proximal SNPs are often in LD, under the null hypothesis this sum is not distributed like a sum of chi-squares of independent random variables. Yet, a change of basis to orthogonal 'eigen-SNPs', which diagonalize the genotypic correlation (LD) matrix, recovers independence with the effect that the sum of independent chi-squares is weighted (with the eigenvalues as weights)[23].

The fast gene scoring is critical as it allows module genes that are in LD, and can thus not be treated independently, to be dynamically rescored. This amounts to fusing the genes of a given module that are in LD and computing a new score that takes the full LD structure of the corresponding locus into account. Pascal tests modules for enrichment in high-scoring (potentially fused) genes using a modified Fisher method, which avoids any *P* value cutoffs inherent to standard binary enrichment tests. The general approach can be summarized in three steps: (1) gene score *P* values of all genes in the background set (here, all genes in a given network) are transformed so that they follow a target distribution respecting their ranking; (2) a test statistic for a given module is computed by summing the transformed scores of module genes (and fusion-genes) and (3) it is evaluated whether the observed test statistic is higher than expected, that is, the module is enriched for trait-associated genes. Specifically, here we employed the 'chi-squared method' implemented in Pascal[23], which transforms gene scores such that they follow a χ^2_1-distribution (gene scores are first rank-transformed to obtain a uniform distribution and then transformed by the χ^2_1-quantile function). χ^2_1-gene scores of a given module of size m are then summed and tested against a χ^2_m-distribution. Since gene scores are first rank-transformed, this is a 'competitive' enrichment test, which evaluates whether the module genes tend to have lower trait-association *P* values than the other genes that are part of the given network. Note that specifying the correct background set is critical for competitive enrichment tests, which here amounts to all genes that are part of the given network (that is, not all genes in the genome), as shown in Supplementary Fig. 4. Last, the resulting nominal module *P* values were adjusted to control the FDR via the Benjamini–Hochberg procedure.

Scoring metric. In Sub-challenge 1, the score for a given network was defined as the number of modules with significant Pascal *P* values at a given FDR cutoff in at least one GWAS (called trait-associated modules, see previous section). Thus, modules that were hits for multiple GWAS traits were only counted once. The reason for this choice is that we do not want to 'overcount' modules that are hits for multiple related GWAS traits compared to modules that are hits for GWASs where few related traits are available (see Fig. 4c). The overall score was defined as the sum of the scores obtained on the six networks (that is, the total number of trait-associated modules across all networks). For the official challenge ranking a 5% FDR cutoff was defined, but performance was further reported at 10, 2.5 and 1% FDR.

Before the challenge, we performed an analysis to explore whether this scoring metric would favor a particular resolution for modules and thus bias results; for example, toward decomposing modules into a larger number of small submodules. To this end, we generated random modules of varying size. We found no systematic bias in the scores for a specific module granularity and this result was confirmed in the challenge (Supplementary Figs. 3 and 4). The key element of the scoring function that was designed to fairly assess module collections with different average module sizes was the higher multiple testing burden applied when a larger number of smaller modules was submitted.

Module predictions in Sub-challenge 2 were scored using the exact same methodology and FDR cutoffs. The only difference to Sub-challenge 1 was that submissions consisted of a single set of modules (instead of one for each network) and there was thus no need to define an overall score. As background gene set, the union of all genes across the six networks was used.

Leaderboard and holdout GWAS datasets. We compiled a collection of 180 GWAS datasets (Supplementary Table 1), including all GWASs for which we could access genome-wide summary statistics (SNP *P* values). We deliberately included both high- and low-powered studies to evaluate whether disease-associated modules could be detected in datasets of varying signal strength. We manually assigned each GWAS dataset to either the leaderboard set used in the leaderboard round or the holdout set used for the final scoring. The assignment was made such that GWASs of closely related traits (for example, height, male height and female height) were either all in the leaderboard set or all in the holdout set, thus avoiding a situation where two very similar GWASs would be found both in the leaderboard and holdout set. This resulted in a leaderboard set of 76 GWASs and a holdout set of 104 GWASs (Supplementary Table 1). Compared to random assignment of GWASs to the leaderboard and holdout set, this setup better tests the robustness of parameters tuned by participants during the leaderboard round.

Robustness analysis of challenge ranking. To gain a sense of the robustness of the ranking with respect to the GWAS data, we subsampled the set of 104 GWASs used for the final evaluation (the holdout set) by drawing $N < 104$ GWASs. Here we used $N = 76$ GWASs (73% of the holdout set) as this is the same size as the leaderboard set, but this choice does not critically affect results. Note that we have to do subsampling rather than resampling of GWASs because the scoring counts the number of modules that are associated to at least one GWAS; that is, including the same GWASs multiple times does not affect the score. We applied this approach to create 1,000 subsamples of the holdout set. The methods were then scored on each subsample.

The performance of every method m was compared to the highest-scoring method across the subsamples by the paired Bayes factor K_m. That is, the method with the highest overall score in the holdout set (all 104 GWASs) was defined as reference (that is, method K1 in Sub-challenge 1). The score $S(m, k)$ of method m in subsample k was thus compared with the score $S(ref, k)$ of the reference method in the same subsample k. The Bayes factor K_m is defined as the number of times the reference method outperforms method m, divided by the number of times method m outperforms or ties the reference method over all subsamples. Methods with $K_m < 3$ were considered a tie with the reference method (that is, method m outperforms the reference in more than one out of four subsamples).

Overview of module identification methods in Sub-challenge 1. Based on descriptions provided by participants, module identification methods were classified into different categories (Fig. 2a). Categories and corresponding module identification methods are summarized in Supplementary Table 2. In the following, we first give an overview of the different categories and top-performing methods, and then describe common pre- and postprocessing steps used by these methods:

- Kernel clustering: instead of working directly on the networks themselves, these methods cluster a kernel matrix, where each entry (i, j) of that matrix represents the closeness of nodes i and j in the network according to the particular similarity function, or kernel that was applied. Some of the kernels that were applied are well-known for community detection, such as the exponential diffusion kernel based on the graph Laplacian[47] employed by method K6. Others, such as the LINE embedding algorithm[48] employed by method K3 and the kernel based on the inverse of the weighted diffusion state distance[24,25] employed by method K1, were newer. Method K1 was the best-performing method of the challenge and is described in detail below.
- Modularity optimization: this method category was, along with random-walk-based methods, the most popular type of method contributed by the community. Modularity optimization methods use search algorithms to find a partition of the network that maximizes the modularity Q (commonly defined as the fraction of within-module edges minus the expected fraction of such edges in a random network with the same node degrees)[43]. The most popular algorithm was Louvain community detection[49]. At least eight teams employed this algorithm in some form as either their main method or one of several methods, including the fourth ranking team. The best-performing modularity optimization method (M1), which ranked second overall, is described in detail below.
- Random-walk-based methods: these methods take inspiration from random walks or diffusion processes over the network. Several teams used the established Walktrap[50], Infomap[51] and Markov clustering algorithms. The top team of this category (method R1, third rank overall) used a sophisticated random-walk method based on multi-level Markov clustering[28], which is described in detail below. While we did not include kernel methods in the 'random walk' category, several of the successful kernel clustering methods used random-walk-based measures within their kernel functions.
- Local methods: only three teams used local community detection methods, including agglomerative clustering and seed set expansion approaches. The top team of this category (method L1, fifth rank overall) first converted the adjacency matrix into a topology overlap matrix[35], which measures the similarity of nodes based on the number of neighbors that they have in common. The team then used the SPICi algorithm[52], which iteratively adds adjacent genes to cluster seeds such as to improve their local density.
- Hybrid methods: seven teams employed hybrid methods that leveraged clusterings produced by several of the different main approaches listed above. These teams applied more than one community detection method to each network to get larger and more diverse sets of predicted modules. The most common methods applied were Louvain[49], hierarchical clustering and Infomap[51]. Two different strategies were used to select a final set of modules for submission: (1) choose a single method for each network according to performance in the leaderboard round, and (2) select modules from all applied methods according to a topological quality score such as the modularity or conductance[10].
- Ensemble methods: much like hybrid methods, ensemble methods leverage clusterings obtained from multiple community detection methods (or multiple stochastic runs of a single method). However, instead of selecting individual modules according to a quality score, ensemble methods merge alternative clusterings to obtain potentially more robust consensus predictions[53]. Our method to derive consensus module predictions from team submissions is an example of an ensemble approach (see below).

Besides the choice of the community detection algorithm, there are other steps that critically affected performance, including preprocessing of the network data, setting of method parameters and postprocessing of predicted modules (Supplementary Table 2):

- Preprocessing: most networks in the challenge were densely connected, including many edges of low weight that are likely noisy. Some of the top teams (for example, M1, R1, L1) benefited from sparsifying these networks by discarding weak edges before applying their community detection methods. An added benefit of sparsification is that it typically reduces computation time. Few teams also normalized the edge weights of a given network to make them either normally distributed or fall in the range between zero and one. Not all methods required preprocessing of networks; for example, the top-performing method (K1) was applied to the original networks without any sparsification or normalization steps.
- Parameter setting: community detection methods often have parameters that need to be specified, typically to control the resolution of the clustering (the number and size of modules). While some methods have parameters that explicitly set the number of modules (for example, the top-performing method K1), other methods have parameters that indirectly control the resolution (for example, the resistance parameter of the runner-up method M1). While there were also methods that had no parameters to set (for example, the classic Louvain algorithm), these methods have an intrinsic resolution that may not always be optimal for a given network and target application.
- Postprocessing: modularization of biological networks often results in highly imbalanced module sizes. That is, some modules may be very small (for example, just one or two genes), while others are extremely large (for example, thousands of genes). Both extremes are generally not useful to gain biological insights at the pathway level. Since current community detection methods generally do not allow constraints on module size to be specified, teams used different postprocessing steps to deal with modules outside the allowed range in the challenge (between 3 and 100 genes). A successful strategy to break down large modules was to recursively apply community detection methods to each of these modules. Alternatively, all modules of invalid size were merged and the method was reapplied to the corresponding subnetwork. Finally, modules with fewer than three genes were often discarded. Some teams also discarded larger modules that were deemed low quality according to a topological metric, although this strategy was generally not beneficial.

Method K1 (first rank in Sub-challenge 1). The top-performing team developed a kernel clustering approach (method K1) based on a distance measure called diffusion state distance (DSD)[24,25], which they further improved for this challenge. DSD produces a more informative notion of proximity than the typical shortest path metric, which measures distance between pairs of nodes by the number of hops on the shortest path that joins them in the network. More formally, consider the undirected network $G(V, E)$ on the node set $V=\{v_1, v_2, v_3,...,v_n\}$ with $|V|=n$. $\mathbf{He}^t(v_x, v_y)$ is defined as the expected number of times that a random walk (visiting neighboring nodes in proportion to their edge weights) starting at node v_x and proceeding for some fixed t steps will visit node v_y (the walk includes the starting point, that is, 0th step). Taking a global view, we define the n-dimensional vector $\mathbf{He}^t(v_x)$ whose ith entry is the $\mathbf{He}^t(v_x,v_i)$ value to network node v_i. Then the DSD^t distance between two nodes v_x and v_y is defined as the $L1$ norm of the difference of their $\mathbf{He}^t$ vectors, that is

$$\mathrm{DSD}^t(v_x, v_y) = ||\mathbf{He}^t(v_x) - \mathbf{He}^t(v_y)||_1$$

It can be shown that DSD is a metric and converges as $t\to\infty$, allowing DSD to be defined independently from the value t (ref. [24].) The converged DSD matrix can be computed tractably, with an eigenvalue computation, as

$$\mathrm{DSD}(v_x, v_y) = ||(\mathbf{1}_x - \mathbf{1}_y)(I - D^{-1}A + W)^{-1}||_1$$

where D is the diagonal degree matrix, A is the adjacency matrix and W is the matrix where each row is a copy of π, the degrees of each of the nodes, normalized by the sum of all the vertex degrees (in the unweighted case, weighted edges can be normalized proportional to their weight) and $\mathbf{1}_x$ and $\mathbf{1}_y$ are the vectors that are zero everywhere except at position x and y, respectively. The converged DSD matrix was approximated using algebraic multigrid techniques. Note that for the signaling network, edge directions were kept and low-weight back edges were added so that the network was strongly connected; that is, if there was a directed edge from v_x to v_y, an edge from v_y to v_x of weight equal to 1/100 of the lowest edge weight in the network was added.

A spectral clustering algorithm[26] was used to cluster the DSD matrix of a given network. Note that the spectral clustering algorithm operates on a similarity matrix (that is, entries that are most alike have higher values in the matrix). However, the DSD matrix is a distance matrix (that is, similar entries have low DSD values). The radial basis function kernel presents a standard way to convert the DSD matrix to a similarity matrix; it maps low distances to high similarity scores and vice versa. Since the spectral clustering algorithm employed uses k-means as the underlying clustering mechanism, it takes a parameter k specifying the number of cluster centers. The leaderboard rounds were used to measure the performance of different k values. Clusters with fewer than three nodes were discarded. Clusters with over 100 nodes were recursively split into two subclusters using spectral clustering (that is, $k=2$) until all clusters had fewer than 100 nodes.

The top-performing team also used a different algorithm to search for dense bipartite subgraph module structure in half of the challenge networks and merged these modules (which were rare) with the clusters generated by their main method[54]. However, a post facto analysis of their results showed that this step contributed few modules and the score would have been similar with this additional procedure omitted.

Method M1 (second rank in Sub-challenge 1). The runner-up team developed a multi-resolution modularity optimization method[27]. The rationale is that in the absence of information on the cluster sizes of the graph, a method should be able to explore all possible topological scales at which clusters may satisfy the definition of module. The multi-resolution method developed by the team works by adding a resistance parameter r to the community detection algorithms. This resistance controls the aversion of nodes to form communities; the larger the resistance, the smaller the size of the modules. For community detection algorithms based on the optimization of the well-known modularity function[43], this resistance takes the form of a self-loop (with a weight equal to r) which is added to all nodes of the network. In this way, all nodes contribute to the internal strength of their modules with a constant amount r. When the resistance is zero, the standard (and implicit) scale of resolution is recovered.

The team first sparsified networks by removing low confidence edges and then applied several well-known modularity optimization algorithms, including: (1) extremal optimization, (2) spectral optimization, (3) Newman's fast algorithm and (4) fine-tuning by iterative repositioning of nodes. The idea is that a combination of several algorithms has fewer chances to get stacked in a suboptimal partition. The resistance parameter r was optimized so as to maximize the proportion of nodes inside communities of the desired sizes defined by the challenge rules, that is, between 3 and 100 nodes (only a handful of values were evaluated due to computational cost, but resulted in much better resolutions than the default of $r=0$). Communities above the size limit (100 nodes) were subdivided recursively.

Method R1 (third rank in Sub-challenge 1). This team used balanced multi-layer regularized Markov clustering (bMLRMCL)[28], an extension of the Markov cluster algorithm (MCL). The algorithm improves three common issues with MCL: (1) scalability for large graphs; (2) fragmented clusters due to the existence of hub nodes and (3) modules of imbalanced size.

Regularized MCL (RMCL) changes the MCL expansion step by introducing a canonical flow matrix, which ensures that the original topology of the graph still influences the graph clustering process beyond the first iteration. Multi-layer RMCL further improves the runtime by first coarsening the graph into multiple layers of smaller graphs to run RMCL on. Last, the balanced version of the algorithm computes a new regularization matrix at each iteration that penalizes big cluster sizes, where the penalty can be adjusted using a balance parameter[28]. Altogether, the method has three parameters: the inflation parameter i, coarsening size c and size balance parameter b. As preprocessing steps, the team first discarded weak edges and then transformed edge weights to integers. Communities with more than 100 nodes were recursively reclustered.

Overview of module identification methods in Sub-challenge 2. There are broadly three different approaches to identify integrated modules across multiple networks: (1) the networks are first merged and then single-network module identification methods are applied on the integrated network, (2) single-network module identification methods are first applied on each individual network and then the resulting modules are merged across networks and (3) dedicated multi-network community detection methods are employed (also called multi-layer or multiplex methods), which are specifically designed to identify modules in layered networks[30,31].

In Sub-challenge 2, the majority of teams employed the first approach. These teams built an integrated network by merging either all six or a subset of the challenge networks, and then applied single-network methods (typically the same method as in Sub-challenge 1) to modularize the integrated network. For example, the team with highest score in Sub-challenge 2 merged the two protein–protein interaction networks and then applied the Louvain algorithm to identify modules in the integrated network. The top-performing team from Sub-challenge 1 also performed competitively in Sub-challenge 2. They applied their single-network method (K1) to an integrated network consisting of the union of all edges from the two protein–protein interaction networks and the co-expression network.

Dedicated multi-network community detection methods were also employed by several teams[30,31]. For example, the runner-up team in Sub-challenge 2 previously extended the modularity measure to multiplex networks and adapted the Louvain algorithm to optimize this multiplex-modularity[31]. For this challenge, the team further improved their method with a randomization procedure, the consideration of edge and layer weights and a recursive clustering of the communities larger than a given size[36].

Similar to Sub-challenge 1, teams used the leaderboard phase to set parameters of their methods. However, besides the parameters of the community detection method, there were additional choices to be made: whether to use all or only a subset of the six networks and how to integrate them.

None of the teams employed the second approach mentioned above, that is, to merge modules obtained from different networks. This approach was only used by the organizers, to form consensus modules in Sub-challenge 2 (see next section). In recent work, Sims et al. intersected brain-specific co-expression modules with generic protein–protein interaction networks, leading to a refined network module enriched for both common and rare variants associated with Alzheimer's disease[55]. Exploring this type of approach with our challenge resources and potentially additional context-specific networks is thus an interesting avenue for future work.

Consensus module predictions. We developed an ensemble approach to derive consensus modules from a given set of team submissions (see Supplementary Fig. 7 for a schematic overview). In Sub-challenge 1, a consensus matrix C^n was defined for each network n, where each element c_{ij} corresponds to the fraction of teams that put gene i and j together in the same module in this network. That is, c_{ij} equals one if all teams clustered gene i and j together, and c_{ij} equals zero if none of the teams clustered the two genes together. The top-performing module identification method (K1) was used to cluster the consensus matrix (that is, the consensus matrix was considered a weighted adjacency matrix defining a functional gene network, which was clustered using the top module identification method of the challenge). Method K1 has only one parameter to set, which is the number of cluster centers used by the spectral clustering algorithm. This parameter was set to the median number of modules submitted by the considered teams for the given network. The consensus module predictions described in the main text were derived from the submissions of the top 50% teams (that is, 21 teams) with the highest overall score on the leaderboard GWAS set.

Multi-network consensus modules were obtained by integrating team submissions from Sub-challenge 1 across all six networks using the same approach (Supplementary Fig. 7). The same set of teams was considered (that is, top 50% on the leaderboard GWAS set). First, a multi-network consensus matrix was obtained by taking the mean of the six network-specific consensus matrices C^n. The multi-network consensus matrix was then clustered using method K1 as described above, where the number of cluster centers was set to the median number of modules submitted by the considered teams across all networks.

Two additional, more sophisticated approaches to construct consensus matrices C^n were tested: (1) normalization of the contribution of each module by the module size led to similar results as the basic approach described above, and (2) unsupervised estimation of module prediction accuracy using the Spectral Meta Learner ensemble method[56]. These methods did not perform well in this context (Supplementary Fig. 7).

Similarity of module predictions. To define a similarity metric between module predictions from different methods, we represented module predictions as vectors. Namely, the set of modules predicted by method m in network k was represented as a prediction vector $\mathbf{P}_{mk}$ of length $N_k(N_k-1)/2$, where N_k is the number of genes in the network. Each element of this vector corresponds to a pair of genes and equals one if the two genes are in the same module and zero otherwise. Accordingly, for any two module predictions (method m_1 applied to network k_1, and method m_2 applied to network k_2), we calculated the distance as follows:

$$D(m_1k_1, m_2k_2) = 1 - \frac{<\mathbf{P}_{m_1k_1}, \mathbf{P}_{m_2k_2}>}{||\mathbf{P}_{m_1k_1}||_2||\mathbf{P}_{m_2k_2}||_2}$$

where <.,.> is the Euclidean inner product, $||.||_2$ is the Euclidean norm and D is the (symmetric) distance matrix between the 252 module predictions submitted in Sub-challenge 1 (that is, 42 methods applied to each of six networks). The distance matrix D was used as input to multidimensional scaling analysis for dimensionality reduction in Fig. 3a.

Overlap between trait-associated modules. Three different metrics were considered to quantify the overlap between trait-associated modules from different methods and networks. The first metric was the Jaccard index, which is defined as the size of the intersection divided by the size of the union of two modules (gene sets) A and B:

$$J(A,B) = \frac{|A \cap B|}{|A \cup B|}$$

The Jaccard index measures how similar two modules are, but does allow the detection of submodules. For example, consider a module A of size ten that is a submodule of a module B of size 100. In this case, even though 100% of genes of the first module are comprised in the second module, the Jaccard index is rather low (0.1). To capture submodules, we thus considered in addition the percentage of genes of the first module that are comprised in the second module:

$$S(A,B) = \frac{A \cap B}{|A|}$$

Last, we also evaluated the significance of the overlap. To this end, we computed the P value p_{AB} for the overlap between the two modules using the hypergeometric distribution. P values were adjusted using Bonferroni correction given the number of module pairs tested.

Based on these three metrics, we categorized the type of overlap that a given trait-module A had with another trait-module B as:

1. strong overlap if $J(A,B) \geq 0.5$ and $p_{AB} < 0.05$;
2. submodule if $J(A,B) < 0.5$ and $S(A,B) - J(A,B) \geq 0.5$ and $p_{AB} < 0.05$;
3. partial overlap if $J(A,B) < 0.5$ and $S(A,B) - J(A,B) < 0.5$ and $p_{AB} < 0.05$;
4. insignificant overlap if $p_{AB} \geq 0.05$.

An additional category, strong overlap and submodule, was defined for trait module A that satisfy both conditions (1) and (2) with two different trait modules B and C. This categorization was used to get a sense of the type of overlap between trait modules from all methods (see Supplementary Fig. 2).

Trait similarity network. We defined a network level similarity between GWAS traits based on overlap between trait-associated modules. To this end, we only considered the most relevant networks for our collection of GWAS traits, that is, the two protein–protein interaction, the signaling and the co-expression network (see Fig. 2d). For a given network, the set of 'trait-module genes' G_T was obtained for every trait T by taking the union of the modules associated with that trait across all challenge methods. If different GWASs were available for the same trait type (see Supplementary Table 1), the union of all corresponding trait-associated modules was taken. The overlap between every pair of trait-module gene sets G_{T_1} and G_{T_2} was evaluated using the Jaccard index $J(G_{T_1}, G_{T_2})$ and the hypergeometric P value $P_{T_1T_2}$ as described in the previous section. P values were adjusted using Bonferroni correction. For the visualization as a trait–trait network in Fig. 4c, an edge between traits T_1 and T_2 was added if the overlap was significant ($P_{T_1T_2} < 0.05$) in at least three out of the four considered networks, and node sizes and edge weights were set to be proportional to the average number of trait-module genes and the average Jaccard index across the four networks, respectively.

Functional enrichment analysis. To test network modules for enrichment in known gene functions and pathways, we considered diverse annotation and pathway databases. GO annotations for biological process, cellular component and molecular functions were downloaded from the GO website (http://geneontology.org, accessed on 20 January 2017). Curated pathways (KEGG, Reactome and BioCarta) were obtained from MSigDB v.5.2 (http://software.broadinstitute.org/gsea). We also created a collection of gene sets reflecting mouse mutant phenotypes, as defined by the Mammalian Phenotype Ontology[57]. We started with data files HMD_HumanPhenotype.rpt and MGI_GenePheno.rpt, downloaded from the Mouse Genome Informatics database (http://www.informatics.jax.org) on 21 February 2016. The first file contains human–mouse orthology data and some phenotypic information; we then integrated more phenotypic data from the second file, removing the two normal phenotypes MP:0002169 (no abnormal phenotype detected) and MP:0002873 (normal phenotype). For each remaining phenotype, we then built a list of all genes having at least one mutant strain exhibiting that phenotype, which we considered as a functional gene set.

Annotations from curated databases are known to be biased toward certain classes of genes. For example, some genes have been much more heavily studied than others and thus tend to have more annotations assigned to them. This and other biases lead to an uneven distribution of the number of annotations per genes (annotation bias). On the other hand, the gene sets (modules) tested for enrichment in these databases typically also exhibit bias for certain classes of genes (selection bias)[58,59]. Standard methods for GO enrichment analysis use the hypergeometric distribution (that is, Fisher's exact test), the underlying assumption being that, under the null hypothesis, each gene is equally likely to be included in the gene set (module). Due to selection bias, this is typically not the case in practice, leading to inflation of P values[58,59]. Following Young et al.[59], we thus used the Wallenius non-central hypergeometric distribution to account for biased sampling. Corresponding enrichment P values were computed for all network modules and annotation terms (pathways). The genes of the given network were used as a background gene set. For each network, module identification method and annotation database, the $M \times T$ nominal P values of the M modules and T annotation terms were adjusted using the Bonferroni correction.

Selection of representative module annotations. Gene-set enrichment analysis methods often identify multiple significantly enriched gene sets with very similar compositions. To annotate our modules with few, but informative, gene sets, we formulate the gene-set enrichment problem within a regression framework[33]. Thus, the problem of gene-set enrichment is transformed into a feature selection problem; that is, the aim is to select the gene sets that best predict the membership of genes in a given module.

We constructed gene sets from the latest version of GO (format v.1.2; data version, releases/15 July 2018) and the Reactome database (download time, 16 July 2018). Genes belonging to a GO term or a Reactome pathway are considered as one gene set, independent of positions of either the term or the pathway in the respective hierarchies. Next, we used the gene sets to construct a gene-by-gene-set binary matrix G, whose rows are genes and columns are gene sets. G_{ij} equals 1 if and only if gene i belongs to gene set j; otherwise G_{ij} equals zero.

Given a module M, as well as the background genes B (the union of genes within GO and Reactome), we construct a vector $\mathbf{y}$ representing all genes in B. We assign $y_i = 1$ if and only if gene g_i belongs to the module M. Next, we train the regression model: $\mathbf{y} = G\boldsymbol{\beta} + \boldsymbol{\varepsilon}$ using elastic net with $\alpha = 0.5$ (the hyperparameter α controls the number of selected gene sets). Gene sets with coefficients larger than zero were selected as representative annotations for the given module[33].

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

Challenge data and results are available from the challenge website (https://synapse.org/modulechallenge). This includes: official challenge rules; gene scores for the compendium of 180 GWASs used in the challenge plus five additional GWASs obtained after the challenge (GWAS SNP P values are available on request); official challenge rules; gene scores for the compendium of 180 GWASs used in the challenge plus five additional GWASs obtained after the challenge (GWAS SNP P values are available on request); the six challenge networks (anonymized and deanonymized versions); the final module predictions of all teams for both sub-challenges; consensus module predictions for both sub-challenges; individual module scores for all GWASs and enriched functional annotations for all modules.

Code availability

Code is available on GitHub for: user-friendly, dockerized versions of the top three methods (https://github.com/BergmannLab/MONET); the latest Pascal version (https://www2.unil.ch/cbg/index.php?title=Pascal, https://github.com/dlampart/Pascal); the regression-based gene-set enrichment analysis (https://github.com/TaoDFang/GeneModuleAnnotation) and, in addition, the scoring scripts, a snapshot of the Pascal version used for the challenge, and module identification method descriptions and code provided by teams are available on the challenge website (https://synapse.org/modulechallenge).

References

46. Barrett, T. et al. NCBI GEO: archive for functional genomics data sets–10 years on. *Nucleic Acids Res* **39**, D1005–D1010 (2011).
47. Kondor, R. I. & Lafferty, J. D. Diffusion kernels on graphs and other discrete input spaces. In *Proc. Nineteenth International Conference on Machine Learning* 315–322 (Morgan Kaufmann Publishers Inc., 2002).
48. Tang, J. et al. LINE: Large-scale information network embedding. In *Proc. 24th International Conference on World Wide Web* 1067–1077 (International World Wide Web Conferences Steering Committee, 2015).

49. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. Fast unfolding of communities in large networks. *J. Stat. Mech. Theory Exp.* **2008**, P10008 (2008).
50. Pons, P. & Latapy, M. Computing communities in large networks using random walks (long version). Preprint at *arXiv* https://arxiv.org/abs/physics/0512106 (2005).
51. Rosvall, M., Axelsson, D. & Bergstrom, C. T. The map equation. *Eur. Phys. J. Spec. Top.* **178**, 13–23 (2009).
52. Jiang, P. & Singh, M. SPICi: a fast clustering algorithm for large biological networks. *Bioinforma.* **26**, 1105–1111 (2010).
53. Lancichinetti, A. & Fortunato, S. Consensus clustering in complex networks. *Sci. Rep.* **2**, srep00336 (2012).
54. Gallant, A., Leiserson, M. D., Kachalov, M., Cowen, L. J. & Hescott, B. J. Genecentric: a package to uncover graph-theoretic structure in high-throughput epistasis data. *BMC Bioinforma.* **14**, 23 (2013).
55. Sims, R. et al. Rare coding variants in PLCG2, ABI3, and TREM2 implicate microglial-mediated innate immunity in alzheimer's disease. *Nat. Genet.* **49**, 1373–1384 (2017).
56. Parisi, F., Strino, F., Nadler, B. & Kluger, Y. Ranking and combining multiple predictors without labeled data. *Proc. Natl Acad. Sci. USA* **111**, 201219097 (2014).
57. Blake, J. A. et al. Mouse genome database (MGD)-2017: community knowledge resource for the laboratory mouse. *Nucleic Acids Res.* **45**, D723–D729 (2017).
58. Glass, K. & Girvan, M. Annotation enrichment analysis: an alternative method for evaluating the functional properties of gene sets. *Sci. Rep.* **4**, 4191 (2014).
59. Young, M. D., Wakefield, M. J., Smyth, G. K. & Oshlack, A. Gene ontology analysis for RNA-seq: accounting for selection bias. *Genome Biol.* **11**, R14 (2010).

nature research

Corresponding author(s): Daniel Marbach

Last updated by author(s): Apr 2, 2019

Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed
☐	☒ The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒ A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒ The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☐	☒ A description of all covariates tested
☐	☒ A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒ A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒ For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☐	☒ For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☐	☒ For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒ Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	Does not apply / no software used for data collection
Data analysis	All code used in the challenge is freely available and open source. The Pascal tool was used for the challenge scoring, which is available from github. In addition, a snapshot of the Pascal version used for the challenge, scoring scripts, and code for module identification methods submitted by teams are available from the challenge website. Code used for functional enrichment analysis of modules is available from github.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

All challenge data and results are available from the challenge website (https://synapse.org/modulechallenge). This includes the challenge networks, module identification method descriptions and code provided by teams, the final module predictions of all teams for both sub-challenges, consensus module predictions for both sub-challenges, method scores at varying FDR cutoffs, individual module scores for all GWASs, enriched functional annotations for all modules, a snapshot of the PASCAL tool and scoring scripts, and the gene score p-values for the compendium of 180 GWASs used in the challenge (plus 5 additional GWASs obtained after the challenge). GWAS SNP p-values are available from the corresponding author (D.M.) upon request.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	Considering the DREAM Challenge a scientific experiment, the number of samples corresponds to the number of final team submissions (Subchallenge 1: N=42; Subchallenge 2: N=33). Since the challenge was open for anyone to participate, the number of participating teams could not be determined in advance.
Data exclusions	No data were excluded.
Replication	Measures taken to verify reproducibility included the scoring of challenge submissions: (1) on a blinded test set, (2) at varying FDR cutoffs and (3) on subsampling of the test set (robustness analysis). The top-performing method scored best at each of these metrics, while performance of other teams varied.
Randomization	Does not apply, there were no experiments performed that could have been randomized.
Blinding	Challenge participants were blinded to the test set used for the scoring.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology
☒	☐ Animals and other organisms
☒	☐ Human research participants
☒	☐ Clinical data

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging