Published in final edited form as:
*Nat Methods*. 2022 October ; 19(10): 1250–1261. doi:10.1038/s41592-022-01616-x.

# BIONIC: biological network integration using convolutions

**Duncan T. Forster**[1,2,3,13], **Sheena C. Li**[2,4,13], **Yoko Yashiroda**[4], **Mami Yoshimura**[4], **Zhijian Li**[2], **Luis Alberto Vega Isuhuaylas**[2], **Kaori Itto-Nakama**[5], **Daisuke Yamanaka**[6], **Yoshikazu Ohya**[5,7], **Hiroyuki Osada**[4], **Bo Wang**[3,8,9,10], **Gary D. Bader**[1,2,9,11,12], **Charles Boone**[1,2,4]

[1]Department of Molecular Genetics, University of Toronto, Toronto, Ontario, Canada.

[2]The Donnelly Centre, University of Toronto, Toronto, Ontario, Canada.

[3]Vector Institute for Artificial Intelligence, Toronto, Ontario, Canada.

[4]RIKEN Center for Sustainable Resource Science, Wako, Saitama, Japan.

[5]Department of Integrated Biosciences, Graduate School of Frontier Sciences, The University of Tokyo, Kashiwa, Japan.

[6]Laboratory for Immunopharmacology of Microbial Products, School of Pharmacy, Tokyo University of Pharmacy and Life Sciences, Hachioji, Tokyo, Japan.

[7]Collaborative Research Institute for Innovative Microbiology, The University of Tokyo, Tokyo, Japan.

[8]Peter Munk Cardiac Center, University Health Network, Toronto, Ontario, Canada.

[9]Department of Computer Science, University of Toronto, Toronto, Ontario, Canada.

[10]Department of Laboratory Medicine and Pathobiology, University of Toronto, Toronto, Ontario, Canada.

[11]The Lunenfeld-Tanenbaum Research Institute, Sinai Health System, Toronto, Ontario, Canada.

**Correspondence and requests for materials** should be addressed to Bo Wang, Gary D. Bader or Charles Boone. bowang@vectorinstitute.ai; gary.bader@utoronto.ca; charlie.boone@utoronto.ca.

Author contributions
D.T.F. conceived and developed the method and computational experiments. S.C.L. and M.Y. performed the chemical–genetic screens. Z.L. provided resources for the TS mutant collection. L.A.V.I. preprocessed and provided the chemical–genetic data. H.O. provided the chemical matter and information about the screened compounds. S.C.L. and Z.L. constructed the drug-hypersensitive TS mutant collection. K.I.-N., D.Y. and H.O. performed the jervine biochemical validation. D.T.F., S.C.L., Y.Y., Y.O., B.W., G.D.B. and C.B. wrote the manuscript. B.W., G.D.B. and C.B. conceived and supervised the project.

Code availability
The BIONIC code is available at https://github.com/bowang-lab/BIONIC[77]. Code to reproduce the main figure analyses (Figs. 2–6) is available at https://github.com/duncster94/BIONIC-analyses[78] and a library implementing the coannotation prediction, module detection and gene function prediction evaluations is available at https://github.com/duncster94/BIONIC-evals [79]. The BIONIC integrated yeast features (PEG features) can be explored at https://bionicviz.com.

Competing interests
The authors declare no competing interests.

Additional information
**Extended data** are available for this paper at https://doi.org/10.1038/s41592–022-01616-x.

**Supplementary information** The online version contains supplementary material available at https://doi.org/10.1038/s41592-022-01616-x.

**Peer review information** *Nature Methods* thanks Kevin Yuk-Lap Yip and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. Primary Handling Editor: Lin Tang, in collaboration with the *Nature Methods* team.

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

[12]Princess Margaret Cancer Centre, University Health Network, Toronto, Ontario, Canada.

[13]These authors contributed equally: Duncan T. Forster, Sheena C. Li.

## Abstract

Biological networks constructed from varied data can be used to map cellular function, but each data type has limitations. Network integration promises to address these limitations by combining and automatically weighting input information to obtain a more accurate and comprehensive representation of the underlying biology. We developed a deep learning-based network integration algorithm that incorporates a graph convolutional network framework. Our method, BIONIC (Biological Network Integration using Convolutions), learns features that contain substantially more functional information compared to existing approaches. BIONIC has unsupervised and semisupervised learning modes, making use of available gene function annotations. BIONIC is scalable in both size and quantity of the input networks, making it feasible to integrate numerous networks on the scale of the human genome. To demonstrate the use of BIONIC in identifying new biology, we predicted and experimentally validated essential gene chemical–genetic interactions from nonessential gene profiles in yeast.

High-throughput functional genomics projects produce massive amounts of biological data for thousands of genes, often represented as gene–gene interaction networks, which link genes or proteins of related function[1]. These functional interaction networks have varying rates of false-positives and -negatives and integrating them promises to generate more accurate and complete functional networks. However, the diversity of experimental methods makes unifying this information a major challenge.

Existing network integration methods have not yet solved this problem. For example, many integration algorithms produce networks that retain only global topological features of the original networks, which can be at the expense of important local relationships[2–5], whereas others fail to effectively integrate networks with partially disjoint node sets[6,7]. Some methods incorporate too much noise in their output, for instance by using more dimensions than necessary to represent their output, which can be detrimental to gene function and functional interaction prediction quality[2–6]. Most data integration approaches do not scale in the number of networks or in the size of the networks required for real world settings[4,6,8]. Supervised methods have traditionally been the most common network integration approach[5,8–10] and, while highly successful, they require labeled training data to optimize their predictions of known gene functions, which may not be available. Moreover, annotations can be biased and limited, working only with known functional descriptions and reinforcing the existing understanding of gene relationships rather than identifying new ones.

To address the potential bias of supervised approaches, unsupervised biological network integration methods have recently been expl ored[2–4,6,7,11]. Theoretically, unsupervised network integration methods can provide a number of desirable features such as automatically retaining high-quality gene relationships and removing spurious ones, inferring new relationships based on the shared topological features of many networks in aggregate and outputting comprehensive results that cover the entire space of information associated with the input data, all while remaining agnostic to any particular view of biological function. However, practically, unsupervised methods still face challenges, such as scalability. Recently, new unsupervised data representation methods have been developed that focus on learning compact features over networks[12,13]. However, this approach produces general-purpose node features that are not necessarily optimal for the task of interest. Advances in deep learning have addressed this shortcoming with the development of the graph convolutional network (GCN), a general class of neural network architectures that are capable of learning features over networks[14–17] in a scalable manner. Compared to general-purpose node feature learning approaches[12,13], GCNs have substantially improved performance for a range of general network tasks[14,17].

Here we present a general, scalable deep learning framework for network integration called BIONIC (Biological Network Integration using Convolutions), which uses GCNs to learn a single, unified feature vector for each gene, given many different input networks. BIONIC addresses the aforementioned limitations of existing integration methods and produces integration results that accurately reflect the underlying network topologies and capture functional information. To demonstrate the use of BIONIC, we integrate three diverse, high-quality gene and protein interaction networks, to obtain integrated gene features that we compare to a range of function prediction benchmarks. We analyze our findings in the context of those obtained from a wide range of integration methodologies[12,17], and we show that BIONIC features perform well at both capturing functional information and scaling in terms of the number of networks and network size, while maintaining gene feature quality. Finally, we applied BIONIC network integration toward the analysis of chemical–genetic interactions[18], which allowed us to make predictions about the cellular targets of previously uncharacterized bioactive compounds.

## Results

### BIONIC architecture

BIONIC uses the GCN neural network architecture to learn optimal gene (protein) interaction network features individually and combines these features into a single, unified representation for each gene (Fig. 1). First, the input data, if not already in a network format, are converted to networks (for example, by gene expression profile correlation) (Fig. 1a). Each input network is then run through a sequence of GCN layers (Fig. 1b) to produce network-specific gene features. The number of GCN layers used (three layers in our experiments: Methods and Supplementary Data File 1) determines the size of the neighborhood (that is, genes directly connected to a given gene) used to update the gene features[14], where one layer would use only the gene's immediate neighbors, two layers would use the second order neighborhood and so on. Residual connections are added from the output of each network-specific GCN layer in the sequence to the output of the final GCN in the sequence (Extended Data Fig. 1). This allows BIONIC to learn gene features based on multiple neighborhood sizes rather than just the final neighborhood, while additionally improving training by preventing vanishing gradients[19]. The network-specific features are then summed through a stochastic gene dropout procedure to produce unified gene features that can be used in downstream tasks, such as functional module detection or gene function prediction. To optimize the functional information encoded in its integrated features, BIONIC must have relevant training objectives that facilitate capturing salient features across multiple networks. Here, BIONIC uses an unsupervised training objective, and if some genes have functional labels (such as complex, pathway or bioprocess membership annotations), BIONIC can also use these labels to update its learned features through a semisupervised objective.

For the unsupervised objective, BIONIC uses an autoencoder design and reconstructs each input network by mapping the integrated gene features to a network representation (decoding) and minimizing the difference between this reconstruction and the original input networks. By optimizing the fidelity of the network reconstruction, BIONIC forces the learned gene features to encode as much salient topological information present in the input networks as possible, which reduces the amount of spurious information encoded. Indeed, for a set of three yeast networks[20–22], inputting these networks into BIONIC individually tends to produce features with higher performance on several benchmarks compared to the original network format (Extended Data Fig. 2). This is likely due to the tendency for BIONIC to progressively embed related genes closer together during the training process, while ensuring unrelated genes remain far apart (Extended Data Fig. 3). By reconstructing the input networks, BIONIC is also trained to model the latent factors from each network that will best reconstruct all input networks.

For the semisupervised objective, BIONIC predicts labels for each gene using the integrated gene features and then updates its weights by minimizing the difference between the predictions and a set of user-specified ground-truth functional labels. Here, BIONIC performs multi-label classification, where a given gene may be assigned more than one class label. BIONIC ignores the classification error for any genes lacking ground-truth labels and so is able to incorporate as much (or as little) labeled information as is available. The semisupervised classification objective is used in conjunction with the unsupervised network reconstruction objective when gene labels are available, and the unsupervised objective is used on its own when no gene labels are available.

## Evaluation criteria

For the following analyses, we assessed the quality of the input networks and network integration method outputs using three evaluation criteria: (1) gene coannotation prediction; (2) gene module detection and (3) supervised gene function prediction. First, we used an established precision-recall evaluation strategy[22,23] to determine how well gene– gene relationships produced by the given method overlapped with gene pairs coannotated to the same term in a particular functional standard. Second, we evaluated the capacity of each method to produce biological modules by comparing clusters computed from the output of each method to known modules such as protein complexes, pathways and biological processes. These two evaluations measure the intrinsic quality of the outputs generated by the integration methods, that is without training any additional models on top of the outputs. Finally, the supervised gene function prediction evaluation determines how discriminative the method outputs are for predicting known gene functions. Here, a portion of the genes and corresponding labels (known functional classes such as protein complex membership) were held out and used to evaluate the accuracy of a support vector machine (SVM) classifier[24], which is trained on the remaining gene features, output from the given integration method, to predict the held-out labels[7]. This constitutes an extrinsic evaluation, indicating how effectively the method outputs can be used in conjunction with an additional classification model.

In the following experiments, to ensure a fair choice of hyperparameters across BIONIC and the integration methods we compared to, we performed a hyperparameter optimization step using an independent set of *Schizosaccharomyces pombe* networks as inputs[25–27] and a set of Gene Ontology (GO) curated pombe protein complexes[28] for evaluation. The best performing hyperparameters for each approach were used (Methods).

## Evaluation of BIONIC features and input networks

We first used the unsupervised BIONIC to integrate three diverse yeast networks: a comprehensive network of correlated genetic interaction profiles (4,529 genes, 33,056 interactions)[22], a coexpression network derived from transcript profiles of yeast strains carrying deletions of transcription factors (1,101 genes, 14,826 interactions)[21] and a protein–protein interaction network obtained from an affinity-purification mass-spectrometry assay (2,674 genes, 7,075 interactions)[20], which combine for a total of 5,232 unique genes and 53,351 unique interactions (Fig. 2 and Supplementary Data File 2). Compared to the input networks, BIONIC integrated features have equivalent or superior performance on all evaluation criteria over three different functional benchmarks: IntAct protein complexes[29], Kyoto Encyclopedia of Genes and Genomes (KEGG) pathways[30] and GO Biological Processes[28] (Fig. 2a and Supplementary Data File 3). The coannotation and module detection benchmarks contain between 1,786 and 4,170 genes overlapping the integration results. The module detection benchmarks define between 107 and 1,809 modules. The IntAct, KEGG and GO Biological Process gene function prediction benchmarks cover 567, 1,770 and 1,211 genes overlapping the integration results, and 48, 53 and 63 functional classes, respectively (Supplementary Data File 2). As an additional test, BIONIC produces high-quality features that accurately predict a diverse set of yeast biological process annotations per gene[22] (Fig. 2b). Some categories in this last test do better than others. These performance patterns were mirrored in the individual input networks, indicating that this is the result of data quality, rather than method bias.

We observed that features obtained through BIONIC network integration often outperformed the individual input networks at capturing functional modules (Fig. 2a) and captured more modules (Fig. 2c and Supplementary Data File 4), demonstrating the use of the combined features over individual networks for downstream applications such as module detection. Here we treated the network adjacency profiles (rows in the adjacency matrix) as gene features. We then examined how effectively the input networks and integrated BIONIC features captured known protein complexes, by matching each individual known complex to its best-matching predicted module and quantifying the overlap (Fig. 2c). We then compared the overlap scores from each network to the BIONIC overlap scores to identify complexes where BIONIC performs either better or worse than the input networks. Of 344 protein complexes tested, BIONIC strictly improved 196, 309 and 222 complex predictions and strictly worsened 82, 17 and 98 complex predictions compared to the input protein–protein interaction, coexpression and genetic interaction networks, respectively. The distributions of complex overlap scores for each dataset indicate that BIONIC predicts protein complexes more accurately than the input networks on average. Indeed, if we use an overlap score of 0.5 or greater to indicate a successfully captured complex, the integrated BIONIC features, containing information from three networks, capture 121 complexes, compared to 88, 3 and 74 complexes for the individual protein–protein interaction, coexpression and genetic interaction networks, respectively (Fig. 2c). We also repeated this module analysis while optimizing the clustering parameters on a per-module basis, an approach that tests how well each network and BIONIC perform at capturing modules under optimal clustering conditions for each module. Here too, the integrated BIONIC features capture more modules and with a greater average overlap score than the individual input networks (Extended Data Figs. 4 and 5 and Supplementary Data File 5).

To better understand how BIONIC is able to improve functional gene module detection compared to the input networks, we examined the LSM2–7 complex, which was identified in our module detection evaluation (Fig. 2a) as an example to show how BIONIC effectively combines gene–gene relationships across different networks and recapitulates known biology. The LSM2–7 complex localizes to the yeast nucleoli and is involved in the biogenesis or function of the small nucleolar RNA *SNR5* (ref. [31]). LSM2–7 is made up of the protein products of six genes: *LSM2*, *LSM3*, *LSM4*, *LSM5*, *LSM6* and *LSM7*. We found that the cluster that best matched the LSM2–7 complex in each input network only captures a subset of the full complex (Supplementary Data File 4). The BIONIC module, however, contains five out of six members of the LSM2–7 complex, along with two additional members: LSM1 and PAT1, which are functionally associated with the LSM2–7 complex[32]. The missing member, LSM5, is in the local neighborhood of the cluster in the BIONIC feature space. We examined the best-matching clusters and their local neighborhood, consisting of genes that show a direct interaction with predicted members of the LSM2–7 complex, in the input networks, and in a profile similarity network obtained from the integrated BIONIC features of these networks (Fig. 2d). We found that both the PPI and genetic interaction networks captured two members of the LSM2–7 complex, with two additional members in the local neighborhood. The coexpression network only identified one complex member, and the local neighborhood of the best-matching module did not contain any additional known complex members. Finally, BIONIC used the interaction information across input networks to better identify the LSM2–7 module, with the addition of two functionally related proteins. This analysis demonstrates the use of BIONIC for identifying meaningful biological modules by effectively combining information across input networks. Indeed, when we optimized the module detection procedure to specifically resolve the LSM2–7 complex, we found that BIONIC was able to capture the complex with a higher overlap score (0.83) than any of the input networks (0.33, 0.17 and 0.50 for the PPI, coexpression and genetic interactions networks, respectively), and it outperformed other integration methods (0.43, 0.22, 0.44, 0.60 and 0.68 for the Union, iCell, deepNF, Mashup and multi-node2vec methods, respectively) (Supplementary Data File 5).

We also performed an analysis to examine how BIONIC features encode the input networks and found that the input networks are generally encoded uniformly across feature dimensions (Supplementary Note 1 and Extended Data Fig. 6).

## Evaluation of BIONIC and established unsupervised integration methods

We compared network integration results from the unsupervised BIONIC (Fig. 2) to those derived from several different established integration approaches: a naive union of networks (Union), a nonnegative matrix tri-factorization approach (iCell)[2], a deep learning multi-modal autoencoder (deepNF)[11], a low-dimensional diffusion state approximation approach (Mashup)[7] and a multi-network extension of the node2vec (ref. [13]) model (multi-node2vec)[33] (Fig. 3). These unsupervised integration methods cover a wide range of methodologies and the main possible output types (networks for Union and iCell, features for deepNF, Mashup and multi-node2vec). BIONIC performs as well as, or better than the tested integration methods across all evaluation types and benchmarks (Fig. 3a). We also evaluated BIONIC and the other integration approaches on a per-biological process basis (Fig. 3b). Here we found BIONIC generally outperforms the established integration approaches on each biological process, with the exception of several biological processes when compared to deepNF. Averaging over the performance for each biological process, we found BIONIC performs on par with deepNF (average precision of 0.53 for BIONIC compared to 0.52 for deepNF). DeepNF performs competitively on the per-biological process evaluations (Fig. 3b), but it underperforms on the global performance evaluations (Fig. 3a). The per-biological process evaluations assess how well a method predicts large-scale biological process coannotation, whereas the global performance evaluations measure how well a method predicts smaller-scale functional modules (that is, protein complexes). This discrepancy in performance indicates deepNF is able to capture broad-scale functional organization, but it fails to resolve smaller functional modules. BIONIC performs well on both of these evaluations, however, indicating it can learn gene features that resolve both broad and detailed functional organization. To ensure these results are consistent under different input networks, we integrated a set of yeast-two-hybrid networks and found similar performance patterns (Supplementary Note 2 and Extended Data Fig. 7).

## Evaluation of BIONIC in a semisupervised setting

We also tested how BIONIC performs in a semisupervised setting (Fig. 4). Here, we compared BIONIC trained with no labeled data (unsupervised), BIONIC trained with a held-out set of functional labels given by IntAct, KEGG and GO (semisupervised), and a supervised integration algorithm using the same labels (GeneMANIA[5]). For each of these methods, we integrated the yeast protein–protein interaction, coexpression and genetic interaction networks from the Fig. 2 analysis. In each benchmark, 20% of genes (IntAct, KEGG, GO) were randomly held out and used as a test set, while the remaining 80% of genes were used for training. The unsupervised BIONIC did not use any gene label information for training, but it was evaluated using the same test set as the supervised methods to ensure a consistent performance comparison. To control for variability in the train-test set partitioning, this procedure was repeated ten times and the average performance across test sets was reported (Methods). We found that adding labeled data can substantially improve the features BIONIC learns and these features also outperform the integration results produced by the supervised GeneMANIA method. We also found that even without labeled data, BIONIC performs as well as, or exceeds, GeneMANIA performance. Notably, the performance of the unsupervised and semisupervised BIONIC is similar for gene function prediction. This indicates unsupervised BIONIC features are already sufficiently discriminative for classifiers to perform well. Thus, BIONIC can be used effectively in both an unsupervised and semisupervised setting, which demonstrates its versatility as a biological network integration algorithm. We also analyzed the performance of BIONIC in a scenario where labels are noisy (Supplementary Note 3 and Extended Data Fig. 8).

## Scalability of BIONIC and established integration approaches

An effective integration algorithm should be able to scale to many network inputs and networks with many nodes. To test network input scalability, we randomly sampled progressively larger sets of yeast gene coexpression networks (Fig. 5a and Supplementary Data File 2) and assessed the performance of the resulting integrations of these sets. We similarly tested node scalability by randomly subsampling progressively larger gene sets of four human protein–protein interaction networks[34–37] (Fig. 5b and Supplementary Data File 2). BIONIC can integrate numerous networks (Fig. 5a), and networks with many nodes (Fig. 5b), outperforming all other methods assessed for progressively more and larger networks. To achieve this scalability, BIONIC takes advantage of the versatile nature of deep learning technology by learning features for small batches of genes and networks at a time, reducing the computational resources required for any specific training step. To learn gene features over large networks, BIONIC learns features for random subsets of genes at each training step and randomly subsamples the local neighborhoods of these genes to perform the graph convolution (Methods), maintaining a small overall computational footprint. This subsampling allows BIONIC to integrate networks with many genes, whereas methods such as Mashup can only do so with an approximate algorithm that reduces integration performance (Supplementary Fig. 1). To integrate many networks, BIONIC uses a network-wise sampling approach, where a random subset of networks is integrated at a time during each training step. This reduces the number of parameter updates required at once, since only GCNs corresponding to the subsampled networks are updated in a given training step. We also tested the extent of BIONIC scalability in terms of computational resources (Supplementary Note 4 and Extended Data Fig. 9).

## BIONIC predictions of chemical–genetic interactions

We asked whether BIONIC can generate new, testable biological hypotheses. Chemical–genetic approaches analyze the effects of mutations on cell growth in response to compound treatment and can be used to systematically predict the molecular targets of uncharacterized compounds[38]. For example, if a conditional temperature sensitive (TS) mutant carries a mutation that compromises the activity of a compound's target gene, it is often specifically hypersensitive to the compound[39,40].

Previously, we generated a dataset of chemical–genetic screens, consisting of a pool of deletion mutants of 289 nonessential genes (diagnostic pool) and 1,522 compounds[18]. Using this data, we used BIONIC to predict chemical sensitivities for a wider set of 873 essential genes across a subset of 50 compounds. For the compound selection procedure, we used the unsupervised BIONIC integrated protein–protein interaction network, coexpression network and genetic interaction network features from the Fig. 2 analysis which we refer to as the physical, expression, and genetic (PEG) features. We selected compounds to study by identifying those that BIONIC predicts well within the diagnostic pool data. We did this by partitioning sensitive genes from each compound into train and test sets, and we used the BIONIC features to predict the test set genes using the training genes as input (Methods). The top 50 compounds, for which sensitive genes were most successfully predicted, were selected for further analysis. Sensitive essential gene predictions for each of the 50 chosen compounds were generated in a similar way to the compound selection procedure, with predictions being made on yeast essential genes rather than the diagnostic pool genes (Methods).

The BIONIC essential gene sensitivity predictions were experimentally validated using profiles for the compound set from a chemical– genetic screen using a collection of TS yeast mutants (Supplementary Data File 6). A DNA barcoded collection of 1,181 mutants containing TS alleles spanning 873 genes was constructed in a yeast genetic background that conferred drug hypersensitivity (*pdr1Δpdr3Δsnq2Δ*). The TS mutant collection was pooled and screened against the compound set. Mutant-specific barcodes were amplified from each compound-treated pool, and Illumina sequencing was used to quantify the relative abundance of TS mutant strains in the presence of each compound. Sequencing data was processed using BEAN-counter software to quantify chemical–genetic interactions and eliminate nonspecific technical effects[41]. Further statistical analysis was conducted to identify chemical–genetic interactions that satisfied a ‘far outlier’ cutoff (Methods), which were then compared to the sensitive genes predicted by BIONIC.

Out of 156 essential genes experimentally identified as sensitive to the set of 50 screened compounds, BIONIC successfully predicted 35. BIONIC significantly predicts sensitive genes for 13 out of 50 compounds under an ordered Fisher’s exact test. We also assessed more broadly whether BIONIC can correctly predict the biological process a given compound’s sensitive genes are annotated to. BIONIC sensitive gene predictions were statistically enriched (Fisher’s exact test) for 27 out of 62 annotated biological processes across compounds (Fig. 6a). We compared the quality of BIONIC’s predictions to a random baseline (Fig. 6b). Here, we generated 1,000 random permutations of the BIONIC PEG feature gene labels and computed sensitive essential gene predictions for the 50 screened compounds, as described previously. We found BIONIC sensitive gene and bioprocess predictions were substantially more accurate than the random permutations, indicating the BIONIC PEG features encode relevant information for the prediction of chemical–genetic interactions. We looked at the 13 significantly predicted compounds in more detail to see which sensitive gene predictions BIONIC correctly predicted and the corresponding ranks of those genes in the prediction list (Fig. 6c). We observed that for eight out of 13 compounds, the correct BIONIC predictions rank in the top ten most sensitive interactions. BIONIC predictions and experimental results for the 50 selected compounds can be found in Supplementary Data File 7.

We examined the best predicted compound, NP329, in more detail. NP329 is a pseudojervine from the RIKEN Natural Product Depository[42], and among its top ten most sensitive interactions with the diagnostic pool mutants were the *FLC2*, *DFG5*, *GAS1* and *HOC1* (ref. [18]) genes. The *FCL2* product is a putative calcium channel involved in cell wall maintenance[43], *DFG5* encodes a glycosylphosphatidylinositol (GPI)-anchored membrane protein required for cell wall biogenesis in bud formation[44], *GAS1* encodes a β-1,3-glucanosyltransferase required for cell wall assembly[45–47], and *HOC1* codes for an alpha-1,6-mannosyltransferase involved in cell wall mannan biosynthesis[48]. By comparing NP329's diagnostic pool gene sensitivity profiles with the compendium of genetic interactions mapped in yeast and analyzing our data using the CG-TARGET software for chemical–genetic profile interpretation[18,49], the top three high-confidence GO bioprocesses predicted to be perturbed by NP329 were 'cell wall biogenesis' (GO:0042546), 'cell wall organization or biogenesis' (GO:0071554) and 'fungal-type cell wall organization or biogenesis' (GO:0071852). This strongly implicates the pseudojervine NP329 as a disrupter of proper cell wall biogenesis in yeast.

To further study this compound–process interaction, we hierarchically clustered the BIONIC PEG features and we focused on the essential genes present in the Fig. 2b 'glycosylation, protein folding/targeting, cell wall biosynthesis' bioprocess (Fig. 6d). We observed that six out of 16 NP329 sensitive essential genes lie in the bioprocess, as do 18 out of 20 BIONIC predicted sensitive essential genes. Within this bioprocess, BIONIC successfully predicts four (*BIG1*, *KRE5*, *KRE9*, *ROT1*) out of the six NP329 sensitive essential genes. These results indicate that BIONIC is able to both predict a relevant biological process targeted by the compound and the specific sensitive genes. Moreover, the four sensitive genes successfully predicted by BIONIC were all closely clustered together based on the integrated BIONIC features (Fig. 6). *ROT1* encodes an essential chaperone required for N- and O-glycosylation in yeast[50] and is required for normal levels of β-1,6-glucan[51]. Both *KRE5* and *BIG1* are also required for proper β-1,6-glucan synthesis[52,53]. These interactions further indicate NP329 can interfere in the proper synthesis of β-1,6-glucan, an essential cell wall component. Since the chemical structure of NP329 is extremely similar to the steroidal alkaloid jervine, we tested the effect of jervine on the production of β-1,6-glucan. *KRE6* is a nonessential gene that, like its paralog *SKN1*, encodes a glucosyl hydrolase required for β-1,6-glucan biosynthesis[54]. We found that treatment of cells with 5 μg ml$^{-1}$ of jervine reduced β-1,6-glucan levels to the same extent as a kre6 deletion mutant, likely by inhibiting *KRE6* and its paralog *SKN1* (Extended Data Fig. 10). In a more detailed analysis, we found that point mutations in *KRE6* or *SKN1* can lead to jervine resistance, which further suggests that jerveratrum-type steroidal alkaloids target Kre6 and Skn1 (ref. [55]). These results show that BIONIC can predict relevant chemical–genetic interactions and has the potential to link compounds to their cellular targets.

## Discussion

BIONIC is a deep learning algorithm that extends the GCN architecture to integrate biological networks. BIONIC produces gene features that capture functional information well when compared to other unsupervised methods[12,17] as determined by a range of benchmarks and evaluation criteria. BIONIC can use labeled data in a semisupervised fashion when it is available, and it can be purely unsupervised otherwise. BIONIC scales to a greater number of input networks and network sizes compared to established unsupervised methods.

BIONIC can be used to predict pairs of related genes (coannotation prediction), identify functional gene modules (module detection) and accurately predict functional gene labels (gene function prediction). One of the main goals of this work is to generate fully integrated features encoding functional information for a particular organism, such that the resulting features can be used to predict numerous different aspects of cell and organism function[56]. As a proof-of-concept, we integrated three different yeast networks, incorporating protein–protein interaction, coexpression and genetic interaction data. We also used BIONIC features to generate predictions for essential gene chemical sensitivities and a substantial number were experimentally validated, indicating BIONIC is effective at prediction and hypothesis generation.

BIONIC is capable of capturing relevant functional information across input networks. However, input networks do not have uniform quality and some networks may only describe certain types of functional relationship effectively (such as those within a particular biological process) while obscuring other relationships. Indeed, while BIONIC is able to capture a greater number of functional modules in an integrated network compared to a single input network (Fig. 2c and Extended Data Fig. 5), it does not capture every functional module present in the input networks (Fig. 2c, Extended Data Fig. 5 and Supplementary Data Files 4 and 5). This is likely due to some networks obscuring signals present in other networks. Implementing more advanced input network feature weighting should ensure that high-quality information is preferentially encoded in the learned features and that low-quality information is not. This may help to identify which functional relationships are driven by which networks and network types, thereby indicating which parts of the functional range have good or poor coverage and identifying areas to target for future experimental work.

The naive union of networks approach performs well, motivating its inclusion as a baseline in any network integration algorithm assessment. While the union network contains all possible relationships across networks, it likely contains relatively more false-positive relationships in the integrated result, since all false-positives in the input networks are retained by the union operation. Thus, the union should work well for high-quality networks, but perform poorly with noisy networks.

Our chemical–genetic analysis demonstrates the potential of BIONIC to provide target predictions from limited experimental data. While BIONIC performs well at predicting essential gene chemical– genetic interactions, further improvements in performance could potentially be made through an optimized choice of input networks that specifically indicate these chemical sensitivities. BIONIC chemical–genetic interaction predictions could also be used to instead generate a set of putative nonsensitive genes for a given compound, indicating bioprocesses where the compound is not active. This would reduce the size of the experimental space when screening, resulting in more rapid and less expensive data generation. Finally, strong BIONIC chemical–genetic interaction predictions that are not reflected in the experimental data could indicate experimental false negatives that require additional investigation.

BIONIC learns gene features based solely on their topological role in the given networks. A powerful future addition to BIONIC would be to include gene or protein features such as amino acid sequence[57], protein localization[58], mutant morphological defect[59] or other nonnetwork features to provide additional context for genes in addition to their topological role. Continued development of integrative gene function prediction using deep learning-based GCN and encoder–decoder technologies will enable us to map gene function more richly and at larger scales than previously possible.

## Online content

Any methods, additional references, Nature Research reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41592-022-01616-x.

## Methods

### BIONIC method overview

An undirected input network can be represented by its adjacency matrix $A$ where $A_{ij} = A_{ji} > 0$ if node $i$ and node $j$ share an edge and $A_{ij} = A_{ji} = 0$ otherwise. BIONIC first preprocesses each input network to contain the union of nodes across all input networks and ensures the corresponding row and column orderings are the same. In instances where networks are extended to include additional nodes not originally present in them (so all input networks share the same union set of nodes), the rows and columns corresponding to these nodes are set to 0.

BIONIC encodes each input network using instances of a GCN variant known as the graph attention network (GAT)[17]. We selected this architecture because of its considerable performance improvements over existing architectures on a range of node classification tasks[17]. The GAT has the ability to learn alternative network edge weights, allowing it to downweight or upweight edges based on their importance for the network reconstruction task. In the original formulation, the GAT assumes binary network inputs. We modify the GAT to consider a priori network edge weights. The GAT formulation is then given by:

$$\mathrm{GAT}(A, H) = \sigma\left(\alpha H W^{\top}\right) \tag{1}$$

where

$$\alpha_{ij} = \frac{A_{ij}\exp\left(\sigma\left(\mathbf{a}^{\top}[W\mathbf{h}_i \parallel W\mathbf{h}_\mathbf{j}]\right)\right)}{\sum_{k=1} A_{ik}\exp\left(\sigma\left(\mathbf{a}^{\top}[W\mathbf{h}_i \parallel W\mathbf{h}_\mathbf{k}]\right)\right)} \tag{2}$$

Here, $W$ is a trainable weight matrix that projects aggregated node features into another feature space, $k$ represents nodes in the neighborhood of $i$, $\mathbf{a}$ is a vector of trainable attention coefficients that determine the resulting edge weighting, $\top$ is the transpose operation, $\mathbf{h}_i$ is the feature vector for node $i$ (that is, the $i$th row of feature matrix $H$), $\|$ denotes the concatenation operation and $\sigma$ corresponds to a nonlinear function (in our case, a leaky rectified linear unit (LeakyReLU)) that produces more sophisticated features than linear maps. Equation (1) corresponds to a node neighborhood aggregation and projection step that incorporates an edge weighting scheme (equation (2)). In practice, several edge weighting schemes (known as attention heads) are learned and combined simultaneously, resulting in:

$$\mathrm{GAT}(A, H) = \|_{k=1}^{K} \sigma\left(\alpha^{(k)} H W^{(k)\top}\right) \tag{3}$$

where $K$ is the number of attention heads. This is done to stabilize the attention learning process, as per the author's original results[17]. In our experiments, we use ten attention heads per GAT encoder, each with a hidden dimension of 68, as per our hyperparameter optimization results (Obtaining integrated results and Supplementary Data File 1).

Initial node features $H_{\mathrm{Init}}$ are one-hot encoded so that each node is uniquely identified (that is, $H_{\mathrm{Init}} = I$ where $I$ is the identity matrix). These features are first mapped to a lower dimensional space through a learned linear transformation to reduce memory footprint and improve training time. BIONIC encodes each network by passing it through several sequential GAT layers to learn node features based on higher-order neighborhoods. Outputs from each GAT pass are then summed to produce the final network-specific features (Extended Data Fig. 1). Based on the hyperparameter optimization results, we used three GAT layers in our experiments. We found BIONIC to be robust to the number of layers (Supplementary Fig. 2). After all networks are separately encoded, the network-specific node features are combined through a weighted, stochastically masked summation given by:

$$H_{\mathrm{combined}} = \sum_{j=1}^{N} s_j \mathbf{m}^{(\mathbf{j})} \odot H^{(j)} \tag{4}$$

Here, $N$ is the number of input networks, $s_j$ is the learned scaling coefficient for feature representations of network $j$, $\odot$ is the element-wise product, $H^{(j)}$ is the matrix of learned feature vectors for nodes in network $j$ and $\boldsymbol{m}^{(j)}$ is the node-wise stochastic mask for network $j$, calculated as:

$$m_i^{(j)} = \begin{cases} 1, & \text{if node } i \text{ is unique to network } j \text{ or } m_i^{(k \neq j)} = 0 \\ 0, & \text{if node } i \text{ is not in unextended network } j \\ \dfrac{X}{\sum_{k=1}^{N} m_i^{(k)}}, & X \sim \text{Bernoulli } (0.5), \text{ otherwise} \end{cases} \tag{5}$$

The mask $m$ is designed to randomly drop node feature vectors produced from networks with the constraint that a node cannot be masked from every network, and node features from nodes not present in the original, unextended networks are dropped. This masking procedure forces the network encoders to compensate for missing node features in other networks, ensuring the encoders learn cross-network dependencies and map their respective node features to the same feature space. The network scaling vector $\mathbf{s}$ in equation (4) enables BIONIC to scale features in a network-wise fashion, affording more flexibility in learning the optimal network-specific node features for the combination step. $\mathbf{s}$ is learned with the constraint that its elements are positive and sum to 1, ensuring BIONIC does not over- or negatively scale the features.

We found that learning the integrated features in this joint manner (learning and combining network-specific features end-to-end) performs better than simply concatenating the network-specific features (that is, late fusion), indicating that BIONIC is able to learn complementary information across input networks (Supplementary Fig. 3).

To obtain the final, integrated node features $F$, BIONIC maps $H_{\text{combined}}$ to a low-dimensional space through a learned linear transformation. In $F$, each column corresponds to a specific learned feature and each row corresponds to a node. We found the quality of the integrated features was generally robust to the number of feature dimensions, with performance saturating at 512 features (Supplementary Fig. 4). We also assessed the denoising capabilities of BIONIC (Supplementary Fig. 5). Here we progressively added false-positive edges to a yeast PPI network[20] and determined how well these noisy networks can predict protein complex coannotation relationships compared to the BIONIC features learned by encoding these same networks. We found that the low-dimensional feature learning approach is more robust to input network noise than the noisy networks themselves.

To obtain a high-quality $F$, BIONIC uses an unsupervised training objective. When gene labels are provided, an additional semisupervised training objective is also used. For the unsupervised training objective, BIONIC decodes $F$ into reconstructions of the original input networks and minimizes the discrepancy between the reconstructions and the inputs. The decoded network reconstruction is given by:

$$\hat{A} = F \cdot F^{\top} \tag{6}$$

The unsupervised loss is then given by:

$$L_{\text{unsupervised}} = \frac{1}{n^2}\sum_{j=1}^{N}\left\|\mathbf{b}^{(j)} \odot \left(\hat{A} - A^{(j)}\right) \odot \mathbf{b}^{(j)\top}\right\|_F^2 \tag{7}$$

where $n$ is the total number of nodes present in the union of networks, $\mathbf{b}^{(j)}$ is a binary mask vector for network $j$ indicating which nodes are present (value of 1) or extended (value of 0) in the network, $A^{(j)}$ is the adjacency matrix for network $j$ and $\| \cdot \|_F$ is the Frobenius norm. This loss represents computing the mean squared error between the reconstructed network $\hat{A}$ and input $A^{(j)}$ while the mask vectors remove the penalty for reconstructing nodes that are not in the original network $j$ (that is, extended), then summing the error for all networks.

For the semisupervised training objective, BIONIC first predicts gene labels by mapping $F$ to a matrix of class predictions as follows:

$$\hat{Y} = S(FW_{\text{classifiler}}) \tag{8}$$

where $S$ is the sigmoid function and $W_{\text{classifiler}}$ is a trainable weight matrix. The resulting class prediction matrix $\hat{Y}$ has genes as rows and class labels as columns. The ground-truth label matrix $Y$ indicates the correct labels for a set of genes in the input networks. $Y$ is extended to include zero vectors for any genes present in the input networks but not present in the labels, ensuring it has the same shape as $\hat{Y}$. The semisupervised loss is then given by:

$$L_{\text{semisupervised}} = \frac{1}{nC} \sum_{i=1}^{n} \sum_{j=1}^{C} b_{\text{labels}_i} \odot -\left(Y_{ij} \log\left(\hat{Y}_{ij}\right) + (1 - Y_{ij}) \log\left(1 - \hat{Y}_{ij}\right)\right) \tag{9}$$

where $n$ is the total number of nodes present in the union of networks, $C$ is the number of classes, $b_{\text{labels}_i}$ is a binary mask indicating whether node $i$ was present in the original label set (value of 1) or was extended (value of 0) and log indicates the natural logarithm. This loss represents the masked binary cross entropy between the predicted labels $\hat{Y}$ and the true labels $Y$ ignoring the loss of any nodes not originally present in $Y$.

The final loss BIONIC trains to minimize is a weighted sum of the unsupervised and semisupervised losses:

$$L = \lambda L_{\text{unsupervised}} + (1 - \lambda) L_{\text{semisupervised}} \tag{10}$$

where $\lambda$ is a value in the range [0, 1] indicating the relative weights of the two losses. When no labeled data are available, $\lambda$ is set to 1.

## Implementation details

BIONIC was implemented using PyTorch[60], a popular Python-based deep learning framework, and relies on functions and classes from the PyTorch Geometric library[61]. It uses the Adam[62] optimizer to train and update its weights. To be scalable in the number of networks, BIONIC uses an optional network batching approach where subsets of networks are sampled and integrated at each training step. The sampling procedure is designed so that each network is integrated exactly once per training step. Network batching yields a constant memory footprint at the expense of increased runtime with no empirical degradation of feature quality. This feature is provided for additional scalability over what is demonstrated in this work and was not used in any of our reported experiments. Additionally, BIONIC is scalable in the number of network nodes. It uses a node sampling approach (equivalent to mini-batch training, where nodes are samples) to learn features for subsets of nodes in a network, and a neighborhood sampling procedure to subsample node neighborhoods. Node sampling ensures only part of a network needs to be retained in memory at a time while neighborhood sampling reduces the effective higher-order neighborhood size in sequential GAT passes, again reducing the number of nodes required to be retained in memory at any given time, further reducing BIONIC's memory footprint.

For very large networks where the initial node feature matrix (that is, the identity matrix) cannot fit into memory due to limitations with PyTorch matrix operations, BIONIC incorporates a singular value decomposition-based approximation. First, the union of networks is computed by creating a network that contains the nodes and edges of all input networks. If an edge occurs in multiple networks, the maximum weight is used. A low-dimensional singular value decomposition approximation of the normalized Laplacian matrix of the union network is computed and used as the initial node features for each network. Finally, BIONIC uses sparse representations of network adjacency matrices (except for the input node feature matrix, see above), further reducing memory footprint. All BIONIC integration experiments in this paper were run on an NVIDIA Titan Xp graphical processing unit with 12 GB of video RAM, no more than 16 GB of system RAM and a single 2.4 GHz Intel Xeon CPU.

## Network preprocessing

The yeast protein–protein interaction network[20] and human protein–protein interaction networks[34–37] were obtained from BioGRID[63], genetic interaction profiles[22] were obtained directly from the published supplementary data of Costanzo et al.[22], and gene expression profiles[21] were obtained from the SPELL database[64]. These networks were chosen since they had the most functional information compared to other networks in their class (that is, protein–protein interaction networks, coexpression networks and genetic interaction networks). To create a network from the genetic interaction profiles, genes with multiple alleles were collapsed into a single profile by taking the maximum profile values across allele profiles. Pairwise Pearson correlation between the profiles was then calculated, and gene pairs with a correlation magnitude greater than or equal to 0.2 were retained as edges, as established[22]. For the gene expression profiles, networks were constructed by retaining gene pairs with a profile Pearson correlation magnitude in the 99.5th percentile. Coexpression and genetic interaction networks had their edge weights normalized to the range [0, 1].

## Obtaining integrated results

The naive union of networks benchmark was created by taking the union of node sets and edge sets across input networks. For edges common to more than one network, the maximum weight was used. For all other methods, automated hyperparameter optimization was performed to ensure hyperparameters were chosen consistently and fairly. Here, a *S. pombe* genetic interaction network[27], coexpression network[26] and protein–protein interaction network[25] were used as inputs to the integration methods. To perform one iteration of the hyperparameter optimization, a random combination of hyperparameters was uniformly sampled over a range of reasonable values for each method and used to integrate the three pombe networks. The integration results were then evaluated using a pombe protein complex standard (obtained from https://www.pombase.org/data/annotations/Gene_ontology/GO_complexes/Complex_annotation.tsv). The evaluations consisted of a coannotation prediction, module detection and gene function prediction assessment (Evaluation methods). This procedure was repeated for 50 combinations of hyperparameters, for each method. For methods that produced features (deepNF, Mashup, multi-node2vec and BIONIC), a feature dimension of 512 was used to ensure results were comparable across methods. For methods that required a batch size parameter (deepNF and BIONIC), the batch size was set to 2,048 to ensure reasonable computation times. Hyperparameter combinations were then ranked for each method across the three evaluation types and the hyperparameter combination corresponding to the highest average rank across evaluation types was chosen. The hyperparameter optimization results are found in Supplementary Data File 1. Note that the union method was not included in the hyperparameter optimization because it has no hyperparameters. Additionally, the Mashup method used 44 hyperparameter combinations rather than 50, as six hyperparameter combinations exhausted the available memory resources and did not complete.

All integration results reported were obtained by integrating networks using the set of hyperparameters identified in the hyperparameter optimization procedure. BIONIC features used in the Figs. 2, 3 and 6 analyses are found in Supplementary Data File 8. Coannotation prediction, module detection and gene function prediction standards used in Figs. 2–5 are found in Supplementary Data File 9.

## Benchmark construction

Functional benchmarks were derived from GO Biological Process ontology annotations, KEGG pathways and IntAct complexes for yeast, and CORUM complexes for human (Supplementary Data File 3). Analyses were performed using positive and negative gene pairs, clusters or functional labels obtained from the standards as follows: the GO Biological Process benchmark was produced by filtering Inferred from Electronic Annotation (IEA) annotations, as they are known to be lower quality, removing genes with dubious open reading frames and filtering terms with more than 30 annotations (to prevent large terms, such as those related to ribosome biogenesis, from dominating the analysis[65]). We found the performance evaluations to be robust to this threshold (Supplementary Fig. 6). For the coannotation benchmark, all gene pairs sharing at least one annotation were retained as positive pairs, while all gene pairs not sharing an annotation were considered to be negative pairs. KEGG, IntAct and CORUM benchmarks were produced analogously, without size filtering.

For the module detection benchmark, clusters were defined as the set of genes annotated to a particular term, for each standard. Modules of size 1 (singletons) were removed from the resulting module sets as they are uninformative. For the per-module analyses in Fig. 2c, Extended Data Figs. 4 and 5 and Supplementary Data Files 4 and 5, we also removed any modules of size 2 since these modules had highly variable Jaccard scores.

The supervised standards were obtained by treating each gene annotation as a class label, leading to genes with multiple functional classes (that is, a multi-label classification problem). The standards were filtered to only include classes with 20 or more members for GO Biological Process and KEGG, or ten members for IntAct. This was done to remove classes with very few data points, ensuring more robust evaluations.

The granular function standard in Figs. 2b, 3b and 6 was obtained from the Costanzo et al.[22] supplementary materials. Any functional category with fewer than 20 gene members was removed from the analysis to ensure only categories with robust evaluations were reported.

## Evaluation methods

We used a precision-recall-based coannotation framework to evaluate individual networks and integrated results. We used precision-recall instead of receiving operator curve because of the substantial imbalance of positives and negatives in the pairwise benchmarks for which the receiving operator curve would overestimate performance. Here, we computed the pairwise cosine similarities between gene profiles in each network or integration result. Due to the high-dimensionality of the datasets, cosine similarity is a more appropriate measure than Euclidean distance since the contrast between data points is reduced in high-dimensional spaces under Euclidean distance[66]. Precision-recall operator points were computed by varying a similarity threshold, above which gene or protein pairs are considered positives and below which pairs are considered negative. Each set of positive and negative pairs was compared to the given benchmark to compute precision and recall values. To summarize the precision-recall curve into a single metric, we computed average precision (AP) given by:

$$\mathrm{AP} = \sum_{i=1}^{n} (R_i - R_{i-1}) P_i \tag{11}$$

where $n$ is the number of operator points (that is, similarity thresholds) and $P_i$ and $R_i$ are the precision and recall values at operator point $i$, respectively. This gives the average of precision values weighted by their corresponding improvements in recall. We chose this measure over the closely related area under the precision-recall curve measure since this interpolates between operator points and tends to overestimate actual performance[67].

The module detection evaluation was performed by clustering the integrated results from each method and comparing the coherency of resulting clusters with the module-based benchmarks. Since the benchmarks contain overlapping modules (that is, one gene can be present in more than one module) that prevent the use of many common clustering evaluation metrics (since these metrics assume unique assignment of gene to cluster), the module sets are subsampled during the evaluation to ensure there are no overlapping modules (the original module sets are used as-is for the per-module-optimized experiments in Extended Data Fig. 5 and Supplementary Data File 5). Next, the integrated results are hierarchically clustered with a range of distance metrics (Euclidean and cosine), linkage methods (single, average and complete) and thresholds to optimize benchmark comparisons over these clustering parameters (this is done for all methods that are compared). The resulting benchmark-optimized cluster sets are compared to the benchmark module sets by computing adjusted mutual information: an information theoretic comparison measure that is adjusted to normalize against the expected score from random clustering. The highest adjusted mutual information score for each integration approach is reported, ensuring the optimal cluster set for each dataset across clustering parameters is used for the comparison and that our results are not dependent on clustering parameters. Finally, this procedure is repeated ten times to control for differences in scores due to the cluster sampling procedure. The sets of clustering parameter-optimized BIONIC clusters obtained from the Fig. 2 integration for each standard are in Supplementary Data File 4.

To perform the supervised gene function prediction evaluation, ten trials of five-fold cross validation were performed using SVM classifiers each using a radial basis function kernel[24]. The classifiers were trained on a set of gene features obtained from the given integration method with corresponding labels given by the IntAct, KEGG and GO Biological Process supervised benchmarks in a one-versus-all fashion (since each individual gene has multiple labels). Each classifier’s regularization and gamma parameters were tuned in the validation step. For each trial, the classifier results were evaluated on a randomized held-out set consisting of 10% of the gene features not seen during training or validation and the resulting classification accuracy was reported. We repeated this entire procedure for a random forest[68] and a gradient boosted trees[69] classifier and found BIONIC also outperforms the compared integration methods, indicating the SVM classifier is not biased toward improving BIONIC performance (Supplementary Fig. 7).

The granular functional evaluations in Figs. 2b and 3b were generated by computing the average precision (as mentioned in the precision-recall evaluation framework description) for the gene subsets annotated to the given functional categories.

To perform the module comparison analysis in Fig. 2c, we additionally applied the module detection analysis performed in Fig. 2a to the input networks. Here, the interaction profiles of the networks were treated as gene features and the clustering parameters were optimized to best match the IntAct complexes standard. We compared the resulting module sets from the input networks and BIONIC features to known protein complexes given by the IntAct standard. For each complex in the standard, we reported the best-matching predicted module in each dataset as determined by the overlap (Jaccard) score between the module and the known complex (Supplementary Data File 4). To generate the Venn diagram, we defined a complex to have been captured in the dataset if it had an overlap score of 0.5 or greater with a predicted module.

To perform the LSM2–7 module analysis in Fig. 2d, we analyzed the predicted module in each dataset that had the highest overlap score with the LSM2–7 complex. We created a network from the BIONIC features by computing the cosine similarity between all pairs of genes and setting all similarities below 0.5 to zero. The resulting nonzero values were then treated as weighted edges to form a network. We extracted a subnetwork from each of the protein–protein interaction, coexpression, genetic interaction and newly created BIONIC networks, consisting of the best scoring predicted module and the genes showing direct interactions with those in the predicted module. We laid out these networks using the spring-embedded layout algorithm in Cytoscape[70]. The edges in the protein–protein interaction network correspond to direct, physical interactions, and the edges in the coexpression and genetic interaction networks correspond to the pairwise Pearson correlation of the gene profiles, as described above.

To perform the semisupervised network integration experiment in Fig. 4, we first generated randomized train and test sets. Here, 20% of genes were randomly held out in each gene function benchmark (IntAct, KEGG and GO Biological Process) separately and retained for downstream evaluations. These benchmarks consist of functional labels for a set of yeast genes (protein complex membership in IntAct, pathway membership in KEGG and biological process annotation in GO Biological Process) and are the same benchmarks used in the gene function prediction evaluation (Figs. 2a and 3a). The remaining 80% of genes were used for training GeneMANIA and BIONIC. To generate test sets for the coannotation prediction benchmarks, we removed any coannotations where both genes were present in the training set. To generate test sets for the module detection benchmarks, we removed any modules consisting entirely of genes in the training set. We then integrated the three yeast networks from the Figs. 2 and 3 analysis (a protein–protein interaction[20], gene coexpression[21] and genetic interaction network[22]) using the supervised GeneMANIA, BIONIC without using any labeled data (unsupervised) and a semisupervised mode of BIONIC that uses the labeled data (semisupervised). Each integration result was then evaluated using the held-out test data. For the coannotation prediction and module detection evaluations, the integrated features from BIONIC (both unsupervised and semisupervised) and the integrated network from GeneMANIA were evaluated. Both GeneMANIA and the semisupervised BIONIC generate gene label predictions directly, without the need for an additional classifier such as in the Fig. 2a gene function prediction evaluation. However, the unsupervised BIONIC does not generate gene label predictions (since it is given no labeled information to begin with). To ensure a consistent comparison with GeneMANIA and the semisupervised BIONIC, we trained a classification head on top of the unsupervised BIONIC. The classification head architecture is identical to the semisupervised BIONIC classification head, however, in the unsupervised case we only allow gradients from the classification loss objective to backpropagate to the classification head, not the rest of the model. This ensures a comparable classification model can be trained on top of the unsupervised BIONIC model, without the labeled data affecting the model weights such as in the semisupervised case. GeneMANIA does not generate multi-label predictions, and so we used GeneMANIA to generate label predictions for each class individually and then performed Platt scaling to convert these binary class predictions to multi-label predictions[7,71]. The gene function prediction evaluations were then performed by comparing the gene label predictions from the integration methods, to the held-out test labels. This entire procedure, starting with the train-test set partitioning, to the final evaluations, was repeated a total of ten times to control for performance variability due to the partitioning procedure.

## Network scaling experiment

To perform the network scaling experiment, we uniformly sampled subsets of the yeast coexpression networks (Supplementary Data File 2). We performed ten integration trials for each network quantity, and these trials were paired (that is, each method integrated the same randomly sampled sets of networks). The average precision scores of the resulting integrations with respect to the KEGG pathways coannotation standard (Supplementary Data Files 3) were then reported. The Mashup method did not scale to the seven-network input size or beyond on a machine with 64 GB of RAM.

## Node scaling experiment

The node scaling experiment was performed by uniformly subsampling the nodes of four large human protein–protein interaction networks[34–37] (Supplementary Data File 2) for a range of node quantities and integrating these subsampled networks. Ten trials of subsampling were performed for each number of nodes (paired, as above) and the average precision scores with respect to the CORUM complexes coannotation standard (Supplementary Data File 3) were reported. The Mashup method did not scale to 4,000 nodes or beyond on a machine with 64 GB of RAM.

## Gene chemical sensitivity predictions

Chemical–genetic profiles against a diagnostic set of 310 nonessential yeast gene deletion mutants were obtained from a previous study[18]. The genes were chosen using the COMPRESS-GI algorithm, which selected a set of 157 genes capturing most of the functional information within genome-wide genetic interaction data[72], along with 153 genes that were manually selected to complement the set. Haploid deletion mutants for the gene set were constructed in a genetic background that conferred drug hypersensitivity (*pdr1Δpdr3Δsnq2Δ*) using synthetic genetic analysis technology, and each mutant strain was barcoded with a unique 20 bp DNA identifier adjacent to a common priming site. The mutant collection was grown and stored as a pooled library in yeast extract peptone-glycerol (15% v/v). A set of approximately 10,000 compounds from the RIKEN Natural Product Depository (NPDepo) were interrogated. Screens were done in 96-well format, where a single well contained the entire pool of 310 mutants at a density of $4.65 \times 10^5$ cells per ml and 196 µl of YPGal media (1% yeast extract, 2% peptone, 2% galactose). Each well was treated with 2 µl of compound (1 mg $\text{ml}^{-1}$ stock dissolved in dimethylsulfoxide (DMSO)). After 48 h of growth in 30 °C, genomic DNA was extracted from each compound-treated pool with an automated high-throughput nucleic acid purification robot (QIAcube HT, Qiagen). Mutant-specific barcodes and well-specific index tags were PCR-amplified using multiplex primers and a communal U2 primer. PCR products were pooled in 768-plex and gel-purified from 2% agarose gels using a Geneclean III kit. Amplicons were quantified using a Kapa quantitative PCR kit and were sequenced with an Illumina Hiseq 2500 machine at the RIKEN Center for Life Science Technologies. Sequencing data was processed using the BEAN-counter software[41], which generated chemical–genetic interaction $z$-scores normalized against DMSO-only (1% DMSO) treated samples. False discovery rates were estimated for biological process[22] predictions, for each compound, and those compounds with a false discovery rate of <25% were retained, resulting in a set of

1,522 compounds and 289 genes (high-confidence set)[18]. Next, interquartile range (IQR) scores were calculated from the chemical–genetic scores as follows:

$$\mathrm{IQRscore}_i = \frac{\mathrm{CGs}_i - \overline{\mathrm{CGs}}}{Q3_{\mathrm{CGs}} - Q1_{\mathrm{CGs}}} \tag{12}$$

Here, $\mathrm{CGs}_i$ is the chemical–geneticscore for the $i$th replicate, $\overline{\mathrm{CGs}}$ is the median of all chemical–genetic scores, $Q3_{\mathrm{CGs}}$ is the 75th percentile of chemical–genetic scores and $Q1_{\mathrm{CGs}}$ is the 25th percentile of chemical–genetic scores. Tukey's test[73] was used to determine outliers based on the IQR of the distribution of IQR scores in the screen. Genes with at least one replicate that had a negative (sensitive) chemical–genetic score more than three times the IQR of the compound profile (that is, 'outlier' genes) were retained.

To predict chemical–genetic interactions using BIONIC, we first selected a set of 50 compounds to generate predictions on and experimentally validate. For each diagnostic pool compound, we filtered out any genes not present in the integrated BIONIC features (the same features used for the Figs. 2 and 3 analyses, referred to as PEG features). Any compounds with fewer than two outlier sensitive genes were then removed. For each of the remaining compounds, we randomly split the sensitive genes into train and test sets. Next, for a given compound, we computed BIONIC predictions for the test set genes. We did this by averaging the corresponding BIONIC PEG features for each gene in the training set under a cosine distance metric to get a representative feature vector in gene feature space for the given compound. The BIONIC predictions for the compound were then obtained by identifying the top 20 nearest genes to this feature vector (excluding genes in the training set).

To obtain a score for the BIONIC predictions, an ordered Fisher's exact test was performed between the test set genes and the BIONIC predictions as follows:

$$p = \min(\{f(n,k) : n, k = (1, k_1), \ldots, (20, k_{20})\}) \tag{13}$$

where

$$f(n,k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}} \tag{14}$$

$p$ corresponds to the minimum $P$ value obtained for progressively larger subsets of BIONIC's 20 predictions, starting from the top prediction to the full set of 20 predictions. $n$ is the number of total predictions made by BIONIC and $k$ is the number of those predictions that are correct. $k_i$ corresponds to the number of correct predictions for the first $i$ genes in the

BIONIC predictions. $f$ is the probability mass function of the hypergeometric distribution. Here, $K$ corresponds to the number of genes found to be sensitive to the given compound. $N$ is the total number of yeast essential genes in the analysis, specifically, essential genes for which TS mutants could be made and are also present in the BIONIC features (847 total genes). We chose the ordered Fisher's exact test over the commonly used unordered version because BIONIC produces a ranked list of predictions. Taking into account the ordering of BIONIC predictions is a fairer assessment, since, for example, a compound may only have a small number of sensitive genes (fewer than 20). In this case, BIONIC's top predictions may include these essential genes, however, an unordered Fisher's exact test would not consider this ranking and treat the full set of 20 predictions as equivalent, whereas the ordered test would consider the ranking.

The above process was repeated five times for new randomly sampled train and test gene splits, or up to the maximum number of train-test splits possible for compounds with fewer than five sensitive genes. Final $P$ values were obtained for each compound by averaging over the $P$ values from each trial. Compounds were ranked by most significant $P$ values and the top 50 compounds were selected for further screening. Sensitive essential gene predictions for a given compound were then generated by using the full set of sensitive diagnostic pool genes as the training set, computing a representative compound feature vector by averaging the training set BIONIC gene features, and identifying the top 20 nearest essential genes to this compound feature vector.

The BIONIC gene chemical sensitivity predictions were benchmarked against experimental data obtained from chemical–genetic screens using a collection of TS mutants for essential genes. We previously constructed a drug-hypersensitive, barcoded set of TS mutants for 1,181 TS alleles spanning 837 essential genes[40]. Similar to the diagnostic set of nonessential genes, this collection also contained the pdr1Δpdr3Δsnq2Δ triple deletion and a 20 bp barcode was inserted next to a common priming site upstream of a natMX cassette integrated at the pdr3Δ locus. We conducted chemical–genetic screens against the 50 compounds initially selected for BIONIC analysis using the same method that was used to generate the diagnostic set profiles, except that the TS mutant pools were incubated at 25 °C instead of 30 °C for 48 h. We calculated chemical–genetic interaction $z$-scores and removed nonspecific technical effects using BEAN-counter software[41]. IQR scores were calculated as described above. Negative (sensitive) interactions that were more than four times the IQR (classified as 'far outliers') were used to validate the gene chemical sensitivities predicted by BIONIC.

The significance of BIONIC sensitive essential gene predictions for each compound was determined by using an ordered Fisher's exact test, as detailed above. The Benjamini–Hochberg procedure[74] was applied to the resulting $P$ values at a false discovery rate of 5%.

To generate biological process[22] predictions as reported in Fig. 6a,b, a Fisher's exact test was performed between the full set of 20 BIONIC gene predictions and biological process gene annotations. We used the same annotations as in Figs. 2b and 3b (ref. [22]). If the BIONIC sensitive gene predictions were enriched for one or more bioprocesses, and these bioprocesses overlapped with the annotated bioprocess of the true sensitive genes, we considered this a correct bioprocess prediction. To generate the random benchmark in Fig. 6b, the gene labels of the BIONIC integrated features were randomly permuted and new essential sensitive gene predictions for the 50 selected compounds were generated in the same manner as the original BIONIC predictions (detailed above). This process was repeated for 1,000 random gene label permutations to generate the benchmark distributions. The circle plot in Fig. 6d was produced by first hierarchically clustering the integrated BIONIC gene features, subsetted to essential genes annotated to the glycosylation, protein folding/targeting, cell wall biosynthesis bioprocess. Two clustering thresholds were chosen to generate clusters, broadly indicating the hierarchical organization of the BIONIC gene features. The first, most granular clustering threshold was adaptively chosen to generate clusters best-matching known protein complexes, as defined by the IntAct complexes standard[29]. For each protein complex in the standard, the clustering threshold was optimized to produce the cluster best matching this protein complex. For clusters not matching known complexes, the largest complex optimized threshold was used. The second, higher clustering threshold was set to a cophenetic distance of 0.9.

The BIONIC essential gene sensitivity predictions can be found in Supplementary Data File 7.

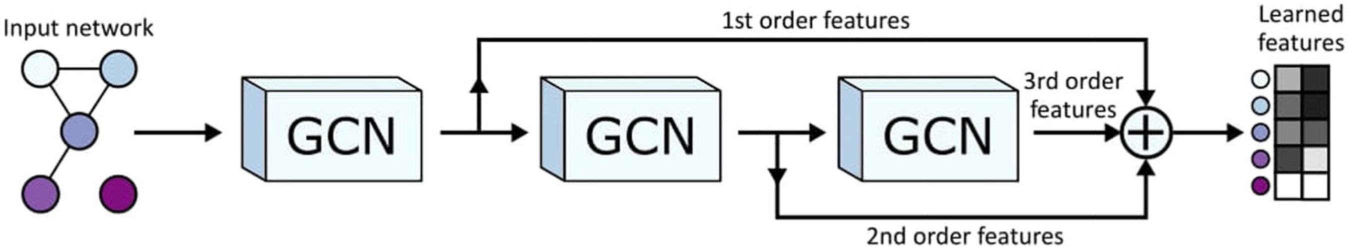
## Quantification of β-1,6-glucan levels

Wild-type (*his3Δ* in the BY4741 background) and the *kre6Δ* strain (YOC5627) of *S. cerevisiae* were grown in yeast extract peptone at 25 °C with shaking at 200 r.p.m. to 1 × $10^7$ cells per ml. Wild-type cells were treated with 5 μg $ml^{-1}$ jervine (J0009; Tokyo Chemical Industry, Tokyo, Japan) for 4 h. We used jervine since it is chemically similar to NP329 and is more commercially available than NP329. The samples were centrifuged at 15,000*g* for 3 min, and the supernatant was discarded. The pellet was washed and suspended in PBS, adjusted to 1 × $10^6$ cells per ml and autoclaved for 20 min. After centrifugation at 15,000*g* for 1 min, the supernatant was stored on ice (sample A) and the pellet was further extracted. The β-1,6-glucan was extracted from the pellet using a slightly modified version of the protocol of Kitamura et al.[75]. First, 500 ml of 10% TCA was added to the culture, which was incubated on ice for 10 min. After centrifugation at 15,000*g* for 3 min, the samples were washed twice with deionized water. The pellet was suspended in 500 μl of 1 N NaOH and incubated at 75 °C for 1 h. The solution was mixed with 500 μl of 1 M HCl and Tris buffer (10 mM Tris-HCl, pH 7). After centrifugation at 15,000*g* for 1 min, the supernatant was stored on ice (sample B). The total amounts of β-1,6-glucan in samples A and B were measured according to the method of Yamanaka et al.[76].

## Reporting summary

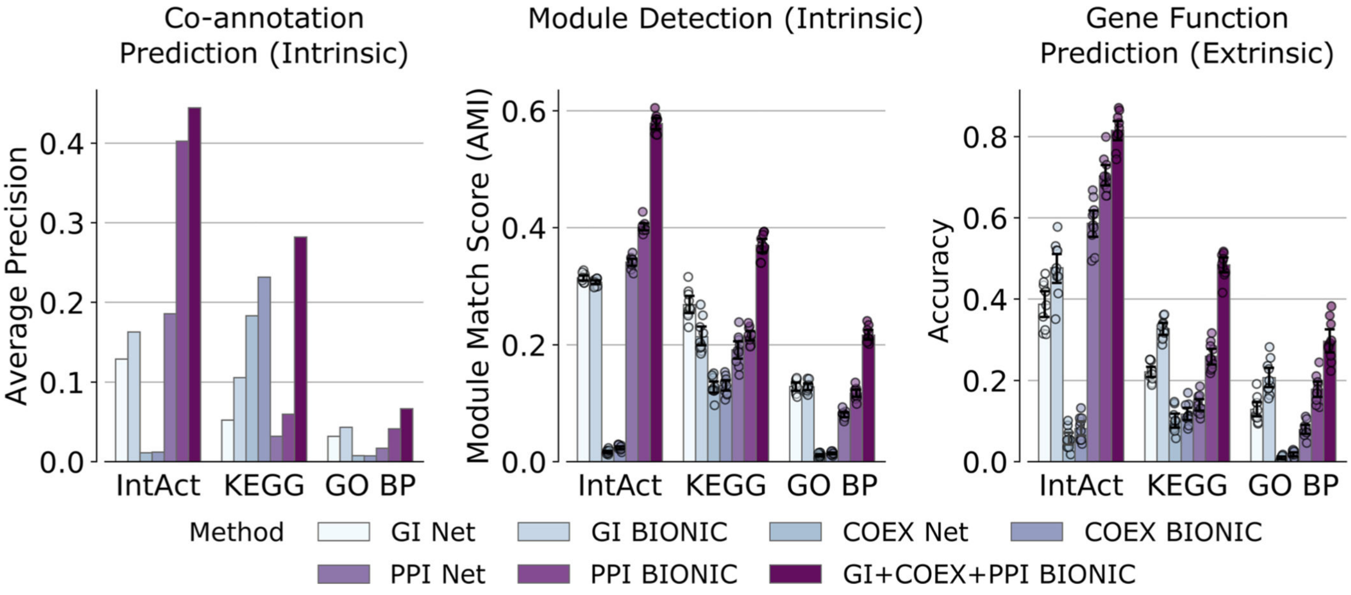
Further information on research design is available in the Nature Research Reporting Summary linked to this article.

## Extended Data



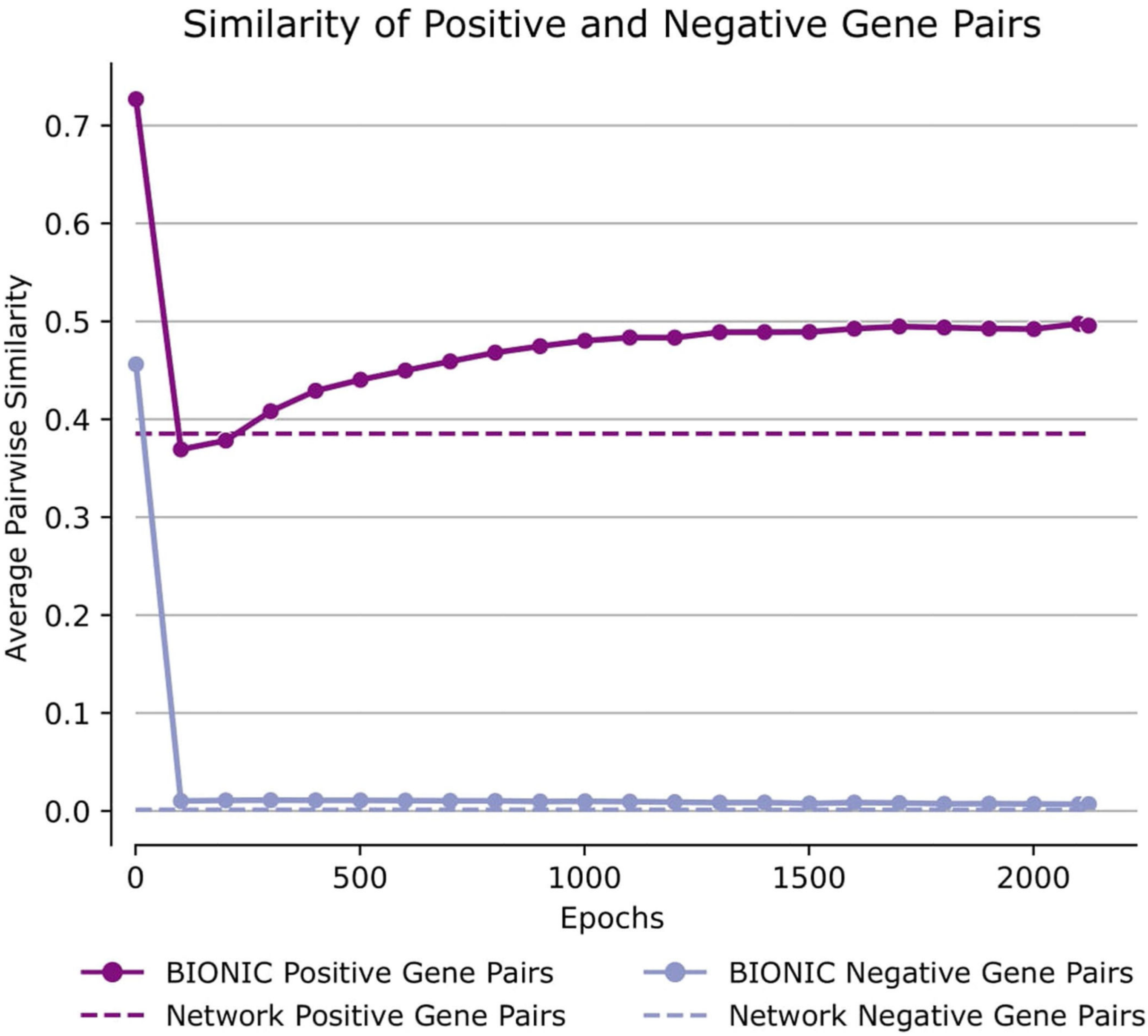
**Extended Data Fig. 1 |. Detailed view of individual BIONIC network encoder.**

A more detailed view of an individual network encoder, including residual connections. A network specific graph convolutional network is used to encode the input network for increasing neighborhood sizes. The first GCN in the sequence learns features for a given node based on the node's immediate neighborhood (1st order features). The next GCN learns features based on the node's second order neighborhood (2nd order features), and so on. The node feature matrices learned by each GCN pass are summed together to create the final learned, network-specific features. Summing the outputs of the various GCNs in this way creates residual connections, allowing features from multiple neighborhood sizes to generate the final learned features, rather than just the final neighborhood size. This figure shows three GCN layers, but BIONIC uses the same pattern of connections for any number of GCN layers. Note that the GCN layers for a given encoder share their weights, so in effect, there is a single GCN layer for each encoder.



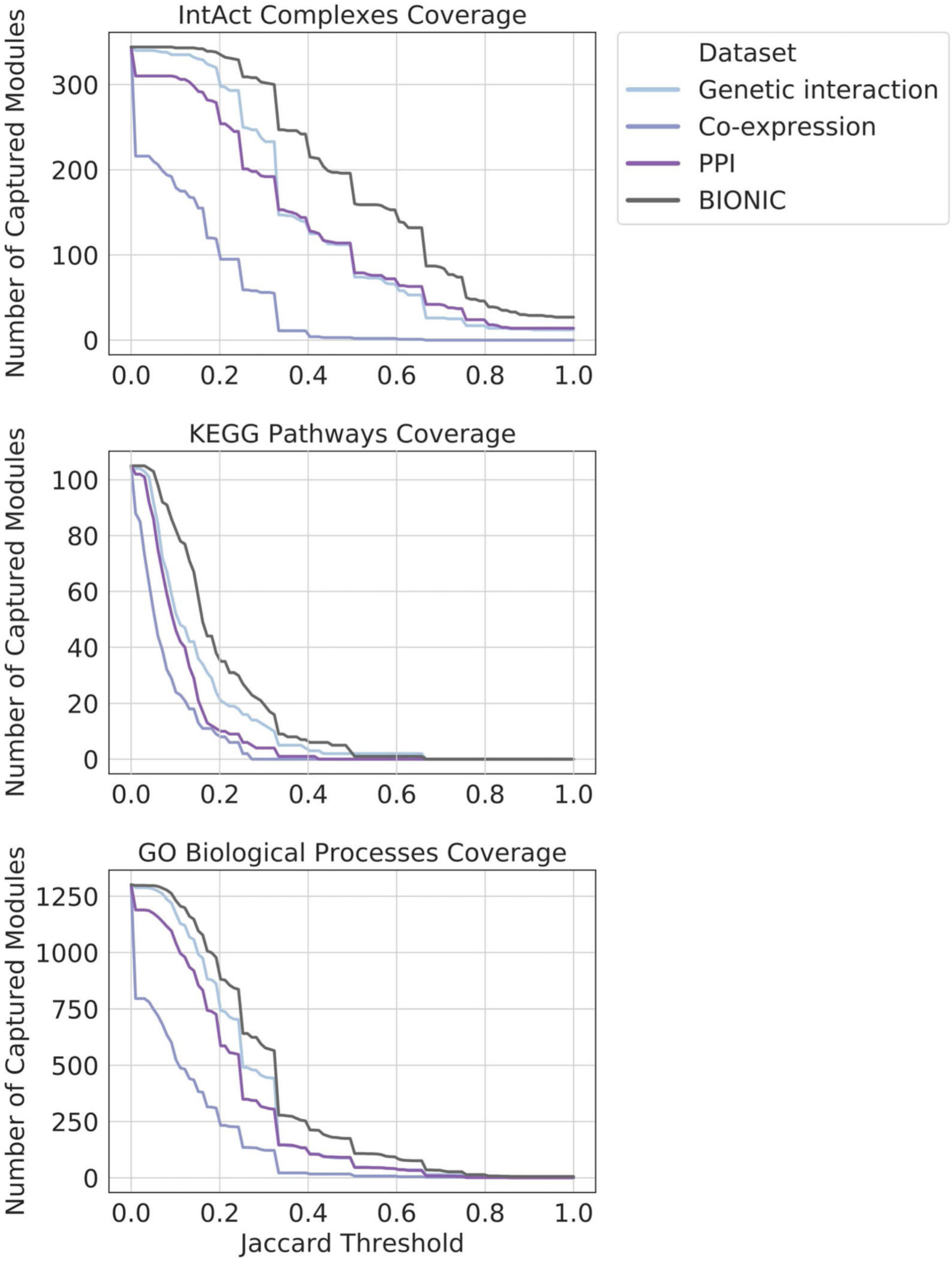
**Extended Data Fig. 2 |. Comparison of individual network features produced by BIONIC.**

A comparison of individual networks (denoted 'Net'), their corresponding features encoded using the unsupervised BIONIC (denoted 'BIONIC'), as well as the BIONIC integration of these networks (denoted 'GI+COEX+PPI BIONIC'). BP = Biological Processes, GI = Genetic Interaction, COEX = Co-expression, PPI = Protein-protein Interaction. These are the same networks and evaluations used in Fig. 2. Data are presented as mean values. Error bars indicate the 95% confidence interval for n = 10 independent samples.



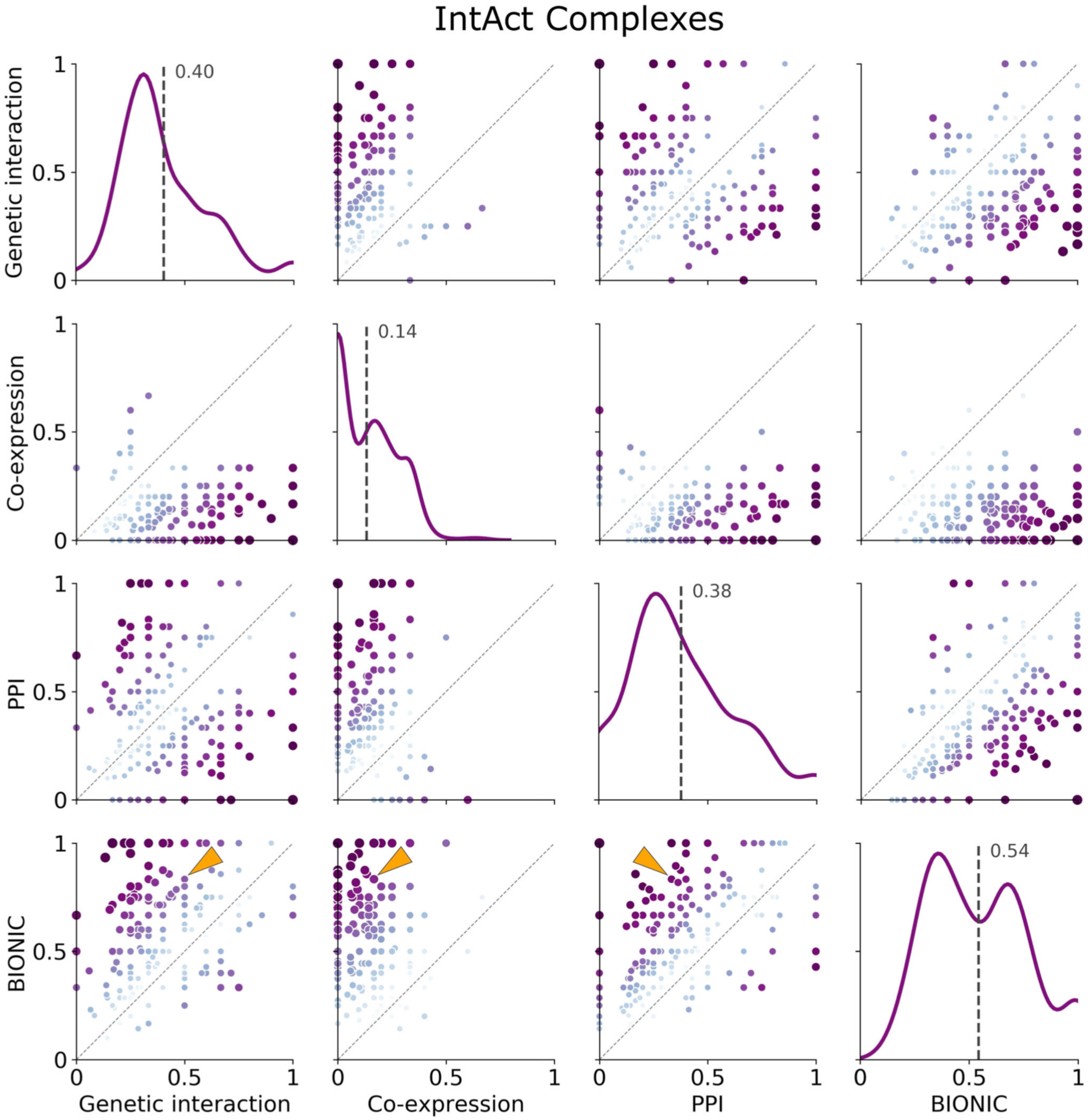
**Extended Data Fig. 3 |. Dynamics of BIONIC feature space through training.**

Comparison of pairwise gene similarities (cosine similarity in the case of BIONIC, direct binary adjacency in the case of the network), as defined by IntAct Complexes for known co-complex relationships (positive pairs) and no co-complex relationships (negative pairs), between a yeast PPI network (as used in the Fig. 2 analyses) and the unsupervised BIONIC features produced from this network. The BIONIC similarities are shown throughout the training process (epochs), whereas the input network is constant so its pairwise similarities do not change. 'Network' denotes the input PPI network, 'BIONIC' denotes the features learned from this network using BIONIC.



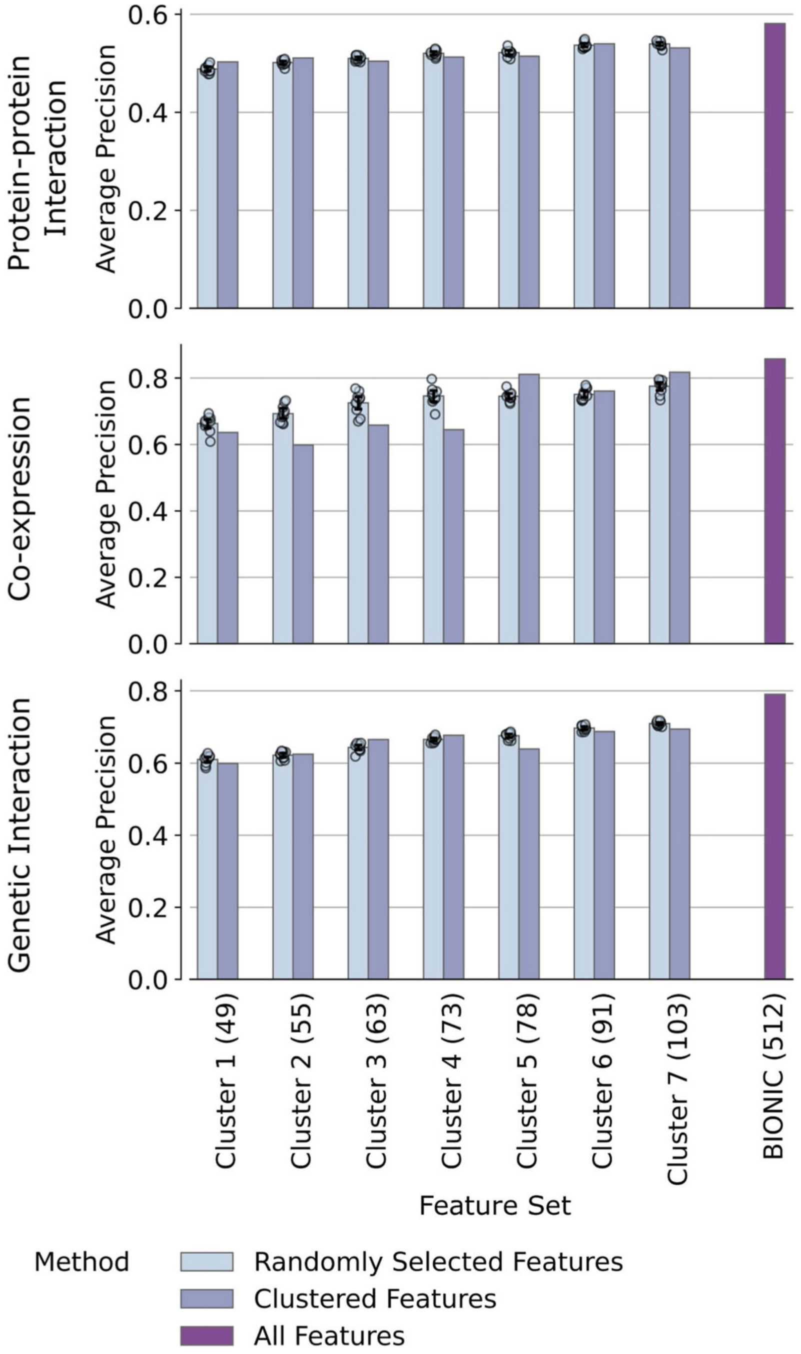
**Extended Data Fig. 4 |. Coverage of BIONIC and input network captured modules.**

Coverage of functional gene modules by individual networks and the unsupervised BIONIC integration of these networks (denoted BIONIC), as determined by a parameter optimized module detection analysis where the clustering parameters were optimized for each module individually. The number of captured modules is reported for a range of overlap scores (Jaccard threshold). Higher threshold indicates greater correspondence between the clusters obtained from the dataset and their respective modules given by the standard. PPI = protein-protein interaction. These are the same networks and BIONIC features as Fig. 2.



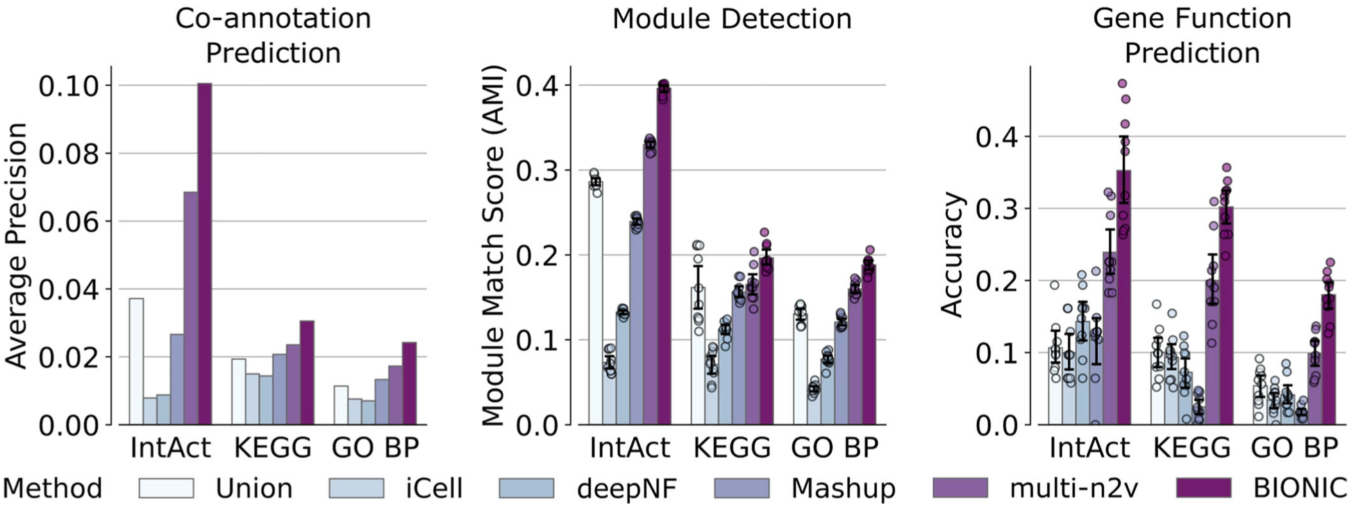
**Extended Data Fig. 5 |. Captured modules comparison for BIONIC and input networks for optimal clustering parameters.**

Known protein complexes (as defined by the IntAct standard) captured by individual networks and the unsupervised BIONIC integration of these networks (denoted BIONIC). Hierarchical clustering was performed on the datasets and resulting clusters were compared to known IntAct complexes and scored for set overlap using the Jaccard score (ranging from 0 to 1). The clustering algorithm parameters were optimized for each module individually, unlike the analysis in Fig. 2 where the clustering parameters were optimized for the standard as a whole. Each point is a protein complex, as in Fig. 2c. The dashed line indicates instances where the given data sets achieve the same score for a given module. Histograms indicate the distribution of overlap (Jaccard) scores for the given dataset, and the labelled dashed line indicates the mean of this distribution. The individual modules shown here as well as for the KEGG Pathways and IntAct Complexes module standards can be found in Supplementary Data File 4. The LSM2–7 complex is indicated by the arrows. PPI = protein-protein interaction. This analysis uses the same networks and BiONIC features as Fig. 2.



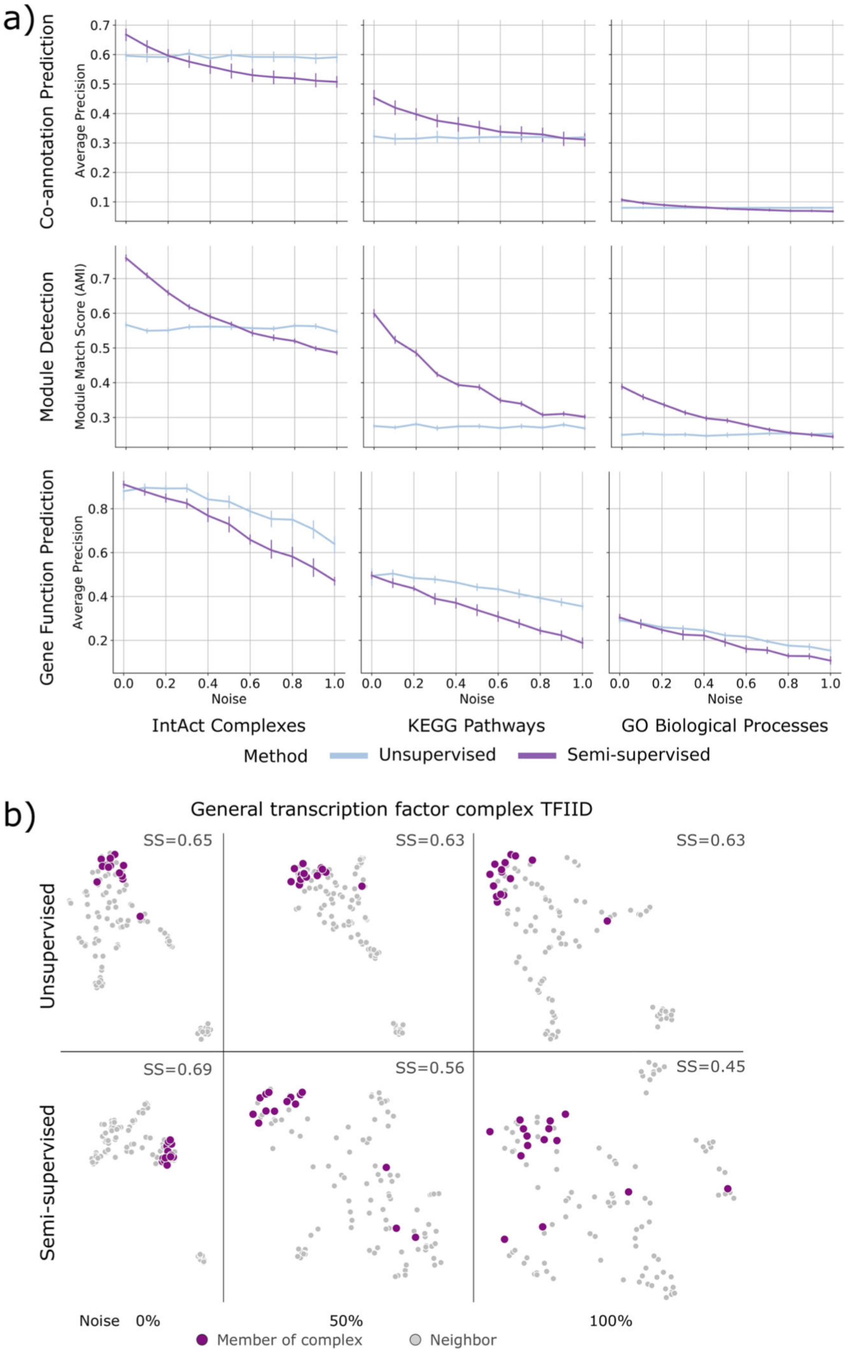
**Extended Data Fig. 6 |. Interpretability of BIONIC feature space.**

Co-annotation evaluations of the unsupervised BIONIC features subset to different clusters of feature dimensions (denoted ‘Cluster’). The number of feature dimensions for each cluster is given in parenthesis. The performance of the original BIONIC features (denoted BIONIC (512)) is also displayed. Data are presented as mean values. Bars indicate 95% confidence interval for n = 10 independent samples.



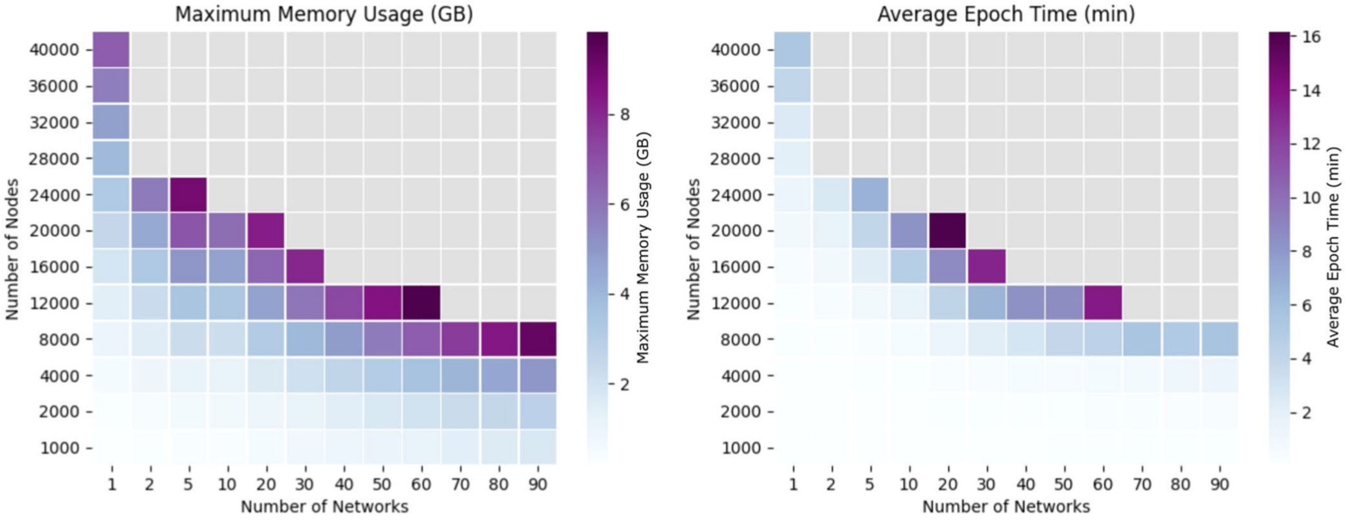
**Extended Data Fig. 7 |. Integration method performance for yeast-two-hybrid network inputs.**

Performance comparison of 5 yeast-two-hybrid network integrations across functional standards, evaluation types and unsupervised integration methods. Data are presented as mean values. Bars indicate 95% confidence interval for n = 10 independent samples. BP = Biological Process, multi-n2v = multi-node2vec.



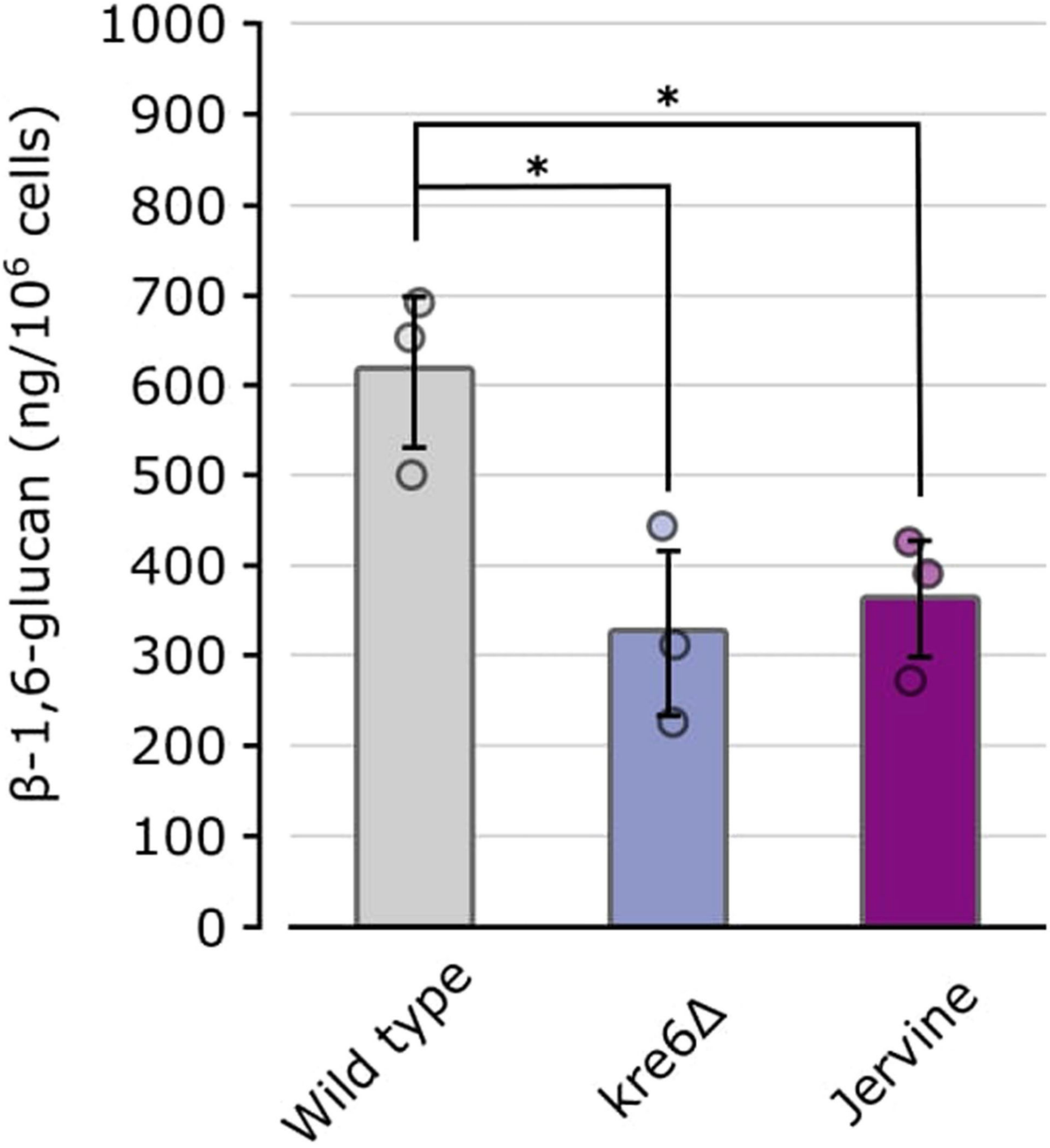
**Extended Data Fig. 8 |. Effects of label poisoning on BIONIC semi-supervised and unsupervised performance.**

Semi-supervised BIONIC comparisons. **a)** A label poisoning experiment, where progressively more permutation noise is added to the label sets the semi-supervised BIONIC is trained on. ‘Noise’ indicates the proportion of permutation noise applied (multiply by 100 for percentages). Data are presented as mean values. Bars indicate 95% confidence interval for n = 10 independent samples. **b)** UMAP plots comparing the embedding space of the TFIID complex and the 100 nearest neighbors of this complex for unsupervised and semi-supervised BIONIC over a range of label noise values. SS = average silhouette score of TFIID members.



**Extended Data Fig. 9 |. Computational scalability of BIONIC.**

Graphics processing unit (GPU) memory usage in gigabytes (left) and average wall clock epoch time in minutes (right) for a range of network sizes and number of networks. GB = gigabyte, min = minutes. Gray squares indicate a scenario where BIONIC exceeded the maximum memory of the GPU and failed to complete. The experiments were run on a Titan Xp GPU with a 2.4 GHz Intel Xeon CPU and 32 GB of system memory.



**Extended Data Fig. 10 |. β-1,6-glucan levels in yeast strains.**

The amount of glucan per cell was calculated using pustulan as a standard. Data are presented as mean values. Error bars indicate standard deviation for n = 3 biologically independent samples. kre6Δ compared to wild type p-value = 0.01473, Jervine compares to wild type p-value = 0.01520. * Significant difference (p-value < 0.05 after Bonferroni correction, Welch's one-sided t-test).

## Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

## Acknowledgements

We thank B. Andrews, M. Costanzo and C. Myers for their insightful comments. We also thank M. Fey for adding important features to the PyTorch Geometric library for us. This work was supported by NRNB (US National Institutes of Health, National Center for Research Resources grant number P41 GM103504). Funding for continued development and maintenance of Cytoscape is provided by the US National Human Genome Research Institute under award number HG009979. This work was also supported by the Canadian Institutes of Health Research Foundation grant number FDN-143264, US National Institutes of Health grant number R01HG005853 and joint funding by Genome Canada (OGI-163) and the Ministry of Economic Development, Job Creation and Trade, under the program Bioinformatics and Computational Biology. This work was supported by the National Research Council of Canada through the AI for Design program. This work was supported by CIFAR AI Chair programs. This work was also supported by JSPS KAKENHI grant numbers JP15H04483 (C.B. and Y.O.), JP17H06411 (C.B. and Y.Y.), JP18K14351 (K.I.-N.), JP19H03205 (Y.O.), JP20K07487 (D.Y.) and a RIKEN Foreign Postdoctoral Fellowship (S.C.L.). This research was enabled in part by support provided by SciNet and the Digital Research Alliance of Canada.

## Data availability

All data, standards, BIONIC yeast features and chemical–genetic interaction data are available in the following Figshare repository: https://figshare.com/projects/BIONIC_Biological_Network_Integration_using_Convolutions/122585. There are no restrictions on the data. Source data are provided with this paper.

## References

1. Fraser AG & Marcotte EM A probabilistic view of gene function. Nat. Genet 36, 559 (2004). [PubMed: 15167932]
2. Malod-Dognin N. et al. Towards a data-integrated cell. Nat. Commun 10, 805 (2019). [PubMed: 30778056]
3. Wang P, Gao L, Hu Y. & Li F. Feature related multi-view nonnegative matrix factorization for identifying conserved functional modules in multiple biological networks. BMC Bioinf. 19, 394 (2018).
4. Argelaguet R. et al. Multi-omics factor analysis—a framework for unsupervised integration of multi-omics data sets. Mol. Syst. Biol 14, e8124 (2018). [PubMed: 29925568]
5. Mostafavi S, Ray D, Warde-Farley D, Grouios C. & Morris Q. GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. Genome Biol. 9, S4 (2008).
6. Wang B. et al. Similarity network fusion for aggregating data types on a genomic scale. Nat. Methods 11, 333 (2014). [PubMed: 24464287]
7. Cho H. et al. Compact integration of multi-network topology for functional analysis of genes. Cell Syst. 3, 540–548.e5 (2016). [PubMed: 27889536]
8. Huttenhower C, Hibbs M, Myers C. & Troyanskaya OG A scalable method for integration and functional analysis of multiple microarray datasets. Bioinformatics 22, 2890–2897 (2006). [PubMed: 17005538]
9. von Mering C. et al. STRING: a database of predicted functional associations between proteins. Nucleic Acids Res. 31, 258–261 (2003). [PubMed: 12519996]
10. Alexeyenko A. & Sonnhammer ELL Global networks of functional coupling in eukaryotes from comprehensive data integration. Genome Res. 19, 1107–1116 (2009). [PubMed: 19246318]
11. Gligorijević V, Barot M. & Bonneau R. deepNF: deep network fusion for protein function prediction. Bioinformatics 34, 3873–3881 (2018). [PubMed: 29868758]

12. Perozzi B, Al-Rfou R. & Skiena S. DeepWalk: online learning of social representations. In Proc. 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (eds Macskassy S. & Perlich C) 701–710 (Association for Computing Machinery, 2014).
13. Grover A. & Leskovec J. node2vec: scalable feature learning for networks. KDD 2016, 855–864 (2016). [PubMed: 27853626]
14. Kipf TN & Welling M. Semi-supervised classification with graph convolutional networks. In Proc. International Conference on Learning Representations (2017).
15. Defferrard M, Bresson X. & Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proc. Advances in Neural Information Processing Systems (NIPS 2016) Vol. 29, 3844–3852 (Curran Associates, Inc., 2016).
16. Hamilton W, Ying Z. & Leskovec J. Inductive representation learning on large graphs. In Proc. Advances in Neural Information Processing Systems (NIPS 2017) Vol. 30, 1024–1034 (Curran Associates, Inc., 2017).
17. Veličković P. et al. Graph attention networks. In Proc. International Conference on Learning Representations (2018).
18. Piotrowski JS et al. Functional annotation of chemical libraries across diverse biological processes. Nat. Chem. Biol 13, 982–993 (2017). [PubMed: 28759014]
19. He K, Zhang X, Ren S. & Sun J. Deep residual learning for image recognition. In Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 770–778 (IEEE, 2016).
20. Krogan NJ et al. Global landscape of protein complexes in the yeast *Saccharomyces cerevisiae*. Nature 440, 637–643 (2006). [PubMed: 16554755]
21. Hu Z, Killion PJ & Iyer VR Genetic reconstruction of a functional transcriptional regulatory network. Nat. Genet 39, 683–687 (2007). [PubMed: 17417638]
22. Costanzo M. et al. A global genetic interaction network maps a wiring diagram of cellular function. Science 353, aaf1420 (2016).
23. Myers CL et al. Discovery of biological networks from diverse functional genomic data. Genome Biol. 6, R114 (2005). [PubMed: 16420673]
24. Cortes C. & Vapnik V. Support-vector networks. Mach. Learn 20, 273–297 (1995).
25. Vo TV et al. A proteome-wide fission yeast interactome reveals network evolution principles from yeasts to human. Cell 164, 310–323 (2016). [PubMed: 26771498]
26. Martín R. et al. A PP2A-B55-mediated crosstalk between TORC1 and TORC2 regulates the differentiation response in fission yeast. Curr. Biol 27, 175–188 (2017). [PubMed: 28041796]
27. Ryan CJ et al. Hierarchical modularity and the evolution of genetic interactomes across species. Mol. Cell 46, 691–704 (2012). [PubMed: 22681890]
28. Ashburner M. et al. Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. Nat. Genet 25, 25–29 (2000). [PubMed: 10802651]
29. Orchard S. et al. The MIntAct project–IntAct as a common curation platform for 11 molecular interaction databases. Nucleic Acids Res. 42, D358–363 (2014). [PubMed: 24234451]
30. Kanehisa M. & Goto S. KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res. 28, 27–30 (2000). [PubMed: 10592173]
31. Fernandez CF, Pannone BK, Chen X, Fuchs G. & Wolin SL An Lsm2-Lsm7 complex in *Saccharomyces cerevisiae* associates with the small nucleolar RNA snR5. Mol. Biol. Cell 15, 2842–2852 (2004). [PubMed: 15075370]
32. Chowdhury A, Mukhopadhyay J. & Tharun S. The decapping activator Lsm1p-7p-Pat1p complex has the intrinsic ability to distinguish between oligoadenylated and polyadenylated RNAs. RNA 13, 998–1016 (2007). [PubMed: 17513695]
33. Wilson JD, Baybay M, Sankar R, Stillman P. & Popa AM Analysis of population functional connectivity data via multilayer network embeddings. Netw. Sci 9, 99–122 (2021).
34. Huttlin EL et al. Architecture of the human interactome defines protein communities and disease networks. Nature 545, 505 (2017). [PubMed: 28514442]
35. Huttlin EL et al. The bioplex network: a systematic exploration of the human interactome. Cell 162, 425–440 (2015). [PubMed: 26186194]

36. Hein MY et al. A human interactome in three quantitative dimensions organized by stoichiometries and abundances. Cell 163, 712–723 (2015). [PubMed: 26496610]

37. Rolland T. et al. A proteome-scale map of the human interactome network. Cell 159, 1212–1226 (2014). [PubMed: 25416956]

38. Roemer T. & Boone C. Systems-level antimicrobial drug and drug synergy discovery. Nat. Chem. Biol 9, 222–231 (2013). [PubMed: 23508188]

39. Ayscough KR et al. High rates of actin filament turnover in budding yeast and roles for actin in establishment and maintenance of cell polarity revealed using the actin inhibitor latrunculin-A. J. Cell Biol 137, 399–416 (1997). [PubMed: 9128251]

40. Persaud R. et al. Clionamines stimulate autophagy, inhibit *Mycobacterium tuberculosis* survival in macrophages, and target Pik1. Cell Chem. Biol 29, 870–882 (2021). [PubMed: 34520745]

41. Simpkins SW et al. Using BEAN-counter to quantify genetic interactions from multiplexed barcode sequencing experiments. Nat. Protoc 14, 415–440 (2019). [PubMed: 30635653]

42. Kato N, Takahashi S, Nogawa T, Saito T. & Osada H. Construction of a microbial natural product library for chemical biology studies. Curr. Opin. Chem. Biol 16, 101–108 (2012). [PubMed: 22406171]

43. Protchenko O, Rodriguez-Suarez R, Androphy R, Bussey H. & Philpott CC A screen for genes of heme uptake identifies the FLC family required for import of FAD into the endoplasmic reticulum. J. Biol. Chem 281, 21445–21457 (2006). [PubMed: 16717099]

44. Kitagaki H, Wu H, Shimoi H. & Ito K. Two homologous genes, DCW1 (YKL046c) and DFG5, are essential for cell growth and encode glycosylphosphatidylinositol (GPI)-anchored membrane proteins required for cell wall biogenesis in Saccharomyces cerevisiae. Mol. Microbiol 46, 1011–1022 (2002). [PubMed: 12421307]

45. Ram AF et al. Loss of the plasma membrane-bound protein Gas1p in *Saccharomyces cerevisiae* results in the release of beta1,3-glucan into the medium and induces a compensation mechanism to ensure cell wall integrity. J. Bacteriol 180, 1418–1424 (1998). [PubMed: 9515908]

46. Tomishige N. et al. Mutations that are synthetically lethal with a gas1Delta allele cause defects in the cell wall of *Saccharomyces cerevisiae*. Mol. Genet. Genomics 269, 562–573 (2003). [PubMed: 12827498]

47. Ragni E, Fontaine T, Gissi C, Latgè JP & Popolo L. The Gas family of proteins of *Saccharomyces cerevisiae*: characterization and evolutionary analysis. Yeast 24, 297–308 (2007). [PubMed: 17397106]

48. Neiman AM, Mhaiskar V, Manus V, Galibert F. & Dean N. *Saccharomyces cerevisiae* HOC1, a suppressor of pkc1, encodes a putative glycosyltransferase. Genetics 145, 637–645 (1997). [PubMed: 9055074]

49. Simpkins SW et al. Predicting bioprocess targets of chemical compounds through integration of chemical-genetic and genetic interactions. PLoS Comput. Biol 14, e1006532 (2018).

50. Pasikowska M, Palamarczyk G. & Lehle L. The essential endoplasmic reticulum chaperone Rot1 is required for protein N- and O-glycosylation in yeast. Glycobiology 22, 939–947 (2012). [PubMed: 22492205]

51. Machi K. et al. Rot1p of *Saccharomyces cerevisiae* is a putative membrane protein required for normal levels of the cell wall 1,6-beta-glucan. Microbiology 150, 3163–3173 (2004). [PubMed: 15470097]

52. Levinson JN, Shahinian S, Sdicu A-M, Tessier DC & Bussey H. Functional, comparative and cell biological analysis of *Saccharomyces cerevisiae* Kre5p. Yeast 19, 1243–1259 (2002). [PubMed: 12271460]

53. Azuma M, Levinson JN, Pagé N. & Bussey H. *Saccharomyces cerevisiae* Big1p, a putative endoplasmic reticulum membrane protein required for normal levels of cell wall beta-1,6-glucan. Yeast 19, 783–793 (2002). [PubMed: 12112232]

54. Roemer T, Delaney S. & Bussey H. SKN1 and KRE6 define a pair of functional homologs encoding putative membrane proteins involved in beta-glucan synthesis. Mol. Cell. Biol 13, 4039–4048 (1993). [PubMed: 8321211]

55. Kubo K. et al. Jerveratrum-type steroidal alkaloids inhibit β-1,6-glucan biosynthesis in fungal cell walls. Microbiol. Spectr 10, e0087321 (2022).

56. Usaj M. et al. TheCellMap.org: a web-accessible database for visualizing and mining the global yeast genetic interaction network. G3 7, 1539–1549 (2017). [PubMed: 28325812]

57. Elnaggar A. et al. ProtTrans: towards cracking the language of lifes code through self-supervised deep learning and high performance computing. IEEE Trans. Pattern Anal. Mach. Intell 10.1109/TPAMI.2021.3095381 (2021).

58. Almagro Armenteros JJ, Sønderby CK, Sønderby SK, Nielsen H. & Winther O. DeepLoc: prediction of protein subcellular localization using deep learning. Bioinformatics 33, 3387–3395 (2017). [PubMed: 29036616]

59. Mattiazzi Usaj M. et al. Systematic genetics and single-cell imaging reveal widespread morphological pleiotropy and cell-tocell variability. Mol. Syst. Biol 16, 30 (2020).

60. Paszke A. et al. Automatic differentiation in PyTorch. in NIPS Autodiff Workshop (2017).

61. Fey M. & Lenssen JE Fast Graph Representation Learning with PyTorch Geometric. in ICLR 2019 Workshop on Representation Learning on Graphs and Manifolds (2019).

62. 1. Kingma DP & Ba J. Adam: A Method for Stochastic Optimization. in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (eds. Bengio Y. & LeCun Y) (2015).

63. Stark C. et al. BioGRID: a general repository for interaction datasets. Nucleic Acids Res. 34, D535–D539 (2006). [PubMed: 16381927]

64. Hibbs MA et al. Exploring the functional landscape of gene expression: directed search of large microarray compendia. Bioinformatics 23, 2692–2699 (2007). [PubMed: 17724061]

65. Myers CL, Barrett DR, Hibbs MA, Huttenhower C. & Troyanskaya OG Finding function: evaluation methods for functional genomic data. BMC Genomics 7, 187 (2006). [PubMed: 16869964]

66. Aggarwal CC, Hinneburg A, Keim DA (2001). On the Surprising Behavior of Distance Metrics in High Dimensional Space. In: Van den Bussche J, Vianu V. (eds) Database Theory — ICDT 2001. ICDT 2001. Lecture Notes in Computer Science, vol 1973. Springer, Berlin, Heidelberg. 10.1007/3-540-44503-X_27

67. Davis J. & Goadrich M. The relationship between Precision-Recall and ROC curves. In Proc. 23rd International Conference on Machine Learning: June 25–29, 2006; Pittsburgh, Pennsylvania (eds Cohen WW & Moore A) 233–240 (ACM Press, 2006).

68. Breiman L. Random forests. Mach. Learn 45, 5–32 (2001).

69. Friedman JH Greedy function approximation: a gradient boosting machine. Ann. Stat 29, 1189–1232 (2001).

70. Shannon P. et al. Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome Res. 13, 2498–2504 (2003). [PubMed: 14597658]

71. Platt JC Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. in Advances in Large Margin Classifiers (eds Smola AJ et al.) 61–74 (MIT Press, 1999).

72. Deshpande R. et al. Efficient strategies for screening large-scale genetic interaction networks. Preprint at bioRxiv 10.1101/159632 (2017).

73. Beyer H. Tukey & John W. Exploratory Data Analysis. Addison-Wesley Publishing Company Reading, Mass.—Menlo Park, cal., London, Amsterdam, Don Mills, Ontario, Sydney 1977, XVI, 688S. Biom. J 23, 413–414 (1981).

74. Benjamini Y. & Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. J. R. Stat. Soc. Ser. B Stat. Methodol 57, 289–300 (1995).

75. Kitamura A, Someya K, Hata M, Nakajima R. & Takemura M. Discovery of a small-molecule inhibitor of β-1,6-glucan synthesis. Antimicrob. Agents Chemother 53, 670–677 (2009). [PubMed: 19015325]

76. Yamanaka D. et al. Development of a novel β-1,6-glucan-specific detection system using functionally-modified recombinant endo-β-1,6-glucanase. J. Biol. Chem 295, 5362–5376 (2020). [PubMed: 32132174]

77. Forster D. Biological Network Integration using Convolutions (BIONIC) v.0.2.4. Zenodo 10.5281/zenodo.6762584 (2022).

78. Forster D. BIONIC analyses v.0.1.0. Zenodo 10.5281/zenodo.6762596 (2022).

79. Forster D. BIONIC evaluations (BIONIC-evals) v.0.1.0. Zenodo 10.5281/zenodo.6762602 (2022).

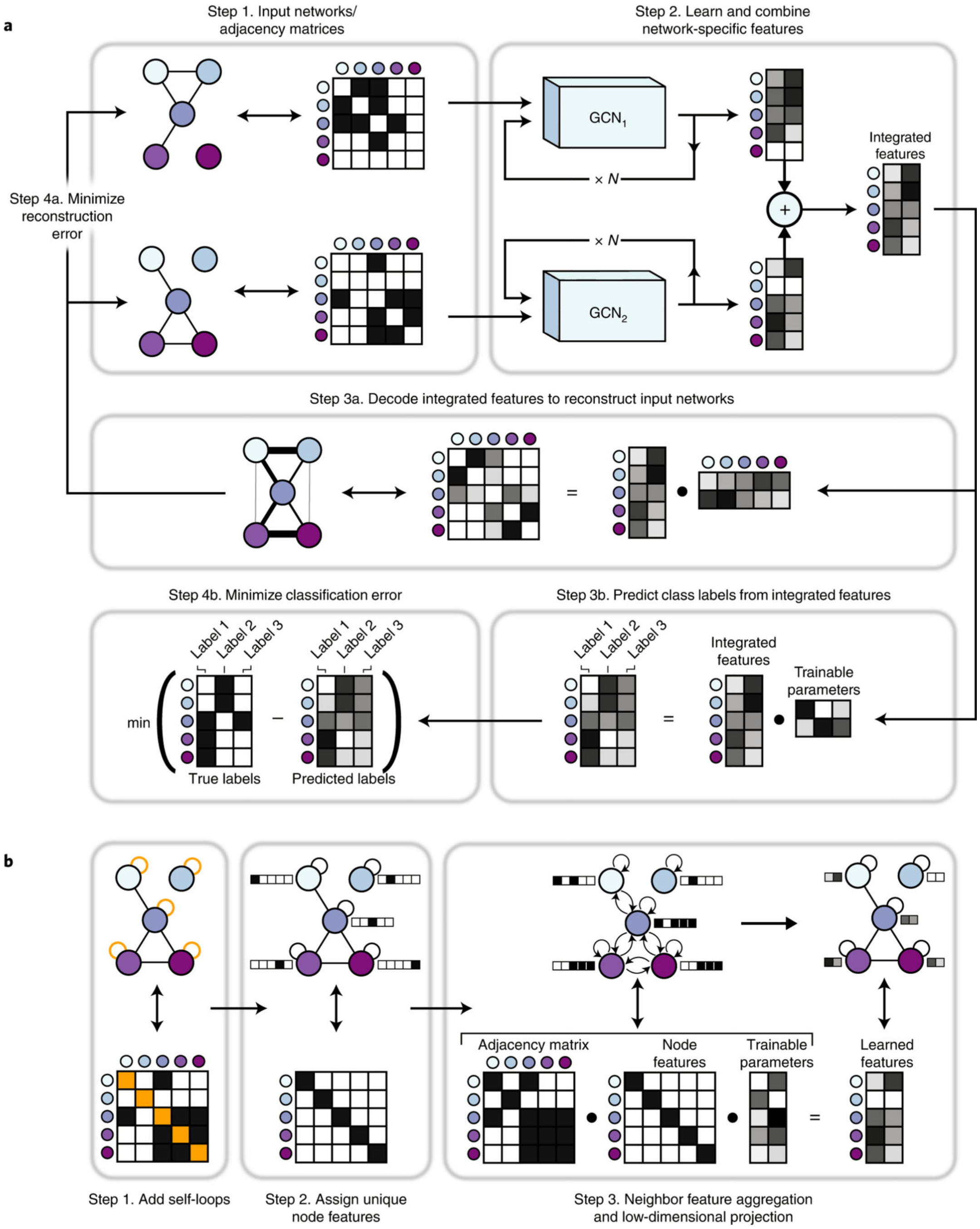

a
Step 1. Input networks/ adjacency matrices
Step 2. Learn and combine network-specific features
GCN$_1$
GCN$_2$
× N
× N
Integrated features
Step 4a. Minimize reconstruction error
Step 3a. Decode integrated features to reconstruct input networks
=
Step 4b. Minimize classification error
Step 3b. Predict class labels from integrated features
Label 1
Label 2
Label 3
min
True labels
Predicted labels
Integrated features
Trainable parameters
b
Adjacency matrix
Node features
Trainable parameters
Learned features
Step 1. Add self-loops
Step 2. Assign unique node features
Step 3. Neighbor feature aggregation and low-dimensional projection

**Fig. 1 |. BIONIC algorithm overview.**

**a**, BIONIC integrates networks as follows: Step 1. Gene interaction networks input into BIONIC are represented as adjacency matrices. Step 2. Each network is passed through a graph convolution network (GCN) to produce network-specific gene features that are then combined into an integrated feature set that can be used for downstream tasks such as functional module detection. The GCNs can be stacked multiple times (denoted by *N*) to generate gene features encompassing larger neighborhoods. Step 3a. Unsupervised. BIONIC attempts to reconstruct the input networks by decoding the integrated features through a dot product operation. Step 4a. Unsupervised. BIONIC trains by updating its weights to reproduce the input networks as accurately as possible. Step 3b. Semisupervised. If labeled data are available, BIONIC predicts functional labels for each gene using the learned gene features. Step 4b. Semisupervised. BIONIC trains by updating its weights to predict the ground-truth labels and minimize classification error. **b**, The GCN architecture functions by: Step 1. Adding self-loops to each network node; Step 2. Assigning a ‘one-hot’ feature vector to each node for the GCN to uniquely identify the nodes; and Step 3. Propagating node features along edges followed by a low-dimensional, learned projection to obtain updated node features that encode the network topology.

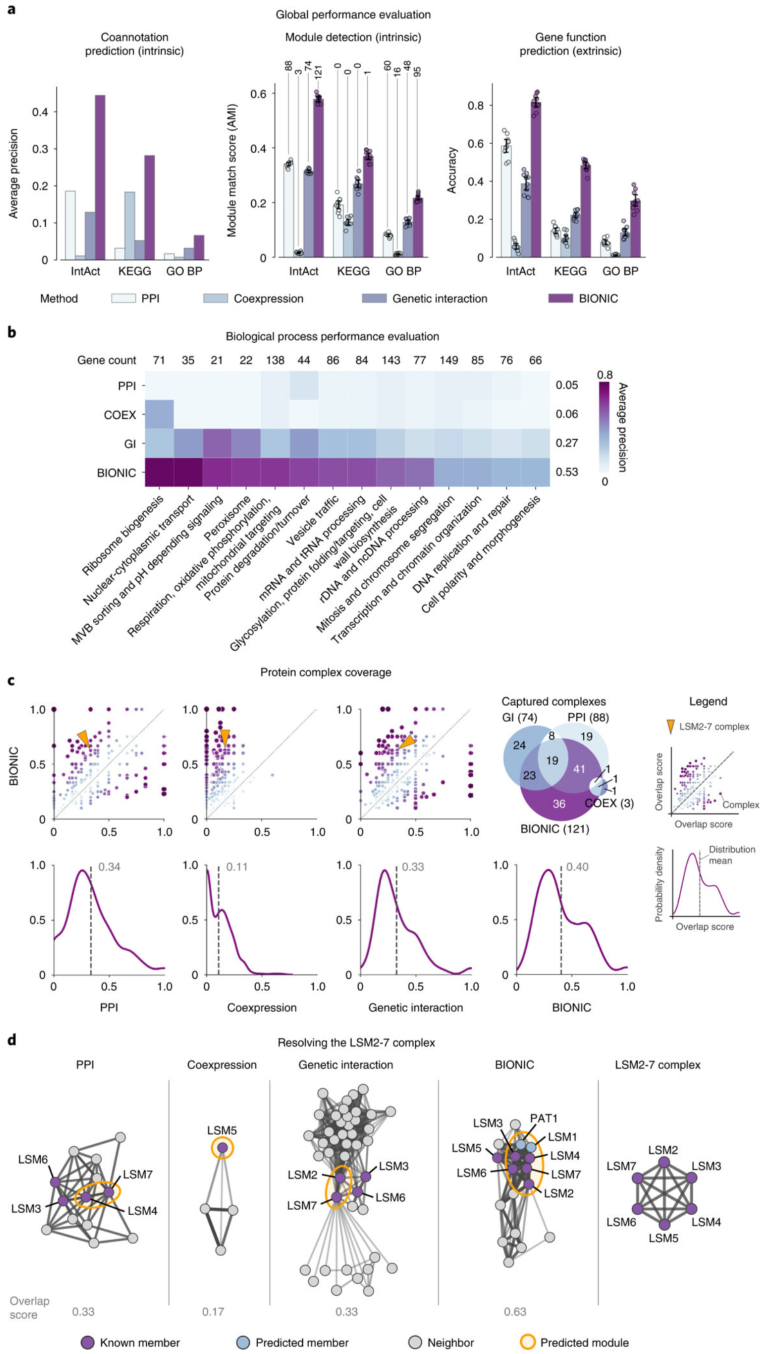


**Fig. 2 |. Comparison of BIONIC integration to three input networks.**

**a**, Functional evaluations for three yeast networks, and unsupervised BIONIC integration. Data are presented as mean values. Error bars indicate the 95% confidence interval for $n = 10$ independent samples. Number of captured modules are indicated above the module detection bars as determined by a 0.5 overlap (Jaccard) score cutoff. **b**, Evaluations over high-level functional categories, split by category. Numbers above columns indicate gene overlap with integration results and the average performance of each method is reported (right of each row). **c**, Top row: comparison of overlap scores between known complexes and predicted modules. Each point is a protein complex. The axes indicate the overlap (Jaccard) score, where 0 indicates no members of the complex were captured, and 1.0 indicates the complex was captured perfectly. The diagonal indicates equivalent performance. Points above the diagonal are complexes where BIONIC outperforms the given network, and points below the diagonal are complexes where BIONIC underperforms. The arrows indicate the LSM2–7 complex, shown in **d**. A Venn diagram describes the overlap of captured complexes (score of 0.5 or higher) between the input networks and BIONIC integration. Numbers in brackets denote the total captured complexes for each method. Bottom row: the distribution of overlap scores between predicted and known complexes. The dashed line indicates the mean. **d**, Functional relationships between predicted LSM2–7 complex members and genes in the local neighborhood, as given by the three input networks and BIONIC integration. The predicted cluster best matching the LSM2–7 complex in each network is circled. The overlap score of the predicted module with the LSM2–7 complex is shown. Edges correspond to protein–protein interactions in PPI[20], Pearson correlation between gene profiles in coexpression[21] and genetic interaction[22] networks, and cosine similarity between gene features in the BIONIC integration. The complete LSM2–7 complex is depicted on the right. Edge weight corresponds to the strength of the functional relationship. PPI, protein–protein interaction; COEX, coexpression; GI, genetic interaction; BP, biological process.

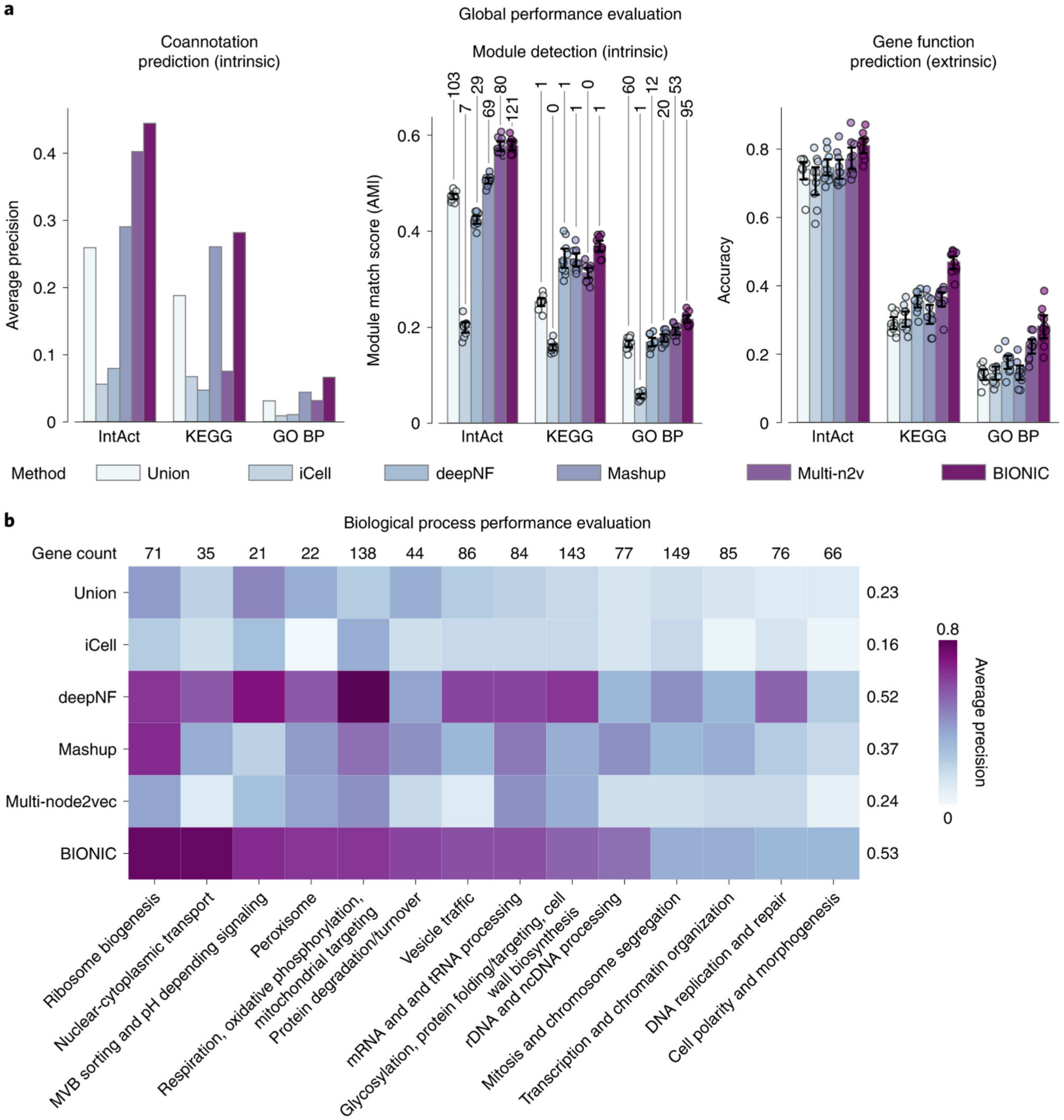


**Fig. 3 |. Comparison of BIONIC to existing integration approaches.**

**a**, Coannotation prediction, module detection and gene function prediction evaluations for three yeast networks integrated by the tested unsupervised network integration methods. The input networks and evaluation standards are the same as in Fig. 2. Data are presented as mean values. Error bars indicate the 95% confidence interval for $n = 10$ independent samples. Numbers above the module detection bars indicate the number of captured modules, as determined by a 0.5 overlap (Jaccard) score cutoff. **b**, Evaluation of integrated features using high-level functional categories, split by category. Numbers above columns indicate gene overlap with integration results and the average performance of each method across categories is reported (right of each row). PPI, protein– protein interaction; BP, biological process.

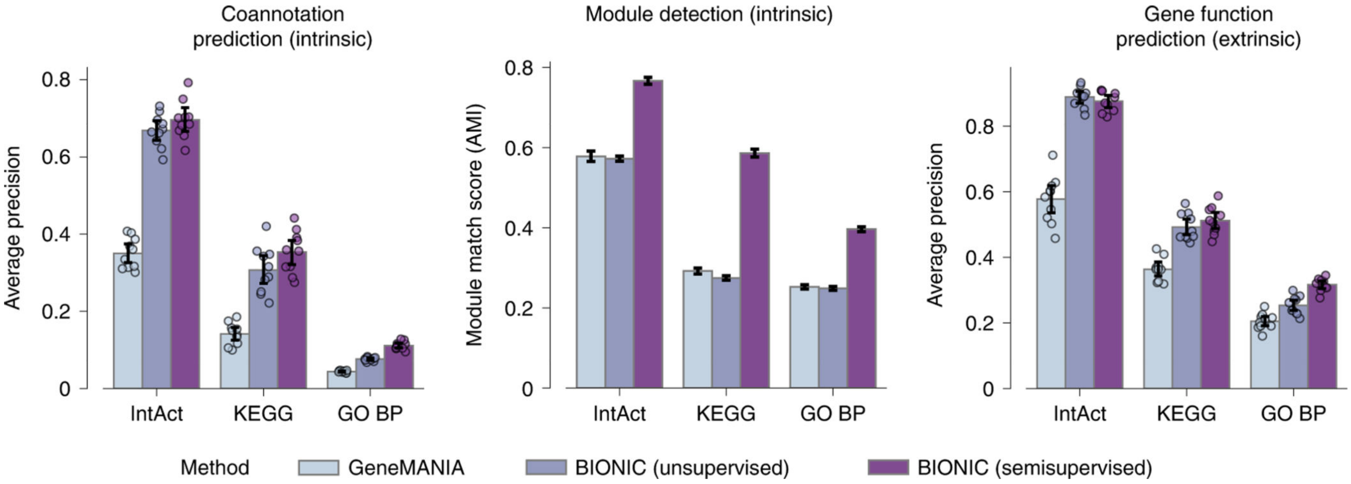


**Fig. 4 |. Supervised performance of BIONIC compared with an existing supervised integration approach.**

Performance comparison between a supervised network integration algorithm trained with labeled data (GeneMANIA), BIONIC trained without any labeled data (unsupervised) and BIONIC trained with labeled data (semisupervised). Bars indicate the average performance over ten trials of random train-test splits for the given benchmark (Methods). Data are presented as mean values. Error bars indicate the 95% confidence interval. $n$ = 10 independent samples for the coannotation prediction and gene function prediction evaluations, and $n$ = 100 for the module detection evaluation. BP, biological process.

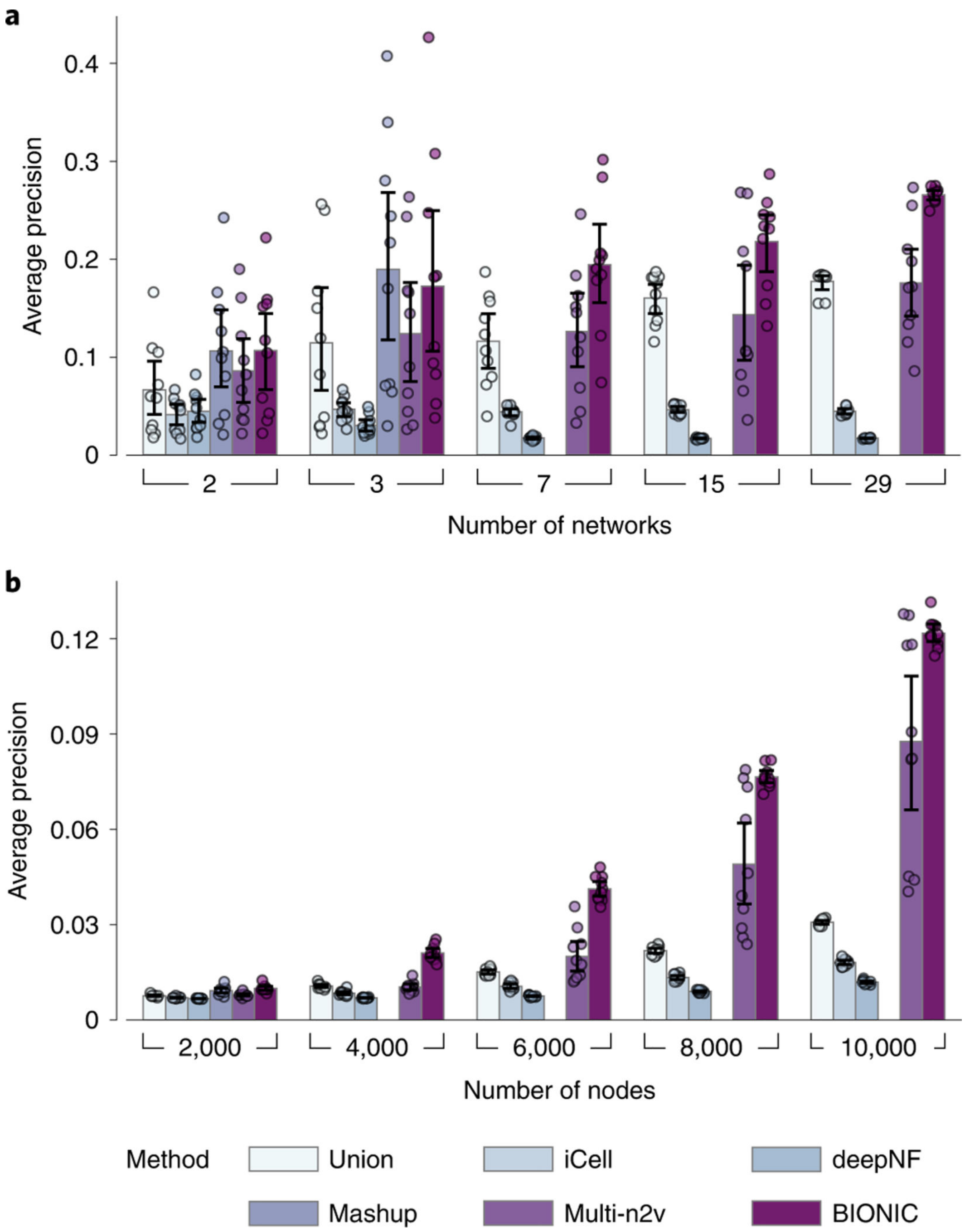


**Fig. 5 |. Network quantity and network size performance comparison across integration methods.**

**a**, Performance comparison of unsupervised integration methods across different numbers of randomly sampled yeast coexpression input networks on KEGG pathways gene coannotations. **b**, Performance comparison of unsupervised integration methods across four human protein–protein interaction networks for a range of subsampled nodes (genes) on CORUM complexes protein coannotations. In these experiments, the Mashup method failed to scale to seven or more networks (**a**) and 4,000 or more nodes (**b**), as indicated by the absence of bars in those cases (Methods). Data are presented as mean values. Error bars indicate the 95% confidence interval for $n = 10$ independent samples. multi-n2v, multi-node2vec.

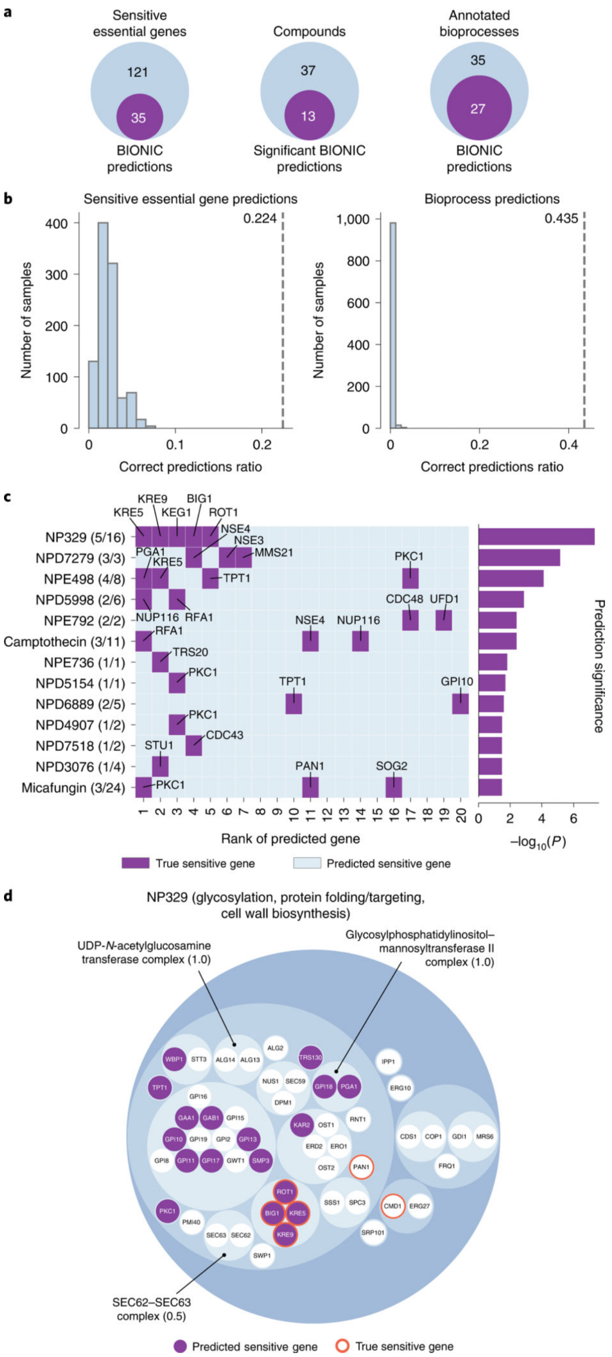


**Fig. 6 |. BIONIC essential gene chemical–genetic interaction predictions.**

**a**, From left to right, the number of correct unsupervised BIONIC sensitive essential gene predictions across the 50 screened compounds, the number of compounds BIONIC significantly predicted sensitive essential genes for (ordered Fisher's exact test) and the number of correctly predicted sensitive essential gene annotated bioprocesses, based on the bioprocess enrichment of BIONIC predictions for each compound. **b**, A comparison of correctly predicted sensitive genes (left) and correctly predicted biological process annotations (right) between BIONIC predictions (dashed line) and $n = 1,000$ random permutations of BIONIC features gene labels (histogram). Correct prediction ratio is the number of correct predictions divided by the number of total sensitive essential genes (left) or annotated biological processes (right) across the 50 screened compounds. **c**, Rank of BIONIC sensitive essential gene predictions for the 13 significantly predicted compounds. The number of correctly predicted genes out of total sensitive genes are shown in parentheses beside each compound name. The statistical significance of the BIONIC predictions for each compound is displayed in the bar plot on the right. **d**, Hierarchical organization of essential genes in the glycosylation, protein folding/targeting, cell wall biosynthesis bioprocess based on integrated BIONIC features. Smallest circles correspond to genes, larger circles indicate clusters of genes. Six genes sensitive to the NP329 compound are indicated with orange borders, and corresponding BIONIC predictions lying in the bioprocess are indicated as purple circles. Captured protein complexes in the bioprocess are annotated and the corresponding overlap score (Jaccard) with the true complex is given in parentheses. Source data for this figure are provided in Supplementary Data File 7.