Bioinformatics, 36(11), 2020, 3457–3465
doi: 10.1093/bioinformatics/btaa150
Advance Access Publication Date: 14 April 2020
Original Paper

Systems biology

Supervised learning is an accurate method for network-based gene classification

Renming Liu[1,†], Christopher A. Mancuso[1,†], Anna Yannakopoulos[1], Kayla A. Johnson[1,2] and Arjun Krishnan [1,2,*]

[1]Department of Computational Mathematics, Science and Engineering and [2]Department of Biochemistry and Molecular Biology, Michigan State University, East Lansing, MI 48824, USA

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Zhiyong Lu

Received on August 26, 2019; revised on December 1, 2019; editorial decision on February 19, 2020; accepted on February 27, 2020

Abstract

Background: Assigning every human gene to specific functions, diseases and traits is a grand challenge in modern genetics. Key to addressing this challenge are computational methods, such as supervised learning and label propagation, that can leverage molecular interaction networks to predict gene attributes. In spite of being a popular machine-learning technique across fields, supervised learning has been applied only in a few network-based studies for predicting pathway-, phenotype- or disease-associated genes. It is unknown how supervised learning broadly performs across different networks and diverse gene classification tasks, and how it compares to label propagation, the widely benchmarked canonical approach for this problem.

Results: In this study, we present a comprehensive benchmarking of supervised learning for network-based gene classification, evaluating this approach and a classic label propagation technique on hundreds of diverse prediction tasks and multiple networks using stringent evaluation schemes. We demonstrate that supervised learning on a gene's full network connectivity outperforms label propagaton and achieves high prediction accuracy by efficiently capturing local network properties, rivaling label propagation's appeal for naturally using network topology. We further show that supervised learning on the full network is also superior to learning on node embeddings (derived using *node2vec*), an increasingly popular approach for concisely representing network connectivity. These results show that supervised learning is an accurate approach for prioritizing genes associated with diverse functions, diseases and traits and should be considered a staple of network-based gene classification workflows.

Availability and implementation: The datasets and the code used to reproduce the results and add new gene classification methods have been made freely available.

Contact: arjun@msu.edu

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

1 Introduction

A grand challenge in the post-genomic era is to characterize every gene across the genome in terms of the cellular pathways they participate in, and which multifactorial traits and diseases they are associated with. Computationally predicting the association of genes to pathways, traits or diseases—the task termed here as 'gene classification'—has been critical to this quest, helping prioritize candidates for experimental verification and for shedding light on poorly characterized genes (Bernardes and Pedreira, 2013; Jiang *et al.*, 2016; Peña-Castillo *et al.*, 2008; Piro and Cunto, 2012; Radivojac *et al.*, 2013; Sharan *et al.*, 2007; Yang *et al.*, 2011). Key to the success of these methods has been the steady accumulation of large amounts of publicly available data collections, such as curated databases of genes and their various attributes (Buniello *et al.*, 2019; Kanehisa *et al.*, 2017, 2019; Kanehisa and Goto, 2000; Liberzon *et al.*, 2011; Piñero *et al.*, 2015, 2017; Smith *et al.*, 2018; Subramanian *et al.*, 2005; Wu *et al.*, 2013; Xin *et al.*, 2016), controlled vocabularies of biological terms organized into ontologies (Ashburner *et al.*, 2000; Schriml *et al.*, 2012; Smith *et al.*, 2004; The Gene Ontology Consortium, 2019), high-throughput functional genomic assays (Athar *et al.*, 2019; Edgar *et al.*, 2002; Leinonen *et al.*, 2011) and molecular interaction networks (Greene *et al.*, 2015; Huang *et al.*, 2018; Li *et al.*, 2017; Stark *et al.*, 2006; Szklarczyk *et al.*, 2015).

While protein sequence and 3D structure are remarkably informative about the corresponding gene's molecular function (Altschul *et al.*, 1990; Jiang *et al.*, 2016; Radivojac *et al.*, 2013; Sleator and Walsh, 2010; Whisstock and Lesk, 2003), the pathways or phenotypes that a gene might participate in significantly depends on the other genes that it works with in a context dependent manner. Molecular interaction networks—graphs with genes or proteins as nodes and their physical or functional relationships as edges—are powerful models for capturing the functional neighborhood of genes on a whole-genome scale (Karaoz *et al.*, 2004; Leone and Pagnani, 2005; Vazquez *et al.*, 2003). These networks are often constructed by aggregating multiple sources of information about gene interactions in a context-specific manner (Greene *et al.*, 2015; Ideker and Sharan, 2008). Therefore, unsurprisingly, several studies have taken advantage of these graphs to perform network-based gene classification (Guan *et al.*, 2010; Köhler *et al.*, 2008; Leiserson *et al.*, 2015; Park *et al.*, 2013; Vanunu *et al.*, 2010; Warde-Farley *et al.*, 2010).

The canonical principle of network-based gene classification is 'guilt-by-association', the notion that proteins/genes that are strongly connected to each other in the network are likely to perform the same functions, and hence, participate in similar higher-level attributes, such as phenotypes and diseases (Wang *et al.*, 2011). Instead of just aggregating 'local' information from direct neighbors (Schwikowski *et al.*, 2000), this principle is better realized by propagating pathway or disease labels across the network to capture 'global' patterns, achieving state-of-the-art results (Cáceres and Paccanaro, 2019; Cowen *et al.*, 2017; Deng *et al.*, 2004; Karaoz *et al.*, 2004; Köhler *et al.*, 2008; Komurov *et al.*, 2010; Leiserson *et al.*, 2015; Leone and Pagnani, 2005; Mostafavi *et al.*, 2008; Murali *et al.*, 2011; Nabieva *et al.*, 2005; Page *et al.*, 1999; Tsuda *et al.*, 2005; Vanunu *et al.*, 2010; Vazquez *et al.*, 2003; Warde-Farley *et al.*, 2010; Zhou *et al.*, 2003; Zhu *et al.*, 2003). These global approaches belong to a class of methods referred to here as 'label propagation'. Distinct from label propagation (LP) is another class of methods for gene classification that relies on the idea that network patterns characteristic of genes associated with a specific phenotype or pathway can be captured using supervised machine learning (Barutcuoglu *et al.*, 2006; Greene *et al.*, 2015; Guan *et al.*, 2010; Krishnan *et al.*, 2016; Lanckriet *et al.*, 2004; Park *et al.*, 2013). While this class of methods—referred to here as 'supervised learning'—has yielded promising results in a number of applications, how it broadly performs across different types of networks and diverse gene classification tasks is unknown. Consequently, supervised learning (SL) is used far less compared to LP for network-based gene classification.

The goal of this study is to perform a comprehensive, systematic benchmarking of SL for network-based gene classification across a number of genome-wide molecular networks and hundreds of diverse prediction tasks using meaningful evaluation schemes. Within this rigorous framework, we compare SL to a widely used, classic LP technique, testing both the original (adjacency matrix A) and a diffusion-based representation of the network (influence matrix I; Fig. 1). This combination results in four methods (listed with their earliest known references): LP on the adjacency matrix (LP-A) (Schwikowski *et al.*, 2000), LP on the influence matrix (LP-I) (Page *et al.*, 1999), SL on the adjacency matrix (SL-A) (Barutcuoglu *et al.*, 2006) and SL on the influence matrix (SL-I) (Lanckriet *et al.*, 2004). Additionally, we evaluate the performance of SL using node embeddings as features, as the use of node embeddings is burgeoning in network biology.

Our results demonstrate that SL outperforms LP for gene-function, gene-disease and gene-trait prediction. We also observe that SL captures local network properties as efficiently as LP, where both methods achieve more accurate predictions for genesets that are more tightly clustered in the network. Lastly, we show that SL using the full network connectivity is superior to using low-dimensional node embeddings as the features, which, in turn, is competitive to LP.

2 Materials and Methods

2.1 Networks

We chose a diverse set of undirected, human gene/protein networks based on criteria laid out in Huang *et al.* (2018) (Fig. 1): (i) networks constructed using high- or low-throughput data, (ii) the type of interactions the network was constructed from and (iii) if annotations were directly incorporated in constructing the network. We used versions of the networks that were released prior to 2017 so as not to bias the temporal holdout evaluations. We used all edge scores (weights) unless otherwise noted, and the nodes in all networks were mapped into Entrez genes using the MyGene.info database (Wu *et al.*, 2013; Xin *et al.*, 2016). If the original node ID mapped to multiple Entrez IDs, we added edges between all possible mappings. The networks used in this study are BioGRID (Stark *et al.*, 2006), the full STRING network (Szklarczyk *et al.*, 2015), as well as the subset with just experimental support (referred to as STRING-EXP in this study), InBioMap (Li *et al.*, 2017) and the tissue-naïve network from GIANT (Greene *et al.*, 2015), referred to as GIANT-TN in this study. These networks cover a wide size range, with the number of nodes ranging from 14 089 to 25 689 and the number of edges ranging from 141 629 to 38 904 929. More information on the networks can be found in the Supplementary Section 1.1.

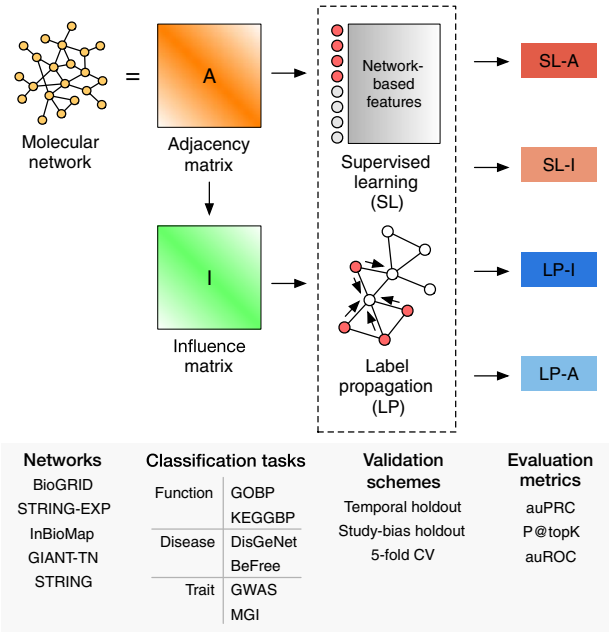

Fig. 1. Workflow for gene classification pipeline. Four methods are compared: SL-A, SL-I, LP-A and LP-I. Model performance on a variety of gene classification tasks is evaluated over a number of different molecular networks, validation schemes and evaluation metrics. Additionally, the performance of SL using node embeddings as features (SL-E) is evaluated (not shown in this figure)

2.2 Network representations

We considered three distinct representations of molecular networks: the adjacency matrix, an influence matrix and low-dimensional node embeddings. Let $G = (V, E, W)$ denote an undirected molecular network, where V is the set of vertices (genes), E is the set of edges (associations between genes) and W is the set of edge weights (the strengths of the associations). G can be represented as a weighted adjacency matrix $A_{i,j} = W_{i,j}$, where $A \in \mathbb{R}^{V \times V}$. G can also be represented as an influence matrix, $F \in \mathbb{R}^{V \times V}$, which can capture both local and global structure of the network. F was obtained using a random walk with restart transformation kernel (Leiserson *et al.*, 2015),

$$F = \alpha[I - (1 - \alpha)W_D]^{-1} \quad (1)$$

where α is the restart parameter, I is the identity matrix and W_D is the degree weighted adjacency matrix given by $W_D = AD^{-1}$, where $D \in \mathbb{R}^{V \times V}$ is a diagonal matrix of node degrees. A restart parameter of 0.85 was used for every network in this study. Detailed

information of how the restart parameter was chosen can be found in Supplementary Section 1.2 and Figure S1.

G can also be transformed into a low-dimensional representation through the process of node embedding. In this study, we used the *node2vec* algorithm (Grover and Leskovec, 2016), which borrows ideas from the *word2vec* algorithm (Mikolov *et al.*, 2013a, 2013b) used in natural language processing. The objective of *node2vec* is to find a low-dimensional representation of the adjacency matrix, $E \in \mathbb{R}^{V \times d}$, where $d \ll V$. This is done by optimizing the following log-probability objective function:

$$E = \max_{f} \sum_{u \in V} \log\left(\Pr\left(N_S(u)|e(u)\right)\right) \quad (2)$$

where $N_S(u)$ is the network neighborhood of node u generated through a sampling strategy S and $e(u) \in \mathbb{R}^d$ is the feature vector of node u. In *node2vec*, the sampling strategy is based on random walks that are controlled using two parameters *p* and *q*, in which a high value of *q* keeps the walk local (a breadth-first search), and a high value of *p* encourages outward exploration (a depth-first search). The values of *p* and *q* were both set to 0.1 for every network in this study. Detailed information of how the node2vec hyperparameters were chosen can be found in the Supplementary Section 1.2.

2.3 Prediction methods

We compared the prediction performance across four specific methods across two classes, LP and SL.

2.3.1 Label propagation

LP methods are the most widely used methods in network-based gene classification and achieve state-of-the-art results (Cowen *et al.*, 2017; Köhler *et al.*, 2008). In this study, we considered two LP methods, LP-A and LP-I. First, we constructed a binary vector of ground-truth labels, $x \in \mathbb{R}^{V \times 1}$, where $x_i = 1$ if gene i is a positively labeled gene in the training set, and 0 otherwise. In LP-A, we constructed a score vector, $S \in \mathbb{R}^{V \times 1}$, denoting the predictions,

$$S = Ax \quad (3)$$

where A is the adjacency matrix. Thus, the predicted score for a gene using LP-A is equal to the sum of the weights of the edges between the gene and its direct, positively labeled network neighbors. In LP-I, the score vector, S, is generated using equation (3), except A is by replaced by F, the influence matrix [equation (1)]. In both LP-A and LP-I, only positive examples in the training set are used to calculate the score vector S to reflect how LP is typically used in practice (Cowen *et al.*, 2017; Köhler *et al.*, 2008; Picart-Armada *et al.*, 2019). Both positive and negative examples in the test set are later used for evaluation.

2.3.2 Supervised learning

SL can be used for network-based gene classification by using each gene's network neighborhoods as feature vectors, along with gene labels, in a classification algorithm. Here, we used logistic regression with L2 regularization as the SL classification algorithm, which is a linear model that aims to minimize the following cost function (Pedregosa *et al.*, 2011):

$$\min_{w,\ c} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \log\left(\exp\left(-y_i\left(X_i^T w + c\right)\right) + 1\right) \quad (4)$$

where $w \in \mathbb{R}^m$ is the vector of weights for a model with *m* features, C determines the regularization strength, *n* the number of examples, y is the ground-truth label, $X \in \mathbb{R}^{n \times m}$ is the data matrix and c is the intercept. After training a model using the labeled genes in the training set, the learned model weights are used to classify the genes in the testing set, returning a prediction probability for these genes that is bounded between 0 and 1. The regularization parameter, C, was set to 1.0 for all models in this study.

In this study, three different network-based gene-level feature vectors were used to train three different SL classifiers: the rows of the adjacency matrix (SL-A), the rows of the influence matrix (SL-I) and the rows of the node embedding matrix (SL-E). Model selection and hyperparameter tuning are described in detail in the Supplementary Section 1.2.

2.4 Geneset-collections

We curated a number of geneset-collections to test predictions on a diverse set of tasks: function, disease and trait (Fig. 1). Function prediction was defined as predicting genes associated with biological processes that are part of the Gene Ontology (referred to here as 'GOBP') (The Gene Ontology Consortium, 2019; Ashburner *et al.*, 2000) obtained from MyGene.info (Wu *et al.*, 2013; Xin *et al.*, 2016), and pathways from the Kyoto Encyclopedia of Genes and Genomes (Kanehisa *et al.*, 2017, 2019; Kanehisa and Goto, 2000), referred to 'KEGGBP' since disease-related pathways were removed from the original KEGG annotations in the Molecular Signatures Database (Liberzon *et al.*, 2011; Subramanian *et al.*, 2005). Disease prediction was defined based on predicting genes associated with diseases in the DisGeNET database (Piñero *et al.*, 2015, 2017). Annotations from this database were divided into two separate geneset-collections: those that were manually-curated (referred to as 'DisGeNet' in this study) and those derived using the BeFree text-mining tool (referred to as 'BeFree' in this study). Trait prediction was defined as predicting genes linked to human traits from Genome-wide Association Studies (GWAS), curated from a community challenge (Choobdar *et al.*, 2019), and mammalian phenotypes (annotated to human genes) from the Mouse Gene Informatics (MGI) database (Smith *et al.*, 2018).

Each of these six geneset-collections contained anywhere from about a hundred to tens of thousands of genesets that varied widely in specificity and redundancy. Therefore, each collection was pre-processed to ensure that the final set of prediction tasks from each source is specific, largely non-overlapping and not driven by multi-attribute genes. First, if genesets in a collection corresponded to terms in an ontology (e.g. biological processes in the GOBP collection), annotations were propagated along the ontology structure to obtain a complete set of annotations for all genesets. Second, we removed genesets if the number of genes annotated to the geneset was above a certain threshold and then compared these genesets to each other in order to remove genesets that were highly-overlapping with other genesets in the collection, resulting in a set of specific, non-redundant genesets. Finally, individual genes that appeared in more than 10 of the remaining genesets in a collection were removed from all the genesets in that collection to remove multi-attribute (e.g. multi-functional) genes that are potentially easy to predict (Gillis and Pavlidis, 2011). Detailed information on geneset pre-processing and geneset attributes can be found in the Supplementary Section 1.3, Table S2 and Figure S3.

Selecting positive and negative examples: In each geneset-collection, for a given geneset, genes annotated to that set were designated as the set of positive examples. The SL methods additionally required a set of negative genes for each given geneset for training, and both SL and LP methods require a set of negative genes for each geneset for testing. A set of negative genes was generated by: (i) finding the union of all genes annotated to all genesets in the collection, (ii) removing genes annotated to the given geneset and (iii) removing genes annotated to any geneset in the collection which significantly overlapped with the given geneset (*P*-value <0.05 based on the one-sided Fisher's exact test).

2.5 Validation schemes

We performed extensive and rigorous evaluations based on three validation schemes: temporal holdout, study-bias holdout and 5-fold cross validation (5FCV). In temporal holdout, within a geneset-collection, genes that only had an annotation to any geneset in the collection after January 1, 2017 were considered test genes, and all other genes were considered training genes. Temporal holdout is the most stringent evaluation scheme for gene classification since it mimics the practical scenario of using current knowledge to predict the future and is the preferred evaluation method used in the CAFA

challenges (Jiang *et al.*, 2016; Radivojac *et al.*, 2013). Since the Gene Ontology was the only source with clear date-stamps for all its annotations, temporal holdout was applied only to the GOBP geneset-collection. For study-bias holdout, genes were ranked by the number of PubMed articles they were mentioned in, obtained from Brown *et al.* (2015). The top two-thirds of the most-mentioned genes were considered training genes, and the rest of the least-mentioned genes were used for testing. Study-bias holdout mimics the real-world situation of learning from well-characterized genes to predict novel un(der)-characterized genes. The last validation scheme is the traditional 5FCV, where the genes are split into five equal folds in a stratified manner (i.e. in each split, the proportion of genes in the positive and negative classes is preserved). In all these schemes, only genesets with at least 10 positive genes in both the training and test sets were considered. More information on the validation schemes is available in the Supplementary Section 1.4.

2.6 Evaluation metrics

In this study, we considered three evaluation metrics: the area under the precision-recall curve (auPRC), the precision of the top-K ranked predictions (P@TopK) and, the area under the receiver-operator curve (auROC). For P@TopK, we set K equal to the number of ground-truth positives in the testing set. Since the standard auPRC and P@TopK scores are influenced by the prior probability of finding a positive example (equal to the proportion of positives to the total of positives and negatives), we expressed both metrics as the logarithm (base 2) of the ratio of the original metric to the prior. More details on the evaluation metrics can be found in the Supplementary Section 1.5.

3 Results

We systematically compare the performance of four gene classification methods (Fig. 1): SL-A, SL-I, LP-A and LP-I. We choose six geneset-collections that represent three prominent gene classification tasks: gene-function (GOBP, KEGGBP), gene-disease (DisGeNet, BeFree) and gene-trait (GWAS, MGI) prediction. We use three different validation schemes: temporal holdout (train on genes annotated before 2017 and test on genes annotated in 2017 or later; only done for GOBP as it has clear timestamps), holdout based on study bias (train on well-studied genes and predict on less-studied genes) and the traditional 5FCV. Temporal holdout and study-bias holdout validation schemes are presented in the main text as they are more stringent and reflective of real-world tasks as compared to 5FCV (Kahanda *et al.*, 2015). To ascertain the robustness of the relative performance of the methods to the underlying network, we choose five different genome-scale molecular networks that differ in their content and construction. To be in concert with temporal holdout evaluation and curtail data leakage, all the networks used throughout this study are the latest versions released before 2017. We present evaluation results based on the auPRC in the main text and results based on the P@topK and auROC in the Supplementary Figures S4–S8. We note that the 5FCV, P@topK and auROC results in the Supplementary Material are, for the most part, consistent with the results presented in the main text of this study.

Our first analysis was to directly compare all four prediction methods against each other for each geneset in a given collection. For each geneset-collection–network combination, we rank the four methods per geneset (based on auPRC) using the standard competition ranking and calculate each method's average rank across all the genesets in the collection (Fig. 2). For function prediction, SL-A is the top-performing method by a wide margin (particularly clear based on GOBP temporal holdout), with SL-I being the second best method. For disease and trait prediction, SL-A and SL-I still outperform LP-I, but to a lesser extent. In all cases, LP-A is the worst performing method. The large performance difference between the SL and LP methods in the GOBP temporal holdout validation is noteworthy since temporal holdout is the most stringent validation scheme and the one employed in community challenges, such as CAFA (Jiang *et al.*, 2016; Radivojac *et al.*, 2013).

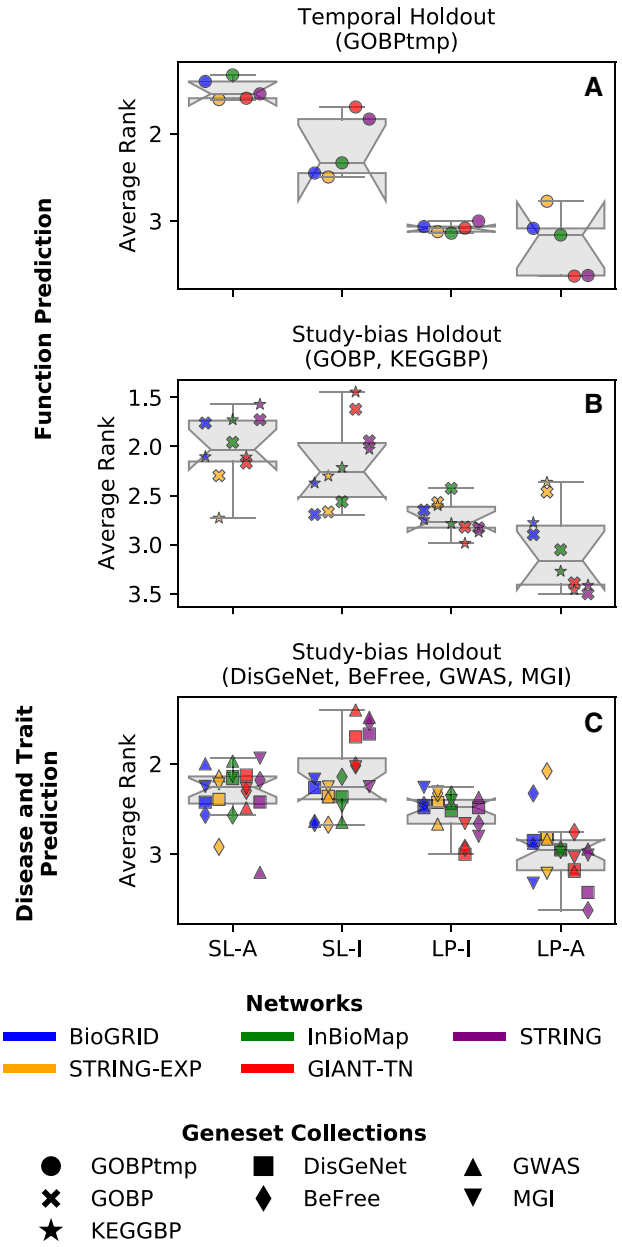

Fig. 2. Average rank across the four methods. Each point in each boxplot represents the average rank for a geneset-collection–network combination, obtained based on ranking the four methods in terms of performance for each geneset in a geneset-collection using the standard competition ranking. (**A**) Functional prediction tasks using GOBP temporal holdout, (**B**) functional prediction tasks using study-bias holdout for GOBP and KEGGBP and (**C**) disease and trait prediction tasks using study-bias holdout for DisGeNet, BeFree, GWAS and MGI. The results are shown for auPRC where different colors represent different networks and different marker styles represent the different geneset-collections. SL methods outperform LP methods for all prediction tasks

Following the observation that SL methods outperform LP methods based on relative ranking, we use a non-parametric paired test (Wicoxon signed-rank test) to statistically assess the difference between specific pairs of methods (Fig. 3A). For each geneset-collection–network combination, we compare the two methods in one class to the two methods in the other class (i.e. we compare SL-A to LP-A, SL-A to LP-I, SL-I to LP-A and SL-I to LP-I). Each comparison yields a *P*-value along with the number of genesets in the collection where one method outperforms the other. After correcting the four *P*-values for multiple hypothesis testing (Benjamini *et al.*, 2006), if a method from one class outperforms both methods from the other

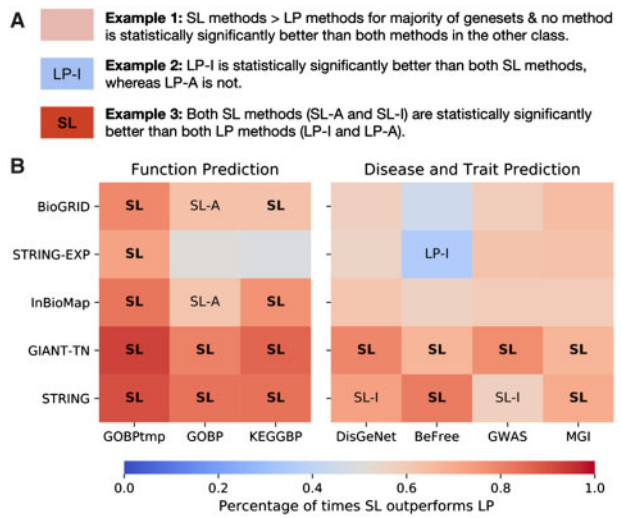

Fig. 3. Testing for a statistically significant difference between SL and LP methods. (**A**) A key on interpreting the analysis. For each network–geneset combination, each method is compared to the two methods from the other class (i.e. SL-A versus LP-I, SL-A versus LP-A, SL-I versus LP-I, SL-I versus LP-A). If a method was found to be significantly better than both methods from the other class (Wilcoxon ranked-sum test with an FDR threshold of 0.05), the cell is annotated with that method. If both models in that class were found to be significantly better than the two methods in the other class, the cell is annotated in bold with just the class. The color scale represents the fraction of genesets that were higher for the SL methods across all four comparisons. The first column uses GOBP temporal holdout, whereas the remaining six columns use study-bias holdout. (**B**) SL methods show a statistically significant improvement over LP methods, especially for function prediction

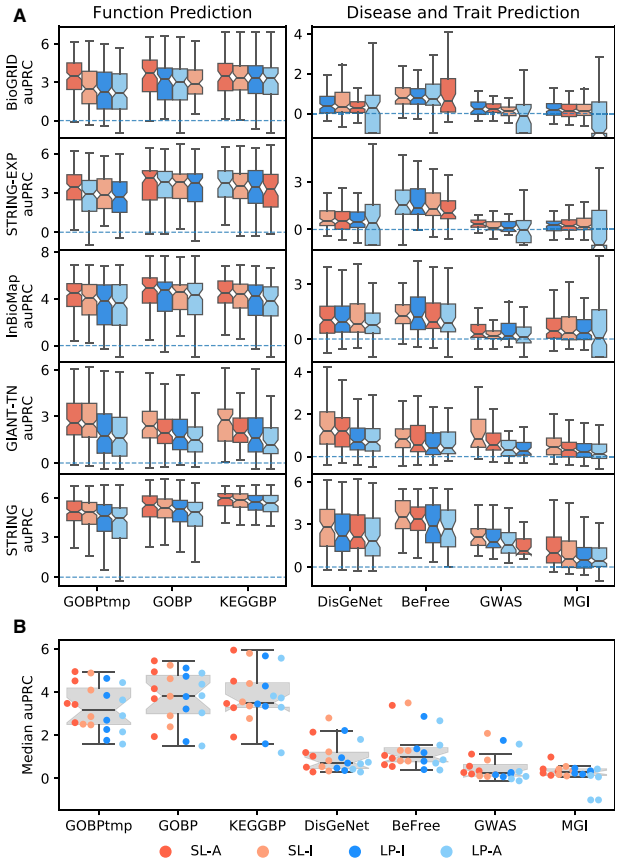

Fig. 4. Boxplots for performance across all geneset-collection–network combinations. (**A**)The performance for each individual geneset-collection–network combination is compared across the four methods; SL-A (red), SL-I (light red), LP-I (blue) and LP-A (light blue). The methods are ranked by median value with the highest scoring method on the left. Results show SL methods outperform LP methods, especially for function prediction. (**B**) Each point in the plot is the median value from one of the boxplots in (A). This shows that both SL and LP methods perform better for function prediction compared to disease/trait prediction. (Color version of this figure is available at *Bioinformatics* online.)

class independently (in terms of the number winning genesets), and if both (corrected) *P*-values are <0.05, we consider a method to have significantly better performance compared to the entire other class.

Additionally, we track the percentage of times the SL methods outperform the LP methods across all four comparisons within a geneset-collection–network combination.

The results show that for function prediction SL is almost always significantly better than LP when considering auPRC (Fig. 3B). Based on temporal holdout on GOBP, both SL-A and SL-I are always significantly better than both LP methods. Based on study-bias holdout, in the 10 function prediction geneset-collections–network combinations using GOBP and KEGGBP, SL-A is a significantly better method 8 times (80%) and SL-I is a significantly better method 6 times (60%). Neither LP-I nor LP-A ever significantly outperforms the SL models. The performance of SL and LP is more comparable for disease and trait prediction, but SL methods still perform better in a larger fraction of genesets. For the 20 disease and trait geneset-collection–network combinations, SL-I is a significantly better method 8 times (40%), and SL-A is a significantly better method 6 times (30%), LP-I is a significantly better method once (5%), and LP-A is never a significantly better method.

To visually inspect not only the relative performance of all four methods, but to also see how well the models are performing in an absolute sense, we examined the boxplots of the auPRC values for every geneset-collection–network combination (Fig. 4). The first notable observation is that, regardless of the method, function prediction tasks have much better performance results than disease/trait prediction tasks (Fig. 4B). Based on temporal holdout for function prediction (GOBPtmp), SL-A is the top-performing model based on the highest median performance for every network. Additionally, for all networks except STRING-EXP, SL-I is the second best performing model. For the 10 combinations of five networks with GOBP and KEGGBP, the top method based on the highest median performance is an SL method all but once, with SL-A being the top model 7 times (70%), SL-I being the top model 2 times (20%, GOBP and KEGGBP on GIANT-TN) and LP-A being the top model once (10%, KEGGBP on STRING-EXP). As noted earlier, for disease and trait prediction, SL and LP methods have more comparable performance. Of the 20 geneset-collection–network combinations, each of SL-A, SL-I, LP-I and LP-A is the top method based on median performance 5 (25%), 10 (50%), 4 (20%) and 1 (5%) times, respectively.

Although the boxplots in Figure 4 can give an idea of effect sizes, to further quantify this, we looked at the ratios of auPRC values across all genesets (Supplementary Section 2.2 and Fig. S10). The results show that SL-A and SL-I both have a substantial effect size compared to LP-I for function prediction. Also, for all prediction tasks, the effect size of SL methods over LP-I is equal to or greater than the effect size of LP-I over LP-A, where LP-I is widely considered a much better model than LP-A and thus, the comparison between LP-I and LP-A can be viewed as a baseline effect size.

Among the two classes of network-based models—SL and LP—it is intuitively clear how LP directly uses network connections to propagate information from the positively labeled nodes to other nodes close in the network. On the other hand, while SL is an accurate method for gene classification, it has not been studied if SL's performance is tied to any traditional notion of network connectivity. To shed light on this problem, we investigated the performance of SL-A and LP-I as a function of three different properties of individual genesets in a collection: the number of annotated genes, edge density (a measure of how tightly connected the geneset is within itself) and segregation (a measure of how isolated the geneset is from the rest of the network). While the performance of neither SL-A nor LP-I has a strong association with the size of the geneset, the

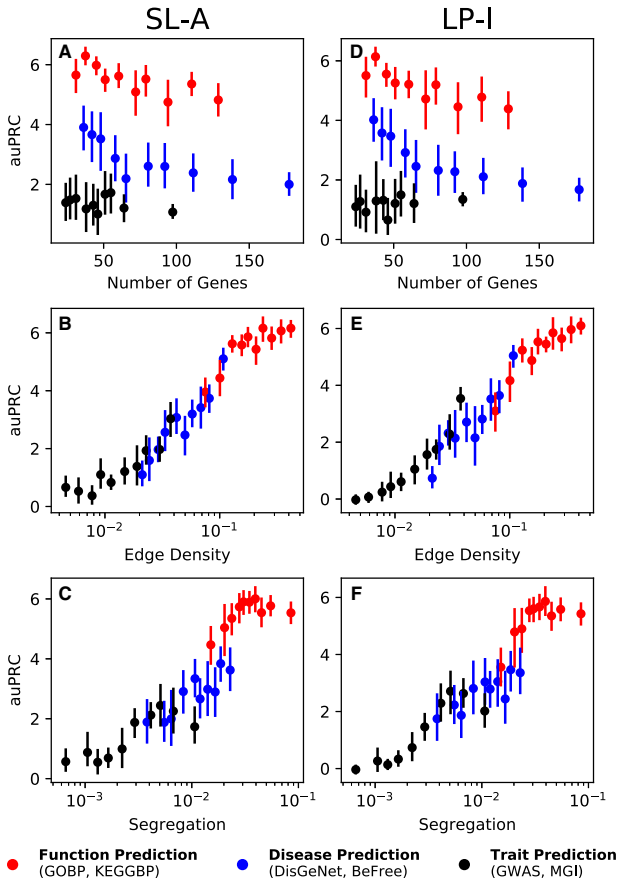

Fig. 5. Performance versus network/geneset properties. SL-A (**A–C**) is able to capture network information as efficiently as LP-I (**D–F**), for the STRING network. There is no correlation between the number of genes in the geneset versus performance (A, D), but there is a strong correlation between the performance and the edge density (B, E) as well as segregation (C, F). The different colored dots represent function genesets (red, GOBP and KEGGBP), disease genesets (blue, DIGenet and BeFree) and trait genesets (black, GWAS and MGI). The vertical line is the 95% confidence interval. Similar trends can be seen for the other networks (Supplementary Fig. S9). (Color version of this figure is available at *Bioinformatics* online.)

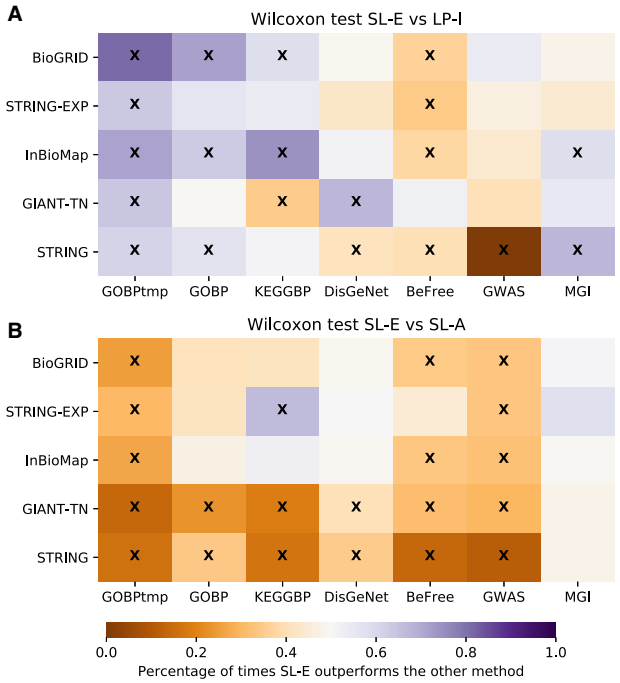

Fig. 6. Performance of SL-E versus LP-I and SL-A. We compare the performance of SL on the embedding matrix (SL-E) versus LP-I and SL-A using a Wilcoxon ranked-sum test. The performance metric is auPRC, the color scale represents the fraction of terms that were higher for the SL-E model (with purple meaning that SL-E had a higher fraction of better performing genesets compared to either LP-I or SL-A) and an 'x' signifies that the *P*-value from the Wilcoxon test was below 0.05. (**A**) Shows that SL-E is quite competitive with the classic method of LP-I and (**B**) shows that SL-A outperforms SL-E in a majority of cases. (Color version of this figure is available at *Bioinformatics* online.)

performance of SL-A has a strong positive correlation with both edge density and segregation of the geneset, similar to what is seen for LP-I (Fig. 5). For visual clarity, Figure 5 presents results for just the STRING network, but very similar results are seen in the other networks as well (Supplementary Fig. S9). Detailed information on how the geneset and network properties are calculated can be found in the Supplementary Section 1.3.

Finally, since machine learning on node embeddings is gaining popularity for network-based node classification, we compared the top SL and LP methods tested here to this approach. Specifically, we compared LP-I and SL-A to an SL method using embeddings (SL-E) obtained from the *node2vec* algorithm (Grover and Leskovec, 2016) (Fig. 6). For function prediction, we observe that SL-E substantially outperforms LP-I. For GOBP temporal holdout, SL-E is always significantly better than LP-I. For the GOBP and KEGGBP study-bias holdout, out of the 10 geneset-collection–network combinations, SL-E is significantly better than LP-I 5 times (50%), whereas the converse is true only once (10%). These patterns nearly reverse for the 20 disease/trait prediction tasks, with LP-I performing significantly better than SL-E 6 times (30%), and SL-E significantly outperforming LP-I 3 times (15%). The comparison between SL-E and SL-A showed that SL-A demonstrably outperforms SL-E for both function and disease/trait prediction tasks. Among the 30 geneset-collection–network combinations, SL-A is a significantly better model 20 times (67%), whereas SL-E comes out on top just once (3%). This shows that although methods that use node embeddings are a promising avenue of research, they should be compared to the strong baseline of SL-A when possible.

4 Discussion

We have conducted the first comprehensive benchmarking of SL for network-based gene classification, establishing it as a leading approach.. Further, to the best of our knowledge, neither the studies that propose new methods nor those that systematically compare existing approaches have directly compared the two classes of methods—SL and LP—against each other. Our work provides this systematic comparison and shows that SL methods demonstrably outperform LP methods for network-based gene classification, particularly for function prediction.

Both SL and LP methods are, in general, more accurate for function prediction than disease and trait prediction. This trend is likely due to the fact that molecular interaction networks are primarily intended, either through curation or reconstruction, to reflect biological relationships between genes/proteins as they pertain to 'normal' cellular function. The utility of network connectivity to gene–disease or gene–trait prediction is incidental to the information the network holds about gene–function associations. This notion is supported by the observation that genesets related to function genesets are more tightly clustered than disease and trait genesets in the genome-wide molecular networks used in this study (Supplementary Fig. S3). Further analysis of prediction accuracy of genesets as a function of their network connectivity lends credence to the use of network structure by SL (Fig. 5 and Supplementary Fig. S9). Part of LP's appeal, widespread use and development is this natural use of network topology to predict gene properties by diffusing information from characterized genes to uncharacterized genes in their network vicinity. Therefore, we expect that genes associated with

tightly clustered pathways, traits or diseases will be easier to predict using LP, which is observed in our analysis (Fig. 5 and Supplementary Fig. S9). On the other hand, since SL (based on the full network) is designed to use global gene connectivity, it has been unclear if there is any association between the local clustering of genesets and their prediction performance using SL. Here, we show that the performance of SL, across networks and types of prediction tasks, is highly correlated with local network clustering of the genes of interest (Fig. 5 and Supplementary Fig. S9). This result substantiates SL as an approach that can accurately predict gene attributes by taking advantage of local network connectivity.

While being accurate, training a SL model on the adjacency matrix (SL-A) can take some computational time and resources as the size of the molecular network increases, thus, considerably differing in speed for, say, STRING-EXP (14 089 nodes and 141 629 unweighted edges) and GIANT-TN (25 689 nodes and 38 904 929 weighted edges). Worthy of note in this context is the recent excitement in deriving node embeddings for each node in a network, concisely encoding its connectivity to all other nodes, and using them as features in SL algorithms for node classification (Cai *et al.*, 2018; Cui *et al.*, 2018; Goyal and Ferrara, 2018; Grover and Leskovec, 2016; Hamilton *et al.*, 2017; Perozzi *et al.*, 2014; Wang *et al.*, 2016). Although we show that SL-A markedly outperforms SL-E (Fig. 6), the unique characteristics of SL-E methods call for further exploration. For instance, the greatly reduced number of features allows SL-E methods to be more readily applicable to classifiers more complex than logistic regression, such as deep neural networks, which are typically ill-suited for problems where the number of features is much greater than the number of training examples. Further, since the reduced number of features allow SL-E methods to be trained orders of magnitude faster than SL-A or SL-I, they can be easily incorporated into ensemble learning models, which combine the results from many shallow learning algorithms. Akin to LP (Valdeolivas *et al.*, 2019; Warde-Farley *et al.*, 2010; Zhao *et al.*, 2019), node embeddings also offer a convenient route to incorporating multiple networks into SL approaches. While methods such as SL-I and SL-A may require concatenating the original networks or integrating them into a single network before learning, recent work has shown that SL-E-based methods can embed information from multiple molecular/heterogeneous networks and learn gene classifiers in tandem (Alshahrani and Hoehndorf, 2018; Ata *et al.*, 2018; Bai *et al.*, 2019; Cho *et al.*, 2016; Gligorijević *et al.*, 2018; Li *et al.*, 2019b; Nelson *et al.*, 2019; Yang *et al.*, 2018; Zitnik and Leskovec, 2017). However, none of these studies have compared the variety of SL-E methods to learning directly on the adjacency matrix. Given our finding here that SL-A greatly outperforms SL-E for function, disease and trait prediction, we advise and urge that every new SL-E method should be compared to SL-A for network-based gene classification.

In past work, SL methods for gene classification have mostly relied on hand-crafting features from graph-theory metrics, such as degree and centrality measures, or combining metrics to expand the feature set, resulting in a feature set size of ∼30 or less (Li *et al.*, 2019a; Zhang *et al.*, 2016). We do not include a comparison to these types of methods in this study because predicting genes to functions or diseases based on generic network metrics, such as high degree, does not capture anything unique about specific functions or diseases. On the other hand, SL models with individual genes as features contain information biologically relevant to the specific prediction task (Lee *et al.*, 2013, 2019).

Critical to all these conclusions is the rigorous preparation of diverse, specific prediction tasks and the choice of meaningful validation schemes and evaluation metrics. Temporal holdout and study-bias holdout validations help faithfully capture the performance of the computational methods when a researcher uses them to prioritize novel uncharacterized genes in existing molecular networks for experimental validation based on a handful of currently known genes. Although we provide all the results for the auROC metric in the Supplementary Materials for completion (Supplementary Figs S4, S5 and S8), we base our conclusions on metrics driven by precision: auPRC and P@topK. While auROC is still commonly used in genomics, it is ill-suited to most biological prediction tasks including gene classification since they are highly imbalanced problems, with negative examples far outnumbering positive examples (Saito and Rehmsmeier, 2015). Optimizing for precision-based metrics, on the other hand, helps control for false-positives among the top candidates (Davis and Goadrich, 2006), an important consideration when providing a list of candidate genes for further study. Accompanying the results in this manuscript, we are providing our comprehensive evaluation framework in the form of data—networks, prediction tasks and evaluation splits—on Zenodo and the underlying code on Github to enable other researchers to not only reproduce our results but also to add new network-based gene classification methods for comparison. Together, the data and the code provide the community a systematic framework to conduct gene classification benchmarking studies. See 'Availability of data and materials' for more information.

In this study, we have presented conclusive evidence that SL is an accurate method for gene classification. However, there exist many possible avenues for future exploration including studying how negative example selection affects gene classification. Although negative example selection has been studied for function prediction in a few studies (Youngs *et al.*, 2013, 2014), there is room for exhaustive testing of how to best generate and incorporate negative examples for gene classification in both LP and SL methods. To begin to address this, we included an analysis using negative examples for LP (Supplementary Section 2.3 and Figs S11–S14), where we show using negative examples slightly improves the performance of LP models, but not enough to change any of the findings in this work. Another avenue of exploration is performing a large model selection and hyperparameter tuning benchmarking study, including more non-linear models and embedding techniques. Lastly, it will be of interest to try network-based SL on other types of tasks, such as predicting enhancer roles in gene regulatory networks or predicting amino acid residue properties from 3D protein structures.

In conclusion, we have established that SL outperforms LP for network-based gene classification across networks and prediction tasks (functions, diseases and traits). We show that SL, in which every gene is its own feature, is able to capture network information just as well as LP. Finally, we show that SL-A demonstrably outperforms SL using node embeddings, and thus we strongly recommend that future work on using node embeddings for gene classification draws a comparison to using SL-A.

Acknowledgements

We are grateful to Daniel Marbach for making the GWAS data available. We thank members of the Krishnan Lab for valuable discussions and feedback on the manuscript.

Author contributions

A.K., R.L. and C.A.M. conceived and designed the experiments; R.L. and C.A.M. performed the experiments; R.L., C.A.M., A.Y., K.A.J. and A.K. processed the networks and geneset-collections and analyzed the data. R.L., C.A.M. and A.K. wrote the manuscript. All authors read, edited and approved the final manuscript.

Funding

This work was primarily supported by US National Institutes of Health (NIH) [grant R35 GM128765 to A.K.]; supported in part by MSU start-up funds (to A.K.); National Institutes of Health [F32 F32GM134595 to C.A.M.]; and MSU Engineering Distinguished Fellowship (to A.Y.).

Availability of data and materials

The data used in this study, including the geneset-collections and networks, are freely available on Zenodo at https://zenodo.org/record/3352348. We note that KEGG and InBioMap data are available only from the original sources due to restrictive licenses. A GitHub repository, GenePlexus, that contains

code to reproduce the results in this study as well as add new gene classification methods is available at https://github.com/krishnanlab/GenePlexus. *Conflict of Interest*: none declared.

References

Alshahrani,M. and Hoehndorf,R. (2018) Semantic Disease Gene Embeddings (SmuDGE): phenotype-based disease gene prioritization without phenotypes. *Bioinformatics*, **34**, i901–i907.

Altschul,S.F. *et al*. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.

Ashburner,M. *et al*. (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet*., **25**, 25–29.

Ata,S.K. *et al*. (2018) Integrating node embeddings and biological annotations for genes to predict disease-gene associations. *BMC Syst. Biol*., **12**, 138.

Athar,A. *et al*. (2019) ArrayExpress update – from bulk to single-cell expression data. *Nucleic Acids Res*., **47**, D711–D715.

Bai,J. *et al*. (2019) HiWalk: learning node embeddings from heterogeneous networks. *Inf. Syst*., **81**, 82–91.

Barutcuoglu,Z. *et al*. (2006) Hierarchical multi-label prediction of gene function. *Bioinformatics*, **22**, 830–836.

Benjamini,Y. *et al*. (2006) Adaptive linear step-up procedures that control the false discovery rate. *Biometrika*, **93**, 491–507.

Bernardes,J.S. and Pedreira,C.E. (2013) A review of protein function prediction under machine learning perspective. *Recent Pat. Biotechnol*., **7**, 122–141.

Brown,G.R. *et al*. (2015) Gene: a gene-centered information resource at NCBI. *Nucleic Acids Res*., **43**, D36–D42.

Buniello,A. *et al*. (2019) The NHGRI-EBI GWAS Catalog of published genome-wide association studies, targeted arrays and summary statistics 2019. *Nucleic Acids Res*., **47**, D1005–D1012.

Cáceres,J.J. and Paccanaro,A. (2019) Disease gene prediction for molecularly uncharacterized diseases. *PLoS Comput. Biol*., **15**, e1007078.

Cai,H. *et al*. (2018) A comprehensive survey of graph embedding: problems, techniques and applications. *IEEE Trans Knowl Data Eng*., **30**, 1616–1637.

Cho,H. *et al*. (2016) Compact integration of multi-network topology for functional analysis of genes. *Cell Syst*., **3**, 540–548.

Choobdar,S. *et al*. (2019) Open community challenge reveals molecular network modules with key roles in diseases. *bioRxiv, 265553*.

Cowen,L. *et al*. (2017) Network propagation: a universal amplifier of genetic associations. *Nat. Rev. Genet*., **18**, 551–562.

Cui,P. *et al*. (2018) A survey on network embedding. *IEEE Trans. Knowl. Data Eng*., **31**, 833–852.

Davis,J. and Goadrich,M. (2006) The relationship between precision-recall and ROC curve. In: *ICML'06: Proceedings of the 23rd International Conference on Machine Learning*. pp. 233–240. ACM, New York, NY, USA.

Deng,M. *et al*. (2004) An integrated probabilistic model for functional prediction of proteins. *J. Comput. Biol*., **11**, 463–475.

Edgar,R. *et al*. (2002) Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res*., **30**, 207–210.

Gillis,J. and Pavlidis,P. (2011) The impact of multifunctional genes on 'guilt by association' analysis. *PLoS One*, **6**, e17258.

Gligorijević,V. *et al*. (2018) deepNF: deep network fusion for protein function prediction. *Bioinformatics*, **34**, 3873–3881.

Goyal,P. and Ferrara,E. (2018) Graph embedding techniques, applications, and performance: a survey. *Knowl.-Based Syst*., **151**, 78–94.

Greene,C.S. *et al*. (2015) Understanding multicellular function and disease with human tissue-specific networks. *Nat. Genet*., **47**, 569–576.

Grover,A. and Leskovec,J. (2016) nodvec: scalable feature learning for networks. In: *KDD'16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 855–864. ACM Press, San Francisco, CA, USA.

Guan,Y. *et al*. (2010) Functional genomics complements quantitative genetics in identifying disease-gene associations. *PLoS Comput. Biol*., **6**, e1000991.

Hamilton,W.L. *et al*. (2017) Representation learning on graphs: methods and applications. *IEEE Data Engineering Bulletin*.

Huang,J.K. *et al*. (2018) Systematic evaluation of molecular networks for discovery of disease genes. *Cell Syst*., **6**, 484–495.

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

Jiang,Y. *et al*. (2016) An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol*., **17**, 184.

Kahanda,I. *et al*. (2015) A close look at protein function prediction evaluation protocols. *Gigascience*, **4**, 41.

Kanehisa,M. and Goto,S. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res*., **28**, 27–30.

Kanehisa,M. *et al*. (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res*., **45**, D353–D361.

Kanehisa,M. *et al*. (2019) New approach for understanding genome variations in KEGG. *Nucleic Acids Res*., **47**, D590–D595.

Karaoz,U. *et al*. (2004) Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc. Natl. Acad. Sci. USA*, **101**, 2888–2893.

Köhler,S. *et al*. (2008) Walking the interactome for prioritization of candidate disease genes. *Am. J. Hum. Genet*., **82**, 949–958.

Komurov,K. *et al*. (2010) Use of data-biased random walks on graphs for the retrieval of context-specific networks from genomic data. *PLoS Comput. Biol*., **6**, e1000889.

Krishnan,A. *et al*. (2016) Genome-wide prediction and functional characterization of the genetic basis of autism spectrum disorder. *Nat. Neurosci*., **19**, 1454–1462.

Lanckriet,G.R.G. *et al*. (2004) Kernel-based data fusion and its application to protein function prediction in yeast. *Pac. Symp. Biocomput*., **9**, 300–311.

Lee,Y. *et al*. (2013) Ontology-aware classification of tissue and cell-type signals in gene expression profiles across platforms and technologies. *Bioinformatics*, **29**, 3036–3044.

Lee,Y. *et al*. (2019) A computational framework for genome-wide characterization of the human disease landscape. *Cell Syst*., **8**, 152–162.

Leinonen,R. *et al*.; on behalf of the International Nucleotide Sequence Database Collaboration. (2011) The sequence read archive. *Nucleic Acids Res*., **39**, D19–D21.

Leiserson,M.D.M. *et al*. (2015) Pan-cancer network analysis identifies combinations of rare somatic mutations across pathways and protein complexes. *Nat. Genet*., **47**, 106–114.

Leone,M. and Pagnani,A. (2005) Predicting protein functions with message passing algorithms. *Bioinformatics*, **21**, 239–247.

Li,T. *et al*. (2017) A scored human protein-protein interaction network to catalyze genomic interpretation. *Nat. Methods*, **14**, 61–64.

Li,X. *et al*. (2019a) Network-based methods for predicting essential genes or proteins: a survey. *Brief. Bioinform*. doi: 10.1093/bib/bbz017.

Li,Y. *et al*. (2019b) PGCN: disease gene prioritization by disease and gene embedding through graph convolutional neural networks. *bioRxiv, 532226*.

Liberzon,A. *et al*. (2011) Molecular signatures database (MSigDB) 3.0. *Bioinformatics*, **27**, 1739–1740.

Mikolov,T. *et al*. (2013a) Distributed representations of words and phrases and their compositionality. In: *NIPS'13: Proceedings of the 26th International Conference on Neural Information Processing Systems*, Vol. 2. pp. 3111–3119. Curran Associates Inc., USA.

Mikolov,T. *et al*. (2013b) Efficient estimation of word representations in vector space. *ArXiv13013781 Cs*.

Mostafavi,S. *et al*. (2008) GeneMANIA: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biol*., **9**, S4.

Murali,T.M. *et al*. (2011) Network-based prediction and analysis of HIV dependency factors. *PLoS Comput. Biol*., **7**, e1002164.

Nabieva,E. *et al*. (2005) Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, **21**, i302–i310.

Nelson,W. *et al*. (2019) To embed or not: network embedding as a paradigm in computational biology. *Front. Genet*, **10**, 381.

Page,L. *et al*. (1999) *The PageRank Citation Ranking: Bringing Order to the Web*. Available at: http://ilpubs.stanford.edu:8090/422/ (July 2019, date last accessed).

Park,C.Y. *et al*. (2013) Functional knowledge transfer for high-accuracy prediction of under-studied biological Processes. *PLoS Comput. Biol*., **9**, e1002957.

Pedregosa,F. *et al*. (2011) Scikit-learn: machine learning in Python. *J. Mach. Learn. Res*., **12**, 2825–2830.

Peña-Castillo,L. *et al*. (2008) A critical assessment of *Mus musculusgene* function prediction using integrated genomic evidence. *Genome Biol*., **9**, S2.

Perozzi,B. *et al*. (2014) DeepWalk: online learning of social representation. In: *KDD'14: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 701–710. ACM, New York, NY, USA.

Picart-Armada,S. *et al*. (2019) Benchmarking network propagation methods for disease gene identification. *PLoS Comput. Biol*., **15**, e1007276.

Piñero,J. *et al*. (2015) DisGeNET: a discovery platform for the dynamical exploration of human diseases and their genes. *Database*, **2015**, bav028.

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

Piro,R.M. and Cunto,F.D. (2012) Computational approaches to disease-gene prediction: rationale, classification and successes. *FEBS J*., **279**, 678–696.

Radivojac,P. *et al*. (2013) A large-scale evaluation of computational protein function prediction. *Nat. Methods*, **10**, 221–227.

Saito,T. and Rehmsmeier,M. (2015) The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLoS One*, **10**, e0118432.

Schriml,L.M. *et al*. (2012) Disease Ontology: a backbone for disease semantic integration. *Nucleic Acids Res*., **40**, D940–D946.

Schwikowski,B. *et al*. (2000) A network of protein–protein interactions in yeast. *Nat. Biotechnol*., **18**, 1257–1261.

Sharan,R. *et al*. (2007) Network-based prediction of protein function. *Mol. Syst. Biol*., **3**, 88.

Sleator,R.D. and Walsh,P. (2010) An overview of in silico protein function prediction. *Arch. Microbiol*., **192**, 151–155.

Smith,C.L. *et al*. (2004) The Mammalian Phenotype Ontology as a tool for annotating, analyzing and comparing phenotypic information. *Genome Biol*., **6**, R7.

Smith,C.L. *et al*.; the Mouse Genome Database Group. (2018) Mouse Genome Database (MGD)-2018: knowledgebase for the laboratory mouse. *Nucleic Acids Res*., **46**, D836–D842.

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

Subramanian,A. *et al*. (2005) Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA*, **102**, 15545–15550.

Szklarczyk,D. *et al*. (2015) STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res*., **43**, D447–D452.

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

Tsuda,K. *et al*. (2005) Fast protein classification with multiple networks. *Bioinformatics*, **21**, ii59–ii65.

Valdeolivas,A. *et al*. (2019) Random walk with restart on multiplex and heterogeneous biological networks. *Bioinformatics*, **35**, 497–505.

Vanunu,O. *et al*. (2010) Associating genes and protein complexes with disease via network propagation. *PLoS Comput. Biol*., **6**, e1000641.

Vazquez,A. *et al*. (2003) Global protein function prediction from protein-protein interaction networks. *Nat. Biotechnol*., **21**, 697–700.

Wang,D. *et al*. (2016) Structural deep network embedding. In: *KDD'16: Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 1225–1234. ACM, New York, NY, USA.

Wang,X. *et al*. (2011) Network-based methods for human disease gene prediction. *Brief. Funct. Genomics*, **10**, 280–293.

Warde-Farley,D. *et al*. (2010) The GeneMANIA prediction server: biological network integration for gene prioritization and predicting gene function. *Nucleic Acids Res*., **38**, W214–W220.

Whisstock,J.C. and Lesk,A.M. (2003) Prediction of protein function from protein sequence and structure. *Q. Rev. Biophys*., **36**, 307–340.

Wu,C. *et al*. (2013) BioGPS and MyGene.info: organizing online, gene-centric information. *Nucleic Acids Res*., **41**, D561–D565.

Xin,J. *et al*. (2016) High-performance web services for querying gene and variant annotation. *Genome Biol*., **17**, 91.

Yang,J. *et al*. (2011) GCTA: a tool for genome-wide complex trait analysis. *Am. J. Hum. Genet*., **88**, 76–82.

Yang,K. *et al*. (2018) HerGePred: heterogeneous network embedding representation for disease gene prediction. *IEEE J. Biomed. Health Inform*., **23**, 1805–1815.

Youngs,N. *et al*. (2013) Parametric Bayesian priors and better choice of negative examples improve protein function prediction. *Bioinformatics*, **29**, 1190–1198.

Youngs,N. *et al*. (2014) Negative example selection for protein function prediction: the NoGO database. *PLoS Comput. Biol*., **10**, e1003644.

Zhang,X. *et al*. (2016) Predicting essential genes and proteins based on machine learning and network topological features: a comprehensive review. *Front. Physiol*., **7**, 75.

Zhao,B. *et al*. (2019) An iteration method for identifying yeast essential proteins from heterogeneous network. *BMC Bioinformatics*, **20**, 355.

Zhou,D. *et al*. (2003) Learning with local and global consistency. In: *NIPS'03: Proceedings of the 16th International Conference on Neural Information Processing Systems*. pp. 321–328. MIT Press, Cambridge, MA, USA.

Zhu,X. *et al*. (2003) Semi-supervised learning using Gaussian fields and harmonic functions. In: *ICML'03: Proceedings of the Twentieth International Conference on International Conference on Machine Learning, Washington DC*. pp. 912–919. AAAI Press.

Zitnik,M. and Leskovec,J. (2017) Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, **33**, i190–i198.

Supervised learning is an accurate method for network-based gene classification

Supplemental Material

Section 1: Methods and Data

Section 1.1: Networks

The networks used in this study are BioGRID, STRING-EXP, InBioMap, GIANT-TN, and STRING. Detailed information about the network properties and sources can be seen in Table S1, with the network construction method and interaction type information coming from (Huang *et al.*, 2018). BioGRID (version 3.4.136) is a low-throughput network that includes both genetic interactions, as well as physical protein-protein interactions (Stark *et al.*, 2006). InBioMap (version 2016_09_12) is a high-throughput, scored network that contains physical protein-protein interactions as well as pathway database annotations incorporated as edges (Li *et al.*, 2017). We used the “final-scores” as the edge weights. STRING (version 10.0) is a high-throughput, scored network that aggregates information from many data sources (Szklarczyk *et al.*, 2015). We used two different STRING networks. First, we used the “combined” network that directly includes database annotations, text-mining, ortholog information, co-expression, and physical protein interactions (referred to as “STRING” in this study). We also used a subset of edges in STRING that had just the “experiments” data, thus restricting the network to one constructed just from physical protein interactions in humans (referred to as “STRING-EXP” in this study). For both networks, we used the corresponding relationship scores as edge weights, after normalizing them to lie between 0 and 1. The GIANT-TN (version 1.0) network is the tissue-naïve network from GIANT (Greene *et al.*, 2015), referred to as the “Global” network on the website, and is constructed from both low- and high-throughput data, and includes information from co-expression, non-protein sources, regulatory data, and physical protein-protein interactions. The GIANT-TN network is a fully connected, scored network. To add sparsity to the GIANT-TN network, we removed all edges with scores below 0.01 (equal to the prior the Bayesian model used to construct the network). It is worth noting here that the purpose of this study is not to compare networks against each other, but rather to determine the performance of SL methods vs LP methods on various types of networks.

Table S1. Information on the molecular networks. LT : low-throughput, HT : high-throughput, G : genetic, P : physical, DA : database annotations, CE : co-expression, NP : non-protein, R : regulation, CC : co-citation, O : orthologous.

Network	Number of Genes	Number of Edges	Edge Density	Network Construction Method	Weighted	Interaction Type
BioGRID	20,558	238,474	1.13e-3	LT	No	G, P
STRING-EXP	14,089	141,629	7.08e-4	HT	Yes	P
InBioMap	17,399	644,862	1.58e-3	HT	Yes	P, DA
GIANT-TN	25,689	38,904,929	1.92e-3	LT, HT	Yes	CE, NP, P, R
STRING	17,352	3,640,737	7.20e-3	HT	Yes	CC, CE, O, DA, P

Section 1.2: Model Selection and Hyperparameter Tuning

The restart hyperparameter α used in generating an influence matrix was determined by doing a grid search over values between 0.55 and 0.95 in 0.1 steps for all networks and geneset collections, optimizing for auPRC using label propagation (Fig. S1). In general, there was not a strong dependence on α. It can be seen in Figure S1 that a higher restart probability resulted in marginally better performance for the larger networks (STRING and GLOBAL), whereas as a smaller restart probability led to nominally better performance for the smaller networks such as BioGRID. In this study, we used $\alpha = 0.85$ for every geneset-collection–network combination, as $\alpha = 0.85$ offered good performance and had low variance. This α was used for both LP-I and SL-I. We stress that the tuning of α was never done for SL-I, and thus, our finding that SL methods generally outperform LP methods is not biased by this parameter tuning.

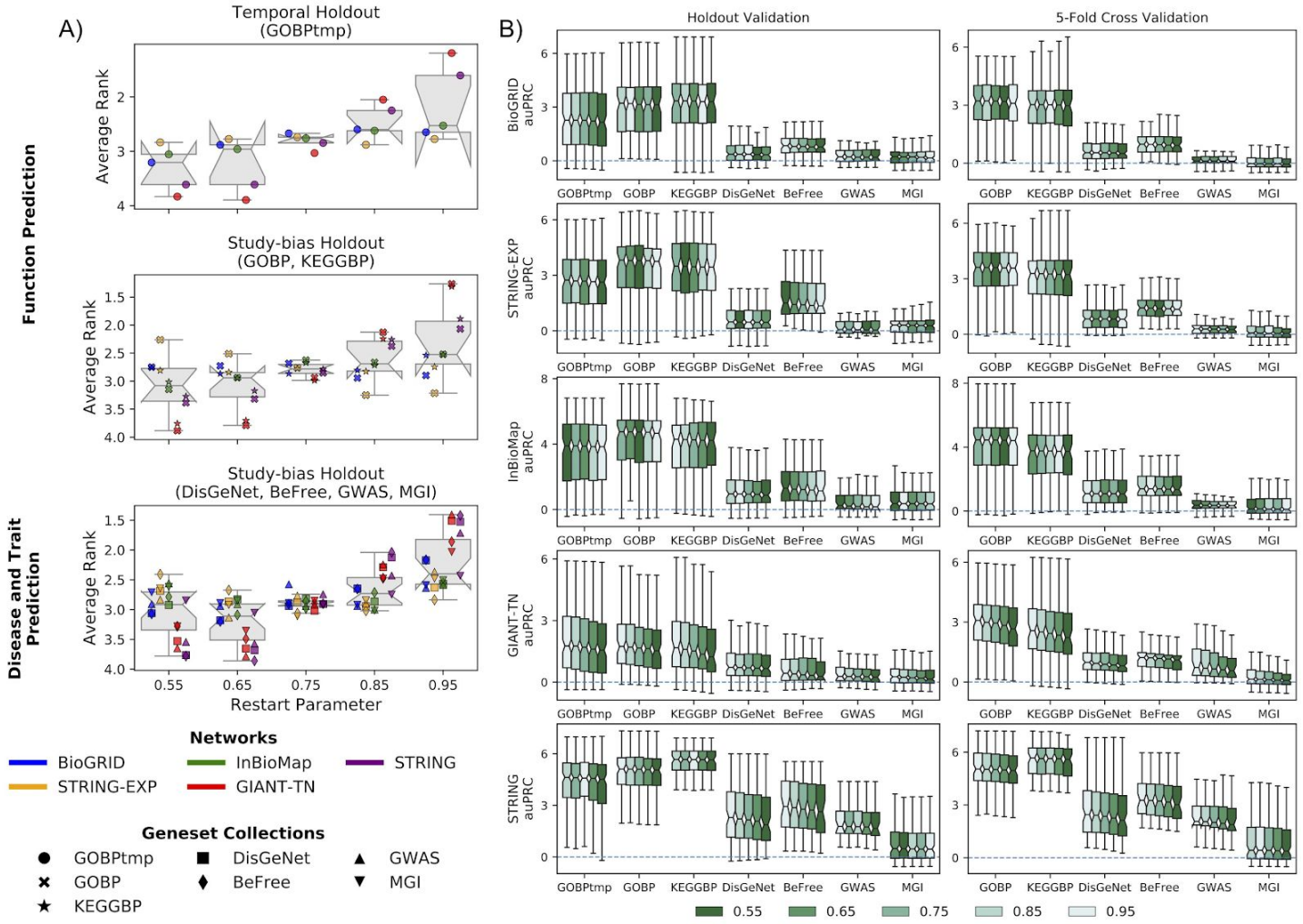

Fig. S1. Tuning the restart probability hyperparameter for label propagation. A) Each point in each boxplot represents the average rank for a geneset-collection–network combination, where the five restart probabilities that were tried were ranked in terms of performance (auPRC) for each geneset in a geneset-collection using the standard competition ranking. A restart probability of 0.85 was chosen for this study as it resulted in good overall performance as well as low variance in performance for the different genset-collection–network combinations. B) The performance for each individual geneset-collection–network combination is compared across the five restart probabilities: 0.55, 0.65, 0.75, 0.85 and 0.95. The methods are ranked by median value of auRPC with the highest scoring method on the left. There is no strong dependence of auRPC on the restart probability.

Model selection of the supervised learning classifier was done by comparing six popular classifiers that are implemented in Python package *Scikit Learn* (Pedregosa *et al.*, 2011). To determine the best supervised learning classifier, we compared their performance over every geneset-collection–network combination using their default hyperparameters in version 0.19 of Scikit Learn (Fig. S2). Logistic regression with L2 regularization is marginally better than linear support vector machines and both these classifiers outperform random forest, logistic regression with L1 regularization, and support vector machines using the radial basis kernel and the 2nd order polynomial kernel. We note that the non-linear SVMs (radial basis and polynomial kernels) took over two orders-of-magnitude longer to train, and thus, those models are not included in the five-fold cross validation results.

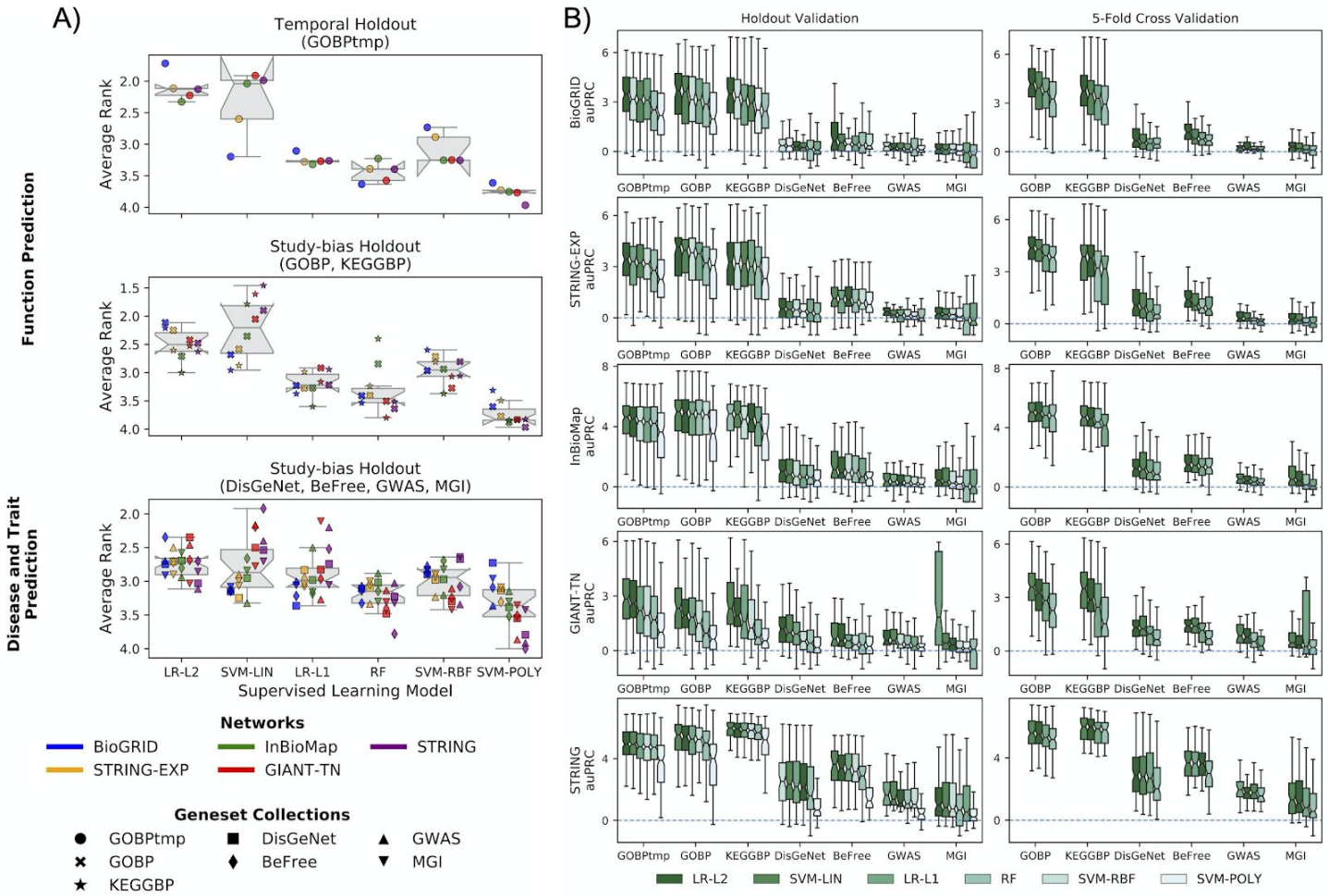

Fig. S2. Comparison of classifiers for supervised learning. A) Each point in each boxplot represents the average rank for a geneset-collection–network combination, where the four classifiers were ranked by the auPRC for each geneset in a geneset-collection using the standard competition ranking. Logistic regression with L2 regularization (LR-L2) was chosen as the classifier for supervised learning as it had slightly better overall performance than a linear support vector machine (SVM). B) The auRPC for each individual geneset-collection–network combination is compared across six supervised learning classifiers: logistic-regression with L1 regularization (LR-L1), LR-L2, SVM models with three kernels (linear; SVM-LIN, radial basis function; SVM-RBF, 2nd order polynomial; SVM-POLY) and a random forest (RF). The classifiers are ranked by median value with the best performing one on the left.

For the model selection of the embedding technique, we chose *node2vec* (Grover and Leskovec, 2016) because its competitive performance and ease of use (Goyal and Ferrara, 2018). The following hyperparameters were tuned based on the aggregated performance across all geneset-collections–network combinations: *p* - the breadth first search parameter, *q* - the depth first search parameter, *d* - embedding size, *l*

- walk length, and *k* - context window size. We left *r* (number of walks per node) at it's default value. Since *p* and *q* are coupled, we performed a grid search for these two parameters leaving all others constant. Each of the other hyperparameters was tuned by leaving the rest at their default values as described in the original *node2vec* publication. The values for the hyperparameters were tuned over are; *p, q* - [0.1. 0.5, 1, 5, 10], *d* - [64, 128, 256, 512, 1024, 2048], *l* - [20, 40, 60, 80, 100, 120, 140, 160, 180, 200], and *k* - [2, 4, 8, 16, 32, 64]. We found that in general there was a large range of values for each parameter where the results were near optimal, and we chose – $p = 0.1$, $q = 0.1$, $d = 512$, $k = 8$, $l = 120$, and $r = 10$ – for every geneset-collection–network combination.

Section 1.3: Geneset-collections

The geneset-collections used in this work are from the Gene Ontology (from version 2 of MyGene.info API with data retrieved on 2018-05-18, GOBPtmp, GOBP) (The Gene Ontology Consortium, 2019; Ashburner *et al.*, 2000; Wu *et al.*, 2013; Xin *et al.*, 2016), Kyoto Encyclopedia of Genes and Genomes (from version 6.1 of MSigDB, KEGGBP) (Kanehisa and Goto, 2000; Kanehisa *et al.*, 2017, 2019), DisGeNet (version 5.0, DisGeNet, BeFree) (Piñero *et al.*, 2017, 2015), GWAS from a community challenge at https://www.synapse.org/#!Synapse:syn11944948 (Choobdar *et al.*, 2019), and the Mouse Gene Informatics database (data retrieved on 2018-10-01, MGI) (Smith *et al.*, 2018).

Pre-processing genesets based on specificity, redundancy, and multi-functionality

Each of these six geneset-collections contained anywhere from about a hundred to tens of thousands of genesets (Table S2) that varied widely in specificity and redundancy. The first pre-processing step we did after downloading the data was to convert the original gene/protein IDs to entrez gene IDs, which was done using gene ID conversions found in MyGene.info (Wu *et al.*, 2013; Xin *et al.*, 2016). If the original ID mapped to more than one entrez ID, all of them were included for further analysis. Next, whenever applicable, annotations to genesets corresponding to terms in a curated ontology were propagated along the *is_a* and *part_of* relationships to ancestor terms in the corresponding ontologies: Gene Ontology (Ashburner *et al.*, 2000) for GOBP, Disease Ontology (Schriml *et al.*, 2019) for DisGeNet and BeFree, and Mammalian Phenotype Ontology (Smith and Eppig, 2009) for MGI.

Subsequent preprocessing steps were designed to ensure that the final set of genesets from each source are specific, largely non-overlapping, and not driven by multi-attribute genes.

Specificity: To select specific biologically-meaningful genesets in each collection, we sorted all the genesets in a collection from the largest to smallest based on the number of annotated genes (geneset size), manually examined their descriptions, and chose a size threshold that roughly separated large, generic genesets from the smaller, specific ones. This threshold was 200 for GOBP and KEGGBP, 300 for MGI, 400 for BeFree, 500 for GWAS and GOBPtmp, and 600 for DisGeNet.

Redundancy: To remove redundant genesets within a collection, first, we calculated the Jaccard index ($|A \cap B| / |A \cup B|$) and the overlap index ($|A \cap B| / min(|A|, |B|)$) between all pairs of genesets (with A and B representing the sets of genes annotated to the genesets). Then, we built a graph with the genesets as the nodes, and added edges between genesets pairs if their Jaccard index was >0.5 *and* their overlap index was >0.7. The geneset graph constructed in this manner contained many connected components, each representing a set of highly overlapping genesets. Finally, we used the following procedure to pick representative genesets within each component: a) calculate a score for each geneset equal to the sum of the proportions of genes in other linked genesets that are contained within it (higher this score, more representative that geneset is), b) create a sorted list of all the genesets in decreasing order of this score, and c) pick the first geneset in the list, remove every subsequent geneset that is connected to it in the graph, and

repeat this step until the sorted list is empty. This procedure resulted in a reasonable number of non-redundant genesets within each collection. The same Jaccard and overlap thresholds were used for all collections except MGI. For MGI, since an overlap cutoff of 0.7 still resulted in thousands of genesets, it was lowered to 0.5.
Multi-attribute genes: Given the set of largely non-overlapping genesets in a collection, individual genes were removed from all genesets if they appeared in more than 10 genesets in that collection. This step ensures that the evaluations are not biased by multi-attribute genes that can potentially be easily predicted in a non-specific manner (Gillis and Pavlidis, 2011).

We also note that we did not include the cellular component (CC) or molecular function (MF) classes of the gene ontology as part of the function classification tasks because two genes that are annotated to the same CC or MF need not be related to each other functionally.

Table S2. Information on the geneset-collections. The last four columns reflect the fact each geneset-collection is slightly different for every network and these values are presented as either a range, a median value, or number of genes in a union across all networks used in this study.

Geneset Collection	Number of Genesets From Original Data	Number of Genesets After Redundant Genesets Removed	Number of Genesets After Holdout Preprocessing	Geneset Sizes	Median Geneset Size	Number of Genes from Union of all Genesets
GOBPtmp	11,574 to 754*	166	(115, 160)	(27, 452)	174	9464
GOBP	11,574	313	(84, 96)	(20, 181)	76	5301
KEGGBP	149	138	(63, 74)	(24, 181)	51	3454
DisGeNet	4030	334	(89, 104)	(21, 368)	67	4689
BeFree	2891	207	(49, 57)	(20, 223)	80	2692
GWAS	169	74	(30, 37)	(24, 431)	94	2134
MGI	10,264	492	(90, 121)	(20, 132)	41	2716

***** The GOBP temporal holdout step had an extra initial preprocessing step to make sure there were at least ten genes in the training and testing sets.

Calculating the network properties of the genesets
To determine how the performance of a given geneset depends on the network, for each geneset we calculated three different properties:

1) For a given geneset, T, the number of genes annotated it is given by $|T|$.
2) For a given geneset, T, the edge density, D_T, is given by

$$D_T = \sum_{(u,v)\in T} W_{uv} / (|T| * (|T| - 1)/2), \quad \text{(eqn. S1)}$$

where W_{uv} is the edge weight between genes u and v. The edge density is a measure of how tightly connected the geneset is within itself.
3) For a given geneset, T, the segregation, S_T, is given by

$$S_T = \sum_{(u,v)\in T} W_{uv} / \sum_{u\in T,\, t\in V} W_{ut}. \quad \text{(eqn. S2)}$$

Segregation is a measure of how isolated the geneset is from the rest of the network.

The three geneset properties are shown for all geneset-collection–network combinations in Fig. S3. In general, there is little difference in the number of genes across the different prediction tasks (i.e. function, disease and trait), except for GOBPtmp, which has the largest number of genes due to the fact the genesets need to be larger to have enough with at least 10 testing genes. Edge density and segregation are highest for the function genesets (GOBPtmp, GOBP, KEGGBP) and lowest for the disease and trait genesets (DisGeNet, BeFree, GWAS, MGI).

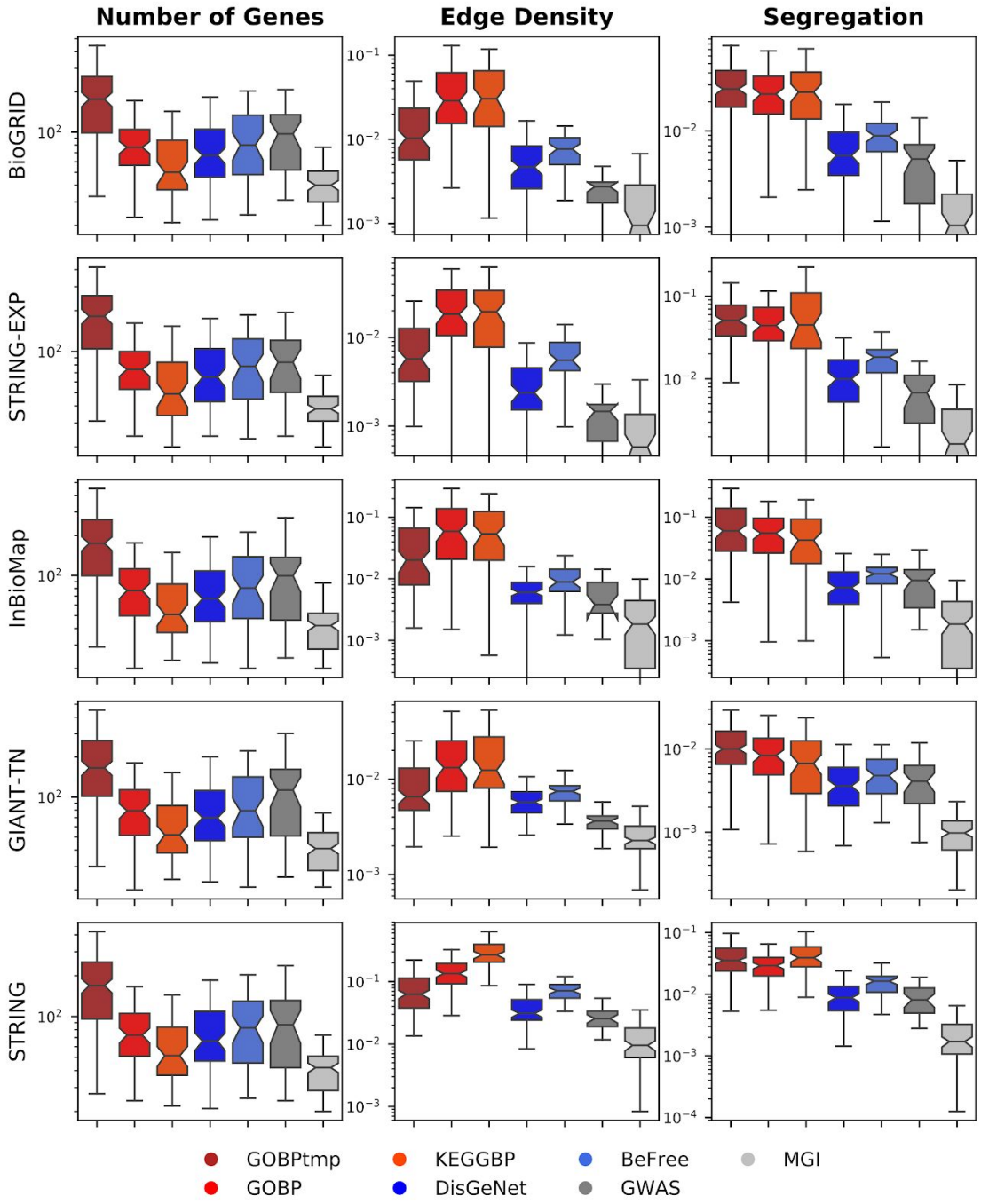

Fig. S3. Network properties for the different geneset-collections. The geneset-collections can be broken up into three prediction tasks; function (GOBPtmp, GOBP, KEGGBP; reds), disease (DiGeNet, BeFree; blues) and trait (GWAS, MGI; greys). In general, there is little difference in the number of genes across the different type prediction tasks (i.e. function, disease and trait), except for GOBPtmp which has the largest number of genes due to the fact the genesets need to be

larger to have enough with at least 10 testing genes. Edge density and segregation are highest for the function genesets (GOBPtmp, GOBP, KEGGBP) and lowest for the disease and trait genesets (DisGeNet, BeFree, GWAS, MGI).

Section 1.4: Validation schemes

We used three different validation schemes to evaluate gene classification.

Temporal holdout validation

Temporal holdout is the most stringent evaluation scheme for gene classification since it mimics the practical scenario of using current knowledge to predict the future. Since Gene Ontology was the only source with clear date-stamps for all its annotations, temporal holdout was applied only to the GOBP geneset-collection. Since the goal of this study is to use relatively recent and widely-used molecular networks, as this would reflect how these models would be deployed in practice, we chose a temporal cutoff point of Jan 1st, 2017. Then, for each geneset-collection, genes that only had an annotation to any geneset in the collection after 2017-01-01 were assigned to the testing set and the remaining genes were assigned to the training set. Since this resulted in the testing set having far fewer genes than the testing set for the other validation schemes, we made the following minor modifications to the geneset pre-processing procedure: GOBP geneset-collection was first filtered to remove any geneset with fewer than ten training genes or had fewer than ten testing genes based on the temporal split and the specificity threshold (maximum number of genes annotated to a geneset) was increased from 200 to 500. Redundancy filtering and multi-attribute gene filtering were unchanged. As noted in Section 1.1, from each network resource considered in this study, we chose the most recent version of the network that was released before 2017-01-01 to ensure no data leakage. Finally, genes were removed from genesets if they were not present in a given network, genesets with fewer than ten training genes or fewer than ten testing genes were filtered out, and the remaining genesets were used to perform the temporal holdout validation.

Study-bias holdout validation

The goal of study-bias validation is to evaluate the scenario that is close to the real-world situation of learning from well-characterized genes to predict novel un(der)-characterized genes. Here, we defined study-bias for each gene as the number of articles in PubMed (http://www.pubmed.gov/) in which that gene was referenced in, as determined in the *gene2pubmed* file (downloaded on 2018-10-30) from the NCBI Gene database (Brown *et al.*, 2015). Using this definition, for each geneset-collection–network combination, we created training-testing splits in the following manner: Genes were removed from genesets if they were not present in the given network. Then, among the remaining genes, a gene was assigned to the training set if it was in the top two-thirds of the list of genes sorted by their PubMed count. The remaining genes were assigned to the testing set. Finally, genesets with fewer than ten training genes or fewer than ten testing genes were filtered out and the remaining genesets were used to perform the study-bias holdout validation.

Five-fold cross-validation

To ensure comparability, we performed 5-fold cross-validation using the same genesets that were used in study-bias holdout, splitting each geneset randomly into five approximately equal folds (with similar proportions of positive and negative examples) and, in rotation, using one fold as the testing set and the remaining four as the training set.

Section 1.5: Evaluation Metrics

In this study, we present results in terms of two popular metrics, auPRC and auROC, as well as the precision of the top *K* ranked predictions (P@TopK). Since, each geneset-collection–network combination has a different number of positive examples (and, hence, different positive:negative proportions), we normalized auPRC and P@topK by the prior. Specifically, auPRC is given by:

$$auPRC = \log_2\left(\frac{auPRC_S}{prior}\right) \quad \text{(eqn. S3)}$$

where $auPRC_S$ is the standard area under the precision-recall curve, and the *prior* is *P*/(*P*+*N*) with *P* being the number of positive ground truth labels, and *N* being the number of negative ground truth labels. The log2 in eqn. S3 allows for the following interpretation: the number of 2-fold increases of the measured $auPRC_S$ over what is expected given the ground truth labels (e.g., a value of 1 indicates a 2-fold increase, a value of 2 indicates a 4-fold increase). Similarly, P@topK is given by:

$$P@topK = \log_2\left(\frac{TP_K}{K \times prior}\right) \quad \text{(eqn. S4)}$$

where K is the number of top-predictions to consider, TP_K is the number true-positives of the top-K predictions, and the *prior* is the same as in eqn. S3. We set K to be the number of ground truth positives in the testing set. P@topK can be thought of as what is the 2-fold increase in the percent of the top-K predictions that were predicted true over the expected value. Of note, it is possible that $TP_k = 0$ if no true positive is captured within the first *K* predictions. This causes P@topK to become $-\infty$. To address this issue, we set such values to be the minimum score obtained across all predictions for that given geneset-collection–network combination.

Precision-based metrics – auPRC and P@topK – are more suitable than the more popular area under the receiver-operating characteristic curve (auROC) for two reasons. First, gene classification is a highly imbalanced problem with many more negative examples than positive examples, and auROC is ill-suited for imbalanced problems (Saito and Rehmsmeier, 2015). Second, precision can control for Type-1 error (false positives) (Davis and Goadrich, 2006). Since the foremost reason for gene classification is to provide a list of candidate genes for further experimental study, it is more important to make sure the top predictions are as correct as possible, as opposed to ensuring that, on average, positive examples are ranked higher than negative examples. However, for completeness, we have provided auROC results in this Supplemental Material (Section 2).

Section 2: Supplementary Results

Section 2.1: Compiled results from all validation schemes in terms of all evaluation metrics

In this section, we present results for the ranking analysis used in Fig. 2, significance test analysis used in Fig. 3, as well as the boxplots representations seen in Fig. 4, based on all evaluation metrics (auPRC, P@TopK, and auROC) as well as all validation schemes (temporal holdout, study-bias holdout and 5FCV) (Figs. S4 - S8). Additionally, in this section, we present the results of how the performance of SL-A and LP-I scale with the number of genes, edge density, and segregation for all networks used in this study (Fig. S9).

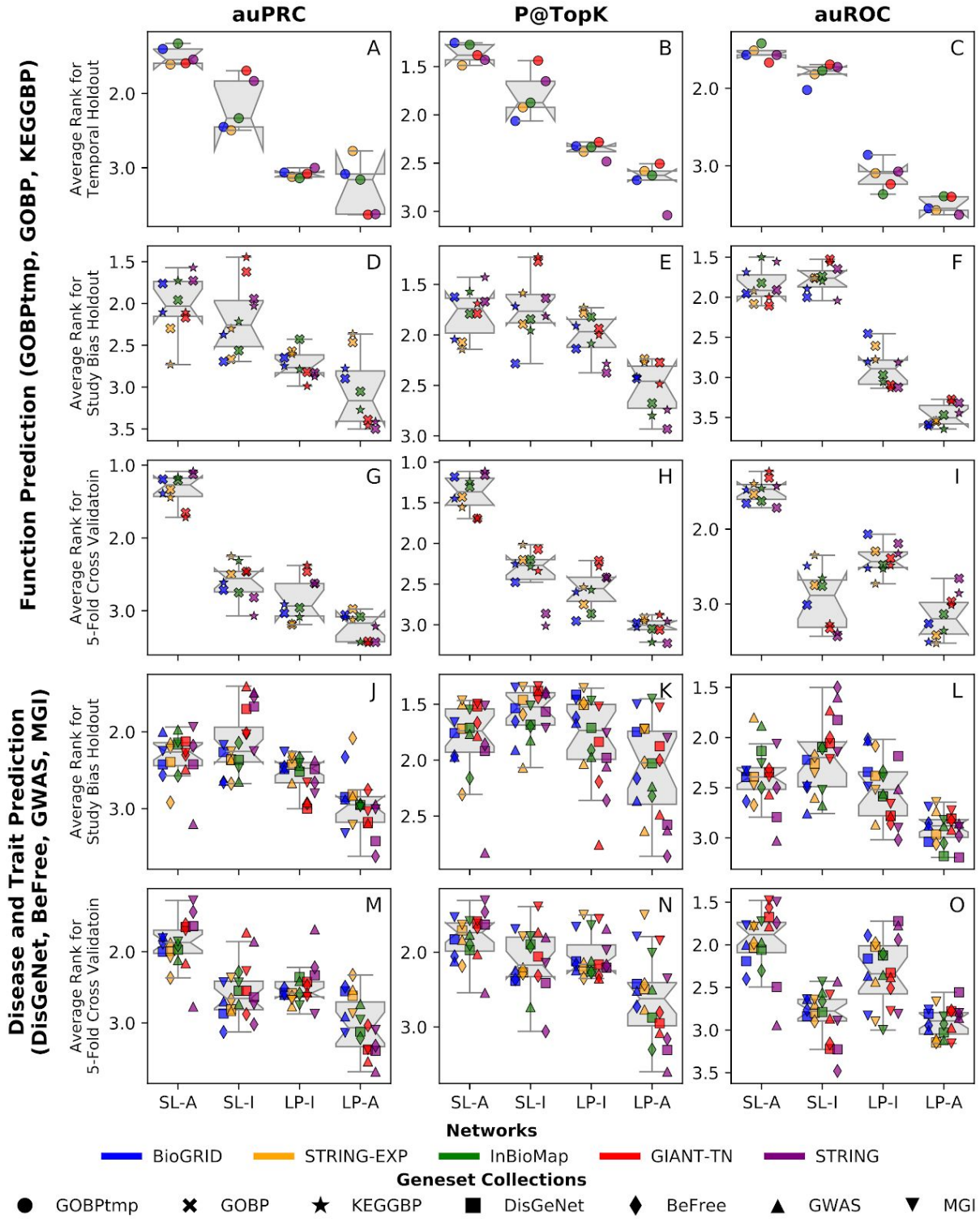

Fig. S4. Average rank of the four methods for all evaluation metrics and validation schemes. Each point in a boxplot represents the average rank for a geneset-collection–network combination, where the four methods were ranked in terms of performance for each geneset in a geneset-collection using the standard competition ranking. Different colors represent different networks and different marker shapes represent different geneset-collections.

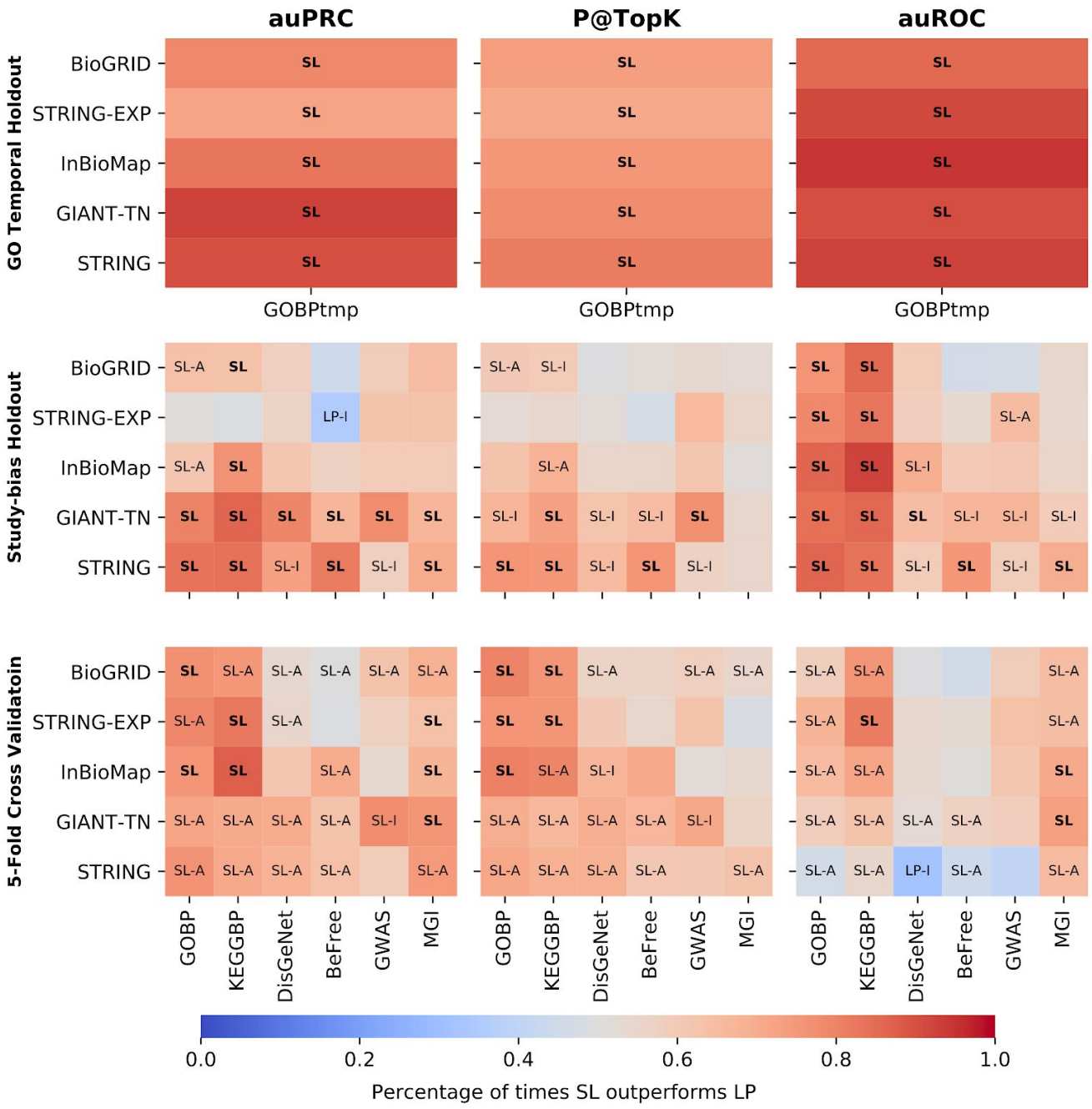

Fig. S5. Testing for a statistically significant difference between SL and LP methods using all evaluation metrics and validation schemes. For each network-geneset combination, each method is compared to the two methods from the other class (i.e. SL-A vs LP-I, SL-A vs LP-A, SL-I vs LP-I, SL-I vs LP-A). If a method was found to be significantly better than both methods from the other class (Wilcoxon ranked-sum test with an FDR threshold of 0.05), the cell is annotated with that method. If both models in that class were found to be significantly better than the two methods in the other class, the cell is annotated in bold with just the class. The color scale represents the fraction of genesets that were higher for the SL methods across all four comparisons. The first column uses GOBP temporal holdout, whereas the remaining 6 columns use study-bias holdout. B) SL methods show a statistically significant improvement over LP methods, especially for function prediction.

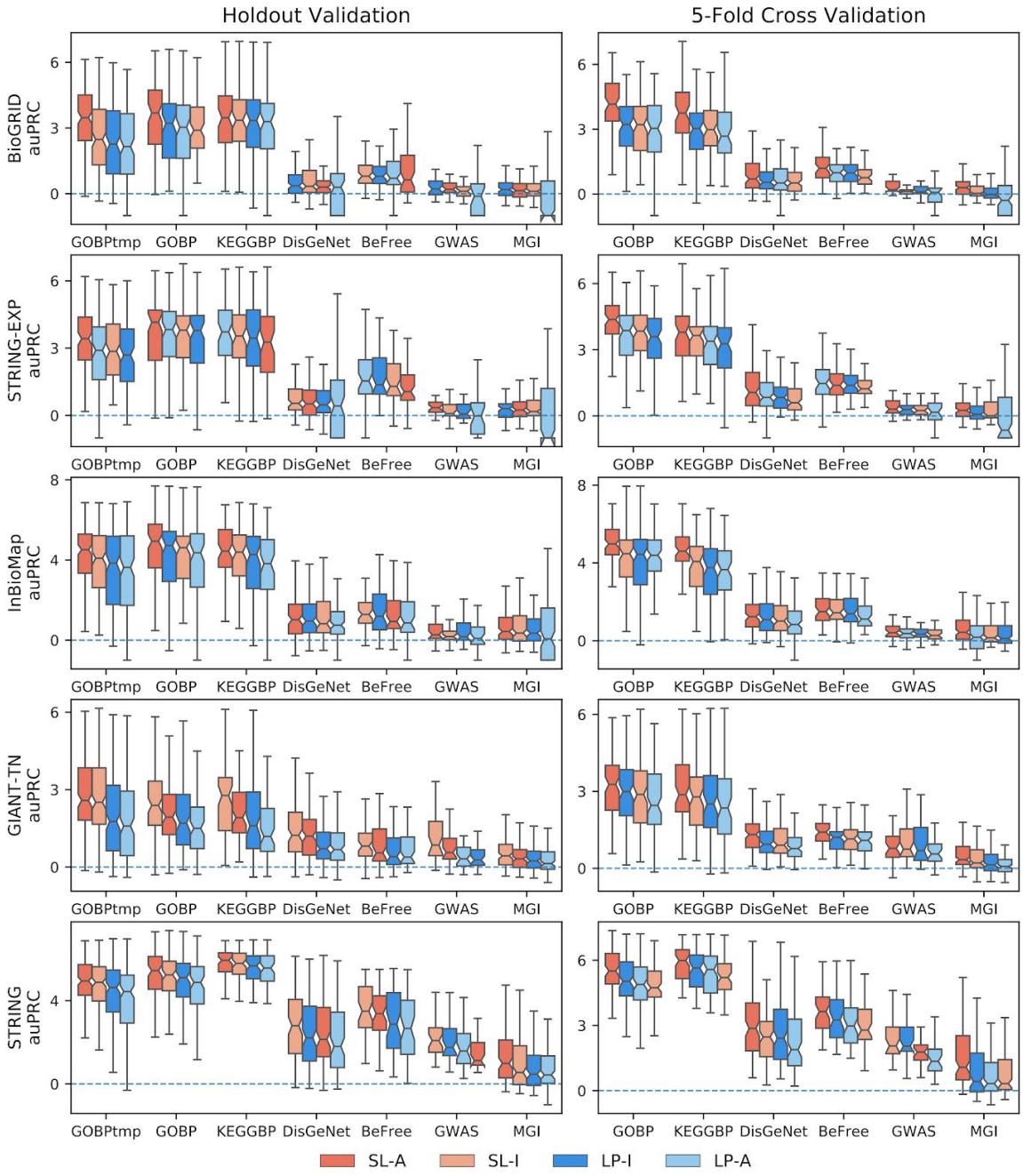

Fig. S6. Boxplots for auPRC performance across all geneset-collection–network combinations. The performance for each individual geneset-collection–network combination is compared across the four methods; SL-A (red), SL-I (light red), LP-I (blue), and LP-A (light blue). The methods are ranked by median value with the highest scoring method on the left. The first column contains temporal and study-bias holdout, and the second column is 5FCV. The scoring metric is auPRC.

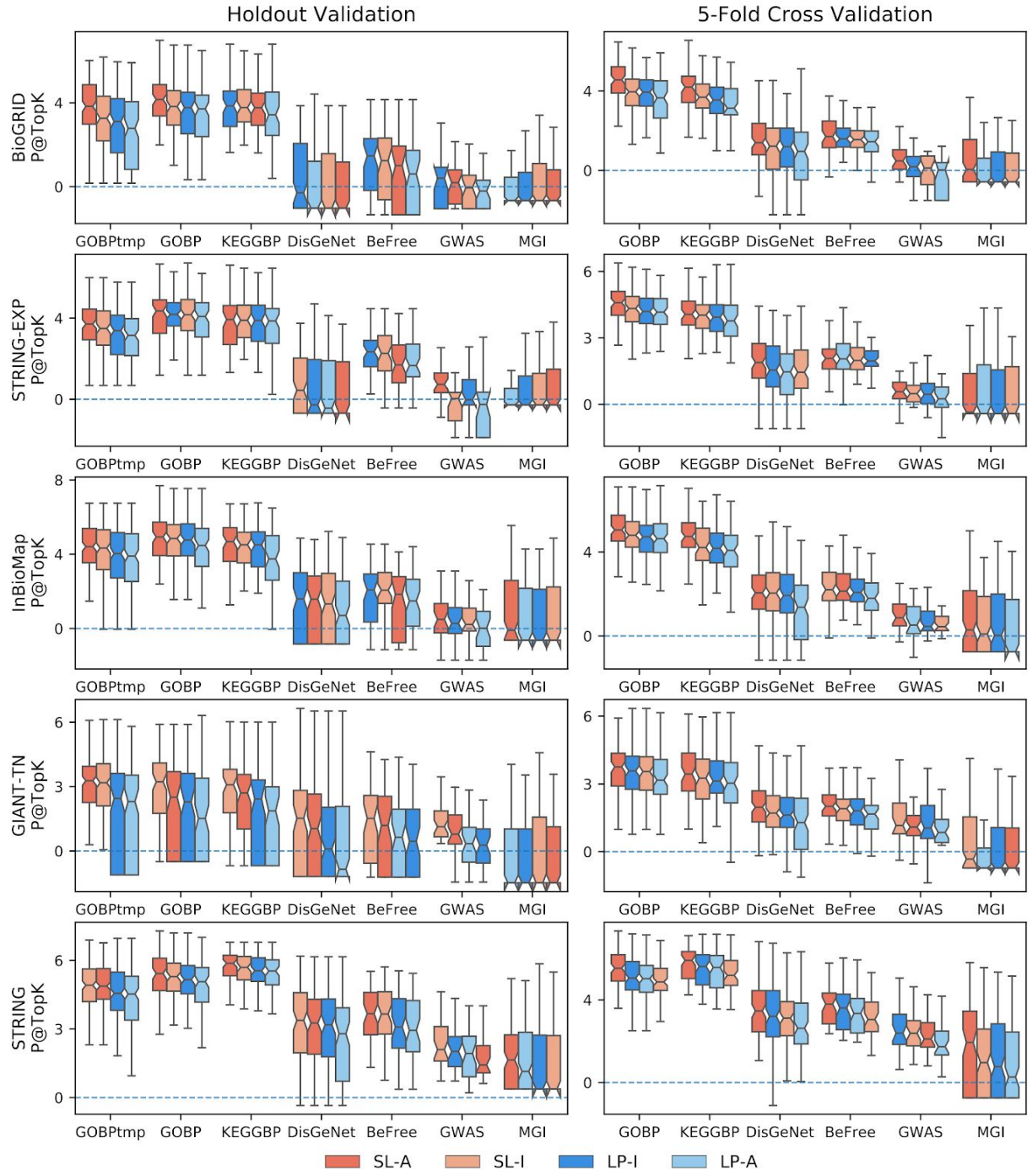

Fig. S7. Boxplots for P@TopK performance across all geneset-collection–network combinations. The performance for each individual geneset-collection–network combination is compared across the four methods; SL-A (red), SL-I (light red), LP-I (blue), and LP-A (light blue). The methods are ranked by median value with the highest scoring method on the left. The first column contains temporal and study-bias holdout, and the second column is 5FCV. The scoring metric is P@TopK.

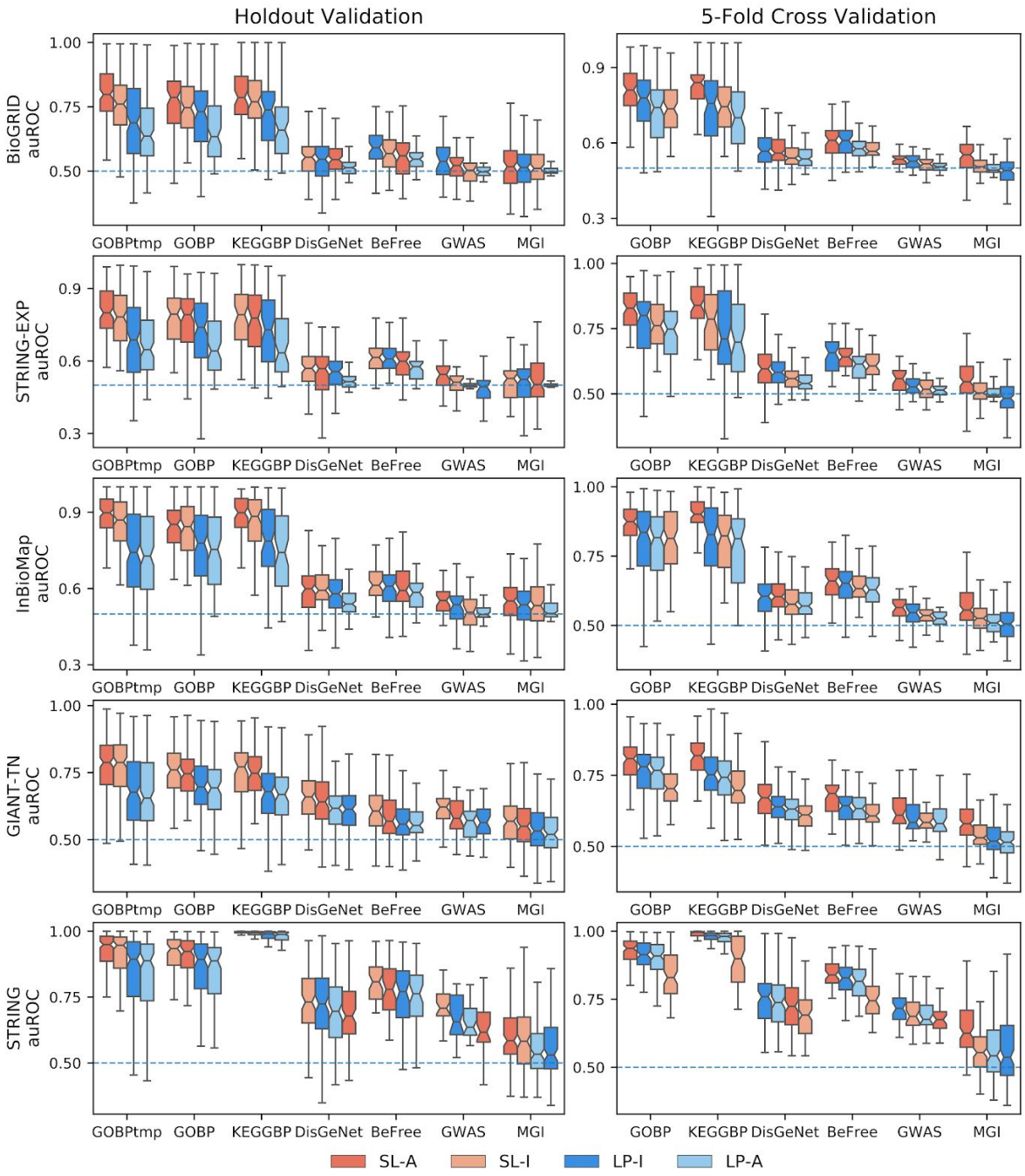

Fig. S8. Boxplots for auROC performance across all geneset-collection–network combinations. The performance for each individual geneset-collection–network combination is compared across the four methods; SL-A (red), SL-I (light red), LP-I (blue), and LP-A (light blue). The methods are ranked by median value with the highest scoring method on the left. The first column contains temporal and study-bias holdout, and the second column is 5FCV. The scoring metric is auROC.

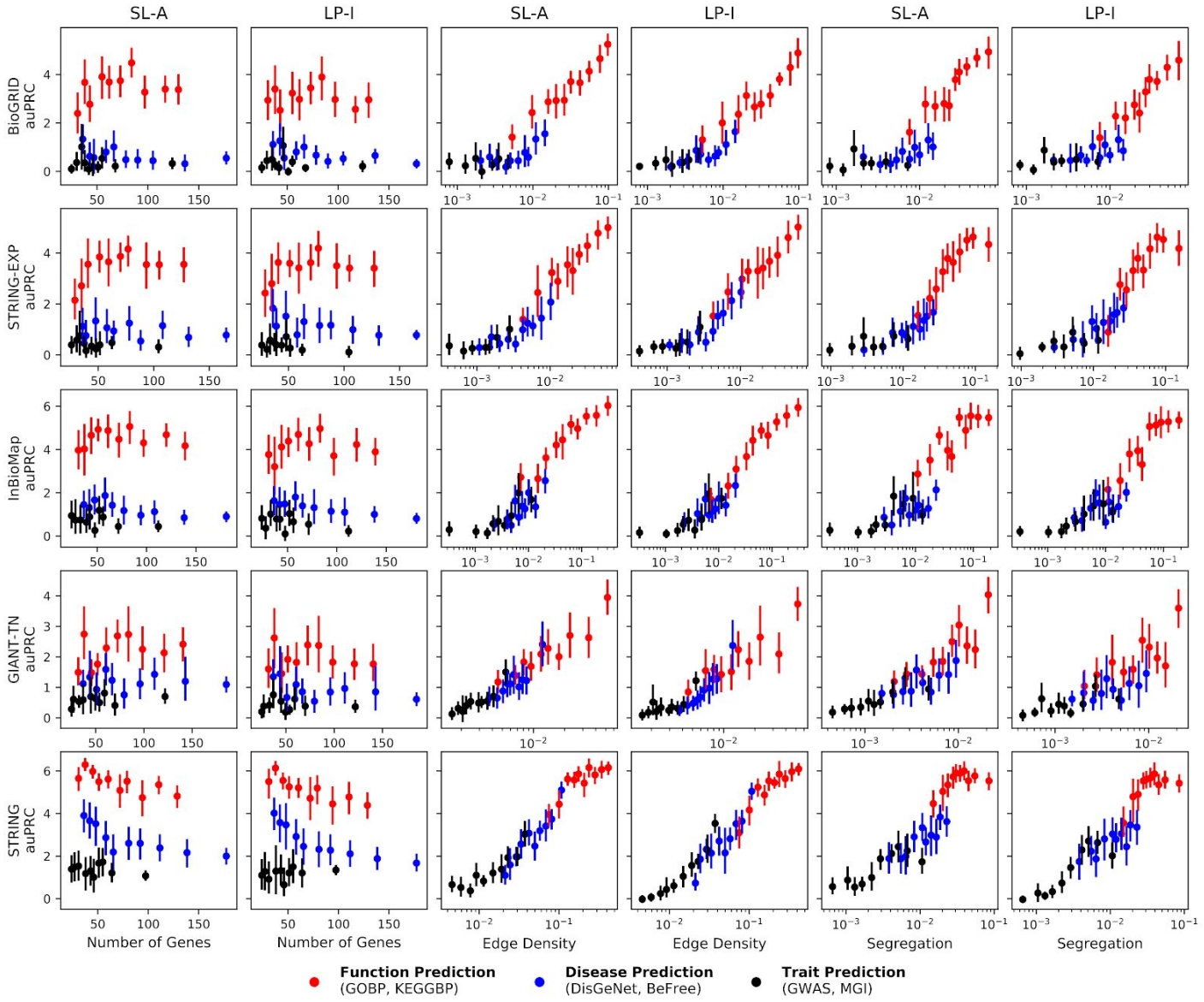

Fig S9. Performance vs Network/Geneset properties for all networks. SL-A is able to capture network information as efficiently as LP-I across all networks. There is no correlation between the number of genes in the geneset versus performance, but there is a strong correlation between the performance and the edge density as well as segregation. The different colored dots represent function genesets (red, GOBP and KEGGBP), disease genesets (blue, DIsGeNet and BeFree), and trait genesets (black, GWAS and MGI). The vertical line is the 95% confidence interval and the performance metric is auPRC.

Section 2.2: Effect Size

In this section, we show results for the effect size between all methods (SL-A, SL-I, LP-I, LP-A). To calculate an effect size, for every geneset we calculate the ratio of auRPC values, find the percent increase/decrease and then take the median value for every geneset-collection–network combination. The results show that SL-A has a significant effect size when compared to LP-I for function prediction (53% for temporal holdout and 19% for Study-bias holdout). Also, for all prediction tasks the effect size seen between the SL methods and LP-I is equal to or greater than the effect size between LP-I and LP-A, where LP-I is widely considered a much better model than LP-A and thus, the comparison between LP-I and LP-A can be viewed as a baseline effect size.

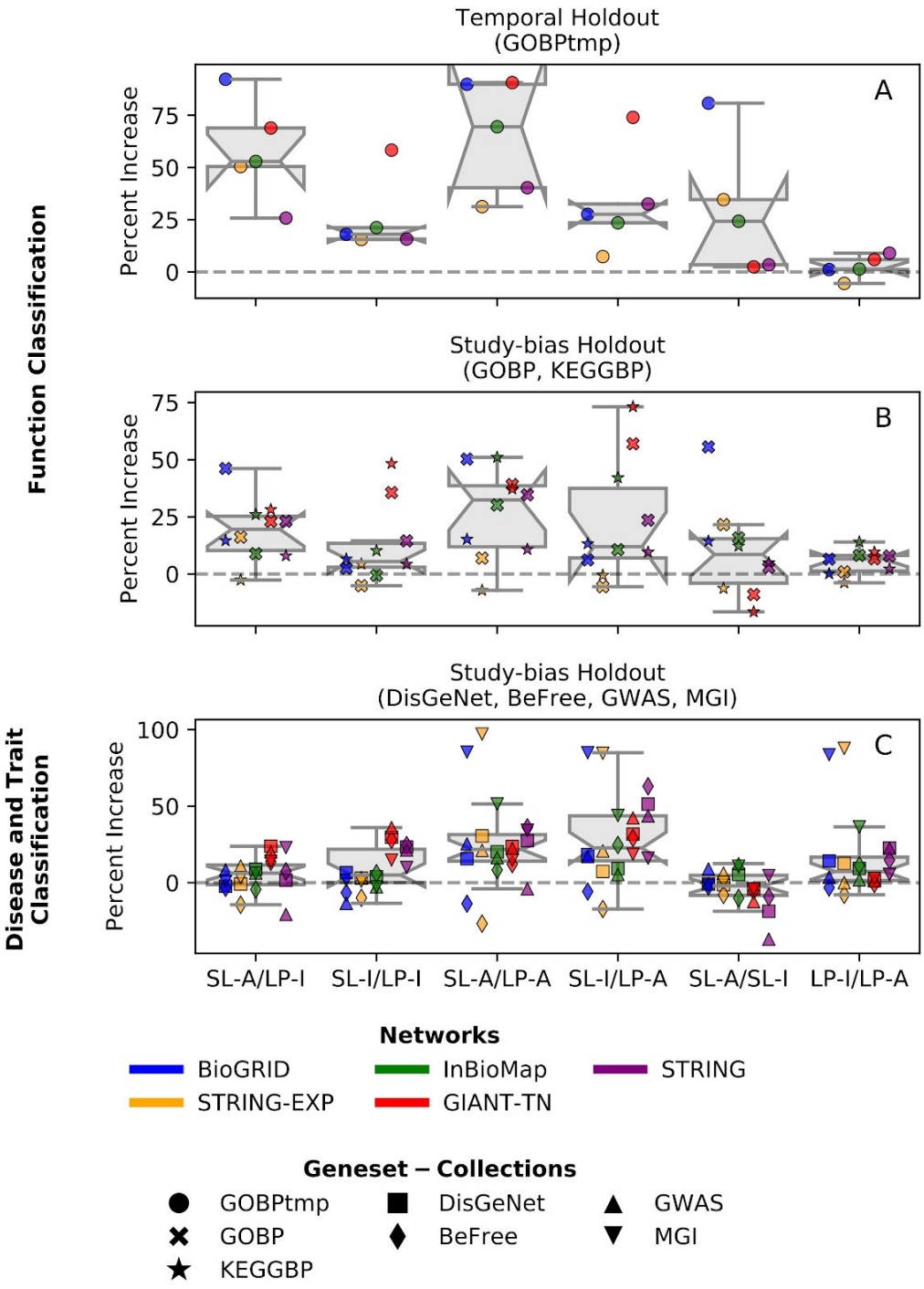

Fig S10. Effect size for every pair of methods. Each point is the median percent increase for every geneset-collection–network combination. (A) Functional prediction tasks using GOBP temporal holdout, (B) Functional prediction tasks using study-bias holdout for GOBP and KEGGBP, and (C) Disease and trait prediction tasks using study-bias holdout for DisGeNet, BeFree, GWAS, and MGI. The results are shown for auPRC where different colors represent different networks and different marker styles represent the different geneset-collections.

Section 2.3: Label Propagation with Negative Examples

In this section, we show results for using negative examples in label propagation (LPN). We performed the same hyperparameter tuning for the restart parameter as described in Supplemental Section 1.2 and find a

restart parameter of 0.45 is optimum when using negative examples (Fig. S11). This optimal value for the restart parameter in LPN is relatively low compared to the optimum value for LP (expect for the GIANT-TN network were both LP and LPN prefer a higher restart value). It is worth noting, that just like with LP, the dependance on the restart parameter is minimal (Fig. S11B). We also include boxplots comparing label propagation with and without negative examples (Fig. S12). Lastly, we show a side by side comparison of the ranking analysis (Fig. S13) and Wilxcon analysis (Fig. S14) using label propagation with and without negative examples. The results show that even though using negative examples slightly increases performance in label propagation, the results when compared against supervised learning remain unchanged.

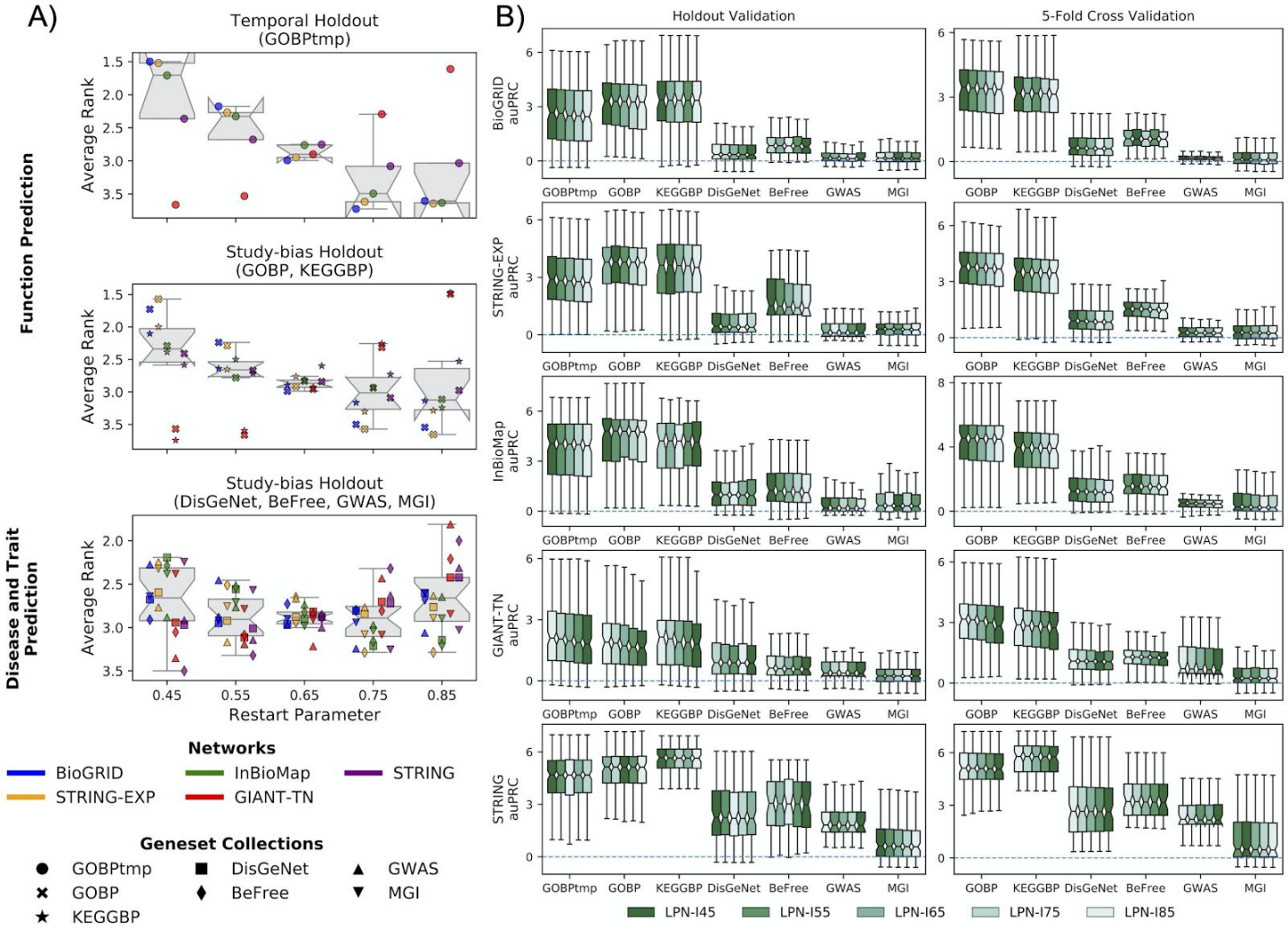

Fig. S11. Tuning the restart probability hyperparameter when using negative examples in label propagation. A) Each point in each boxplot represents the average rank for a geneset-collection–network combination, where the five restart probabilities (0.45, 0.55, 0.65, 0.75 and 0.85) were ranked in terms of performance (auPRC) for each geneset in a geneset-collection using the standard competition ranking. A restart probability of 0.45 was chosen as optimal. B) The performance for each individual geneset-collection–network combination is compared across the five restart probabilities. The methods are ranked by median value of auRPC with the highest scoring method on the left. There is no strong dependence of auRPC on the restart probability.

Fig. S12. Boxplots for performance across all geneset-collection–network combinations for label propagation on the influence matrix with and without using negative examples. A) The performance for each individual geneset-collection–network combination is compared for label propagation with negative examples (LPN-I, blue) and label propagation without negative examples (LP-I, green). The methods are ranked by median value with the highest scoring method on the left. Results show LPN-I has a moderately increased performance when compared to LP-I. B) Each point in the plot is the median value from one of the boxplots in A. This shows that both LPN and LP methods perform better for function prediction compared to disease/trait prediction.

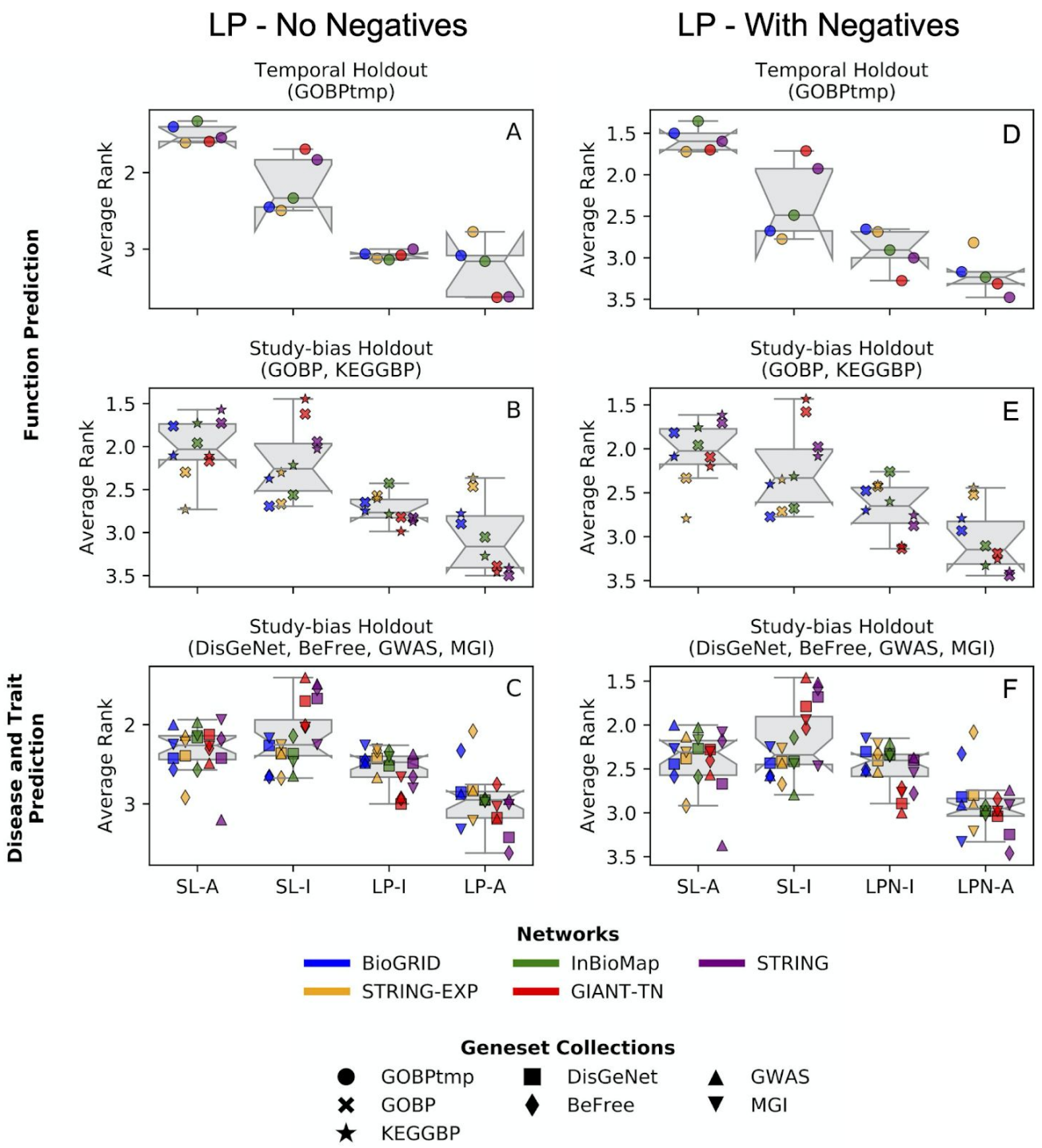

Fig. S13. Comparing results from average rank analysis with and without using negative examples in label propagation. The left column has label propagation without negative examples (LP) and the right column has label propagation with negative examples (LPN). Each point in each boxplot represents the average rank for a geneset-collection–network combination, obtained based on ranking the four methods in terms of performance for each geneset in a geneset-collection using the standard competition ranking. (A, D) Functional prediction tasks using GOBP temporal holdout, (B, E) Functional prediction tasks using study-bias holdout for GOBP and KEGGBP, and (C, F) Disease and trait prediction tasks using study-bias holdout for DisGeNet, BeFree, GWAS, and MGI. The results are shown for auPRC where different colors represent different networks and different marker styles represent the different geneset-collections. The results show that no substantial difference can be seen between using or not using negative examples in label propagation.

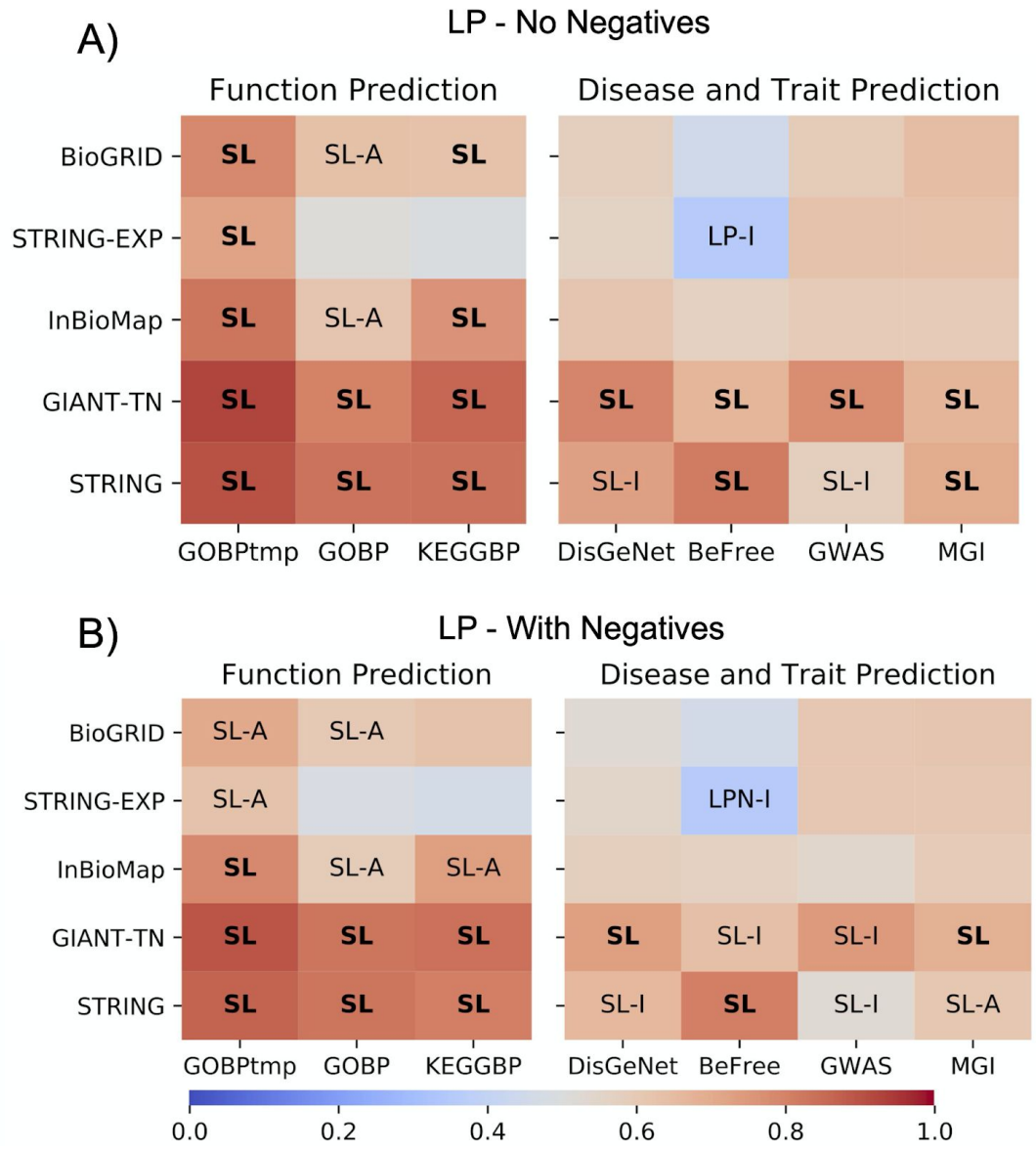

Fig. S14. Comparing the Wilcoxon statistical test analysis with and without using negative examples in label propagation. A) Label propagation without negative examples (LP) and B) label propagation with negative examples (LPN). For each network-geneset combination, each method is compared to the two methods from the other class. If a method was found to be significantly better than both methods from the other class (Wilcoxon ranked-sum test with an FDR threshold of 0.05), the cell is annotated with that method. If both models in that class were found to be significantly better than the two methods in the other class, the cell is annotated in bold with just the class. The color scale represents the fraction of genesets that were higher for the SL methods across all four comparisons. The first column uses GOBP temporal holdout, whereas the remaining 6 columns use study-bias holdout. The results show that no substantial difference can be seen between using or not using negative examples in label propagation.

References

Ashburner,M. *et al.* (2000) Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.*, **25**, 25–29.

Brown,G.R. *et al.* (2015) Gene: a gene-centered information resource at NCBI. *Nucleic Acids Res.*, **43**, D36–D42.

Choobdar,S. *et al.* (2019) Open Community Challenge Reveals Molecular Network Modules with Key Roles in Diseases. *bioRxiv*, 265553.

Davis,J. and Goadrich,M. (2006) The Relationship Between Precision-Recall and ROC Curves. In, *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06. ACM, New York, NY, USA, pp. 233–240.
Gillis,J. and Pavlidis,P. (2011) The Impact of Multifunctional Genes on 'Guilt by Association' Analysis. *PLOS ONE*, **6**, e17258.
Goyal,P. and Ferrara,E. (2018) Graph Embedding Techniques, Applications, and Performance: A Survey. *Knowl.-Based Syst.*, **151**, 78–94.
Greene,C.S. *et al.* (2015) Understanding multicellular function and disease with human tissue-specific networks. *Nat. Genet.*, **47**, 569–576.
Grover,A. and Leskovec,J. (2016) node2vec: Scalable Feature Learning for Networks. In, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*. ACM Press, San Francisco, California, USA, pp. 855–864.
Huang,J.K. *et al.* (2018) Systematic Evaluation of Molecular Networks for Discovery of Disease Genes. *Cell Syst.*, **6**, 484-495.e5.
Kanehisa,M. *et al.* (2017) KEGG: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Res.*, **45**, D353–D361.
Kanehisa,M. *et al.* (2019) New approach for understanding genome variations in KEGG. *Nucleic Acids Res.*, **47**, D590–D595.
Kanehisa,M. and Goto,S. (2000) KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.*, **28**, 27–30.
Li,T. *et al.* (2017) A scored human protein-protein interaction network to catalyze genomic interpretation. *Nat. Methods*, **14**, 61–64.
Pedregosa,F. *et al.* (2011) Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.
Piñero,J. *et al.* (2017) DisGeNET: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Res.*, **45**, D833–D839.
Piñero,J. *et al.* (2015) DisGeNET: a discovery platform for the dynamical exploration of human diseases and their genes. *Database*, **2015**.
Saito,T. and Rehmsmeier,M. (2015) The Precision-Recall Plot Is More Informative than the ROC Plot When Evaluating Binary Classifiers on Imbalanced Datasets. *PLOS ONE*, **10**, e0118432.
Schriml,L.M. *et al.* (2019) Human Disease Ontology 2018 update: classification, content and workflow expansion. *Nucleic Acids Res.*, **47**, D955–D962.
Smith,C.L. *et al.* (2018) Mouse Genome Database (MGD)-2018: knowledgebase for the laboratory mouse. *Nucleic Acids Res.*, **46**, D836–D842.
Smith,C.L. and Eppig,J.T. (2009) The mammalian phenotype ontology: enabling robust annotation and comparative analysis. *Wiley Interdiscip. Rev. Syst. Biol. Med.*, **1**, 390–399.
Stark,C. *et al.* (2006) BioGRID: a general repository for interaction datasets. *Nucleic Acids Res.*, **34**, D535–D539.
Szklarczyk,D. *et al.* (2015) STRING v10: protein-protein interaction networks, integrated over the tree of life. *Nucleic Acids Res.*, **43**, D447-452.
The Gene Ontology Consortium (2019) The Gene Ontology Resource: 20 years and still GOing strong. *Nucleic Acids Res.*, **47**, D330–D338.
Wu,C. *et al.* (2013) BioGPS and MyGene.info: organizing online, gene-centric information. *Nucleic Acids Res.*, **41**, D561–D565.
Xin,J. *et al.* (2016) High-performance web services for querying gene and variant annotation. *Genome Biol.*, **17**, 91.