Bioinformatics, 37(16), 2021, 2414–2422
doi: 10.1093/bioinformatics/btab098
Advance Access Publication Date: 12 February 2021
Original Paper

Systems biology

NetQuilt: deep multispecies network-based protein function prediction using homology-informed network similarity

Meet Barot [1,*], Vladimir Gligorijević[2], Kyunghyun Cho[1] and Richard Bonneau[1,2,*]

[1]Center for Data Science, New York University, New York, NY 10011, USA and [2]Center for Computational Biology, Flatiron Institute, New York, NY 10010, USA

*To whom correspondence should be addressed.

Associate Editor: Pier Luigi Martelli

Received on August 18, 2020; revised on February 4, 2021; editorial decision on February 7, 2021; accepted on February 9, 2021

Abstract

Motivation: Transferring knowledge between species is challenging: different species contain distinct proteomes and cellular architectures, which cause their proteins to carry out different functions via different interaction networks. Many approaches to protein functional annotation use sequence similarity to transfer knowledge between species. These approaches cannot produce accurate predictions for proteins without homologues of known function, as many functions require cellular context for meaningful prediction. To supply this context, network-based methods use protein-protein interaction (PPI) networks as a source of information for inferring protein function and have demonstrated promising results in function prediction. However, most of these methods are tied to a network for a single species, and many species lack biological networks.

Results: In this work, we integrate sequence and network information across multiple species by computing IsoRank similarity scores to create a meta-network profile of the proteins of multiple species. We use this integrated multispecies meta-network as input to train a maxout neural network with Gene Ontology terms as target labels. Our multispecies approach takes advantage of more training examples, and consequently leads to significant improvements in function prediction performance compared to two network-based methods, a deep learning sequence-based method and the BLAST annotation method used in the Critial Assessment of Functional Annotation. We are able to demonstrate that our approach performs well even in cases where a species has no network information available: when an organism's PPI network is left out we can use our multi-species method to make predictions for the left-out organism with good performance.

Availability and implementation: The code is freely available at https://github.com/nowittynamesleft/NetQuilt. The data, including sequences, PPI networks and GO annotations are available at https://string-db.org/.

Contact: mmb557@nyu.edu or rb133@nyu.edu

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

1 Introduction

Sequences have been the primary source of information protein function prediction, mainly because of their abundance and the ease with which many models can incorporate large amounts of sequence data. However, for function prediction, sequence information fails to give the context of a protein in an organism; this context can be highly relevant in determining the protein's function. Protein interaction networks, on the other hand, offer a way to understand how proteins function in cellular pathways, and thus have been a powerful source of information for inferring the functions of unannotated proteins (Chen *et al.*, 2014; Cho *et al.*, 2016; Milenković and Pržulj, 2008; Mostafavi*et al.*, 2008;Sharan*et al.*, 2007).

In community benchmarks, such as the Critical Assessment of Functional Annotation (CAFA), the best-performing methods rely on multiple complementary data sources—protein sequence, structure and network information—in order to make more accurate predictions (Radivojac*et al.*, 2013; Rentzsch and Orengo, 2009; Zhou *et al.*, 2019). There are many reviews of protein function prediction methods in general (Friedberg, 2006; Kihara, 2016; Lee *et al.*, 2007; Rentzsch and Orengo, 2009). Most previous network-based approaches integrate different types of networks containing complementary information to achieve state-of-the-art performance (Cho

et al., 2016; Gligorijević et al., 2018; Mostafavi et al., 2008), but are limited to training on and making predictions for a single organism's proteins. Methods for sequence and structure-based function prediction are numerous (Cozzetto et al., 2016; Gligorijevic et al., 2019; Gong et al., 2016; Kulmanov and Hoehndorf, 2020); these methods are inherently able to predict functions for proteins of multiple organisms, and can have certain other advantages such as region specificity for predictions (Gligorijevic et al., 2019; Koo and Bonneau, 2019; Vacic et al., 2010). A remaining challenge is using the vast amounts of network information from multiple species in a single model.

Our method, NetQuilt, accomplishes several important goals in function prediction. First, NetQuilt allows for the integration of sequences and networks, which allows the limited knowledge of the homology between proteins to be supplemented by knowledge of the network topology, and vice versa—incomplete protein-protein interaction networks are supplemented by homology. NetQuilt also creates protein features that are not tied to single species and that include evolutionary and functional information. As a result of the increased training examples in the multispecies setting compared to methods considering only single species, rarer Gene Ontology (GO) (Ashburner et al., 2000) terms are able to be trained on. The much larger set of training examples also serves to improve prediction on more abundant terms. Most importantly, our method enables network-based function prediction even for species for which knowledge of their protein interaction networks is limited. We demonstrate the achievement of these goals in several settings. We compare the quality of protein features of a single organism in a single-species versus a multispecies setting. We show that multispecies features are more indicative of a protein's function than single-species features. We also test the model's ability to predict functions of a species whose entire PPI network is missing, with the model trained on all other species in the set being considered, in an approach termed 'leave one species out' (LOSO). We demonstrate that our model is capable of using information from other species to correctly infer functions of the missing species.

2 Related work

Protein function prediction using PPI networks is a node classification problem, the methods for which can be categorized into two groups: label-propagation methods, and classifiers trained on graph features. Label propagation methods propagate labels from labeled nodes to unlabeled nodes via random walks; this strategy is used to predict protein function in a method called GeneMANIA (Mostafavi et al., 2008). Another approach, FunctionalFlow, uses the idea of network flow to propagate labels based on simple local rules (Nabieva et al., 2005). The category of classifiers trained on graph features can be split further into two categories: those that manually engineer features from the network data, or those methods that learn network embeddings of nodes in order to be used in a classifier. The manually engineered graph features can be based on graph measures such as node degree, neighborhood size within some number of steps, number of shortest paths, etc. Other features that can be constructed over nodes include graphlets (Milenković and Pržulj, 2008; Pržulj, 2007; Vacic et al., 2010), and random walk profiles of nodes within their graph, which have been extended and applied to heterogeneous and multiplex biological networks (Li and Patra, 2010; Valdeolivas et al., 2019). Network embedding has been extensively used in protein functional analysis and includes methods based on matrix factorization (Cho et al., 2016), graph kernels (Fan et al., 2019) and deep learning (Gligorijević et al., 2018; Wan et al., 2019; Zitnik and Leskovec, 2017). A comprehensive review of network embedding in computational biology compared to other types of network-based algorithms for several applications can be found in Nelson et al. (2019), and reviews of network representation learning methods in general can be found in Hamilton et al. (2017) and Goyal and Ferrara (2018).

Our previous study (Gligorijević et al., 2018) introduced a method called deepNF (deep Network Fusion), which involves using a multimodal autoencoder to create embeddings of nodes from different types of protein-protein interaction networks of an organism. These embeddings are then used to train support vector machines (SVM) to predict GO terms. This method outperformed other methods using different types of interaction networks to predict function, including Mashup (Cho et al., 2016) and GeneMANIA (Mostafavi et al., 2008), all of which had access to six STRING network types {'experimental', 'coexpression', 'coocurrence', 'neighborhood', 'fusion' and 'database'}. This work demonstrated that multimodal autoencoder neural networks could effectively extract functionally informative features from graphs with multiple edge types. Another method, STRING2GO, uses maxout neural networks in order to create functional representations of proteins from protein interaction networks of a single species (Wan et al., 2019). The maxout network is trained to predict GO terms from Mashup or Node2Vec (Grover and Leskovec, 2016) node embeddings, and the representations of each protein is taken from the layer before the output predictions. These representations are then used to train SVMs to predict GO terms. The authors show that these representations are able to outperform the original Mashup and Node2Vec embeddings of PPI networks when used to train SVMs for the function prediction task. In Zitnik and Leskovec (2017), an unsupervised neural network is used to learn embeddings from a tissue-specific multi-layer PPI graph. These task-independent embeddings are then used to predict multi-cellular function.

However, these methods are limited to using information from single organisms for prediction, because they operate on a feature space common only to proteins of that organism. A better approach would be to take into account information from proteins of many different organisms at once in order to take advantage of large-scale training sets.

A few methods make use of information from protein interaction networks of multiple species. One such method is NetGO, an ensemble learning-to-rank method that combines six component methods, one of which is a k-nearest-neighbors method that uses PPI networks of multiple species (You et al., 2019). One drawback to this method is that it is unable to use the homology information in any way beyond direct transfer of annotation between homologues. Ideally, a protein function prediction method should be able to use homology information to supplement network information even on proteins whose sequences are not similar to the training set protein sequences. In addition, MetaGO (Zhang et al., 2018) is a method that combines scores of sequence homology, structure alignment and homologues of PPI network neighbors combined with logistic regression in order to transfer functional annotations. This method is unable to predict function for a protein without either a sequence homolog, a structurally similar protein in the training set or with a network neighbor with a training set homolog. Another method, MUNK, is a kernel-based method that produces functional embeddings used for predicting synthetic lethality for pairs of proteins of multiple species (Fan et al., 2019); they additionally demonstrate that proteins close in this embedding space are similar in function. The key idea of their approach is that proteins from different species are embedded in the same vector space using graph kernels with landmark proteins in the networks of the two species that perform the same functions.

The problem of network alignment is to find topological and functional similarities between nodes of different networks. Local network alignment algorithms aim to find subgraphs which are conserved between input networks, while the goal of global network alignment algorithms is to find mappings of all nodes between the input networks. Most network alignment methods focus on this latter goal (Gligorijević et al., 2016; Liao et al., 2009; Malod-Dognin and Pržulj, 2015; Patro and Kingsford, 2012; Saraph and Milenković, 2014; Singh et al., 2008; Vijayan et al., 2015).IsoRank (Singh et al., 2008) is a global network alignment algorithm used to align multiple PPI networks. This is done in two stages: first by solving an eigenvalue problem across all pairs of input networks to obtain protein similarity scores, and then by using k-partite matching to obtain the final alignment of all organisms, giving sets of functional orthologs across species. IsoRankN (Liao et al., 2009) was developed as an improvement to the alignment extraction portion of

IsoRank in which instead of k-partite matching, spectral clustering was applied to the meta-graph of all organisms' proteins induced by the similarity scores given by the eigenvalue problem. More recent global network alignment algorithms include L-GRAAL (Malod-Dognin and Pržulj, 2015), which uses a graphlet similarity-scoring function used with a search heuristic based on Lagrangian relaxation, and GHOST, whose key step uses a signature of nodes based on the spectrum of the normalized Laplacian of local subgraphs; this signature is then used to measure topological similarity of networks (Patro and Kingsford, 2012). Fuse (Gligorijević*et al.*, 2016) is another network alignment method consisting of two steps. The first step calculates functional similarity between proteins using a weighted sum of scores from a non-negative matrix tri-factorization of all considered PPI networks and sequence similarity. The second step constructs an edge-weighted k-partite graph (where k is the number of PPI networks) from these similarities and then obtains the one-to-one network alignment using an approximate maximum weight k-partite matching solver. A comprehensive review of biological network alignment can be found in Faisal *et al.* (2015). Other algorithms for network alignment include those that focus on finding small network region similarities conserved among networks, unconstrained by the assumption of one-to-one mapping of nodes. These algorithms fall into the local network alignment category. A comparison study of local and global network alignment methods can be found in Meng*et al.* (2016), where it was found that network topology has additional biological knowledge compared to sequence data; additionally, global and local network alignment methods may give complementary information for protein function prediction.

In this study, we use the first step of IsoRank to integrate sequence homology information with PPI network information to generate functionallyinformative similarity scores between species as well as within species themselves. We use these similarity scores for every protein as its feature representation to enable the training of a neural network with proteins coming from many different organisms' PPI networks in the same input space.

3 Materials and methods

In this section, we describe the problem of protein function prediction from PPI network and homology information, define our performance measures and outline the components of our method, NetQuilt. These components are the global network alignment algorithm for creating both intranetwork (within-species) and internetwork (between proteins in different species) node-similarity profiles, and the maxout neural network, which uses the concatenated aligned-network vectors to predict Gene Ontology (GO) terms. See Figure 1for an overview of the procedure.

3.1 Problem specification

Consider a set of N_{org} undirected graphs, where each graph is a protein-protein interaction network of a different organism. The graphs each have a set of nodes representing proteins for each organism, and a set of edges representing the interactions between these proteins. The graphs are represented by adjacency matrices $\{\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_{N_{org}}\}$. Consider further that we have a set $\{\mathbf{R}_{1,1}, \mathbf{R}_{1,2}, \mathbf{R}_{1,3}, \ldots, \mathbf{R}_{1,N_{org}}, \mathbf{R}_{2,2}, \mathbf{R}_{2,3}, \mathbf{R}_{2,4} \ldots, \mathbf{R}_{N_{org},N_{org}}\}$ of edges representing homology links, between all proteins of all species. Our objective is to assign a predicted GO score vector $\hat{y}_i \in \mathbb{R}^c$ to each protein i, where c is the number of considered terms of a particular GO branch, and each entry $\hat{y}_{ij}$ in $\hat{y}_i$ is a score between 0 and 1 representing the confidence of assigning the jth GO term to protein i.

3.2 Evaluation metrics

We evaluate our predictions with three function-centric measures; precision recall curve (AUPR) under macro and micro averaging, as well as function-centric F1-score, and two protein-centric measures; accuracy, and F-max score.

Under macro averaging, AUPR is calculated for each GO term label in the prediction matrix, and then averaged across all terms. Under micro averaging, the label and prediction matrices are vectorized, and then AUPR is computed across the resulting label and prediction vectors. We calculate F-1 score as in Gligorijević*et al.* (2018) and as previously introduced in Cho *et al.* (2016): we take the top three scoring terms for each protein as 'positive' predictions, and calculate the geometric mean of precision and recall under'micro' averaging for all terms. We have chosen AUPR, rather than the area under the ROC curve, because the ROC can mask poor classification performance in datasets where there is an imbalance of positive labels, which is the case in protein function prediction (Saito and Rehmsmeier, 2015).

The remaining two, accuracy and F-max score, are protein-centric measures. We define accuracy to be the proportion of proteins that were assigned all of their correct GO terms, with no additional terms, using a threshold of 0.5 for assignment. F-max is calculated as in the CAFA competition (Zhou *et al.*, 2019): for each protein, calculate the precision and recall of all GO term predictions for a given threshold between 0 and 1, averaging across all proteins, and compute the F-1 score for all thresholds. F-max is then the maximum of these F-1 scores.

3.3 Creating multispecies similarity profiles with IsoRank

Our method computes profiles of the nodes in all species' networks, creating a shared feature space for all proteins, which we then use to train a maxout neural network to predict protein function. We first compute similarity scores between proteins of different species in a way derived from the IsoRank method of multispecies network alignment (Singh *et al.*, 2008). The scores are given by the following recurrence equation:

$$\mathbf{S}_{ij}^{(t+1)} = \alpha \hat{\mathbf{A}}_i^T \mathbf{S}_{ij}^{(t)} \hat{\mathbf{A}}_j + (1-\alpha)\mathbf{R}_{ij} \quad (1)$$

where:

- $\mathbf{S}_{ij}^{(t)}$ is the similarity matrix between networks (species) i and j after t steps of diffusion;
- $\mathbf{R}_{ij}[k,l] = -\log(eval[k,l])$ is the blast e-value similarity between protein k in network (species) i and protein l in network (species) j, with a maximum e-value cutoff of 1e-3 and with the log score scaled between 0 and 1.; and
- $\hat{\mathbf{A}}_i$, $\hat{\mathbf{A}}_j$are the row-normalized adjacency matrices of networks (species) i and j.

Starting with $\mathbf{S}^{(0)} = \mathbf{I}^{n_i \times n_j}$, we iterate this calculation (Equation 1) until convergence with respect to the norm of the difference between the previous matrix $\mathbf{S}_{i,j}^{(T-1)}$ and the current matrix $\mathbf{S}_{i,j}^{(T)}$. We then calculate IsoRank similarity scores between proteins *within* each species. This computes'alignment' scores between a network and itself, integrating sequence homology scores computed using BLAST and protein-protein interactions.

We can now construct a large symmetric matrix $\mathbf{S}$ in which the IsoRank similarity matrices of all species with themselves are placed along the diagonal, resulting in a block-diagonal matrix. Next, each interspecies protein similarity matrix $\mathbf{S}_{i,j}$ is placed on the off-diagonal, comprising the submatrix with row indices of the proteins of species i and column indices of the proteins of species j. Refer to steps B, C, D, E and F in Figure 1 for a visual description of this matrix construction. $\mathbf{S}$ now contains the information from all the individual protein interaction networks as well as the links between them, integrated with sequence-similarity information. We finally use this matrix as input to a maxout neural network, with each row of the matrix $\mathbf{S}$ being used as a single training sample. We note that since the maxout neural network input depends on the dimensionality of $\mathbf{S}$, the total number of proteins considered by the algorithm is limited by the available GPU memory to contain a batch of training samples.

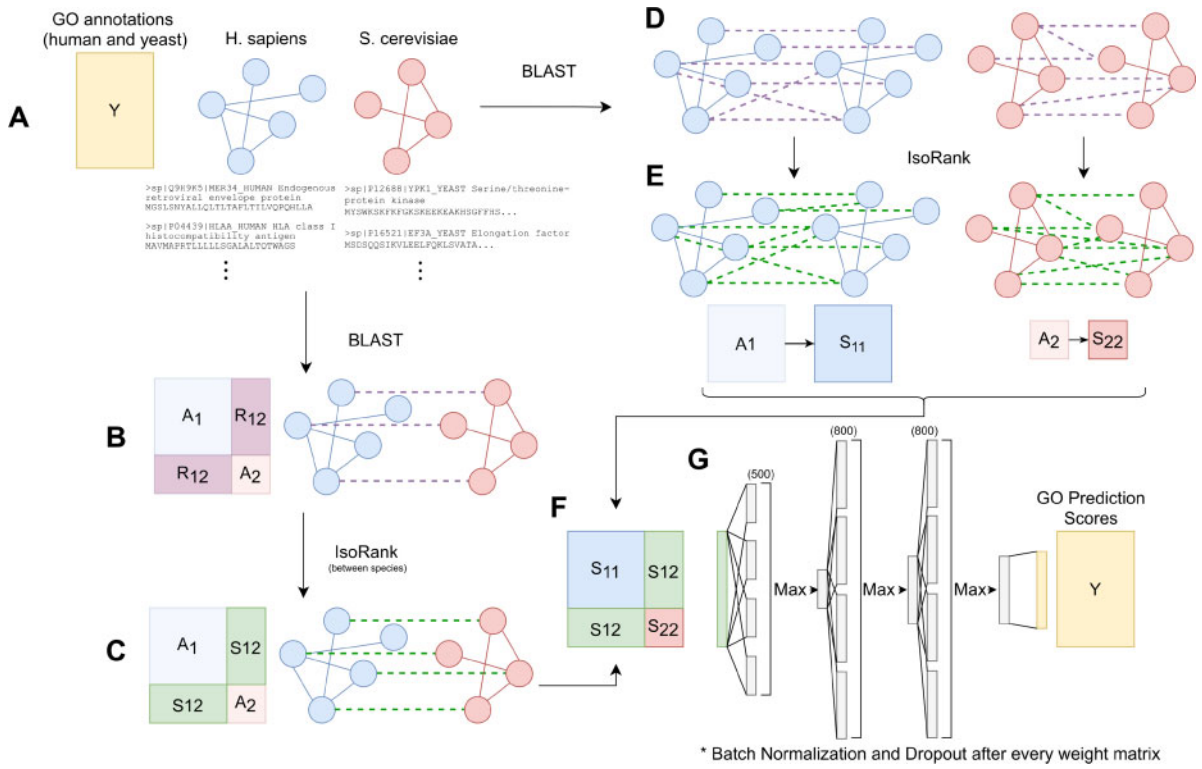

Fig. 1. Overview of our method for running on two organisms (human and yeast). (**A**) For each taxonomy ID, download network, annotation and sequence files from the STRING-db static website (version 11). (**B**) Use BLAST to create sequence identity links between proteins of pairs of different species. (**C**) Compute IsoRank scores between proteins of different species, using BLAST sequence identity values and the organisms' networks to create a combination of network and homology information. (**D**) Use BLAST to create sequence identity links among proteins of each individual species. (**E**) Compute IsoRank alignment scores between proteins of the same species, creating denser matrices S_{11} and S_{22} from weighted adjacency matrices A_{11} and A_{22} and sequence identity matrices R_{11} and R_{22}. (**F**) Concatenate all IsoRank matrices between all species to make the full S matrix. (**G**) Train maxout neural network with the S matrix as features and the annotation matrix as labels

3.4 Using maxout neural networks to predict protein function from aligned Meta-network features

Maxout neural networks, introduced in Goodfellow*et al.* (2013), are neural networks whose layers have the maxout activation function. The maxout activation of a layer is the element-wise maximum of a set of affine transformations to the input of that layer. More explicitly, a maxout layer's ith output value $h_i \in \mathbb{R}^m$ given an input $x \in \mathbb{R}^d$ is defined as:

$$h_i(x) = \max_{j \in 1,k} z_{i,j}$$

where $z_{ij} = x^T W_{:,ij} + b_{i,j}$ is the ith element of the jth affine transformation of the input vector with learned parameters $W \in \mathbb{R}^{d \times m \times k}$ and $b \in \mathbb{R}^{m \times k}$. Maxout activation functions are able to approximate arbitrary convex functions given a sufficient number of maxout units, (i.e. affine transformations), and therefore enable the neural network to learn not only relationships between hidden units but also the activation functions themselves. This provides additional flexibility, which enables the neural network to learn features that are more specifically tailored to a prediction task.

Goodfellow*et al.* (2013) also demonstrated that maxout networks more precisely approximate the average over all neural networks with randomly dropped out connections every iteration. This can be interpreted as a more effective approximation of an ensemble of these neural networks. This applies to the ReLU activation function as well: in fact, maxout activation can be seen as a generalization of ReLU, which is itself a piecewise linear function. However, maxout activation does not have the problem of output units 'dying'—becoming and staying at 0 during optimization.

The architectures for our models are listed in Table 1 (see also part G in Fig. 1). To avoid overfitting, we use early stopping with the criterion of improving AUPR calculated over a validation set consisting of 20% of the training data, with patience 30 (i.e. if the AUPR score does not improve in 30 consecutive epochs, the training is stopped).

Table 1. Model architectures for Eukaryote and Bacteria datasets (see Section 3.5 for a description of these datasets)

Hyperparameters	Bacteria	Eukaryotes
Hidden layer dimensions	[500, 800, 800]	[500, 800, 800]
Maxout units	3	4
Dropout	0.2	0.2
Batch normalization	True	True
Learning rate	0.01	0.01
Batch size	16	32
Max number of epochs	100	300
Optimizer	AdaGrad	AdaGrad

Note: 'Maxout units' refers to the number of separate weight matrices for a given layer; the element-wise max is computed over the product of the weight matrices with the outputs of the previous layer. Batch normalization (Ioffe and Szegedy, 2015) and AdaGrad (Duchi*et al.*, 2011) were used for both sets of species.

The architectures were chosen using cross-validation performance on datasets for eukaryotes and bacteria using a previous version of the STRING (v10.5) database (Szklarczyk*et al.*, 2017) for annotations and network information. The hyperparameter search started with an architecture based on Wan *et al.* (2019), with three rounds of random search, trying 1% of possible models each round. We include a list of hyperparameter ranges for these rounds, as well as a description of this process, in Supplementary Section S5. Empirically, maxout neural networks performed better than neural networks with sigmoid or ReLU activation functions for this task. Other benefits of maxout neural networks include fast gradient

computations relative to other activation functions, e.g. sigmoid, and fewer choices of hyperparameters, since the activation function is learned. The models were implemented using Keras (Chollet*et al.*, 2015).

3.5 Datasets

We conduct our analyses on both a collection of eukaryote networks and a separate collection of bacteria networks. Each dataset consists of STRING PPI networks, of which we use only the 'experimental' category for our method, and Gene Ontology annotations of each organism retrieved from STRING version 11 (Szklarczyk*et al.*, 2017). The statistics on the organisms we include in our study are given in Supplementary Figures S1 and S2, which show the networks' largest connected component ratios and the annotation percentages of proteins present in STRING. The numbers of nodes and experimental PPI edges, for bacteria and eukaryotes, are shown in Supplementary Tables S1 and S2, respectively. In order to select the value of the α parameter for our experiments for each set, we tested several values in a single-species cross-validation setting (see Supplementary Figs S3–S5 for the results of the search). The chosen organisms come from the set of organisms that were evaluated in CAFA 4. For the bacteria, all of the organisms from CAFA 4 were used in our pipeline; for the eukaryotes, we selected a subset to conserve memory when training our models (*Caenorhabditis elegans*, *Drosophila melanogaster*, *Danio rerio*, *Homo sapiens*, *Sus scrofa*, *Mus musculus* and *Rattus norvegicus*). We use GO terms that cover between 0.5% - 5% of the species' proteins in its PPI network (including IEA annotations), and remove proteins without annotations of these GO terms from training and evaluation sets. We note that GO terms, organized in a hierarchy, are dependent on each other, and so average performance across all terms can be influenced by these relationships. A table of the number of GO terms that we consider for both cross-validation and leave-one-species-out validation for each organism can be found in Supplementary Table S3. However, by choosing specific GO terms, with annotations covering between 0.5%-5% of a given organism's proteome, we reduce the influence of the hierarchy on the aggregated performance as a result of removing the more general terms.

3.6 Cross-validation

In our first set of evaluations, in order to compare with single-species methods, we perform cross-validation on a single test species at a time. The performance is averaged over 5 repetitions with 20% of data used as the test set. We train our models, as well as the BLAST baseline, on GO term annotations of any evidence code(Ashburner*et al.*, 2000), but evaluate our predictions with annotations of the evidence codes EXP, IDA, IPI, IMP, IGI, IEP, TAS and IC, as previously used in CAFA papers (Radivojac*et al.*, 2013). Since, realistically, our method has access to more training examples than the single-species methods, we include three benchmark versions of our method:

1. NetQuilt trained on a subsampled set of multispecies annotations, where we randomly subsample training examples equal to the number of training examples we would have if only considering the species being tested on
2. NetQuilt trained on single-organism annotations, in which we take only rows corresponding to the particular organism being evaluated from the original matrix S containing protein similarities among all organisms (for example, training the maxout neural network only on the rows corresponding to human proteins in the block $\mathbf{S}$ matrix represented in Fig. 1B)
3. Single-species Maxout, in which we take only the IsoRank-score matrix for integrating the single organism's PPI network with sequence homology information from BLAST, but not including similarities to any other organisms' proteins (for example, training the maxout neural network only on the $\mathbf{S}_{11}$ matrix for human proteins represented in Fig. 1E)

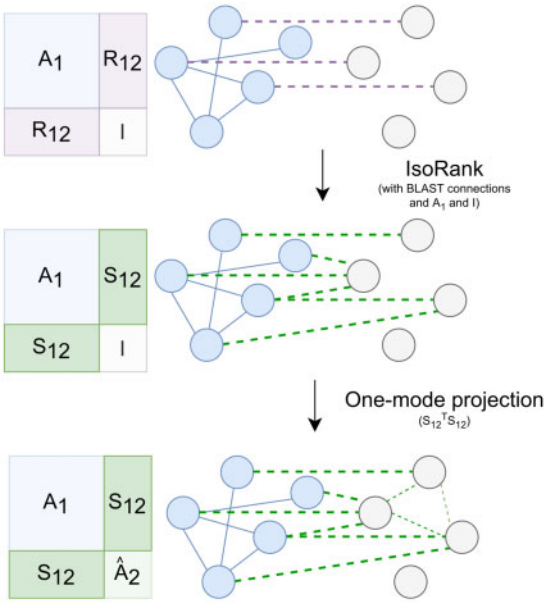

Fig. 2. Procedure for predicting a network to be used in the leave-one-species-out validation setting, where we assume no knowledge of the PPI network for one organism. First, BLAST connections (represented as purple dashed lines) between the proteins of the known network and the left-out network are created. IsoRank is then run for the interspecies matrix, using the known network A_1 and the left-out network given by the identity matrix I, giving the IsoRank connections S_{12} depicted by the large green dashed lines. We finally obtain a predicted network by taking the one-mode projection of the IsoRank connections: $S_{12}^T S_{12} = \hat{A}_2$. In the case of multiple known organisms, we simply take the average of all organisms' one-mode projections with the left-out organism

These benchmarks allow us to disentangle the effects that the number of training examples and the addition of new features have on performance. In addition to these, we also include deepNF, BLAST [propagating labels from training to test proteins based on sequence similarity as in CAFA (Radivojac*et al.*, 2013)], DeepGOPlus (Kulmanov and Hoehndorf, 2020) and MetaGO. deepNF includes information from STRING network types not used by our models: i.e. the coexpression, cooccurrence, neighborhood, fusion and database networks. BLAST, like our main multispecies model, uses proteins from all organisms in the set of chosen species to make predictions on the cross-validation test proteins. DeepGOPlus is a method combining predictions from a deep convolutional neural network and homology-based annotation transfer. DeepGOPlus was trained on its original training set described in Kulmanov and Hoehndorf (2020) with the default parameters, but with proteins present in our test sets removed for each evaluation. We also include the PPI-network and homology-based scoring pipelines of MetaGO, as a method that uses similar input data. These pipelines of MetaGO made predictions with the default settings for all evaluations. In order to make the comparison to our method fair we excluded the structure-based pipeline from MetaGO as our method uses only sequence and PPI information.

3.7 Leave-one-species-out validation

The next set of experiments we performed simulate a scenario in which we use the networks of multiple species in order to predict the functions of proteins of an organism with no PPI network available (a reasonably common occurrence for non-model species). An outline of the procedure is shown in Figure 2.

We first take a single organism with its annotations left out from training and used as the test set, and leave out the network for that organism. In order to construct the features of the organism for use

in the maxout neural network, we first need to obtain interspecies connections between the test organism and all other organisms in the dataset. To do this, we first calculate the sequence similarity between the test organisms' proteins and all other organisms' proteins, and run IsoRank in the previously described way, except that we use the identity matrix in place of the PPI network of the left-out organism. We obtain an $n_i \times n_{\text{test}}$ interspecies protein similarity matrix $S_{i,\text{test}}$ relating each species' n_i proteins with the test species' n_{test} proteins. We then perform a one-mode projection, given by $S_{i,\text{test}}^T S_{i,\text{test}}$, which predicts connections between the nodes of the test species from their shared neighbors (through the IsoRank connections) in other species. Since we have a prediction matrix for every other species in the set besides the test species, we take the element-wise mean of these different matrices to get the predicted network $\hat{A}_{\text{test}}$. Finally, using this matrix as a proxy for a real PPI network, we run IsoRank on the matrix with itself, combined with its own species' BLAST connections, to obtain the matrix $S_{\text{test,test}}$. In these LOSO evaluations, we did not remove any network information from MetaGO. It was run under default settings to predict function for the given organisms.

4 Results

In the following sections, we present the performance of our method in two evaluation settings. The first setting is cross-validation over the annotations of a single species, in which we can compare our method to single-species network-based methods. The second setting is leave-one-species-out (LOSO) evaluation, in which we leave out both a species' PPI network and its annotations while using the rest of the organisms to train, as outlined in the previous section.

4.1 Cross validation over annotations of one species

We present the performance of our method in cross-validation on **human**, **fly**, **mouse** and *E.coli*. We summarize our results using AUPR under micro and macro averaging, accuracy score (Acc), F1-score and F-max, as described in Section 3.2. We show results separately for the three different branches of Gene Ontology, molecular function (MF), biological process (BP) and cellular component (CC).

In Figures 3–5, we see that the NetQuilt network trained on model bacteria proteins outperforms the other methods across the three branches of Gene Ontology for E. coli, human and mouse, for macro and micro AUPRs, F1 score and F-max. This can primarily be attributed to the large number examples included in the training set compared to the benchmark versions of NetQuilt and deepNF, which can only run on a single organism. In addition, the diversity of training examples across multiple species also serves to increase performance, as indicated by the higher performance of the maxout network trained on subsampled sets of annotations from multiple species equal in size to the training set for a single species. As for the methods taking multiple species' annotations into account, NetQuilt has several advantages allowing it to perform better. Compared to DeepGOPlus, NetQuilt has access to PPI information of several species, whereas DeepGOPlus only uses sequence information. Compared to MetaGO, NetQuilt's high-capacity neural network is able to learn more complex dependencies between homology and network topology to predict function. However, for the accuracy measure, NetQuilt performs worse than the other methods. It is likely that the 0.5 cutoff, which we use to consider a GO term 'predicted' in the accuracy measure, is not optimal for NetQuilt, as its predictions are not necessarily calibrated for classification for that particular cutoff.

For fly, shown in Figure 6, deepNF outperforms our method in the biological process and cellular component branches for the macro and micro AUPR, accuracy and F1 scores. We note that

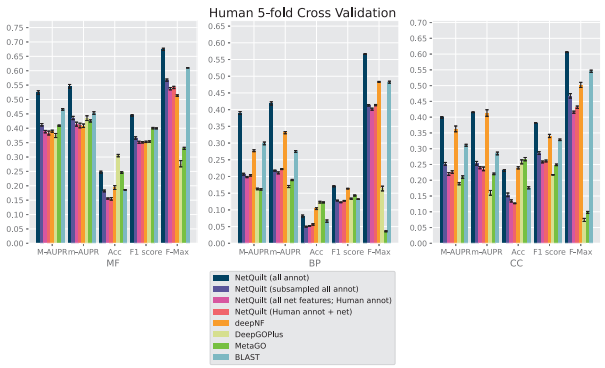

Fig. 4. Performance comparison of NetQuilt method with baselines. Methods shown: NetQuilt trained on model eukaryote annotations; NetQuilt trained on subsampled model eukaryote annotations; NetQuilt trained only on human examples; single-species NetQuilt (taking only the human IsoRank matrix and annotations); deepNF (single-species, but integrating 6 STRING network types); DeepGOPlus (trained on original dataset with our test set proteins removed); MetaGO (predicted with PPI+homology pipelines only, using its default dataset of annotations); and CAFA BLAST annotation transfer method using all selected eukaryote annotations

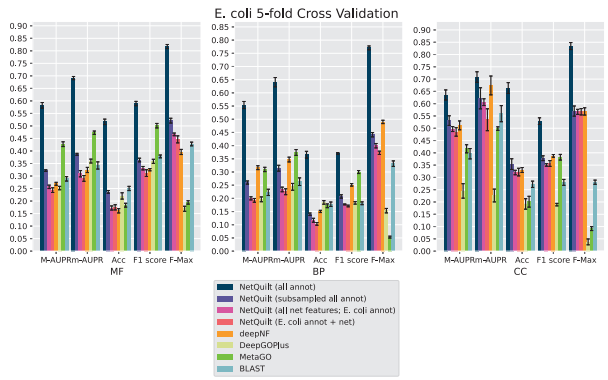

Fig. 3. Performance comparison of NetQuilt method with baselines. Methods shown: NetQuilt trained on model bacteria annotations; NetQuilt trained on subsampled model bacteria annotations; NetQuilt trained only on *E.coli* str. K-12 substr. MG1655 examples; single-species NetQuilt (taking only the *E.coli*IsoRank matrix and annotations); deepNF (single-species, but integrating 6 STRING network types); DeepGOPlus (trained on original dataset with our test set proteins removed); MetaGO (predicted with PPI+homology pipelines only, using its default dataset of annotations); and CAFA BLAST annotation transfer method using all selected bacteria annotations

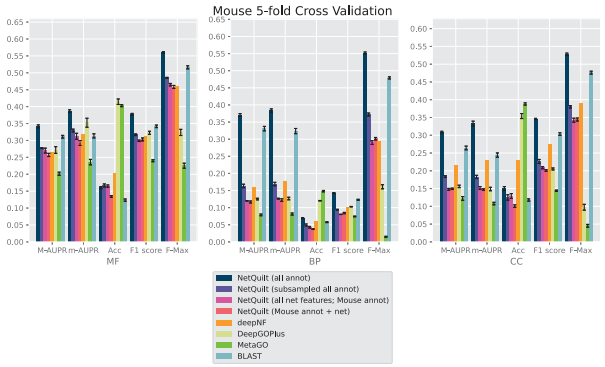

Fig. 5. Performance comparison of NetQuilt method with baselines. Methods shown: NetQuilt trained on model eukaryote annotations; NetQuilt trained on subsampled model eukaryote annotations; NetQuilt trained only on *Musmusculus* examples; single-species NetQuilt (taking only the mouse IsoRank matrix and annotations); deepNF (single-species, but integrating 6 STRING network types); DeepGOPlus (trained on original dataset with our test set proteins removed); MetaGO (predicted with PPI+homology pipelines only, using its default dataset of annotations); and CAFA BLAST annotation transfer method using all selected eukaryote annotations

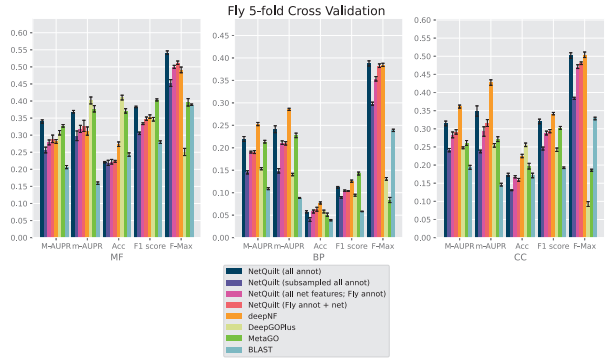

Fig. 6. Performance comparison of NetQuilt method with baselines. Methods shown: NetQuilt trained on model eukaryote annotations; NetQuilt trained on subsampled model eukaryote annotations; NetQuilt trained only on *D.melanogaster* examples; single-species NetQuilt (taking only the fly IsoRank matrix and annotations); deepNF (single-species, but integrating 6 STRING network types); DeepGOPlus (trained on original dataset with our test set proteins removed); MetaGO (predicted with PPI+homology pipelines only, using its default dataset of annotations); and CAFA BLAST annotation transfer method using all selected eukaryote annotations

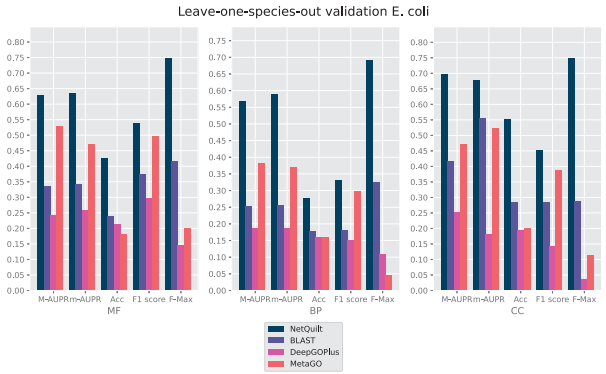

Fig. 7. *E.coli* annotations. Training set included all other species listed in Supplementary Figure S2 besides *E.coli* K-12 substr. MG1655, which was the test organism. No PPI network information of the test organism was used for NetQuilt, BLAST and DeepGOPlus.The PPI-network and homology-based scoring pipelines of MetaGO were used to make predictions with the default data and settings for all evaluations

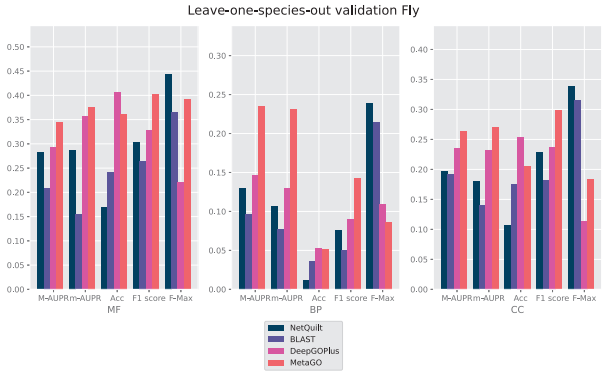

Fig. 8. *D.melanogaster* annotations. Training set included all other species listed in Supplementary Figure S1 besides *D.melanogaster*, which was the test organism. No PPI network information of the test organism was used for NetQuilt, BLAST and DeepGOPlus. The PPI-network and homology-based scoring pipelines of MetaGO were used to make predictions with the default data and settings for all evaluations

deepNF has additional information—the coexpression, cooccurrence, neighborhood, fusion and database networks—in addition to the experimental PPI network from STRING, while our method incorporates only the experimental network and BLAST connections. The performance of the CAFA BLAST baseline method also performs poorly for fly, which reflects the smaller number and magnitude of BLAST connections between fly and the other organisms (see SupplementaryFig. S7 for network and homology comparisons between eukaryotes). Similarly, for biological process and cellular component, both DeepGOPlus and MetaGO perform relatively poorly compared to their performance in molecular function. This indicates that the homology of the organisms in the set does not give as much information as the other sources of information that deepNF takes into account for the fly protein function prediction task. Since our method also relies on homology information, we expect a corresponding decrease in performance when such information is not as salient to the classification task. We see this effect also in the maxout network trained in the subsampled setting, where homology information from the proteins of other organisms is included in the training data at the expense of other proteins in the fly network.

For all organisms, NetQuilt trained only on a single species' annotations performs similarly whether it uses multispecies features or single-species features.

For *E.coli* and human, training on multispecies features gives slightly better performance with regard to the molecular function ontology than training on single-species features. However, for cross-validation on human in the biological process ontology, the multispecies features actually decrease performance.

This is because adding a significantly larger number of features without increasing the number of training examples has limited benefits, with a higher number of parameters needing more samples to train on. On the other hand, both of these baseline models' performances are comparable to that of deepNF for the molecular function ontology for all of the considered organisms. This suggests that the features based on PPI networks integrated with homology through our method can enable the neural network to have competitive performance even without large numbers of training examples.

4.2 Leave-one-species-out validation

In order to explore the performance of our method in a situation in which no PPI interaction network is known for an organism but homology information is present, we present results for E. coli and fly LOSO validation in Figures 7 and 8, and for human and mouse in Supplementary Figures S3 and S4. This setting often describes the case for many newly sequenced species; mass spectrometry or yeast two-hybrid data may not be available for such organisms.

For *E.coli*, we see that our model outperforms the CAFA BLAST labeling method, DeepGOPlus and MetaGO. There are annotations available from all other bacteria, including another well-annotated substrain of *E.coli* (K-12 substr. W3110; see SupplementaryFig. S2). BLAST can use these presumably useful homologs in transferring annotations to the *E.coli* K-12 substr. MG1655, our test organism. However, even with this information, our method outperforms BLAST by more than double in the macro-AUPR performance for biological process, and by similarly large margins in the molecular function and cellular component ontologies. MetaGO does do better than the other two benchmark methods, likely because the *E.coli* PPI network information, which was removed for NetQuilt, is quite relevant to the function prediction task. In addition, MetaGO has access to annotations of some test set proteins, given that the default dataset included with the method was not modified.

For fly, we see NetQuilt generally outperforming the CAFA BLAST labeling method, though for cellular component, the improvement is not as significant. In terms of F-max score, NetQuilt outperforms all other benchmark methods, but MetaGO and DeepGOPlus outperform NetQuilt in the other measures. We note that for MetaGO, the PPI network for fly was not removed, as it was run with its default dataset and settings. This likely contributed to MetaGO's performance, since NetQuilt outperformed MetaGO

when both methods had access to the fly network in the cross-validation setting.

On human and mouse (see Supplementary Figs S3 and S4), our model performs approximately as well as the CAFA BLAST labeling method. The BLAST labeling method performs much better for these organisms than it does for fly and *E.coli*. When homology information is highly informative, as is the case in human and mouse, BLAST is difficult to improve upon. However, in cases where homology is not as informative for the annotation task, the complementary PPI data used by our model allows for significant improvements in performance.

We observe consistent underperformance of DeepGOPlus across E. coli, human and mouse organisms in LOSO which could be explained by the fact the DeepGOPlus was trained only on experimental annotations and the removal of the entire organism greatly impairs its performance. MetaGO, too, relies only on experimental evidence codes to transfer annotations to the test proteins. This could be one reason that both MetaGO and DeepGOPlus perform worse than NetQuilt and the BLAST baseline for human and mouse.

These results show that our method of integrating multiple species' PPI networks and their homology link information can be used effectively to annotate proteins for organisms for which neither PPI network nor annotations are available. In particular, it shows that we can outperform strictly homology-based predictions when there is PPI network information available for species related to the organism we want to annotate.

5 Conclusion

With the arrival of high-throughput experimental techniques came large PPI network datasets of thousands of organisms. Many function prediction algorithms use PPI information for function prediction using a single species at a time. In order to fully exploit this rich source of information, new protein function prediction algorithms should be designed so that multiple PPI networks can be integrated, along with the most abundant source of protein information: homology. We present here a method that is the first of its kind: a multi-species network-based deep learning method for protein function prediction that effectively integrates PPI network information and homology. The integration of multiple PPI networks is based on IsoRank, a PPI network alignment technique that uses homology to transfer topological similarity scores between nodes of different networks. We use the integrated similarity scores as input to a maxout neural network in order to accurately predict protein function. We demonstrate the superiority of our method in Gene Ontology term prediction to single-species network-based approaches, the homology transfer method from the Critical Assessment of Function Annotation (CAFA), the deep learning sequence-based method DeepGOPlus, and the PPI and homology-based pipeline of MetaGO using a cross-validation evaluation.

The multispecies approach enables us not only to produce better predictions in situations involving completing the annotations of a single species using its PPI network, but also to make accurate network-informed predictions on species for which the organism has either an incomplete or an entirely non-existent PPI network. We show this capability through a leave-one-species-out validation whereby we leave out a species' network and annotations and train our model on multiple other species, and then evaluate our function predictions on the left-out species. We show that our method can be at least as good as the CAFA homology transfer method in settings in which homology is very informative, and is a great improvement over the CAFA homology transfer method in settings in which homology information is not enough to produce accurate predictions. We show performance increase in most comparisons to DeepGOPlus and MetaGO under this setting as well.

This method shows promise for training deep learning models on large multispecies PPI network datasets. In light of the informative representations learned by deep-learning algorithms trained on sequence datasets with millions of training examples, we have a vision of applying deep learning techniques similarly to the millions of nodes in all PPI networks. In future work, we hope to explore principled ways of integrating much larger numbers of PPI networks with homology information for function prediction.

Acknowledgements

The authors thank Nicholas Carriero and Ian Fisk of the Flatiron Insitute for discussion and help with high performance computing.

Funding

R.B. and V.G. acknowledge funding from the Simons Foundation. M.B., K.C. and R.B. acknowledge funding from National Science Foundation (NSF) [1922658] and New York University. R.B. and M.B. acknowledge funding from NSF Chemical, Bioengineering, Environmental and Transport Systems (CBET)[CBET-1728858], National Institutes of Health (NIH) Centers for Excellence in Genomic Science [RM1HG011014], NIH [R01HD096770 and R01CA229235]. K.C. acknowledges funding from Samsung Research (Improving Deep Learning using Latent Structure).

Conflict of Interest: The authors declare that there is no conflict of interest regarding this work.

References

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

Chen,B. *et al.* (2014) Identifying protein complexes and functional modules-from static PPI networks to dynamic PPI networks. *Brief. Bioinf.*, **15**, 177–194.

Cho,H. *et al.* (2016) Compact integration of multi-network topology for functional analysis of genes. *Cell Syst.*, **3**, 540–548.e5.

Chollet,F. *et al.* (2015) Keras. https://keras.io (accessed July 2020).

Cozzetto,D. *et al.* (2016) Ffpred 3: feature-based function prediction for all gene ontology domains. *Sci. Rep.*, **6**, 1–11.

Duchi,J. *et al.* (2011) Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, **12**, 2121–2159.

Faisal,F.E. *et al.* (2015) The post-genomic era of biological network alignment. *EURASIP J. Bioinf. Syst. Biol.*, **2015**, 3.

Fan,J. *et al.* (2019) Functional protein representations from biological networks enable diverse cross-species inference. *Nucleic Acids Res.*, **47**, e51.

Friedberg,I. (2006) Automated protein function prediction-the genomic challenge. *Brief. Bioinf.*, **7**, 225–242.

Gligorijević,V. *et al.* (2016) Fuse: multiple network alignment via data fusion. *Bioinformatics*, **32**, 1195–1203.

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

Gligorijevic,V. *et al.* (2019) Structure-based function prediction using graph convolutional networks. *bioRxiv.* DOI: 10.1101/786236.

Gong,Q. *et al.* (2016) Gofdr: a sequence alignment based method for predicting protein functions. *Methods*, **93**, 3–14.

Goodfellow,I.J. *et al.* (2013) Maxout networks. International Conference on Machine Learning. PMLR **28**:1319–1327.

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

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*. San Francisco, California, USA: Association for Computing Machinery (ACM), pp. 855–864.

Hamilton,W.L. *et al.* (2017) Representation learning on graphs: methods and applications. *IEEE Data Engineering Bulletin,* New York, USA: IEEE, vol. 40, pp.52–74.

Ioffe,S. and Szegedy,C. (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. International conference on machine learning. Lille, France: PMLR, vol. 37, pp. 448–456.

Kihara,D. (2016) Computational protein function predictions. *Methods*, **93**, 1–2.

Koo,D.C.E. and Bonneau,R. (2019) Towards region-specific propagation of protein functions. *Bioinformatics*, **35**, 1737–1744.

Kulmanov,M. and Hoehndorf,R. (2020) Deepgoplus: improved protein function prediction from sequence. *Bioinformatics*, **36**, 422–429.

Lee,D. *et al.* (2007) Predicting protein function from sequence and structure. *Nat. Rev. Mol. Cell Biol.*, **8**, 995–1005.

Li,Y. and Patra,J.C. (2010) Genome-wide inferring gene–phenotype relationship by walking on the heterogeneous network. *Bioinformatics*, **26**, 1219–1224.
Liao,C.-S. *et al.* (2009) Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, **25**, i253–i258.
Malod-Dognin,N. and Pržulj,N. (2015) L-graal: Lagrangiangraphlet-based network aligner. *Bioinformatics*, **31**, 2182–2189.
Meng,L. *et al.* (2016) Local versus global biological network alignment. *Bioinformatics*, **32**, 3155–3164.
Milenković,T. and Pržulj,N. (2008) Uncovering biological network function via graphlet degree signatures. *Cancer Inf.*, **6**, CIN.S680.
Mostafavi,S. *et al.* (2008) Genemania: a real-time multiple association network integration algorithm for predicting gene function. *Genome Biol.*, **9**, S4.
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.
Patro,R. and Kingsford,C. (2012) Global network alignment using multiscale spectral signatures. *Bioinformatics*, **28**, 3105–3114.
Pržulj,N. (2007) Biological network comparison using graphlet degree distribution. *Bioinformatics*, **23**, e177–e183.
Radivojac,P. *et al.* (2013) A large-scale evaluation of computational protein function prediction. *Nat. Methods*, **10**, 221–227.
Rentzsch,R. and Orengo,C.A. (2009) Protein function prediction—the power of multiplicity. *Trends Biotechnol.*,**27**, 210–219.
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**, e0118432–21.
Saraph,V. and Milenković,T. (2014) MAGNA: Maximizing Accuracy in Global Network Alignment. *Bioinformatics*, **30**, 2931–2940.
Sharan,R. *et al.* (2007) Network-based prediction of protein function. *Mol. Syst. Biol.*, **3**, 88.
Singh,R. *et al.* (2008) Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proc. Natl. Acad. Sci. USA*, **105**, 12763–12768.
Szklarczyk,D. *et al.* (2017) The STRING database in 2017: quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Res.*, **45**, D362–D368.
Vacic,V. *et al.* (2010) Graphlet kernels for prediction of functional residues in protein structures. *J. Comput. Biol.*, **17**, 55–72.
Valdeolivas,A. *et al.* (2019) Random walk with restart on multiplex and heterogeneous biological networks. *Bioinformatics*, **35**, 497–505.
Vijayan,V. *et al.* (2015) MAGNA++: Maximizing Accuracy in Global Network Alignment via both node and edge conservation. *Bioinformatics*, **31**, 2409–2411.
Wan,C. *et al.* (2019) Using deep maxout neural networks to improve the accuracy of function prediction from protein interaction networks. *PLoSOne*, **14**, e0209958.
You,R. *et al.* (2019) NetGO: improving large-scale protein function prediction with massive network information. *Nucleic Acids Res.*, **47**, W379–W387.
Zhang,C. *et al.* (2018) Metago: predicting gene ontology of non-homologous proteins through low-resolution protein structure prediction and protein–-protein network mapping. *J. Mol. Biol.*, **430**, 2256–2265.
Zhou,N. *et al.* (2019) The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol.*, **20**, 1–23.
Zitnik,M. and Leskovec,J. (2017) Predicting multicellular function through multi-layer tissue networks. *Bioinformatics*, **33**, i190–i198.