Bioinformatics Advances
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

OXFORD

Systems Biology and Networks

MUNDO: Protein Function Prediction Embedded in a Multi-species World

Victor Arsenescu Kapil Devkota Mert Erden Polina Shpilker Matthew Werenski Lenore J. Cowen*

Department of Computer Science, Tufts University, Medford, MA, 02155, USA

***To whom correspondence should be addressed: cowen@cs.tufts.edu**

Associate Editor: XXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Leveraging cross-species information in protein function prediction can add significant power to network-based protein function prediction methods, because so much functional information is conserved across at least close scales of evolution. We introduce MUNDO, a new cross-species co-embedding method that combines a single network embedding method with a co-embedding method to predict functional annotations in a target species, leveraging also functional annotations in a model species network.

Results: Across a wide range of parameter choices, MUNDO performs best at predicting annotations in the mouse network, when trained on mouse and human PPI networks, in the human network, when trained on human and mouse PPIs, and in Baker's yeast, when trained on Fission and Baker's yeast, as compared to competitor methods. MUNDO also outperforms all the cross-species methods when predicting in Fission yeast when trained on Fission and Baker's yeast; however, in this single case, discarding the information from the other species and using annotations from the Fission yeast network alone usually performs best.

Availability: All code is publicly available and can be accessed here: github.com/v0rtex20k/MUNDO

Contact: cowen@cs.tufts.edu

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

1 Introduction

A great many sophisticated and effective algorithms have been developed, based on techniques such as network propagation, to infer function from protein-protein interaction networks (Can *et al.*, 2005; Cao *et al.*, 2014, 2013; Chua *et al.*, 2006; Cowen *et al.*, 2017; Deng *et al.*, 2002; Letovsky and Kasif, 2003; Nabieva *et al.*, 2005; Voevodski *et al.*, 2009). Many of these function prediction methods focus solely on the network of interactions within a single species, using the interconnectivity pattern of Protein-Protein Interaction (PPI) data from that species alone to predict functional labels of proteins of unknown function. One large class of such methods that have recently gained a lot of attention are embedding methods. Embedding methods transform the network topology from a graph into a similarity measure between nodes in a vector space, such that the space around a given node is likely to be enriched for nodes of the same or related function (Choobdar *et al.*, 2019; Grover and Leskovec, 2016; Nelson *et al.*, 2019). For example, Cao *et al.* (2013) use a network-propagation measure, *Diffusion State Distance* (DSD) to embed the network, and then perform a simple k-nearest neighbor algorithm to arrive at its functional predictions. Alternative embeddings, and alternative, more sophisticated classifiers to assign protein function based on the embedded space (see for example, (Grover and Leskovec, 2016)) have also been used.

Until recently, these so-called embedding methods were only designed to leverage known functional label information from a single species. However, by recognizing the course of evolution, one can use homology between orthologous proteins in related species to provide additional information when inferring function (Altschul *et al.*, 1990; Fan *et al.*, 2019; Hamp *et al.*, 2013; Loewenstein *et al.*, 2009; Radivojac *et al.*, 2013a; Singh *et al.*, 2008). Indeed, the most popular and common way to predict the function of an unknown protein is to BLAST (Altschul *et al.*, 1990; Madden, 2013) its sequence against the database of all sequenced proteins in multiple species and transfer functional annotation from its closest annotated match to predict its function. This method leverages the vast power of evolution, but ignores network information entirely. Recently,

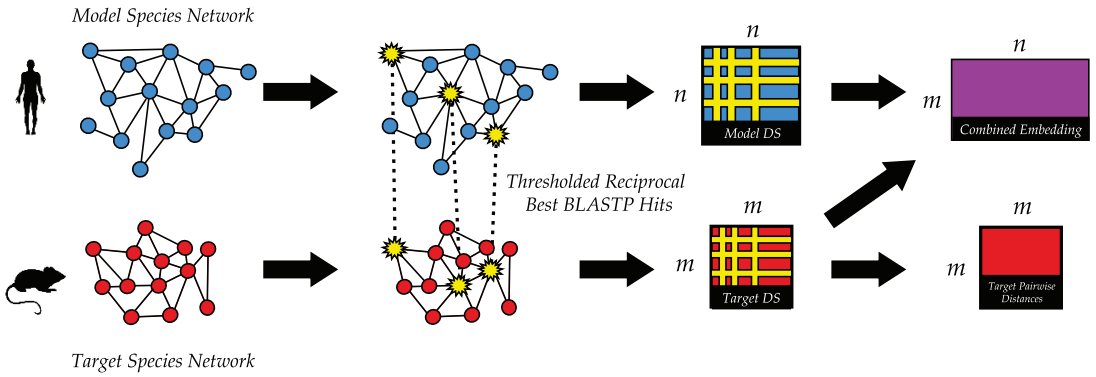

Fig. 1. Overview of each step of the MUNDO algorithm, when leveraging the human PPI network to better predict functional annotations in mouse. The human (blue) and mouse (red) PPI network are given. The red square represents the DSD single-network embedding of the mouse network, and the purple rectangle represents the MUNK co-embedding of the mouse and human networks. MUNDO assigns the functional label of a protein of unknown function based on a weighted vote of its c closest MUNK neighbors and its d closest DSD neighbors. The landmarks are represented by the corresponding yellow stars in each network. The method of using thresholded reciprocal best BLASTP hits to determine the landmarks for the MUNK embedding is described in detail below in Section 3.2. Note that only a relatively small subset of the nodes in each network are paired with nodes in the other, since many nodes' best hits do not meet our stringent thresholds.

Fan *et al.* (2019) introduced MUNK, a novel co-embedding method to transfer functional annotations between multiple species. By treating some of the easiest to recognize orthologous proteins as *landmarks*, identifying them and mapping them together in the embedded space, MUNK is able to *co-embed* multiple PPI networks into the same vector space, combining the power of single-network embedding methods with the possibility of transferring annotation across species.

In this paper, we study how best to utilize the power of co-embeddings to predict GO (Botstein *et al.*, 2000) functional label annotations in one (possibly more sparsely annotated) species by also using the annotations and interactions in a related, (possibly better annotated) species. To test our methods, we mimic the MUNK paper and consider the same pairs of species: *Mus musculus* (mouse) and *H. Sapiens* (human), and the two yeast species *S. cerevisiae* and *S. pombe*. We introduce MUNDO, a method that simultaneously combines functional predictions derived from the single PPI network in the target species with functional predictions derived from the a hybrid embedding; currently we use the MUNK embedding directly as our co-embedding for MUNDO (Figure 1). We tune parameters that determine the relative weight that is given to the function information in the single and hybrid networks, and demonstrate a large range of settings for which MUNDO improves on MUNK alone. We also compare MUNK and MUNDO to a range of simpler baseline approaches that transfer information between the two protein networks on a protein-by-protein basis (see Figure 2). In all four co-species experiments (mouse/human, human/mouse, bakers/fission, fission/bakers) we find that MUNDO performs better than MUNK and the other baseline multi-species approaches. In three of the four experiments MUNDO has the best mean predictive accuracy; in the fourth (bakers/fission) the single network DSD method of Cao *et al.* (2013) which discards all cross-species information outperforms all the tested methods that try to leverage it except MUNDO itself when the training data is the sparsest (only 1/10 of the labels used for training). Finally, we discuss the settings in which MUNDO can be advantageously deployed.

1.1 Algorithm Overview

As shown above, in Figure 1, MUNDO is concerned with two separate embeddings: a single network embedding and a combined embedding. The single network embedding (red box in Figure 1) encodes the relationships between nodes in the target network alone. MUNDO uses the DSD metric introduced by Cao *et al.* (2013) for its single network embedding function prediction method (see Section 2.1) The co-network embedding (purple box in Figure 1), is the MUNK embedding from Fan *et al.* (2019). The input to MUNK is a set of corresponding *landmarks* that are identified to create the co-embedding (yellow starred nodes in Figure 1). We describe the method we use to choose landmarks in Section 3.2 and then describe the resulting MUNK co-embedding component of MUNDO in Section 2.5.1.

1.1.1 Single Network Embedding

The target species PPI network is embedded using the Diffusion State Distance (DSD) metric introduced by Cao *et al.* (2013). DSD captures a fine-grained representation of the local topology of a network through a series of random walks and tends to reduce the influence of hub nodes. A detailed review of DSD is presented in Section 2.1 below.

1.1.2 Multi-Network Embedding

The MUNK algorithm by Fan *et al.* (2019) produces a combined embedding of the model and target species when given a subset of protein pairs to be identified across the networks to act as *landmarks* to join the two networks. We identify landmarks by searching for the reciprocal best BLAST hits, the recommended method from Fan et al's paper. The combined embedding captures the relative similarities between nodes across the two networks, and is based on Fan et al.'s MUNK algorithm which hinged on the use of homologs as *landmarks* to transfer functional information between two PPI networks. The method we use to select MUNK landmarks is described in Section 3.2 and a detailed review of the MUNK co-embedding follows in Section 2.2.

1.1.3 MUNDO Label Assignment

Once the DSD and MUNK embeddings are constructed, MUNDO employs a variation of the simple Majority Vote algorithm (Schwikowski *et al.*, 2000). MUNDO then predicts a GO functional label for each node based on a weighted majority vote among the c closest combined network and d closest DSD single network neighbors in the embedded spaces.

1.2 Paper Outline

The remainder of this paper is structured as follows: Section 2.1 gives a detailed review of DSD, and Section 2.2 gives a detailed review of MUNK. Section 2.3 presents six different alternative methods (in a direct, more simplistic way than MUNK or MUNDO) to predict function taking homology relationships between proteins in the two species into

account. Section 2.4 explores different parameter settings for how *many* neighbors in each of the networks should be voting, as well as how heavily weighted the votes from one network should be compared to the other. In MUNDO and the other methods, we find a large range of parameters where the good performance of MUNDO is quite robust (see Section 4 and supplementary tables). Section 4 compares the performance of MUNDO against standalone DSD (a single network method that does not use cross-species information), MUNK, and the simple cross-species methods. We show that in every case, MUNDO outperforms MUNK and the other cross species methods in a stringent cross-validation experiment. Furthermore, the margin of superiority of MUNDO's performance increases as the size of the available training set decreases (see Section 4). In all but one case MUNDO is the best performer overall; in one case (Baker's yeast as the model species and Fission yeast as the target species), we find the DSD single network method outperforms all the cross-species methods, including MUNDO. Finally we discuss the evolutionary distance between species where a MUNDO approach is currently feasible, and possible extensions to more distant species in Section 5.

2 Methods

2.1 Diffusion State Distance Overview (Cao et al 2013):

Consider the undirected graph $G(V, E)$ on the vertex set $V = \{v_1, v_2, v_3, \ldots, v_n\}$ and $|V| = n$. We define $He^{\{t\}}(A, B)$ to be the expected number of times that a random walk starting at node A and proceeding for t steps, will visit node B. In what follows, assume t is fixed, and when there is no ambiguity in the value of t, we will denote $He^{\{t\}}(A, B)$ by $He(A, B)$.

We define the n-dimensional Diffusion State (DS) vector $He(v_i)$,$\forall v_i \in V$, where

$$He(v_i) = (He(v_i, v_1), He(v_i, v_2), \ldots, He(v_i, v_n)) \quad (1)$$

Then the Diffusion State Distance (DSD) between two vertices u and v, $\forall u, v \in V$ is defined as:

$$DSD(u, v) = ||He(u) - He(v)||_1 \quad (2)$$

where $||He(u) - He(v)||_1$ denotes the l_1 norm of the He vectors of u and v.

In Cao *et al.* (2013) it is proved that DSD is a metric (including satisfying triangle inequality), and furthermore, that DSD converges as the t in $He^{\{t\}}(A, B)$ goes to infinity, allowing us to define DSD independent from the value t.

2.2 MUNK Overview (Fan et al 2019):

Consider a model network $G_1 = (V_1, E_1)$ and a target network $G_2 = (V_2, E_2)$ with $|V_1| = m$ and $|V_2| = n$. MUNK requires in addition *landmark sets* L_1 and L_2 to be specified, with $L_1 \subset V_1$ and $L_2 \subset V_2$, with $|L_1| = |L_2|$, and a bijection between them. We discuss how L_1, L_2 are chosen and how the associated bijection is constructed below.

MUNK constructs kernel (similarity) matrices $D_1 \in \mathbb{R}^{m \times m}$ and $D_2 \in \mathbb{R}^{n \times n}$ corresponding to G_1 and G_2 . Next, it constructs the Reproducing Kernel Hilbert Space (*RKHS*) vector representations C_1 for nodes in the model network G_1 from the factorization $D_1 = C_1 C_1^T$. Let C_{1L} be the subset of the rows of C_1 corresponding to landmarks, and let D_{2L} be the subset of the rows of D_2 corresponding to landmarks (in corresponding order). The key step then is to construct the vector representations of the nodes in the target network G_2. To do this, we treat the similarity scores D_{2L} in the target network as if they applied to the corresponding landmarks in the model network G_1. For a given node in the target network, we want to find a vector for the node such that its inner product with each model landmark vector is equal to its diffusion score to the corresponding target landmark. This implies that the RKHS vectors, $\hat{C}_2$, for nodes in the target network G_2 should satisfy $D_{2L} = C_{1L}\hat{C}_2^T$. This under-determined linear system has solution set

$$\hat{C}_2^T = \hat{C}_{1L}^{\dagger} D_{2L} + (I - \hat{C}_{1L}^{\dagger} C_{1L})W \quad (3)$$

where $\hat{C}_{1L}^{\dagger}$ is the Moore–Penrose pseudo-inverse of C_{1L}, and W is an arbitrary matrix. We choose the solution corresponding to W = 0, meaning that the vectors $\hat{C}_2^T$ are the solutions having minimum norm.

The resulting solution, $\hat{C}_2$, represents the embedding of the nodes of G_2 (the target) into the same space as the nodes of G_1 (the model). We can then compute similarity scores for all pairs of nodes across the two networks as $D_{12} = C_1 \hat{C}_2^T$. This yields D_{12}, an $m \times n$ matrix of similarity scores between nodes in the model and target networks. D_{12} is shown in Figure 1 as the "Combined Embedding" (purple).

2.2.1 Landmark Set Construction

In order to finish specifying the MUNK co-embedding, it is necessary to describe how the landmark sets and landmark bijection is constructed. The set of landmarks is based on *reciprocal best BLAST hits*: for each node u in the target network, we BLAST u (using BLASTP with a BLOSUM62 matrix and word size 3) against all nodes w in the model network, and for each node v in the model network, we BLAST v against all nodes z in the target network. A pair of nodes (u, v) with u in the target network and v in the model network for which u's best BLASTP hit is v and v's best BLASTP hit is u is called a reciprocal best BLAST hit. We set thresholds of percent query coverage and percent sequence identity and keep all the reciprocal best BLAST hits that meet these thresholds as landmarks for MUNK (or the MUNK portion of MUNDO) (see Section 2.4.1 for how we set these parameters in our experiments).

2.3 Simple Homology Transfer Methods

In addition to the MUNK co-embedding method, and the DSD single network method, we also compare MUNDO to six different simple homology transfer methods that use top BLAST hits to transfer functional information between networks in a direct point-to-point fashion. For all six methods: for each node u in the target network, we BLAST u (using BLASTP with a BLOSUM62 matrix and word size 3) against all nodes in the model network. A DS embedding is computed for each network individually, just as in the single network approach discussed above. Finally, pairwise distance matrices for the target and model networks are generated from each embedding matrix. These are depicted above in Figure 2 as the final blue and red matrices, respectively.

Let u refer to the node in the target network whose function we are trying to predict:

1. **Parallel Network Top Blast Hit**: The predicted function for node u is determined by a majority vote amongst its d nearest neighbors in the target network along with u's top BLAST hit v in the model network. Each vote from u's d nearest neighbors is weighted by a factor of α, to v's single vote.

2. **Parallel Network Top Blast Hit + Neighborhood**: The predicted function for node u is determined by a majority vote amongst its d nearest neighbors in the target network, u's top BLAST hit v in the model network, and the c nearest neighbors *of the top BLAST hit* v in the *model* network. Each vote from u's d nearest neighbors is weighted by a factor of α, compared to v's single vote along with the votes of its c nearest neighbors.

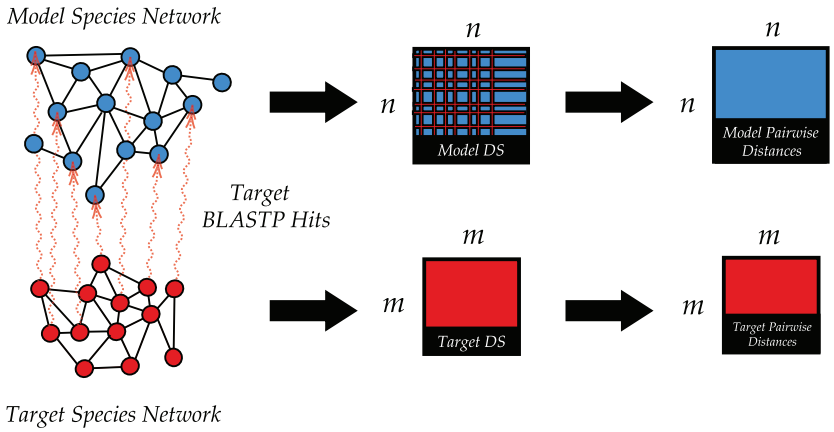

Fig. 2. Simple Homology Transfer Methods. For the alternative simple methods that incorporate homology that we also compare to, the two network embeddings are single network embeddings. In the model (blue) species, the neighborhood of the top BLAST hit in the model (blue) network to the target protein in the target network is combined (in six different proposed methods) with its neighborhood in the target network (red). A DS embedding for each network is then used to generate a pairwise DSD distance matrix for each network. BLAST hits are represented by the connections between red and blue nodes in the leftmost diagram.

3. **Parallel Network All Blast Hits**: The predicted function for node u is determined by a majority vote amongst its d nearest neighbors in the target network and u's first 100 BLAST hits (the default number of hits returned by BLAST). Each vote from u's d nearest neighbors is weighted by α, compared to each of the 100 BLAST hit votes.

4. **Parallel Network All Blast Hits + Neighborhoods**: The predicted function for node u is determined by a majority vote amongst its d nearest neighbors in the target network and u's first 100 BLAST hits (the default number of hits returned by BLAST). Each of the unfiltered one hundred BLAST hits' c nearest neighbors in the *model* network are also included in the vote. Each vote from u's d nearest neighbors is weighted by α, compared to each of the 100 BLAST hit votes and the votes of each hit's c nearest neighbors.

5. **Parallel Network Thresholded Blast Hits**: The predicted function for node u is determined by a majority vote amongst its d nearest neighbors in the target network and all of u's BLAST hits that meet a minimum threshold for query coverage and percent identity are included in the vote. In this case, $p = 85\%$ percent id and $q = 90\%$ query coverage thresholds were used. Each vote from u's d nearest neighbors is weighted by α, compared to each of the thresholded BLAST hits' votes.

6. **Parallel Network Thresholded Blast Hits + Neighborhoods**: The predicted function for node u is determined by a majority vote amongst its d nearest neighbors in the target network and all of u's BLAST hits that meet a minimum threshold for query coverage and percent identity are included in the vote. Furthermore, each of the one thresholded BLAST hits' d nearest neighbors in the *model* network are included. Once again, $p = 85\%$ percent id and $q = 90\%$ query coverage thresholds were used. Each vote from u's d nearest neighbors is weighted by α, compared to each of the thresholded BLAST hits' votes and the votes of each hit's c nearest neighbors.

2.4 MUNDO and Competing Methods Parameters

MUNDO needs to set five parameters: p, q, c, d, and α where p and q concern the quality of the RBH landmarks (for MUNDO and also MUNK; for the simple homology methods they are a threshold on the quality of the ordinary BLAST hits as described in Section 2.3), and thus effect the embeddings, but c, d, α only effect which neighbors vote with what weight. MUNDO, and all the methods we compare MUNDO to, except MUNK, set a parameter d, that represents the size of the DSD neighborhood in the target species network that is considered. MUNDO and MUNK also set a parameter c, which controls the size of the DSD neighborhood in the combined embedding – some of the other competitor methods consider network neighborhood in the source network, and for these we also term the parameter that controls the size of this network neighborhood c.

2.4.1 Reciprocal Blast Hit Landmark Quality

Our setting of p and q ensures that the landmarks are strong reciprocal BLAST hits (RBH) for the human/mouse network co-embedding in both MUNK and MUNDO. We requires that the *coverage* (defined as the proportion of the sequence BLAST aligns) is at least 90% and the *percent sequence identity* is at least 85% . This gives us 309 landmarks between human and mouse. This matches the recommended size and quality of the landmark set in Fan *et al.* (2019). For the two yeast networks, because of the large evolutionary distance between *S. cerevisiae* and *S. pombe*, keeping the same p and q thresholds would only result in 8 landmarks. To increase the number of landmarks it is necessary to relax the thresholds on reciprocal BLAST hits. However, as the thresholds are lowered, the amount of noise in the co-embedding increases, and predictive accuracy is negatively impacted. Therefore, we searched for settings that would keep the percent coverage and sequence identity reasonably high, while giving us at least 60 landmarks (where 60 was chosen to match the mouse/human mapping since the number of proteins in the yeast network is roughly a fifth the size of the number of proteins in our human PPI network - network statistics appear in Section 3.1). We ended up choosing (q, p) = (75%, 50%), resulting in 61 landmarks (We also tried an embedding based on the 79 landmarks obtained with (q, p) set to (50%, 50%), and found that performance was very similar - see Supplement).

2.4.2 Majority Vote Parameters

In order to explore the remaining method parameters d, c and α in a principled way, we randomly split the labeled nodes 50/50 for the first species co-embeddings experiment (human as model species; mouse as target) into separate training and validation sets. Parameters were then explored only on the training set, which was further split into 80% training and 20% testing folds for standard 5-fold cross validation experiments (see Figure 4) We first look at the effect of varying the c and d parameters in *all* methods (noting DSD rewrites $c = 0$, and MUNK rewrites $d = 0$),

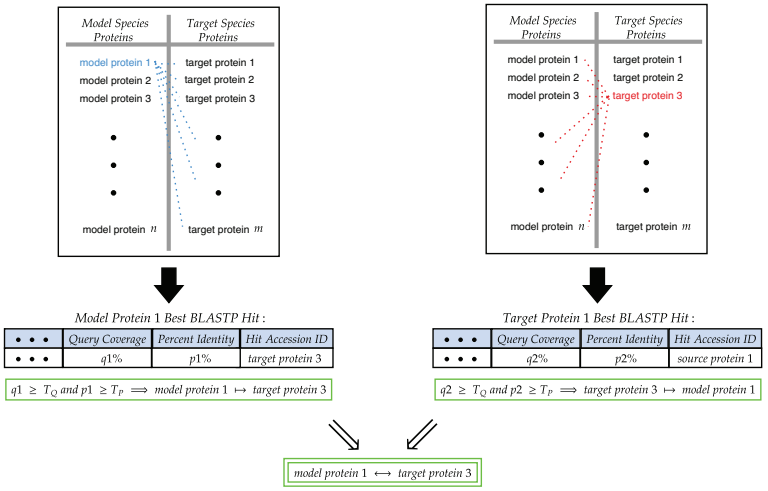

Fig. 3. The schematic above details the landmark selection process, beginning with the identification of reciprocal Best BLAST Hits and ending with their thresholding. Each protein in the target network (shown in red) is BLASTed against all nodes in the model network (shown in blue), and vice versa. A protein from the model species, and its top-ranked BLASTP hit in the target species form a Reciprocal Best BLASTP Hit (RBH) if when the top-ranked BLASTP hit from the target species is BLASTed back against all nodes in the model species, the top-ranked BLASTP hit is the original protein from the target species. RBHs are added to the set of landmarks only if the query coverage and percent identity between the two proteins meet certain thresholds, T_Q and T_P, which are tunable hyperparameters for both MUNK and MUNDO methods (hereafter referred to as q and p, respectively).

by first setting α to be 1.0 in MUNDO, so that votes from the d closest proteins in the individual and c closest neighbors in the combined network had equal weight votes. Table S1 gives the results of a grid search over the parameters $\{5, 10, 20\}$ for both c and d on the training set in 5-fold cross validation. As can be seen in Table S1, over MUNDO and all competitor methods, we find that setting $d = 20$ for the number of neighbors in the target species outperforms all other d settings (note that MUNK does not set the d parameter, only c).

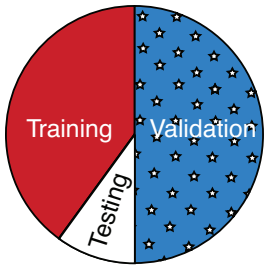

Fig. 4. Illustration of the Train-Test-Validation Split described in Section 2.4. As shown, the dataset was initially split in half, with 50% being reserved for validation. 5-fold cross validation was then performed on the remaining 50% of the data, with 80% of the remaining half being used for training and 20% of the remaining half being used for testing.

We also explored whether weighting votes differently according to whether they came from the d target species neighbors or the c combined species network neighbors would matter. Table S2 in the supplement explores different settings for α, the relative weight that is given to votes from functional labels in the target versus combined networks. Based on this performance, we recommend default MUNDO parameters of $d = 20$, and $\alpha = 1.5$ for human-mouse. We report the performance of MUNDO and all competitor methods over all three c choices in our independent validation set in Table 1 of our main results. As can be seen in Tables 1 and S1-4, MUNDO results were relatively robust to different c values, but $c = 10$ performs slightly better than other settings. We then keep these same parameter values (i.e. $d = 20, c = 10, \alpha = 1.5$) for our mouse/human and both yeast/yeast experiments.

2.5 Full Details of MUNDO

2.5.1 Co-embedding

We give more technical details for how the MUNK co-embedding is computed. Let M be the $n \times n$ matrix consisting of the diffusion state vectors for the model species network and T be the $m \times m$ matrix consisting of the diffusion state vectors for the target species network. These matrices are depicted in Figure 1 as the blue and red square matrices, respectively.

Define $\Lambda = \{\lambda_1, \lambda_2, \ldots \lambda_n\}$ and $\Upsilon = \{v_1, v_2, \ldots, v_n\}$ to be the set of eigenvalues and eigenvectors of the model network M, respectively. We use these to compute the model network embedding N relative to which T will be co-embedded as follows:

$$N = \Upsilon \cdot \Lambda^{1/2}\, \mathbb{I}_n = \begin{bmatrix} | & & | \\ v_1 & \ldots & v_n \\ | & & | \end{bmatrix} \cdot \begin{bmatrix} \lambda_1^{1/2} & 0 & \ldots & 0 \\ 0 & \lambda_2^{1/2} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & \lambda_n^{1/2} \end{bmatrix} \tag{4}$$

Define r_i to be the i^{th} reciprocal best hit for a given network, and let there be p reciprocal best hit pairs in total Let $A_{[i\ldots j]}$ be a matrix comprised of rows $i, i+1, \ldots, j$ of a given matrix A. Finally, let $A^\dagger$ denote the pseudo-inverse (Moore-Penrose inverse (Barata and Hussein, 2012)) of a given matrix A. The target network embedding C relative to the model network embedding N can therefore be computed as:

$$C^T = \left(N^T_{[r_i \ldots r_p]}\right)^\dagger \cdot T_{[r_i \ldots r_p]} \tag{5}$$

In addition to the $m \times n$ Combined Embedding matrix, C, we also compute P, an $m \times m$ pairwise distance matrix derived from T, the matrix of diffusion state vectors. For each pair of m-dimensional vectors $t_i, t_j \in T$, the l_1-norm is computed and stored in P as a scalar value $P_{i,j}$. By definition, P is necessarily square. Matrices C and P are shown in Figure 1 as the rightmost purple and red matrices, respectively.

After sorting both output matrices P and M by distance, we are able to quickly identify the d nearest DSD neighbors and c nearest Combined

Embedding neighbors of each node i in the target network whose function we would like to predict. Respectively, these neighbors can be found at $P_{i,[0\ldots d-1]}$ and $M_{i,[0\ldots c-1]}$.

3 Experimental Setup

3.1 Networks

The Human PPI Network consists of the 25,672 unique nodes and 487,840 unique interaction edges downloaded from BioGRID version 3.5.188, excluding self loops. Of the included edges, 479,400 encoded physical interactions and 8,440 encoded genetic interactions. Taking the largest connected component yields 17,017 nodes and 407,653 edges.

The Mouse PPI Network consists of the 15,979 unique nodes and 74,966 unique interaction edges downloaded from BioGRID version 3.5.188, excluding self loops. Of the included edges, 74,646 encoded physical interactions and 320 encoded genetic interactions. Taking the largest connected component yields 8,543 nodes and 45,605 edges.

The *S. cerevisiae* PPI Network consists of the 4,701 unique nodes and 71,688 unique interaction edges downloaded from BioGRID version 4.2.192, excluding self loops. Of the included edges, 14,008 encoded physical interactions and 57,680 encoded genetic interactions. Taking the largest connected component yields 4,335 nodes and 61,856 edges.

The *S. pombe* PPI Network consists of the 7,335 unique nodes and 607,030 unique interaction edges downloaded from BioGRID version 4.2.192, excluding self loops. Of the included edges, 123,802 encoded physical interactions and 483,228 encoded genetic interactions. Taking the largest connected component yields 5,817 nodes and 538,250 edges.

3.2 Landmark Selection

The quality of the co-embedding performed later in the process depends heavily on the "overlap" between the model and target networks. Traditionally, overlap refers to the set of nodes that exist in both networks. In this case, each network belongs to a distinct species, so no protein can exist in both networks simultaneously. We therefore redefine overlap in this context to refer to the set of reciprocal best BLAST hits between two networks. Any hit which does not meet a minimum similarity threshold set by the user is filtered out. The number of hits can vary greatly between organisms, but experimental results have shown that roughly 300 RBHs are sufficient for a high-quality embedding. This is in line with the minimum number of landmarks recommended in the MUNK (Fan *et al.*, 2019) algorithm. A schematic of the selection process is provided above in Figure 3.

This voting algorithm allows nodes from a different species' network to vote alongside the closest neighbors of a given node, each with their own tunable weight hyperparameters. Once the final list of votes $\mathcal{F}(v)$ has been compiled, the most frequently appearing vote in the list is used as the ultimate prediction for the function of v. Note that this is quite similar to the traditional weighted majority vote algorithm, except each species has its own weight parameter. The optimal values of α, setting the weight of the model species votes as compared to the target species votes, are explored in the supplement.

3.3 Functional Labels

Functional labels for all species were downloaded from EMBL-EBI's Uniprot GOA database, version 201. We considered GO Labels from the Biological Process (BP) and Molecular Function (MF) hierarchies. The set of GO terms was filtered to those in an intermediate range of specificity, retaining GO terms that annotate between between 50 and 500 nodes in the target network for Human $\longleftrightarrow$ Mouse and between 50 and 300 nodes in the target network for *S. pombe* $\longleftrightarrow$ *S. cerevisiae*.

3.4 Performance Measures

We measure the performance of the different methods in three ways. The simplest way, percent accuracy only considers the top prediction for each node. The others consider a ranked list of the top three predictions, which in GO are often more or less specific and related terms: the F1 score, however, does not explicitly take into account the hierarchical structure of GO, whereas the Resnik score measures the information content in reference to the GO hierarchy.

Evaluation Method 1: Percent Accuracy This metric simply measures the percent of nodes whose top predicted functional label is correct, meaning it is among the set of true functional labels assigned to that node.

Evaluation Method 2: Hierarchy Agnostic $F1^*$ Method This evaluation metric, which corresponds to the protein-centric evaluation method in the CAFA challenge (Radivojac *et al.*, 2013b; Zhou *et al.*, 2019), scores a multi-label function prediction set, but still ignores the hierarchical nature of the GO annotations while scoring predictions. For a particular protein i, let T_i be the set representing its true GO annotation and $P_i(\tau)$ represent the set of GO annotations predicted by the Function Prediction method with likelihood greater than the confidence threshold τ. Then, we can compute the precision and recall for the protein i at the threshold τ as

$$\mathrm{prec}_i(\tau) = \frac{|P_i(\tau) \cap T_i|}{|P_i(\tau)|} \tag{6}$$

$$\mathrm{recall}_i(\tau) = \frac{|P_i(\tau) \cap T_i|}{|T_i|} \tag{7}$$

The average precision and recall for a particular confidence threshold τ is:

$$\mathrm{prec}(\tau) = \frac{1}{M}\sum_{i=1}^{M} \mathrm{prec}_{\alpha_\tau(\mathrm{i})}(\tau) \tag{8}$$

$$\mathrm{recall}(\tau) = \frac{1}{N}\sum_{i=1}^{N} \mathrm{recall}_{\mathrm{i}}(\tau) \tag{9}$$

where α_τ represents the set of all proteins which have at least one GO annotation predicted at the confidence interval τ ($\alpha_\tau(i)$ represents its i^{th} member), M is the size of the set $\alpha_\tau(i)$ and N is the total number of proteins in the test set.

We can then compute the $F1$ score at confidence τ, and $F1^*$ as

$$F1(\tau) = 2\frac{\mathrm{prec}(\tau)\cdot\mathrm{recall}(\tau)}{\mathrm{prec}(\tau)+\mathrm{recall}(\tau)} \tag{10}$$

$$F1^* = \max_\tau F1(\tau) \tag{11}$$

3.4.1 Evaluation Method 3: Resnik Similarity Metric

This metric models the hierarchical nature of GO by introducing the information content of a GO-term (Jiang *et al.*, 2016) in the context of its ancestors. Let ℓ be a GO-term and L be the subgraph generated by all its ancestor labels, including ℓ. The information content of ℓ, is defined formally as

$$i(\ell) = -\log(Pr(L)) \tag{12}$$

where the joint probability $Pr(L)$ is computed as

$$Pr(L) = \prod_{v\in L} Pr(v|\mathcal{P}(v)) \tag{13}$$

The term $Pr(v|\mathcal{P}(v))$, v being a GO-term and $\mathcal{P}(v)$ representing the parents of v, denotes the probability that we get v from $\mathcal{P}(v)$ after further

	DSD	Top Blast Hit	Top Blast Hit + Neighbors	All Blast Hits	All Blast Hits + Neighbors	Thresholded Blast Hits	Thresholded Blast Hits + Neighbors	MUNK	MUNDO
$(\mathbf{d}, \mathbf{c}) = (\mathbf{20}, \mathbf{5})$ Accuracy	13.23	14.12	14.07	13.46	13.16	13.67	13.69	10.87	**15.05**
F1-max	10.03	10.67	10.69	10.20	10.01	10.34	10.42	7.55	**11.25**
MF Resnik	2.71	2.73	2.23	2.63	2.07	2.72	2.67	2.52	**2.97**
BP Resnik	2.11	2.25	2.24	2.25	1.83	2.11	2.45	**3.57**	2.45
$(\mathbf{d}, \mathbf{c}) = (\mathbf{20}, \mathbf{10})$ Accuracy	13.23	14.12	13.72	13.46	12.76	13.67	13.58	10.69	**15.12**
F1-max	10.03	10.67	10.42	10.20	9.81	10.34	10.40	8.36	**11.41**
MF Resnik	2.71	2.73	1.99	2.63	1.80	2.72	2.63	2.82	**3.17**
BP Resnik	2.11	2.25	2.49	2.25	2.26	2.11	2.71	**3.79**	2.82
$(\mathbf{d}, \mathbf{c}) = (\mathbf{20}, \mathbf{20})$ Accuracy	13.23	14.12	12.93	13.46	12.11	13.67	13.42	11.15	**14.42**
F1-max	10.03	10.67	9.93	10.20	9.28	10.34	10.27	8.24	**10.87**
MF Resnik	2.71	**2.73**	2.01	2.63	1.93	2.72	2.63	1.94	2.66
BP Resnik	2.11	2.25	2.49	2.25	2.07	2.11	2.59	2.27	**2.76**

Table 1. Mean percent accuracy, F1-max, and Resnik scores reported for H. sapiens → M. Musculus. The performance of each method is reported over the independent validation set (half the size of the original data set), with the entirety of the training set used for labeled nodes. All competing methods shown with $\alpha = 1$, $d = 20$; and $c = \{5, 10, 20\}$ where note that only the "+ Neighbors" columns, MUNK, and MUNDO have a source or combined network in which to set a separate c parameter. MUNDO, based on supplementary table S2 sets $\alpha = 1.5$. MUNDO performs best across the board, except for BP Resnik, where MUNK is often better. For results over all tested parameter settings, see the Supplement.

ontological specialization. Expression (9) can be further simplified using expression (10) to obtain

$$i(\ell) = -\sum_{v \in L} \log(Pr(v|\mathcal{P}(v))) = \sum_{v \in L} ia(v) \tag{14}$$

The term $ia(v)$, referred to as information accretion of the annotation v, denotes the increase in the information obtained through the addition of child GO-term (v) to the set of its parent terms (or $\mathcal{P}(v)$).

Resnik similarity ($\mathrm{res_{tt}}$) between two GO-terms, x and y, is

$$\mathrm{res_{tt}(x, y)} = i(\mathrm{lca}(x, y)) \tag{15}$$

where $\mathrm{lca(x, y)}$ represents the least common ancestor between x and y.

We next extend this similarity measure between GO terms to a similarity metric between two sets of GO-terms, using a similar method to Zhao and Wang (2018). Define $\mathrm{res_{st}}(X, y)$, which takes a GO-set X, and a GO-term y as

$$\mathrm{res_{st}}(X, y) = \max_{x \in X} \mathrm{res_{tt}}(x, y) \tag{16}$$

Then, the Resnik similarity $\mathrm{res_{ss}}$ for GO-sets X and Y can be defined as

$$\mathrm{res_{ss}}(X, Y) = \frac{\sum_{x \in X} \mathrm{res_{st}}(Y, x) + \sum_{y \in Y} \mathrm{res_{st}}(X, y)}{|X| + |Y|} \tag{17}$$

Let Q be the set containing all the test proteins, T_q be the true GO-terms and $P_q(\tau)$ be the predicted GO-terms at the confidence interval τ, for a protein $q \in Q$. We compute:

$$\mathrm{RES} = \max_{\tau} \frac{1}{|Q|} \sum_{q \in Q} \mathrm{res_{ss}}(T_q, P_q(\tau)) \tag{18}$$

We measure the Resnik similarity over both the Molecular Function (MF) and Biological Process (BP) hierarchies of GO terms.

3.5 Inverted Cross Validation Experiment

In the most common form of k-fold cross validation, $k-1$ of the folds are used for *training* and the remaining fold is used for *testing*. Because we were especially interested in performance of function prediction methods when the amount of training data is sparse, we copied the experimental setup of Lazarsfeld *et al.* (2019), which "inverts" the relative sizes of the training and test data, so that, only one of the folds is used for training, and the other $k-1$ have all their annotations removed and are placed in the testing set. Thus, for our inverted cross-validation experiment, the larger k is, the smaller the size of the training set. Mean and standard deviation of percent accuracy, calculated as the percent of time a protein in the test set is assigned a label which is correctly among its true annotations, is computed over 5 different runs of k-fold cross validation, for $k = 2, 4, 6, 10$ in our experiments. In addition, we compute precision and recall by looking at the top r predicted labels (in our case, we present results for $r = 3$) using the method of Deng *et al.* (2004), and report the maximum F1 score according to their first recommended method, and also compute Resnik scores for both the MF and BP hierarchies.

4 Results

Because in our first species experiment (*human* as the model species and *mouse* as the target species) we needed to tune parameters, we did standard 5-fold cross validation to tune those parameters for MUNDO and all its competitors, and then presented the results on the independent validation set. As shown in Table 1, MUNDO with parameters set as described in section 2.4 produces substantially better percent accuracy and F1 Max scores than DSD, MUNK, and all six simple homology methods.

In the other species experiments, we decided to fix MUNDO's parameters to the recommended defaults based on the first species experiment. This eliminates the need for an independent validation set for these experiments. For these experiments, we therefore decided to run *inverted* cross validation experiments (see Section 3.5 about our unusual design of inverted cross-validation: the 10-fold experiment inverts the usual role of training and test sets: with 1 fold used for training and 9 folds used for test). We compared MUNDO to DSD, MUNK, and the top performing variant of the simple methods in the first experiment, namely Top BLAST hit + Neighborhood, in 2-fold, 4-fold, 6-fold and 10-fold inverted cross-validation experiments and results appear in Tables 2-4. MUNDO performs best or second best in all three experiments; it is best by most measures when mouse is the model species and human is the target, and when *pombe* is the model species and *cerevisiae* is the target, but is outperformed by single network DSD in the target species when *cerevisiae* is the model species and *pombe* is the target, except when the size of the training set is set to the smallest size we tested (the 10-fold experiment). Interestingly, we note that the margin of improvement for MUNDO **increases** as the number of proteins that contribute functional labels in the training data **decreases**. Looking at the runner up methods is also interesting: The single network method is slightly more accurate than MUNK when we reserve half the nodes for the training set; MUNK becomes comparable or very slightly more accurate compared to the single network in the 10-fold experiment when the amount of training data is much lower. Top Blast Hit + Neighborhood (Simple Method 2) is found to be best performing of the simple homology transfer methods; outperforming both single network and MUNK (but never MUNDO). All

methods are shown with parameter settings of c, d and α, as noted; (full parameter results appear in the supplement).

		DSD	Top Blast Hit + Neighborhood	MUNK	MUNDO
2-fold	Accuracy	10.98 ± 0.17	10.65 ± 0.35	8.35 ± 0.05	$\mathbf{11.64 \pm 0.20}$
	F1-max	6.48 ± 0.01	6.48 ± 0.06	4.47 ± 0.05	$\mathbf{7.22} \pm 0.01$
	MF Resnik	4.22 ± 0.31	$\mathbf{4.69 \pm 0.01}$	3.76 ± 0.01	4.41 ± 0.28
	BP Resnik	3.81 ± 0.08	3.97 ± 0.26	2.11 ± 0.57	$\mathbf{4.07 \pm 0.46}$
4-fold	Accuracy	9.60 ± 0.17	9.24 ± 0.13	8.34 ± 0.08	$\mathbf{10.94 \pm 0.13}$
	F1-max	5.83 ± 0.06	5.86 ± 0.21	4.46 ± 0.03	$\mathbf{6.62 \pm 0.02}$
	MF Resnik	4.18 ± 0.31	4.12 ± 0.62	3.95 ± 0.13	$\mathbf{4.19 \pm 0.12}$
	BP Resnik	$\mathbf{4.48 \pm 0.16}$	3.32 ± 0.75	2.72 ± 0.27	4.29 ± 0.82
6-fold	Accuracy	8.82 ± 0.13	8.52 ± 0.10	8.28 ± 0.04	$\mathbf{10.39 \pm 0.10}$
	F1-max	5.34 ± 0.12	5.29 ± 0.14	4.46 ± 0.03	$\mathbf{6.39 \pm 0.07}$
	MF Resnik	4.15 ± 0.42	3.81 ± 0.53	4.00 ± 0.13	$\mathbf{4.32 \pm 0.53}$
	BP Resnik	$\mathbf{4.28 \pm 0.84}$	3.67 ± 1.56	2.46 ± 0.34	3.89 ± 0.35
10-fold	Accuracy	7.79 ± 0.09	7.47 ± 0.13	8.30 ± 0.02	$\mathbf{9.86 \pm 0.08}$
	F1-max	4.82 ± 0.14	4.71 ± 0.20	4.45 ± 0.02	$\mathbf{6.08 \pm 0.12}$
	MF Resnik	3.51 ± 1.07	3.31 ± 1.01	$\mathbf{3.97 \pm 0.07}$	3.78 ± 0.46
	BP Resnik	$\mathbf{3.59 \pm 1.12}$	2.61 ± 1.53	2.51 ± 0.38	3.35 ± 0.54

Table 2. Mean percent accuracy, F1-max, and Resnik scores (standard dev.) reported for M. Musculus → H. sapiens over 10 runs of inverted (see Section 3.5) k-fold cross validation. **Method Settings** For all methods: $d = 20$, $c = 10$, $\alpha = 1.5$. Additional parameter settings explored in the supplement.

		DSD	Top Blast Hit + Neighborhood	MUNK	MUNDO
2-fold	Accuracy	$\mathbf{25.90 \pm 0.83}$	20.62 ± 0.52	11.10 ± 0.34	25.14 ± 0.60
	F1-max	21.01 ± 0.22	15.91 ± 0.13	11.96 ± 0.17	$\mathbf{21.02 \pm 0.27}$
	MF Resnik	1.34 ± 0.10	1.33 ± 0.03	1.29 ± 0.08	$\mathbf{1.90 \pm 0.15}$
	BP Resnik	0.93 ± 0.01	$\mathbf{0.94 \pm 0.01}$	0.50 ± 0.06	0.93 ± 0.01
4-fold	Accuracy	$\mathbf{23.22 \pm 0.29}$	17.95 ± 0.38	11.09 ± 0.17	22.77 ± 0.30
	F1-max	$\mathbf{19.15 \pm 0.63}$	14.17 ± 0.51	11.73 ± 0.19	19.04 ± 0.78
	MF Resnik	1.41 ± 0.14	1.41 ± 0.13	1.16 ± 0.02	$\mathbf{1.78 \pm 0.30}$
	BP Resnik	0.87 ± 0.02	$\mathbf{0.89 \pm 0.02}$	0.56 ± 0.06	0.88 ± 0.01
6-fold	Accuracy	$\mathbf{21.94 \pm 0.30}$	16.06 ± 0.27	11.11 ± 0.09	21.49 ± 0.31
	F1-max	17.90 ± 0.57	12.77 ± 0.73	11.72 ± 0.15	$\mathbf{18.02 \pm 0.66}$
	MF Resnik	1.25 ± 0.10	1.27 ± 0.09	1.15 ± 0.01	$\mathbf{1.31 \pm 0.15}$
	BP Resnik	0.82 ± 0.04	$\mathbf{0.84 \pm 0.04}$	0.61 ± 0.06	0.82 ± 0.04
10-fold	Accuracy	19.56 ± 0.29	13.64 ± 0.30	11.07 ± 0.07	$\mathbf{19.74 \pm 0.27}$
	F1-max	15.93 ± 0.66	10.57 ± 0.47	11.70 ± 0.07	$\mathbf{16.68 \pm 0.51}$
	MF Resnik	1.28 ± 0.11	1.27 ± 0.08	1.14 ± 0.01	$\mathbf{1.50 \pm 0.37}$
	BP Resnik	0.73 ± 0.03	$\mathbf{0.75 \pm 0.03}$	0.61 ± 0.04	0.73 ± 0.03

Table 3. Mean percent accuracy, F1-max, and Resnik scores (standard dev.) reported for S. cerevisiae → S. pombe over 10 runs of inverted (see Section 3.5) k-fold cross validation. **Method Settings** For all methods: $d = 20$, $c = 10$, $\alpha = 1.5$ Additional parameter settings explored in the supplement.

4.1 Running time

The most computationally expensive step of MUNDO is to compute the single DSD network embedding of the target species, and the combined MUNK embedding of both species. We make the following remarks: 1) the embedding step only has to be done once for a species pair, 2) we were able to compute both the MUNK and DSD embedding steps for our experiments even using exact, converged DSD, in approximately 8 hours for the human/mouse embedding, and only a couple of hours for the much smaller yeast/yeast networks on a desktop computer: all others steps take seconds to assign functional labels, and 3) if even the exact DSD and combined embedding computations are considered too expensive, we could instead compute an approximate DSD much faster using methods described in Lin *et al.* (2018).

5 Discussion

We have introduced MUNDO, a new co-embedding method that leverages the power of multiple species to improve functional label prediction. We optimized MUNDO for predicting mouse functional labels, when trained on mouse and human PPI networks, but showed that the same parameter settings give good performance when predicting human functional labels (trained on mouse and human) and predicting *S. cerevisiae* labels when trained on both *S. cerevisiae* and *S. pombe*. However, MUNDO did not perform as well as the single network method in predicting *S. pombe* annotations when also including the *S. cerevisiae* network in the co-embedding, and in fact, performance degraded for all the methods we compared against that tried to incorporate the second species for this case (except for MUNDO when looking at the sparsest number of training labels, where MUNDO actually slightly outperformed the single network method). It is not clear why that might be, but the most likely explanation is that the *S. cerevisiae* data is somehow noisier or differently structured: we note (see Section 3.1) that we included genetic interactions as well as physical interactions in our PPI networks, but *S. cerevisiae* has the largest proportion of genetic interactions, so if they distort the network co-embedding, this could be a possible explanation for the weaker performance. We note that the performance under all three measures we consider largely followed the same trends; in cases with more training data, MUNDO's Resnik score was sometimes only second-best (in these cases the best performer was split among all the other methods).

		DSD	Top Blast Hit + Neighborhood	MUNK	MUNDO
2-fold	Accuracy	8.64 ± 0.31	9.04 ± 0.54	9.39 ± 0.51	$\mathbf{9.62 \pm 0.57}$
	F1-max	4.44 ± 0.18	4.46 ± 0.27	3.23 ± 0.05	$\mathbf{4.59 \pm 0.01}$
	MF Resnik	1.06 ± 0.33	$\mathbf{1.09 \pm 0.29}$	0.55 ± 0.01	0.69 ± 0.01
	BP Resnik	0.39 ± 0.02	$\mathbf{0.45 \pm 0.02}$	0.39 ± 0.02	0.40 ± 0.01
4-fold	Accuracy	8.35 ± 0.32	8.43 ± 0.24	9.20 ± 0.15	$\mathbf{9.80 \pm 0.41}$
	F1-max	4.54 ± 0.37	4.49 ± 0.08	3.26 ± 0.10	$\mathbf{4.76 \pm 0.16}$
	MF Resnik	0.82 ± 0.25	$\mathbf{0.93 \pm 0.25}$	0.56 ± 0.01	0.69 ± 0.01
	BP Resnik	$\mathbf{0.67 \pm 0.43}$	0.51 ± 0.15	0.40 ± 0.01	$\mathbf{0.67 \pm 0.43}$
6-fold	Accuracy	8.50 ± 0.24	8.33 ± 0.17	9.30 ± 0.13	$\mathbf{9.93 \pm 0.27}$
	F1-max	4.49 ± 0.34	4.44 ± 0.19	3.23 ± 0.09	$\mathbf{4.73 \pm 0.20}$
	MF Resnik	0.72 ± 0.07	0.91 ± 0.27	0.56 ± 0.01	$\mathbf{0.97 \pm 0.39}$
	BP Resnik	0.40 ± 0.08	0.62 ± 0.27	0.40 ± 0.01	$\mathbf{0.41 \pm 0.06}$
10-fold	Accuracy	8.30 ± 0.15	8.25 ± 0.18	9.29 ± 0.05	$\mathbf{9.94 \pm 0.13}$
	F1-max	4.46 ± 0.23	4.34 ± 0.24	3.26 ± 0.08	$\mathbf{4.61 \pm 0.22}$
	MF Resnik	0.96 ± 0.46	0.97 ± 0.38	0.56 ± 0.01	$\mathbf{0.98 \pm 0.45}$
	BP Resnik	0.39 ± 0.04	$\mathbf{0.49 \pm 0.19}$	0.40 ± 0.01	0.39 ± 0.04

Table 4. Mean percent accuracy, F1-max, and Resnik scores (standard dev.) reported for S. pombe → S. cerevisiae over 10 runs of inverted (see Section 3.5) k-fold cross validation. **Method Settings** For all methods: $d = 20$, $c = 10$, $\alpha = 1.5$. Additional parameter settings explored in the supplement.

We are also interested in applying MUNDO to other PPI networks and other pairs of species, but we note that the limiting factor, for both MUNDO and its predecessor MUNK, is finding pairs of species that are sufficiently evolutionarily close that a robust set of landmarks can be found to accomplish the mapping. Using the proposed RBH method, this is not currently possible, for example to map between human and yeast; the species distance is just too great. It is an interesting area of open research to determine if other methods of landmark selection that relax the RBH condition can be learned to map between more distant species in order to pursue a MUNDO approach. One strategy would be to adapt existing algorithms for the global network alignment problem (such as (El-Kebir *et al.*, 2015; Hashemifar and Xu, 2014; Kuchaiev and Pržulj, 2011; Neyshabur *et al.*, 2013; Patro and Kingsford, 2012; Sahraeian and Yoon, 2013; Singh *et al.*, 2008; Vijayan *et al.*, 2015), or see Guzzi and Milenković (2018) for a recent survey), most of which construct their overall alignment in two steps; first they compute a global similarity score, and then they apply some sort of exact or heuristic weighted matching algorithm to produce the correspondence. The most straightforward approach would be to take matched node pairs whose similarity score is above some threshold

as the landmark set. We will explore the performance of various global network alignment methods and similarity measurement and matching strategies to produce good landmarks for MUNDO embeddings of more evolutionary distant species in future work.

MUNDO is freely available on the public facing GitHub repository, together with the networks and embeddings discussed in this paper. MUNDO is easy to run on all your favorite PPI networks, since the software currently supports a wide range of popular PPI network formats, including BioGRID, BioPlex, DIP, GeneMANIA, GIANT, HumanNET, Reactome, and STRING. Each database has its own disjoint set of protein identifiers called its *namespace*, making it difficult to directly compare one network to another. To allow MUNDO to compare proteins in different namespaces, all protein identifiers are mapped to the Refseq namespace (Pruitt *et al.*, 2002), which is the format used by NCBI's BLAST suite. Depending on the database of origin of each species, proteins are either directly mapped to Refseq from their original format or indirectly mapped to the ubiquitous Uniprot (Apweiler *et al.*, 2004) namespace, and then from Uniprot to Refseq. Once all the proteins in both networks have been BLASTed against one another, their original Uniprot identifiers are reused to map each node to its appropriate GO labels.

Acknowledgements

Thanks to the Tufts Bioinformatics and Computational Biology Group for helpful feedback. We additionally thank the anonymous referees for suggestions that greatly improved the paper.

Funding

This work was supported by the National Science Foundation [grant numbers 1812503, 1934553]

References

Altschul, S. F. *et al.* (1990). Basic local alignment search tool. *Journal of Molecular Biology*, **215**(3), 403–410.

Apweiler, R. *et al.* (2004). UniProt: the universal protein knowledgebase. *Nucleic Acids Research*, **32**(suppl_1), D115–D119.

Barata, J. C. A. and Hussein, M. S. (2012). The Moore–Penrose pseudoinverse: A tutorial review of the theory. *Brazilian Journal of Physics*, **42**(1-2), 146–165.

Botstein, D. *et al.* (2000). Gene Ontology: tool for the unification of biology. *Nature Genetics*, **25**(1), 25–9.

Can, T. *et al.* (2005). Analysis of protein-protein interaction networks using random walks. In *Proceedings of the 5th International Workshop on Bioinformatics*, pages 61–68.

Cao, M. *et al.* (2013). Going the distance for protein function prediction: a new distance metric for protein interaction networks. *PloS One*, **8**(10), e76339.

Cao, M. *et al.* (2014). New directions for diffusion-based network prediction of protein function: incorporating pathways with confidence. *Bioinformatics*, **30**(12), i219–i227.

Choobdar, S. *et al.* (2019). Assessment of network module identification across complex diseases. *Nature Methods*, **16**(9), 843–852.

Chua, H. N. *et al.* (2006). Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. *Bioinformatics*, **22**(13), 1623–1630.

Cowen, L. *et al.* (2017). Network propagation: a universal amplifier of genetic associations. *Nature Reviews Genetics*, **18**(9), 551.

Deng, M. *et al.* (2002). Prediction of protein function using protein-protein interaction data. In *Proceedings. IEEE Computer Society Bioinformatics Conference*, pages 197–206. IEEE.

Deng, M. *et al.* (2004). Mapping gene ontology to proteins based on protein–protein interaction data. *Bioinformatics*, **20**(6), 895–902.

El-Kebir, M. *et al.* (2015). Natalie 2.0: sparse global network alignment as a special case of quadratic assignment. *Algorithms*, **8**(4), 1035–1051.

Fan, J. *et al.* (2019). Functional protein representations from biological networks enable diverse cross-species inference. *Nucleic Acids Research*, **47**(9), e51–e51.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. *Association for Computing Machinery*, pages 855–864.

Guzzi, P. H. and Milenković, T. (2018). Survey of local and global biological network alignment: the need to reconcile the two sides of the same coin. *Briefings in bioinformatics*, **19**(3), 472–481.

Hamp, T. *et al.* (2013). Homology-based inference sets the bar high for protein function prediction. In *BMC Bioinformatics*, volume 14-S3, page S7. Springer.

Hashemifar, S. and Xu, J. (2014). Hubalign: an accurate and efficient method for global alignment of protein–protein interaction networks. *Bioinformatics*, **30**(17), i438–i444.

Jiang, Y. *et al.* (2016). An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biology*, **17**(1), 184.

Kuchaiev, O. and Pržulj, N. (2011). Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, **27**(10), 1390–1396.

Lazarsfeld, J. *et al.* (2019). Majority vote cascading: A semi-supervised framework for improving protein function prediction. In *Proceedings of the 10th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*, pages 51–60. Association for Computing Machinery.

Letovsky, S. and Kasif, S. (2003). Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, **19**(suppl_1), i197–i204.

Lin, J. *et al.* (2018). Computing the diffusion state distance on graphs via algebraic multigrid and random projections. *Numerical Linear Algebra with Applications*, **25**(3), e2156.

Loewenstein, Y. *et al.* (2009). Protein function annotation by homology-based inference. *Genome biology*, **10**(2), 1–8.

Madden, T. (2013). The BLAST sequence analysis tool. In *The NCBI Handbook [Internet]. 2nd edition*. National Center for Biotechnology Information (US).

Nabieva, E. *et al.* (2005). Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, **21**(suppl_1), i302–i310.

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

Neyshabur, B. *et al.* (2013). Netal: a new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics*, **29**(13), 1654–1662.

Patro, R. and Kingsford, C. (2012). Global network alignment using multiscale spectral signatures. *Bioinformatics*, **28**(23), 3105–3114.

Pruitt, K. *et al.* (2002). The reference sequence (RefSeq) database. *The NCBI Handbook*, **2**.

Radivojac, P. *et al.* (2013a). A large-scale evaluation of computational protein function prediction. *Nature Methods*, **10**(3), 221–227.

Radivojac, P. *et al.* (2013b). A large-scale evaluation of computational protein function prediction. *Nature Methods*, **10**(3), 221–227.

Sahraeian, S. M. E. and Yoon, B.-J. (2013). Smetana: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. *PloS one*, **8**(7), e67995.

Schwikowski, B. *et al.* (2000). A network of protein–protein interactions in yeast. *Nature Biotechnology*, **18**(12), 1257–1261.

Singh, R. *et al.* (2008). Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, **105**(35), 12763–12768.

Vijayan, V. *et al.* (2015). Magna++: Maximizing accuracy in global network alignment via both node and edge conservation. *Bioinformatics*, **31**(14), 2409–2411.

Voevodski, K. *et al.* (2009). Spectral affinity in protein networks. *BMC Systems Biology*, **3**(1), 1–13.

Zhao, C. and Wang, Z. (2018). GOGO: An improved algorithm to measure the semantic similarity between gene ontology terms. *Scientific Reports*, **8**(1), 15107.

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 Biology*, **20**(1), 244.