Published online XX XXXX XXXX

Nucleic Acids Research, XXXX, Vol. XX, No. XX ***1–18***
doi:10.1093/nar/gkn000

Functional Protein Representations from Biological Networks Enable Diverse Cross-Species Inference

Jason Fan [1], Anthony Cannistra [2], Inbar Fried [3], Tim Lim [4], Thomas Schaffner [5], Mark Crovella [4], Benjamin Hescott [6,†], and Mark D.M. Leiserson[1,†*]

[1]Department of Computer Science and Center for Bioinformatics and Computational Biology, University of Maryland, College Park, [2]Department of Biology, University of Washington, [3]University of North Carolina Medical School, [4]Department of Computer Science, Boston University, [5]Department of Computer Science, Princeton University, and [6]College of Computer and Information Science, Northeastern University
[†]Equal contribution

Received XXX; Revised XXX; Accepted XXX

ABSTRACT

Transferring knowledge between species is key for many biological applications, but is complicated by divergent and convergent evolution. Many current approaches for this problem leverage sequence and interaction network data to transfer knowledge across species, exemplified by network alignment methods. While these techniques do well, they are limited in scope, creating metrics to address one specific problem or task.

We take a different approach by creating an environment where multiple knowledge transfer tasks can be performed using the same protein representations. Specifically, our kernel-based method, MUNK, integrates sequence and network structure to create functional protein representations, embedding proteins from different species in the same vector space. First we show proteins in different species that are close in MUNK-space are functionally similar. Next, we use these representations to share knowledge of synthetic lethal interactions between species. Importantly, we find that the results using MUNK-representations are at least as accurate as existing algorithms for these tasks. Finally, we generalize the notion of a phenolog ("orthologous phenotype") to use functionally similar proteins (i.e. those with similar representations). We demonstrate the utility of this broadened notion by using it to identify known phenologs and novel non-obvious ones supported by current research.

*To whom correspondence should be addressed: mdml@cs.umd.edu

INTRODUCTION

A primary challenge of research with model organisms is to transfer knowledge of genetics – i.e. a mapping of genotype to phenotype – between model organisms and humans. The main promise of researching model organisms stems from researchers' ability to measure the organisms in ways that are infeasible in humans. To realize the promise of this research, it is crucial to transfer knowledge between species – ideally, in two directions. First, discoveries in model organisms can be transferred to improve knowledge of human genetics (e.g. via homology). Second, knowledge of human genetics can be transferred to design better experiments in model organisms (e.g. for disease models).

More specifically, cross-species knowledge transfer can enable a wide variety of applications. First and foremost is the large-scale annotation of protein function by transferring function annotations (e.g. from the Gene Ontology (3)). Addressing this problem remains valuable, even in the era of high-throughput genomics, as fewer than 1% of protein sequences in Uniprot have experimentally-derived functional annotations (59). Another application of cross-species knowledge transfer is for *pairwise* gene function (*genetic interactions*). Knowledge of synthetic lethal genetic interactions is crucial for the study of functional genomics and disease (44, 46). Since genome-wide measurement of synthetic lethal interactions in humans is currently infeasible, computationally transferring knowledge of these interactions from model organisms (such as yeast or mouse) to humans (and human cancers) has become a focus of recent research. A third but less well-explored application is in predicting human disease models through 'orthologous phenotypes' or *phenologs* (37). McGary, et al. (37) reasoned that while conserved genes may retain their *molecular* functions across species, conserved molecular function may manifest as different "species-level" phenotypes. As such, they introduced a statistical test to identify such phenologs.

Cross-species knowledge transfer is quite challenging because many model organisms diverged from humans millions of years ago and have fundamentally different genetic architectures. In many cases, only a relatively small subset of genes between species have sequence homologs. Further, as species diverge, protein functions change and are re-purposed

(e.g. (19)) through divergent and convergent evolution, and genetic interactions are often rewired (53, 58).

Existing computational approaches to transfer knowledge across species rely on matching a subset of genes (proteins) in different species by heredity (genetic orthology) or function (functional orthology). One class of computational approach uses sequence data to match genes (proteins) (2, 54). A second class of methods expands beyond sequence by using proteomics data to match proteins, through protein structure prediction (e.g. (25)) or alignment of protein-protein interaction networks (26, 35, 36, 39, 43, 55), commonly called the network alignment problem.

Many cross-species biological problems cannot be formulated as a matching problem; for example, genetic interactions and phenologs are fundamentally measures of *sets* of genes. This motivates the idea of creating general-purpose multi-species protein representations. These in turn could be used to generate a matching, but could also be interpreted as a vector space or used as input to a learning algorithm. General-purpose representations are fast becoming adopted in different areas of machine learning, from natural language processing (e.g. (38)) to network science (e.g. (22)), and recently have begun to be adopted for biological networks (10, 20).

However, the problem of learning multi-species protein representations from network and sequence data remains largely unexplored. Jacunski, et al. (27) showed that protein representations derived from graph theoretic measures of network structure can be used to transfer knowledge of synthetic lethal interactions across species. However, their approach creates the representations in each network independently, does not use sequence data at all, and uses a set of handcrafted features chosen for a particular task. Gligorijević et al (21) use matrix factorization based on sequence similarity with PPI-based Laplacian smoothing to cluster cross-species protein pairs. However that method does not embed nodes in a common vector space, instead computing scores for a subset of protein pairs that are used as an input to max-weight matching for network alignment. More recently, Khurana, et al. (29) developed an embedding for proteins in multiple species for an application concerning neurodegenerative diseases.

Contributions

In this work, we address the limitations of task-specific protein representations, in a way that allows us to move beyond simple matching of proteins across species. We combine protein-protein interaction networks and sequence data from multiple species into unified, biologically meaningful protein representations using network diffusion. Network diffusion is a natural tool for capturing aspects of local and global network structure that correlate with functional similarity of nodes (14). The key insight of our approach is that homologous proteins can serve as *landmarks* for relating proteins in different species. We then show the similarity scores derived from these representations as well as the representations themselves are useful for distinct tasks.

Our method makes only two assumptions. First, it assumes that protein function can be captured using a similarity score that is a *kernel*, which encompasses a broad class of useful metrics. Second, it assumes that sequence homology is known for some subset of landmark proteins across the different species.

In this paper, we use a *diffusion* kernel to create functional protein representations and call the resulting method MUNK *(MUlti-Species Network Kernel)*. We then evaluate the MUNK protein representations on three multi-species tasks.

1. **Multi-species functional similarity.** We show that cross-species matchings and similarity scores derived from the MUNK representations are significantly correlated with cross-species protein function, and achieve comparable performance to two existing network alignment matching methods. We also perform a proof-of-concept experiment using MUNK to predict protein function in humans by simultaneously leveraging networks from humans and multiple model organisms.

2. **Multi-species synthetic lethality.** We train classifiers on MUNK-representations for *pairs* of genes in order to predict synthetic lethal interactions (SLI) in multiple species. We find that classifiers accurately identify SLI in multiple species *simultaneously*, and that they achieve comparable performance to the SINATRA algorithm.

3. **Phenologs (orthologous phenotypes).** We generalize the notion of orthologous phenotypes beyond evolutionarily conserved sequence with a broader definition of functional similarity. We then identify phenologs between human and mouse that are statistically significant for this broader similarity level. A subset of our predicted phenologs match those identified in (37); additionally, we also predict many new phenologs and support these new predictions with biological literature.

Together, these tasks encompass transferring knowledge both between model organisms, and between model organisms and humans.

MATERIALS AND METHODS

The central contribution we make in this paper is to introduce *network-based functional representations for proteins in different species.* MUNK, as shown in Figure 1, leverages properties of kernel functions as tools for measuring the similarity of nodes in a network and for creating embeddings. While the use of kernels for the study of individual networks is well known (14), it remains an open problem to construct network-based kernels that capture the similarity of nodes between *different* networks. This is the challenge that MUNK addresses.

Starting from a given kernel (node similarity function), and given the PPI networks for a source and a target species, MUNK starts by performing a kernel embedding of the source species nodes (proteins). That is, for each node v_i in the source species, MUNK computes a vector $\phi(v_i)$ such that the vector inner product for any two nodes is equal to the kernel similarity of the nodes. The vectors $\phi(v_i)$ can be thought of as an embedding of the source network into a geometric space.

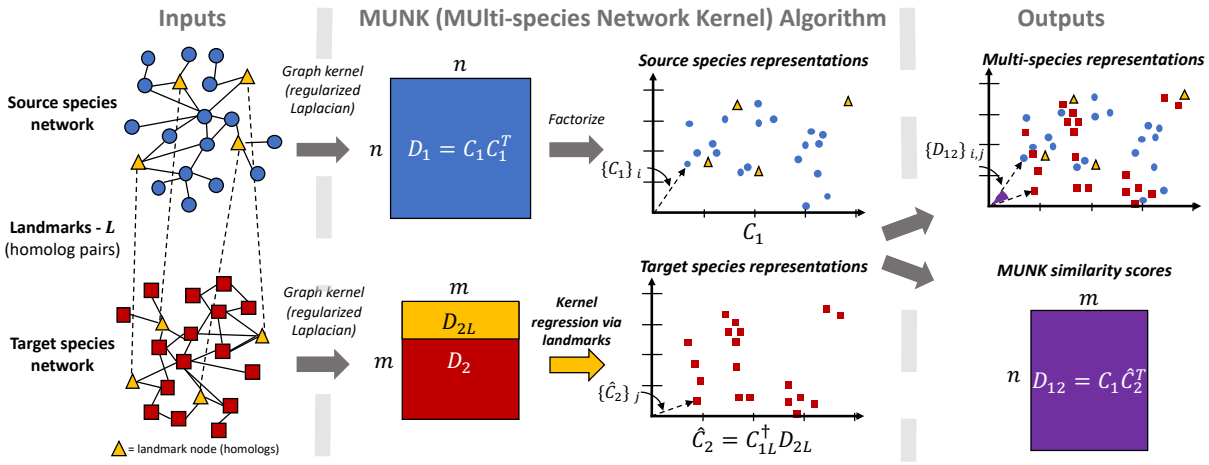


Figure 1. Given a source PPI network, a target PPI network, and a set of landmark (homolog) pairs across species, MUNK computes diffusion kernels for each network. Then, MUNK factorizes the diffusion kernel for the source species into its reproducing kernel Hilbert space (RKHS). Finally, MUNK solves a linear system of the source species' RKHS and the target species' diffusion kernel to create a multi-species vector embedding of source and target proteins. The inner products of these embeddings correlate with functional similarities and the embeddings themselves allow for functional comparisons between proteins across the two networks.

The key step in MUNK is to also embed the nodes of the target species in the *same* vector space, thus creating unified multi-species protein representations. MUNK does this through the use of landmarks – nodes that are known to be the same in both species. For PPI networks, homologous proteins play the role of landmarks. MUNK then places target nodes in the vector space so as to capture their similarity to the landmark nodes.

The result is a joint embedding of both networks in the same vector space. Because both source and target nodes are placed in a way that captures similarity to the same set of landmarks, it becomes possible to score and estimate similarities between nodes in different species. It is also useful to create representations for pairs of proteins, in which case we simply add together the embeddings of two proteins. More details and rationale are provided below.

While MUNK can be used with any network kernel, in this study, we use the regularized Laplacian network kernel in order to capture functional similarity between proteins in different species. The regularized Laplacian kernel is a natural choice for this task because of its close relationship to the principle of "guilt-by-association" often used by protein function prediction methods (34), and to network diffusion methods (e.g. see (60)).

Multi-Species Network Kernel (MUNK) Embedding

We start by noting that there are a large variety of kernels derived from networks (30), (17, Ch. 2), and they can model processes such as random walks, heat diffusion, PageRank, electrical resistance, and other ways of capturing node similarity in a network. Many kernels derived from networks have been applied successfully for a wide range of problems associated with biological network analysis (e.g., see review in (14)).

Though many previous studies have used graph kernels to compare nodes *within* biological networks, to our knowledge, few methods have utilized kernels to fulfill the goal of comparing nodes *across* multiple biological networks. To do so, MUNK relies on a basic property of a kernel: any kernel is also an *inner product* in a particular space. That is, for any kernel $\kappa(\cdot,\cdot)$, there is a function $\phi(\cdot)$ that assigns vectors to nodes such that that $\kappa(i,j) = \phi(i)^T \phi(j)$. The corresponding vector space (termed the *reproducing kernel Hilbert space* (RKHS)) introduces a geometric interpretation for the kernel function. In the context of a kernel for network nodes, the RKHS representation can be thought of as an *embedding* of the network into a vector space in a manner that captures node similarity via inner product.

Conceptually, MUNK approaches the multi-network challenge by constructing a *joint embedding* of the nodes of two (or more) networks in a *single* RKHS. The key to the MUNK method is that, within this RKHS, the similarity of nodes from *different* networks is still captured by inner product, resulting in a *multi-network kernel*.

Given a *source* network G_1, a *target* network G_2, and a kernel κ, the strategy taken by MUNK is to start by embedding the nodes of the source network G_1 using the associated function, ϕ. As described above, this means that inner product between embedded G_1 nodes will capture similarity as described by κ. Next, MUNK makes use of *landmarks* – pairs of nodes in the source and target networks with identical function. The idea is to embed the nodes of the target network G_2 into the same space as the nodes of the source network G_1, such that their position in that space reproduces their similarity to the landmarks of G_2. Essentially, we posit that locating a node from G_2 based on its similarity to the set of landmarks in G_1 will also establish its similarity with the *non-landmark* nodes in G_1. As a result, MUNK creates a multi-network kernel – a single kernel function that captures both the similarity of nodes to each other in the source network G_1, and the similarity of nodes between the source and target networks G_1 and G_2.

This is a fundamentally different strategy than has been used in past manifold-alignment methods (24, 61), in which alignment is based on Euclidean distance. Aligning on Euclidean distance does not respect inner product, and so the similarity captured by the kernel is not preserved in the alignment.

We now define the MUNK approach formally. Let the matrix $K \in \mathbb{R}^{n \times n}$ hold the values of the similarity function $\kappa(i,j)$ for all pairs of n proteins from a particular species. For any such kernel matrix, we can write $K = CC^T$ where C is an $n \times k$ matrix, uniquely defined up to an orthogonal transformation, with $k \leq n$. This follows from the fact that K is positive semidefinite, and means that $\kappa(i,j) = c_i^T c_j$, where c_i is the ith row of C, represented as a column vector. As explained above, the similarity between nodes v_i and v_j is exactly given by the inner product of their corresponding vectors, c_i and c_j.

Now consider a source network $G_1 = (V_1, E_1)$ and a target network $G_2 = (V_2, E_2)$ with $|V_1| = m$ and $|V_2| = n$. We assume the existence of some (small set of) nodes that correspond between G_1 and G_2. In the case where G_1 and G_2 are PPI networks, these can be orthologous proteins. For example, for orthologous proteins in different networks, it is well known that evolutionary rates differ over a wide range of magnitudes (6). Some proteins are highly conserved, and their orthologs will have substantial sequence similarity between G_1 and G_2. Thus, there is generally a small subset of proteins that can be confidently mapped between G_1 and G_2 based on the magnitude and uniqueness of the similarity of their sequence information which we refer to as landmarks.

We then proceed as follows. First, we construct 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, we construct RKHS vector representations C_1 for nodes in the source 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 landmarks in the source 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 *source* 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 underdetermined linear system has solution set,

$$\hat{C}_2^T = C_{1L}^{\dagger} D_{2L} + (I - C_{1L}^{\dagger} C_{1L}) W, \qquad \textbf{(1)}$$

where $C_{1L}^{\dagger}$ is the Moore-Penrose pseudoinverse 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 source). 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 source and target networks.

MUNK and the Regularized Laplacian

While our method can be used with any kernel, in this paper we focus on using a kernel intended to capture functional similarity of proteins. To motivate our choice of kernel function $k(\cdot,\cdot)$, we consider the function prediction problem on a single network, $G = (V,E)$, with $|V| = n$, where G has adjacency matrix A with entries a_{ij}. For simplicity we consider G to be unweighted, so $a_{ij} \in \{0,1\}$. Extensions of our arguments to weighted graphs are straightforward.

A central idea used throughout network-based functional prediction methods is that of *guilt by association* – that is, two nodes that are near each other are more likely to share the same label than two nodes that are far apart. In the context of protein function prediction, this principle has been well established. For example, the authors in (34) show that two neighbors in the protein interaction network are more likely to have the same function than a randomly chosen pair.

Consider the case of determining whether nodes should receive a particular function label where we label a node with a 1 if it should receive the label, and 0 otherwise. We are interested in the case in which the label is rare, and we believe that nodes may be mislabeled (e.g., some nodes labeled 0 should actually be labeled 1). We assume that there is some current labeling which is incomplete; that is, most nodes are currently labeled 0, and some nodes labeled 0 should actually be labeled 1. Define the vector y such that $y_i = 1$ if node v_i has label 1, and zero otherwise. The goal of the function prediction problem is to estimate a new $\hat{y}$ that is a better labeling of nodes in V.

To address this problem, we proceed as follows (4, 16). First, we posit that y should not differ too much from $\hat{y}$ – nodes should tend to be given the labels they already have. Second, we also posit that neighbors in G should tend to be given the same label – this is the guilt by association principle.

Note that these two goals are in conflict: fully following the first principle leaves all labels unchanged, while fully following the second principle makes all labels the same (either 0 or 1). To balance these, we define the following optimization:

$$\hat{y} = \arg\min_{y'} \underbrace{\sum_{i=1}^{n} (y'_i - y_i)^2}_{\text{quality of fit}} + \lambda \underbrace{\sum_{i=1}^{n} \sum_{j=1}^{n} a_{ij} (y'_i - y'_j)^2}_{\text{smoothness}}, \qquad \textbf{(2)}$$

in which we use λ to control the tradeoff between the two principles.

This expression can be compactly expressed using the Laplacian of G: $L = D - A$ in which D is a diagonal matrix

with node degree on the diagonal: $D_{ii}=\sum_j a_{ij}$. Then,

$$\begin{aligned} f(\hat{y}) &= \sum_{i=1}^{n}(\hat{y}_i-y_i)^2+\lambda\sum_{i=1}^{n}\sum_{j=1}^{n}a_{ij}(\hat{y}_i-\hat{y}_j)^2 \\ &= ||\hat{y}-y||^2+\lambda\hat{y}^T L\hat{y}, \\ \frac{df}{d\hat{y}} &= 2\hat{y}-2y+2\lambda L\hat{y}=0, \\ \hat{y} &= (I+\lambda L)^{-1}y. \end{aligned}$$

The matrix $(I+\lambda L)^{-1}$ is the *regularized Laplacian* of G (63). It is a positive semidefinite matrix and hence a kernel. In addition to the "guilt by association" argument, we note an additional reason from (7, 60) that the regularized Laplacian is an appropriate tool for functional inference on protein interaction networks: it also naturally discounts paths that pass through high-degree nodes.

The combination of the multi-network kernel embedding described in the previous section with the Regularized Laplacian constitutes MUNK, and the resulting cross-species similarity scores are MUNK *scores*. We denote the MUNK score of two proteins p_i and p_j as d_{ij}, we refer to the RKHS in which G_1 and G_2 are embedded as MUNK-*space*, and the MUNK-representations are given by the rows of C_1 and $\hat{C}_2$ (as defined in the previous section).

Representations for protein (gene) pairs

We also find it useful to develop representations for pairs of nodes (proteins or genes). These can be used to capture functional similarity between two *pairs* of nodes across species. Further, pair-representations can then be used to predict outcomes for pairs of genes (e.g. synthetic lethality). Given two pairs of nodes $(v_i,v_j),(v_k,v_\ell)$, we define a pairwise similarity metric such that the score is large only if d_{ik} and $d_{j\ell}$ (or $d_{i\ell}$ and d_{jk}) are both large. This reflects the hypothesis that synthetic lethal interactions occur within pathways, and between pathways that perform the same/similar essential biological function (5, 40).

Hence, to represent a pair, we simply sum the MUNK-representations for the nodes in the pair. We then compare pairs by computing MUNK scores in the usual way. Given a matrix C of MUNK-representations for nodes, we define the MUNK-representations for a *pair* of nodes (v_i,v_j) as $P_C(v_i,v_j)=c_i+c_j$. Computing similarity for a two pairs (v_i,v_j) and (v_k,v_ℓ) then yields:

$$(c_i+c_j)^T(c_k+c_\ell)=c_i^T c_k+c_i^T c_\ell+c_j^T c_k+c_j^T c_\ell.$$

In general we expect each of the terms on the right hand side to be close to zero *unless* there is functional similarity between the corresponding nodes, because in high dimension, independent random vectors tend to be nearly orthogonal. We note that the pair-similarity scores and the pair-representations themselves can be used for a variety of tasks.

The utility of this approach is informed by recent work on predicting synthetic lethal interactions from network diffusion (10).

Comparing to existing methods

We benchmark against state-of-the-art methods developed for network alignment and multi-species synthetic lethal classification. For network alignment, we compare MUNK to two standard network alignment algorithms, ISORANK and HUBALIGN (26, 35). ISORANK was the first to combine local topology and sequence for the global network alignment problem (35). HUBALIGN identifies "topologically important" genes and aligns these first, before aligning the remaining genes (26). Both these methods serve as a natural comparison point for MUNK in that they use cross-species protein metrics that can be interpreted as similarity scores. We use default parameters for both ISORANK (using $\alpha=0.6$) and HUBALIGN (using $\alpha=0.7$ and $\lambda=0.1$) using the authors' publicly available software to produce cross-species matchings and similarity scores. We modified the HUBALIGN source code to output the similarity scores, since it did not do so by default.

For multi-species synthetic lethal classification, we compare to SINATRA. SINATRA computes 'connectivity profiles' from a given network by computing graph theoretic measures of topology for each node, and then trains classifiers to predict synthetic lethal interactions across species from rank-normalized the connectivity profiles. We implemented SINATRA in Python 3, as the authors did not make any software publicly available.

In order to demonstrate the advantages of the general-purpose representations constructed by MUNK we try to use all these existing algorithms for both cross-species functional similarity and predicting synthetic lethals. However, there is no obvious way to use the network alignment algorithms for multi-species synthetic lethal prediction. We do benchmark cross-species functional similarity using the connectivity profiles generated by SINATRA.

Data

For the experiments in this study, we study the human (*Homo sapiens*, or *H.s.*), mouse (*Mus musculus*, or *M.m.*), baker's yeast (*S. cerevisiae*, or *S.c.*) and fission yeast (*S. pombe*, or *S.p.*) PPI networks.

Protein-protein interaction networks. We downloaded and processed PPI networks for human and mouse from the STRING database (18), PPI networks for *S.c.* and *S.p.* from BioGRID database (56) (refer to Supplemental Information S2.1 for details). We restricted each network to the two-core of the largest connected component, and report summary statistics of the networks in Table S1. We use the two-core of the graph because topologically indistinguishable nodes (nodes that participate in an automorphism of the graph) will necessarily have identical MUNK homology scores. A large class of topologically indistinguishable nodes includes many of the leaf nodes in the graph (degree-1 nodes). That is, in any case where there are two or more leaf nodes attached to the same parent, the nodes are topologically indistinguishable. Protein names were standardized by mapping to UniProt Accession IDs using mappings provided by the UniProt consortium (59).

Sequence homologs. For each combination of two organisms, we identify homologous pairs of genes using NCBI's Homologene database (54).

Protein function annotations. Protein functions were determined using the Gene Ontology database (GO) (3) using the annotations contained in the gene-to-term files. Currently, GO contains more than 40,000 biological concepts over three domains: Molecular Function (MF), Biological Process (BP) and Cellular Component (CC). We use GO annotation corpora downloaded from SGD (9) for yeast, and UniProt (59) for all other species. We exclude annotations based on IEA or IPI evidence due to their lower associated confidence levels. The gene-to-term mapping file from GO does not always include the more general terms implied by a specific gene-term pair (i.e. the ancestors of the term). Thus, following (62) and unless otherwise noted, we post-process obtained GO annotations by propagating and adding missing GO labels over 'has part', 'part of' and 'is a' GO term relations.

Synthetic lethal interactions. We constructed datasets of synthetic lethal interactions (SLI) and non-interactions (non-SLI) from two high-throughput studies of analogous proteins in baker's (*S.c.*) and fission (*S.p.*) yeast (12, 53). Also, following Jacunski, et al. (27), we constructed datasets of SLI from BioGRID (v3.4.157) in *S.c.* and *S.p.*, sampling an equivalent number of non-SLI from pairs in the PPI network without an SLI. We report the size of the datasets in Table S2, and additional details in the Supplemental Information S2.2.

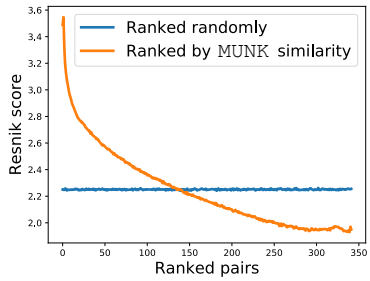
Implementation

We implemented MUNK in Python 3 using the open-source NetworkX, NumPy, and SciPy libraries (23, 45). We executed software pipelines for performing the experiments in part using the `Snakemake` software (31). The source code for MUNK and experiments is available at `https://github.com/lrgr/munk`.

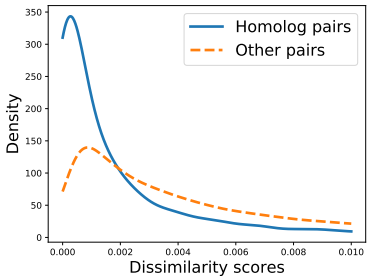
MUNK runs in a practical amount of time for networks of various sizes. Using an Intel Xeon E6-2660 v2 processor with 20 hyper-threaded cores (40 threads) and 94GB of memory, MUNK-representations from human to mouse, mouse to human, human to *S.c.*, *S.c.* to *S.p.*, and *S.p.* to *S.c.*, were computed in 249.1, 3.5, 5.4, 4.1, and 0.25 minutes, respectively.

RESULTS

We demonstrate that MUNK-representations encode functional relationships between proteins in different species by performing three tasks: cross-species protein function annotation, multi-species synthetic lethal classification, and phenolog discovery. We also investigate procedures for choosing the homolog pairs that serve as landmarks, which we describe in *Landmark selection* below. Based on these investigations, we use the same set of 400 homolog-pairs at random to serve as landmarks in all other experiments.



(a)



(b)

Figure 2. The relationship between MUNK similarity scores and functional similarity for the human (source) to mouse (target) embeddings. (a) Relationship between functional similarity measured by Resnik score (y-axis) and protein pairs ranked (x-axis) by MUNK similarity (shown in orange) and ranked randomly (shown in blue; included as a baseline). (b) Distribution of MUNK dissimilarity scores for homologous protein pairs compared to distribution for other (non-homologous) protein pairs.

MUNK-representations capture functional similarity across species

Our results show that the similarity scores given by MUNK-representations are strongly correlated with functional similarity between human and mouse proteins. For this section we evaluate results for pairs of proteins (p_i, p_j), where p_i and p_j are from human (source) and mouse (target), respectively. We only include pairs for which neither p_i or p_j are part of a landmark pair.

We use the Resnik score (51) as a quantitative measure of functional similarity. The Resnik score between two Gene Ontology (GO) (3) terms is the information content of their most informative common ancestor in the GO hierarchy; to compare two proteins we take the maximum Resnik score over all pairs of GO terms. The Resnik score has been shown to be one of the best performing metrics for capturing functional similarity within the GO hierarchy (49).

To demonstrate the relation between MUNK similarity scores and functional similarity, we order each pair according to their MUNK scores, and plot rankings against the Resnik score of the pair. The results (smoothed over non-overlapping windows of 100,000 observations) are shown in Figure 2a. Other cases are shown in Supplementary Information S3.1.

The figure shows that MUNK scores are strongly correlated with functional similarity across the entire range of scores. Furthermore, the very largest MUNK scores are indicative of protein pairs with particularly high functional similarity.

Homolog pairs have distinct MUNK *similarity scores.* Next, we show that pairs that are known to be functionally related are distinguishable by their MUNK similarity scores. For this purpose, we separate pairs (p_i,p_j) where p_i and p_j are homologous proteins in different organisms from other pairs.

In Figure 2b, we show the distribution of MUNK dissimilarity among known homolog pairs, as compared to the distribution of scores across other pairs. In this figure, we use reciprocal scores (dissimilarities), meaning that small scores are associated with high functional similarity. Only the left side of the distributions are shown, as the distribution of all pairs extends far to the right and obscures the homolog distribution on the left. As suggested by the plots, the mean MUNK dissimilarity scores for human-mouse homologs are 36% lower than the mean across other protein pairs. Other cases are shown in Supplementary Information S3.1. These results provide additional evidence that MUNK scores are correlated with functional similarity across species.

MUNK *captures shared biological information beyond node degree.* We find that MUNK captures shared biological information from network topology and not only from node degrees. We assess the statistical significance of the difference in MUNK similarity scores between homologous and non-homologous pairs by generating 1000 pairs of random networks in which each node is given degree very close to that in the original network, but in which edges have been randomized. Specifically, we follow the method of Newman et. al (42) to generate graphs with given degree distributions and remove self loops and parallel edges afterwards. We then compute an empirical P-value by counting the number of pairs of random networks for which the difference in mean MUNK similarity scores between homologs and non-homologs is greater than that observed in real PPI networks. Given the expense of generating many permutations of the large human network, we instead assess two yeast (*S.c.* and *S.p.*) networks. We find that homologous pairs have statistically significantly higher MUNK similarity scores ($P=0.002$ for embedding *S.c.* to *S.p.*, and $P=0.005$ for embedding *S.p.* to *S.c.*). Consequently, we conclude that the differences in the distributions of MUNK similarity scores for homologs and non-homologs is unlikely to be due to node degree alone, but instead is a result of MUNK capturing more detailed network topology.

Comparing MUNK *to other methods.* We compare MUNK to the ISORANK (55) and HUBALIGN (26) network alignment algorithms in their ability to predict cross-species functional similarity of proteins. We focus on two different measures of functional similarity. First is the Gene Ontology consistency (GOC) measure which is commonly used to evaluate functional network alignments produced by a matching algorithm. GOC measures the Jaccard index of the GO terms assigned to each pair of *matched* proteins (p_i,p_j). For MUNK and ISORANK, we generate a matching of all proteins in the smaller network to one protein in the larger network using the Hungarian algorithm applied to the similarity scores. HUBALIGN automatically produces such a matching. Unlike MUNK, the protein similarity scores used by ISORANK and HUBALIGN are a convex combination of BLAST sequence similarity and network topological similarity between each cross-species pair of proteins. The main contribution of the algorithms we study here is in defining the cross-species network similarity, so it is crucial when evaluating alignment methods to control the relative amount of sequence versus network information used by the various methods. Accordingly, we compare the results for MUNK when using 400 cross-species landmark pairs with those of ISORANK and HUBALIGN when using sequence similarity scores only between the same 400 landmarks. However, for context, we also report results for ISORANK and HUBALIGN using all sequence similarity scores.

Species	Algorithm	GOC	k-FS AUPR
human→ mouse	MUNK	0.178	0.060
	ISORANK	**0.200** (0.515)	**0.075** (0.060)
	HUBALIGN	0.115 (0.262)	0.060 (0.067)
human→ baker's yeast	MUNK	**0.156**	**0.034**
	ISORANK	−−− (0.112)	−−− (0.022)
	HUBALIGN	0.119 (0.223)	0.025 (0.030)
baker's → fission yeast	MUNK	**0.275**	0.067
	ISORANK	0.266 (0.288)	**0.070** (0.045)
	HUBALIGN	0.193 (0.238)	0.046 (0.054)

Table 1. Results for functional similarity measures of MUNK and network alignment algorithms HUBALIGN and ISORANK. GO consistency results are reported first restricting the algorithms to only use the same amount of sequence information (BLAST scores for the 400 "landmark" homolog pairs used by MUNK), and then using all available sequence information for HUBALIGN and ISORANK. The reported results for MUNK are for embedding the smaller of the two networks into the larger one. ISORANK did not produce a matching in human-baker's yeast with restricted sequence information.

The second measure we use is k-functional similarity, which evaluates the *space* induced by the protein similarities and was used (without naming it) in (27). We define a pair of proteins (p_i,p_j) from two species to be k-functionally similar if both p_i and p_j are annotated by the same GO term and, in each species, that GO term is associated with at most k proteins (see Data for details on processing of GO). We then rank cross-species protein pairs by similarity scores obtained from MUNK and other benchmarked algorithms and compute enrichment of k-functional similar pairs at k=100 in the top ranked sets. We evaluate enrichment using area under the precision-recall curve (AUPR), which is an appropriate measure when there is a large class imbalance in the data (15), as is the case here (e.g. 3.9% out of 36,877,467 human-mouse protein pairs are classified as k-functionally similar at k=100). We include comparisons to SINATRA (27) and report the functional consistency (FC) of the computed matchings – an additional measure for evaluating network alignments (11, 26) – in Tables S4 and S5.

Table 1 shows functional measures for each algorithm for pairs of proteins from human-mouse, human-baker's yeast, and baker's-fission yeast. When using all sequence similarity scores, ISORANK outperforms the other methods in terms of GO consistency (GOC) for human-mouse (0.515 for ISORANK versus 0.262 and 0.178 HUBALIGN and MUNK, respectively) and baker's-fission yeast (0.288

for IsoRank versus 0.238 and 0.275 for HubAlign and MUNK, respectively), while HubAlign performs the best for human-baker's yeast (0.223 for HubAlign versus 0.112 and 0.156 for IsoRank and MUNK, respectively). This may be due to IsoRank relying more on sequence similarity than the other methods, particularly for human-mouse, where 86% (3,621/4,217) of proteins in the mouse network have a homolog in the human network. When the algorithms are compared using the same amount of sequence information, MUNK outperforms HubAlign and IsoRank for human-baker's yeast (0.156 for MUNK versus 0.119 for HubAlign) and baker's-fission yeast (0.275 for MUNK versus 0.266 and 0.193 for IsoRank and HubAlign, respectively), and IsoRank outperforms for human-mouse (0.200 for IsoRank versus 0.178 and 0.115 for MUNK and HubAlign, respectively). Note that the IsoRank software did not produce a matching in human-baker's yeast with restricted sequence information.

Using k-functional similarity, we find that MUNK-space captures functional similarity comparably to HubAlign and IsoRank in different pairs of species (Table 1). For human-mouse, MUNK performs comparably to HubAlign but is outperformed by IsoRank (0.060 for MUNK versus 0.060 and 0.075 for HubAlign and IsoRank, respectively). In human-baker's yeast, MUNK outperforms HubAlign by 26% (0.034 for MUNK versus 0.025 for HubAlign). In baker's-fission yeast, MUNK performs comparably to IsoRank (achieving within 4% of IsoRank's AUPR) and outperforms HubAlign by a large margin (0.067 for MUNK versus 0.070 and 0.046 for IsoRank and HubAlign, respectively). Thus, MUNK compares favorably to standard approaches for computing cross-species gene/protein similarity, but has a key additional advantage of creating general-purpose protein representations.

As a baseline, we use BLAST bit-scores between protein sequences as similarity scores and evaluate k-functional similarity using these scores. MUNK and IsoRank perform comparably to BLAST which achieved AUPR of 0.064, 0.029, 0.062 for human-mouse, human-baker's yeast, and baker's-fission yeast, respectively.

To investigate novel functional matches between cross-species pairs with low sequence similarity, we repeated the GO consistency analysis for only non-homologous pairs. Table S3 shows the results. Each method identified different fractions of these potential novel matches, as each method identified different numbers of homolog pairs. As expected, the average GOC scores dropped for each method, but the relative ranking of the methods remained consistent with only one change; IsoRank performed better than MUNK for baker's-fission yeast.

Landmark selection. We investigate two ways of choosing landmark pairs of homologs for each pair of species we analyzed. Motivated by the idea that hubs may serve as good landmarks as they are close to most nodes, we select landmark pairs in order of their average rank by degree in each respective species. We also select landmark pairs completely at random. We compare the two approaches while varying the number of landmarks by computing the average GO consistency (GOC) score of the MUNK-space given by the set of landmarks. For each number of landmarks, we sample ten random sets. We note that we are limited in the number of landmarks we can use to the number of homologs shared by the pair of species, and restrict our experiments to using up to 1000 landmarks. We further note that homologous nodes tend to have higher degree than non-homologous nodes in each species (Figure S3).

Figure S4 shows that there is little difference between the two selection procedures. Selecting landmarks by degree only seems to provide an advantage when the number of landmarks is small (≤ 100), possibly indicating that choosing >100 nodes at random means most nodes are close to at least one landmark. We also find that GOC performance plateaus in human-mouse, human-baker's yeast, and baker's-fission yeast species when we include 1000, 200, and 300 landmarks, respectively. Interestingly, this does not indicate that larger networks require more landmarks as, out of the three pairs of species tested, human-baker's yeast has the most combined number nodes (18,481) but plateaus with the fewest landmarks. However, the human-mouse network did not have an obvious plateau, this is perhaps because a very large fraction of mouse nodes have a human homolog (86%). In all our remaining experiments, we use 400 random landmarks, as in each pair of species MUNK using 400 random landmarks achieved a GOC score within 0.02 of the maximum.

Leveraging multiple model organisms for function prediction We also study the potential for leveraging annotations from multiple model organisms simultaneously using MUNK-representations. This problem has not previously been explored extensively, with the exception of (41), which took a Bayesian approach. Our intent is not to propose a new method for function prediction, but to further demonstrate the value of cross-species information as obtained via MUNK.

We focus on function prediction in three species: human, mouse, and baker's yeast (*S.c.*). Following the approach taken in (34), we focus on predicting rare GO terms – in particular we form predictions for all GO terms that occur between 2 and 300 times in the annotation corpus of one species, using the specific annotations from the GO gene-to-term mapping files. We study the information content of multiple MUNK scores by constructing a convex combination $\alpha_H,\alpha_Y,\alpha_M$ of a given species' diffusion score D_1 with the MUNK similarity scores D_{12} of the two other species.

We perform a binary classification for each GO term. We rank proteins by the total scores contributed by other proteins – within the same and across species – annotated with that term, weighted by $\alpha_H,\alpha_Y,\alpha_M$. We then assess performance using the maximum F_1 score averaged over all GO terms. We test whether such a convex combination has greater predictive power for functional inference than just using information from any single species. Details of our method are given in Supplementary Information S1.2.

Figure 3 shows the prediction performance obtained over the simplex $(\alpha_H,\alpha_Y,\alpha_M)$ such that $\sum\alpha_i=1$. Contours show F_1 scores, and the point of maximum F_1 score is plotted. For comparison purposes we provide results for the case in which MUNK scores are randomly permuted. The figure shows that in each case, the greatest improvement in functional prediction comes when making use of information from both additional species. The improvements in functional prediction are shown in Table S8.

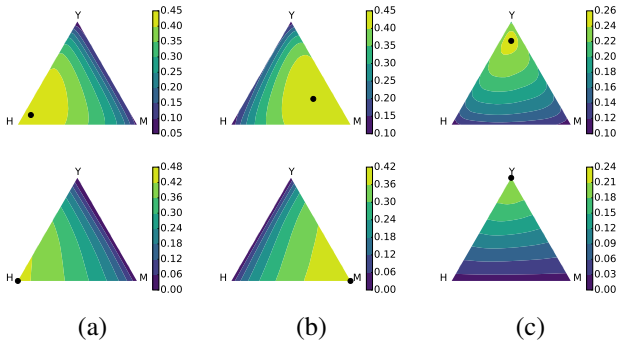


Figure 3. F_1 score of functional inference using multiple additional species. Inferring protein functions for (a) human – H, (b) mouse – M, and (c) baker's yeast – Y. Upper row: using MUNK scores; Lower row: null hypothesis, using randomly permuted MUNK scores.

Multi-species synthetic lethal classification with MUNK-representations

In this section we demonstrate the advantages of general-purpose cross-species protein representations by using MUNK-representations to predict synthetic lethal interactions (SLI) in multiple species simultaneously. Existing matching-based network alignment methods are unable to generalize to this problem, since SLI are a property of *pairs* of genes. Similarly, most existing methods for creating network-based representations are also ill-suited for this problem, since genes in different species are in different vector spaces with different dimensions. The exception to this is the SINATRA algorithm (27), which we benchmark against.

We show that classifiers trained on MUNK-representations can accurately predict SLI in two different species of yeast – *S. cerevisiae* (*S.c.*) and *S. pombe* (*S.p.*) – simultaneously, providing evidence that gene pairs with SLI in different species are co-located in MUNK-space. More specifically, we train a random forest (RF) to classify gene pairs as SLI or non-SLI within both species *simultaneously*, using the source embedding (given by P_{C_1}) and the target embedded into source space (given by $P_{\hat{C}_2}$); see the discussion under *Representations for protein (gene) pairs* for details. We perform 4-fold cross-validation, fixing the relative fraction of pairs from each species, and assess the degree of separation between SLI and non-SLI in MUNK-space by evaluating the RF classifications with maximum F_1 score (the harmonic mean of precision and recall), the area under the ROC curve (AUROC), and the area under the precision-recall curve (AUPR). We report the average across the four folds, separating the results by species. We use a nested cross-validation strategy to choose the number of trees for the RF that maximizes the held-out AUPR. For simplicity, all of our experiments in this section use *S.c.* as the source and *S.p.* as the target.

We first train classifiers on an SLI dataset of high-confidence, low-throughput interactions from BioGRID (8) (see *Data* for additional details of the dataset). This dataset is the most recent update of the dataset used by Jacunski, et al. (27), and we follow their approach by creating a dataset with an equal number of non-SLI sampled randomly from pairs in the PPI networks without SLI. Table 2 shows the results, which demonstrate that classifiers trained on MUNK-representations are very accurate at discriminating between SLI and non-SLI. The RFs achieve average AUROC of 0.933 in *S.c.* (0.876 in *S.p.*), AUPR of 0.933 (0.877), and maximum F_1 score of 0.860 (0.814). We then compare the results using MUNK features to SINATRA (which also uses a RF), using the same protocol as above but train RFs with the rank-normalized features produced by SINATRA for cross-species predictions from (27). SINATRA achieves average AUROC of 0.908 in *S.c.* (0.880 in *S.p.*), AUPR of 0.907 (0.892), and maximum F_1 score of 0.834 (0.808). Thus, while both sets of classifiers make highly accurate predictions, the RFs trained using MUNK-representations as features outperform SINATRA on four of six measures.

Next, we train linear support vector machines (SVMs) instead of RFs to learn hyperplanes separating SLI and non-SLI in MUNK-space, and only see a small degradation in performance compared to the RF (Table S6). This further supports the case that SLI in different species are co-located in MUNK-space because, unlike the decision boundary learned by the RF, the SVM learns a linear decision boundary. Therefore, because the linear SVM also classifies SLI with high accuracy in both species simultaneously, there is evidence for a direction in MUNK-space capturing synthetic lethality.

We then train random forest classifiers with matched high-throughput datasets from *S.c.* and *S.p.* These datasets consist of SLI and non-SLI pairs among 743 *S.c.* genes (12) and 550 *S.p.* genes (53) involved in chromosome biology (see *Data* for additional details of the dataset). The key differences between the chromosome biology SLI datasets and the BioGRID datasets are that the chromosome biology datasets are restricted to functionally similar genes, include 5.5% SLI and 94.5% measured non-SLI in *S.c.* and 10.6% SLI and 89.4% measured non-SLI in *S.p.* (unlike the BioGRID data which only measured SLs), and were generated through high-throughput experiments.

Table 2 shows that the RFs trained on MUNK-representations achieve significant predictive performance on held-out data from the chromosome biology dataset, with an AUROC of 0.864 in *S.c.* (0.822 in *S.p.*), AUPR of 0.402 (0.370), and maximum F_1 score of 0.421 (0.423). While these results show significant predictive power, the performance of the classifiers is poorer than on the BioGRID data. This is likely due to a combination of the noisy, high-throughput

Table 2. Results training classifiers for synthetic lethal interactions on baker's yeast (*S.c.*) and fission yeast (*S.p.*) data *simultaneously*. We compute performance separately for each species (indicated by "Test species"). For each statistic, we report the average on held-out data from 4-fold cross-validation, and bold the highest (best) score.

Dataset	Test	Features	AUROC	AUPR	Max F_1
BioGRID (8)	*S.c.*	MUNK	**0.933**	**0.933**	**0.860**
		SINATRA	0.908	0.907	0.834
	S.p.	MUNK	0.876	0.877	**0.814**
		SINATRA	**0.880**	**0.892**	0.808
Chromosome biology (12, 53)	*S.c.*	MUNK	**0.864**	**0.402**	**0.421**
		SINATRA	0.788	0.201	0.282
	S.p.	MUNK	0.822	0.370	0.423
		SINaTRA	**0.837**	**0.393**	**0.437**

nature of the data measurements and the class imbalance. Interestingly, on the chromosome biology dataset, MUNK outperforms SINATRA by a large margin for predictions in *S.c.*, while SINATRA outperforms MUNK by a smaller margin for predictions in *S.p.* Finally, we repeat our analyses to evaluate how well our classifiers generalize to held-out genes on both datasets, and while there is a drop in performance, they retain significant predictive power (see Supplementary Information S3.2 and Table S7).

Together, these results show that synthetic lethal interactions are significantly clustered *across species* in MUNK-space, and that by using MUNK-representations, which leverage knowledge of a subset of homologous genes across species, classifiers can make accurate predictions for other homologous and non-homologous gene pairs.

Using MUNK-homologs to find known and novel phenologs

As a third demonstration of the utility of general-purpose protein representations, we use MUNK to develop a broadened notion of *phenolog*, generalizing the definition in (37). As put forward in (37), a phenolog is a pair of homologous phenotypes in different species, whose identification can yield, e.g., non-obvious disease models. That paper operationally defined a phenotype based on over-representation of homologous proteins associated with each phenotype. Here, we argue that over-representation of *functionally-similar* proteins constitutes a more powerful operational definition for a phenolog.

Because MUNK scores are strongly correlated with functional similarity across a range of scales (Figure 2a), MUNK scores generalize the relationship of two proteins in different species from a binary value (homologous or not) to a continuous value (degree of functional similarity). This is useful because, while reliance on homologs is a good start for determining phenotypic preservation, the requirement of sequence preservation may be too restrictive if the primary goal is to study function across species (1, 50, 52). Accordingly, in comparing two phenotypes in different species, we can ask about over-representation of functionally-similar proteins over a wide range of functional similarity. We hypothesize that examining different degrees of functional similarity can expose different kinds of phenologs.

We investigate whether additional phenologs may be discovered through the expanded notion of homology that is provided by MUNK (rather than reproduce the results in (37) with a different methodology). As a proof of concept, we show results using human-mouse MUNK scores. To make comparisons with (37) we use the same phenotype to gene association datasets as in that study.

We threshold MUNK similarity scores, leading to a binary classification of each cross-species gene pair as to whether it is functionally-similar at the chosen threshold level. In the results we report here, we set the threshold so as to output a small set of phenolog predictions. As discussed, other threshold settings would potentially produce different functional classes of phenologs as the pairs of genes considered close changes. To compare a phenotype in one species to a phenotype in a different species we count the number of functionally-similar pairs across species in the two associated phenotypes. We then use the same procedure as in (37), with functionally-similar pairs playing the role that homologs did in that work; details are provided in Supplementary Information S1.3.

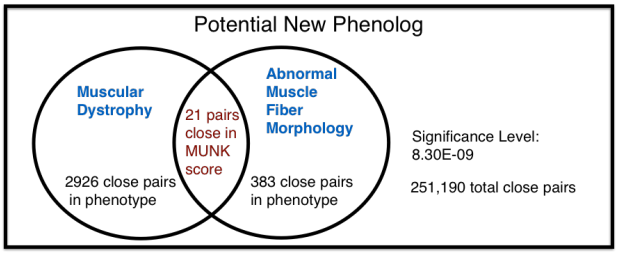


Figure 4. A potential new phenolog, not found by (37), relating the phenotypes Muscular Dystrophy (human) and Abnormal Muscle Fiber Morphology (mouse).

Table 3. Matched mouse phenotypes per human disease among newly-identified phenologs.

Human Disease	Obvious Match	Similar Match	Novel Match
Systemic lupus erythematosus	1	0	0
Dilated cardiomyopathy	4	0	1
Zellweger syndrome	2	1	2
Dysfibrinogenemia	0	0	1
Myopathy	1	0	0
Bardet-Biedel syndrome	6	2	1
Adrenoleukodystrophy	0	3	1
Muscular Dystrophy	2	0	1
Total	**16**	**6**	**7**

By using a stringent threshold on MUNK scores, we sharply limit the size of the set of phenologs predicted. Whereas (37) reported 3634 phenotype pairs passing significance testing, our results show 47 pairs of phenotypes to be significantly associated. Within this set were 18 phenologs previously reported by (37), which is not surprising given that many homolog pairs are ranked highly by MUNK (e.g. see Figure 2b). However our primary interest are the matches that are not part of the homolog-based phenologs reported in (37).

We find that MUNK-based similarity can uncover many new phenologs that are not statistically significant when using homologs. As an example, we show in Figure 4 details of a phenolog found using MUNK, but not found in (37). This phenolog matches *Abnormal Muscle Fiber Morphology* in the mouse with *Muscular Dystrophy* in human. The example is illustrative as it shows that false negatives can occur when using only homologs, even for straightforward matches such as this one. In fact the homolog-based method used in (37) finds only 1 homolog in common, while MUNK detects 21 functionally-similar pairs of proteins. The larger set of functionally-similar protein pairs uncovered by MUNK gives greater statistical power for cases such as this one, where there is only a small number of homologs shared between the two phenotypes.

We find that many of the remaining 29 phenotype pairs identified using MUNK are potentially valid phenologs. To

demonstrate, we compare the human phenotype (disease) with mouse phenotype (symptom). The 29 pairs of phenotypes ranged over 8 unique human diseases and 24 unique mouse phenotypes. We obtained the description of each disease from the Genetics Home Reference[1] and compared the disease description to the matched mouse phenotypes.

Table 3 groups the results into three categories: obvious symptom match, possible symptom match, and novel match. For each disease term and mouse phenotype pair, if the name of the mouse phenotype was indicated as a symptom in the disease description, we considered an obvious match; if the disease description contained a symptom that was similar to the mouse phenotype, we considered it a possible match; and if there was no similarity in the phenotype and the disease description we considered it as a novel match. Table 3 shows that, for over 75% of the new phenologs predicted using MUNK, there is evidence in the literature supporting the association.

In summary, we find that functionally-similar pairs as identified via MUNK can indeed provide a basis for expanding the set of phenologs discovered by previous methods. We anticipate future research developing methods to uncover phenologs may use previous methods in tandem with MUNK-based methods to more thoroughly explore the space of possible phenologs.

DISCUSSION AND FUTURE WORK

We introduce a novel, kernel-based algorithm to create general-purpose, multi-species protein representations using biological networks and sequence data. We use a particular diffusion kernel – the regularized Laplacian – to create functional representations, and use the resulting algorithm, MUNK, to embed proteins from humans, mice, and yeast into shared spaces. We evaluate the MUNK-representations on cross-species functional similarity and multi-species synthetic lethal prediction, showing the MUNK-representations lead to comparable performance as specialized methods for these tasks. We also use MUNK to expand the notion of orthologous phenotypes beyond evolutionarily conserved sequence and identify known and novel phenologs, providing evidence for non-obvious human disease models. Importantly, in these tasks, we transfer knowledge both from humans to model organisms and from model organisms to humans. Thus, MUNK represents a new direction towards realizing the crucial goal of algorithms for transferring knowledge of genetics across species.

The phenologs found using MUNK representations motivate a larger investigation into non-obvious cross-species phenotype relationships. While the seminal work of McGary, et al. (37) was based on the hypothesis that conserved molecular functions can produce different "organism-level" phenotypes, our results show that phenotypes can have statistically significant relationships at a broader level of protein function. This may suggest that the functional relationships between phenotypes in different species exist on a continuum, resulting in different types of disease models.

Our approach of creating cross-species protein representations can be seen as a component of a *transfer learning* (47) approach for cross-species inference. The promise of transfer learning – using knowledge gained in solving one task to aid in solving a different task – for cross-species inference is to leverage species where data is widely available for predictions in species where data is sparse. For example, this is the case for genetic interactions, where approximately 90% of pairwise genetic interactions have been measured in baker's yeast (13), while fewer than 1% of pairs have been tested in humans. Transfer learning is often approached by finding appropriate transformations of data features (e.g. 'domain adaptation', e.g., see (57)). For MUNK, methods for aligning the source and target embeddings may be required to make such a transfer learning approach possible. At the same time, we showed that methods for transferring knowledge across species can be useful even when there is a wealth of data in the target species. Thus, to achieve optimal performance, supervised learners may need to train on multiple species simultaneously.

Beyond kernels derived from protein interaction, there are a wide range of other kernels that can inform biological function assessment, including kernels derived from co-expression, genetic interaction, metabolic pathways, domain structure, and sequence (7, 32, 33). Because MUNK is a method for creating a new kernel encompassing the nodes of multiple networks, it holds potential as a new tool for kernel learning methods such as support vector machines in a wide variety of applications beyond cross-species function prediction.

ACKNOWLEDGEMENTS

This work was supported in part by NSF grants IIS-1421759 and CNS-1618207 (to M.C.) and by a grant from the Boston University Undergraduate Research Opportunities Program (to T.L.). We thank Simon Kasif, Evimaria Terzi, Prakash Ishwar, Lenore Cowen, Donna Slonim, and the Tufts BCB group for helpful discussion on this work.

COMPETING INTERESTS

None to declare.

[1] https://ghr.nlm.nih.gov/

REFERENCES

1. Adrian M Altenhoff, Romain A Studer, Marc Robinson-Rechavi, and Christophe Dessimoz. Resolving the ortholog conjecture: orthologs tend to be weakly, but significantly, more similar in function than paralogs. *PLoS Comput Biol*, 8(5):e1002514, 2012.
2. Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.
3. Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, et al. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000.
4. Yoshua Bengio, Olivier Delalleau, and Nicolas Le Roux. Label propagation and quadratic criterion. In Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien, editors, *Semi-Supervised Learning*, chapter 11. The MIT Press, 2006.
5. Charles Boone, Howard Bussey, and Brenda J Andrews. Exploring genetic interactions and networks with yeast. *Nature Reviews Genetics*, 8(6):437–449, 7 2007.
6. Kevin R Brown and Igor Jurisica. Unequal evolutionary conservation of human protein interactions in interologous networks. *Genome biology*, 8(5):R95, 2007.
7. Mengfei Cao, Hao Zhang, Jisoo Park, Noah Daniels, Mark Crovella, et al. Going the distance for protein function prediction: A new distance metric for protein interaction networks. *PLOS One*, 8(10):e76339, October 2013.
8. Andrew Chatraryamontri, Rose Oughtred, Lorrie Boucher, Jennifer Rust, Christie Chang, et al. The biogrid interaction database: 2017 update. *Nucleic Acids Research*, 45(D1):D369–D379, 2017.
9. JM Cherry, C Adler, C Ball, SA Chervitz, SS Dwight, ET Hester, Y Jia, G Juvik, T Roe, M Schroeder, S Weng, and D Botstein. SGD: Saccharomyces genome database. *Nucleic Acids Res*, 26:73–79, 1998.
10. Hyunghoon Cho, Bonnie Berger, and Jian Peng. Compact integration of multi-network topology for functional analysis of genes. *Cell Systems*, 3(6):540–548.e5, 2016.
11. Connor Clark and Jugal Kalita. A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics*, 30(16):2351–2359, 2014.
12. Sean R Collins, Kyle M Miller, Nancy L Maas, Assen Roguev, Jeffrey Fillingham, et al. Functional dissection of protein complexes involved in yeast chromosome biology using a genetic interaction map. *Nature*, 446(7137):806–810, 7 2007.
13. Michael Costanzo, Benjamin VanderSluis, Elizabeth N Koch, Anastasia Baryshnikova, Carles Pons, et al. A global genetic interaction network maps a wiring diagram of cellular function. *Science*, 353(6306):aaf1420, 2016.
14. Lenore Cowen, Trey Ideker, Benjamin J Raphael, and Roded Sharan. Network propagation: a universal amplifier of genetic associations. *Nature Reviews Genetics*, 2017.
15. Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. pages 233–240, 2006.
16. François Fouss, Kevin Francoisse, Luh Yen, Alain Pirotte, and Marco Saerens. An experimental investigation of kernels on graphs for collaborative recommendation and semisupervised classification. *Neural Netw.*, 31:53–72, July 2012.
17. François Fouss, Marco Saerens, and Masashi Shimbo. *Algorithms and Models for Network Data and Link Analysis*. Cambridge University Press, 2016.
18. Andrea Franceschini, Damian Szklarczyk, Sune Frankild, Michael Kuhn, Milan Simonovic, Alexander Roth, Jianyi Lin, Pablo Minguez, Peer Bork, Christian von Mering, and Lars J. Jensen. String v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Research*, 41(D1):D808–D815, 2013.
19. Adam Frost, Marc G Elgort, Onn Brandman, Clinton Ives, Sean R Collins, et al. Functional repurposing revealed by comparing s.pombe and s.cerevisiae genetic interactions. *Cell*, 149(6):1339–1352, 11 2012.
20. Vladimir Gligorijevic, Meet Barot, and Richard Bonneau. deepnf: Deep network fusion for protein function prediction. *Bioinformatics (Oxford, England)*, 2018.
21. Vladimir Gligorijević, Noël Malod-Dognin, and Nataša Pržulj. Fuse: multiple network alignment via data fusion. *Bioinformatics*, 32(8):1195–1203, 2016.
22. Aditya Grover and Jure Leskovec. Node2vec: Scalable feature learning for networks. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 855–864, New York, NY, USA, 2016. ACM.
23. Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.
24. Jihun Ham, Daniel D Lee, and Lawrence K Saul. Semisupervised alignment of manifolds. In *AISTATS*, pages 120–127, 2005.
25. Hitomi Hasegawa and Liisa Holm. Advances and pitfalls of protein structural alignment. *Current Opinion in Structural Biology*, 19(3):341–348, 9 2009.
26. Somaye Hashemifar and Jinbo Xu. Hubalign: an accurate and efficient method for global alignment of protein–protein interaction networks. *Bioinformatics*, 30(17):i438–i444, 2014.
27. Alexandra Jacunski, Scott J. Dixon, and Nicholas P. Tatonetti. Connectivity homology enables inter-species network models of synthetic lethality. *PLOS Computational Biology*, 11(10):e1004506, 2015.
28. Lars J. Jensen, Michael Kuhn, Manuel Stark, Samuel Chaffron, Chris Creevey, Jean Muller, Tobias Doerks, Philippe Julien, Alexander Roth, Milan Simonovic, Peer Bork, and Christian von Mering. String 8–a global view on proteins and their functional interactions in 630 organisms. *Nucleic acids research*, 37(Database issue):D412–416, January 2009.
29. Vikram Khurana, Jian Peng, Chee Chung, Pavan K Auluck, Saranna Fanning, et al. Genome-scale networks link neurodegenerative disease genes to -synuclein through specific molecular pathways. *Cell Systems*, 4(2):157–170.e14, 2017.
30. Risi Imre Kondor and John Lafferty. Diffusion kernels on graphs and other discrete input spaces. In *ICML*, volume 2, pages 315–322, 2002.
31. Johannes Kster and Sven Rahmann. Snakemake – a scalable bioinformatics workflow engine. *Bioinformatics*, 28(19):2520–2522, 12 2012.
32. G. R. Lanckriet, M. Deng, N. Cristianini, M. I. Jordan, and W. S. Noble. Kernel-based data fusion and its application to protein function prediction in yeast. In *Pacific Symposium on Biocomputing*, pages 300–311, 2004.
33. Christina S. Leslie, Eleazar Eskin, Adiel Cohen, Jason Weston, and William Stafford Noble. Mismatch string kernels for discriminative protein classification. *Bioinformatics*, 20(4):467, 2004.
34. Stanley Letovsky and Simon Kasif. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, 19(suppl 1):i197–i204, 2003.
35. Chung-Shou Liao, Kanghao Lu, Michael Baym, Rohit Singh, and Bonnie Berger. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–i258, 2009.
36. Noël Malod-Dognin and Nataša Pržulj. L-graal: Lagrangian graphlet-based network aligner. *Bioinformatics*, page btv130, 2015.
37. Kriston L McGary, Tae Park, John O Woods, Hye Cha, John B Wallingford, and Edward M Marcotte. Systematic discovery of nonobvious human disease models through orthologous phenotypes. *Proceedings of the National Academy of Sciences*, 107(14):6544–6549, 10 2010.
38. Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. arXiv, 2013.
39. Tijana Milenković, Weng Leong Ng, Wayne Hayes, and Nataša Pržulj. Optimal network alignment with graphlet degree vectors. *Cancer Informatics*, 9:121–137, 2010.
40. Florian L. Muller, Elisa A. Aquilanti, and Ronald A. DePinho. Collateral lethality: A new therapeutic strategy in oncology. *Trends in Cancer*, 1(3):161–173, 2015.
41. Naoki Nariai. *Probabilistic Integration of Heterogeneous, Contextual, and Cross-Species Genome-Wide Data for Protein Function Prediction*. PhD thesis, Boston University, 2010.
42. M. E. J. Newman, S. H. Strogatz, and D. J. Watts. Random graphs with arbitrary degree distributions and their applications. *Phys. Rev. E*, 64:026118, Jul 2001.
43. Behnam Neyshabur, Ahmadreza Khadem, Somaye Hashemifar, and Seyed Shahriar Arab. Netal: a new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics*, 29(13):1654–1662, 2013.
44. Sebastian M.B. Nijman. Synthetic lethality: General principles, utility and detection using genetic screens in human cells. *FEBS Letters*, 585(1):1–6, 11 2011.
45. Travis E. Oliphant. Python for scientific computing. *Computing in Science and Engg.*, 9(3):10–20, May 2007.
46. Nigel J O'Neil, Melanie L Bailey, and Philip Hieter. Synthetic lethality

and cancer. *Nature Reviews Genetics*, 2017.
47. Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, October 2010.
48. Yungki Park and Edward M Marcotte. Flaws in evaluation schemes for pair-input computational predictions. *Nature Methods*, 9(12):1134–1136, 12 2012.
49. Catia Pesquita, Daniel Faria, Andr O. Falco, Phillip Lord, and Francisco M. Couto. Semantic similarity in biomedical ontologies. *PLoS Comput Biol*, 5(7):1–12, 07 2009.
50. Mark E Peterson, Feng Chen, Jeffery G Saven, David S Roos, Patricia C Babbitt, and Andrej Sali. Evolutionary constraints on structural similarity in orthologs and paralogs. *Protein Science*, 18(6):1306–1315, 2009.
51. Philip Resnik. Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *J. Artif. Intell. Res.(JAIR)*, 11:95–130, 1999.
52. Igor B Rogozin, David Managadze, Svetlana A Shabalina, and Eugene V Koonin. Gene family level comparative analysis of gene expression in mammals validates the ortholog conjecture. *Genome biology and evolution*, 6(4):754–762, 2014.
53. Assen Roguev, Sourav Bandyopadhyay, Martin Zofall, Ke Zhang, Tamas Fischer, et al. Conservation and rewiring of functional modules revealed by an epistasis map in fission yeast. *Science*, 322(5900):405–410, 8 2008.
54. Eric W Sayers, Tanya Barrett, Dennis A Benson, Evan Bolton, Stephen H Bryant, et al. Database resources of the national center for biotechnology information. *Nucleic acids research*, 39(suppl 1):D38–D51, 2011.
55. Rohit Singh, Jinbo Xu, and Bonnie Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences*, 105(35):12763–12768, 2008.
56. C. Stark, B.J. Breitkreutz, T. Reguly, L. Boucher, A. Breitkreutz, and M. Tyers. BioGRID: a general repository for interaction datasets. *Nucleic Acids Res*, 34(Database Issue):D535–D539, 2006.
57. Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 2058–2065. AAAI Press, 2016.
58. Mark G. F. Sun, Martin Sikora, Michael Costanzo, Charles Boone, and Philip M. Kim. Network evolution: Rewiring and signatures of conservation in signaling. *PLoS Computational Biology*, 8(3):e1002411, 12 2012.
59. The UniProt Consortium. Uniprot: the universal protein knowledgebase. *Nucleic Acids Research*, 45(D1):D158–D169, 2017.
60. Fabio Vandin, Eli Upfal, and Benjamin J. Raphael. Algorithms for detecting significantly mutated pathways in cancer. *Journal of Computational Biology*, 18(3):507–522, 2016/11/07 2011.
61. Chang Wang and Sridhar Mahadevan. Manifold alignment using procrustes analysis. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1120–1127, 2008.
62. Michael Yu, Michael Kramer, Janusz Dutkowski, Rohith Srivas, Katherine Licon, et al. Translation of genotype to phenotype by a hierarchy of cell subsystems. *Cell Systems*, 2(2):77–88, 2016.
63. D. Zhou and B. Schölkopf. A regularization framework for learning from graph data. In *ICML Workshop on Statistical Relational Learning and Its Connections to Other Fields*, pages 132–137, 2004.

SUPPLEMENTARY INFORMATION

S1 Methods

S1.1 Parameter choice For all regularized Laplacians, we used a value of $\lambda = 0.05$. We found that the resulting relationship between Resnik similarity and MUNK similarity score did not vary significantly for λ values between 0.005 and 0.1.

S1.2 Function Prediction Methods We assess prediction accuracy using leave-one-out cross validation. Let $K^s = \{k^s_{ij}\}$ denote the regularized Laplacian for species s. For GO term g, let $\mathcal{G}^s_g$ be the set of proteins in species s that are annotated with g. The same-species annotation score for a given protein p and GO term g is:

$$c^s(p,g) = \frac{1}{|\mathcal{G}^s_g|} \sum_{i \in \mathcal{G}^s_g} k^s_{pi}$$

in which p is excluded from the sum (i.e., if it is contained in $\mathcal{G}^s_g$). We also construct a cross-species annotation score for each protein, in which MUNK scores with respect to proteins in the other organism are used:

$$c^{s_0,s_1}(p,g) = \frac{1}{|\mathcal{G}^{s_1}_g|} \sum_{i \in \mathcal{G}^{s_1}_g} d_{pi}$$

where $d_{pi} = D_{12}(p,i)$ is the MUNK score for protein p in species s_0 and protein i in species s_1. The prediction score is then $h(p,g) = \alpha c^{s_0}(p,g) + (1-\alpha) c^{s_0,s_1}(p,g)$. To use multiple cross-species annotations, say n, we generalize $h(p,g)$ to a convex combination of the same- and cross-species annotation scores: $h(p,g) = \alpha_0 c^{s_0}(p,g) + \sum_{i=1}^{n} \alpha_i c^{s_0,s_i}(p,g)$ such that $\sum_{i=0}^{n} \alpha_i = 1$.

We evaluate predictions using area under the receiver operating curve (AUC) and maximal F-score (over all detection thresholds). Since we are concerned with predicting rare GO terms, we find that maximal F-score is generally a more discriminative metric. We set the convex coefficients $\{\alpha_i\}$ via cross-validation.

S1.3 Phenolog Discovery Our method matches that used in (37), using protein pairs with high MUNK similarity scores rather than homologs obtained from Homologene.[2] Specifically, let P_1 be the genes associated with the phenotype in species 1 and P_2 be the genes associated with the phenotype in species 2. Our contingency table consists of the counts of the number of "MUNK-homologs" involving $P_1 \cap P_2$, $P_1 \setminus P_2$, $P_2 \setminus P_1$, and $(\Omega \setminus P_1) \setminus P_2$, with Ω denoting the set of all close pairs. We used a Fisher exact test to measure significance, and considered the match significant if the uncorrected P-value was less than 0.05. We corrected for multiple testing using a Bonferroni correction; there were $1{,}278{,}312$ possible phenotype matches so we set the significance level at 3.9×10^{-8}.

[2]Note that none of the landmarks (which are a subset of the homologs) are used to discover new phenologs.

S2 Data

S2.1 Protein-protein interaction networks We constructed protein-protein interaction (PPI) networks in *S.c.*, *S.p.*, mouse, and human. The *S.c.* and *S.p.* networks were obtained from the Biological General Repository for Interaction Datasets (BioGRID) (8) version 3.4.157. Mouse and human PPIs were obtained from the STRING database version 9.1 (18). PPI networks obtained were processed by mapping the protein names to the same namespace. Genes that could not be mapped via the UniProt database were removed from the PPI networks entirely. We provide further details of the network processing in *Data*. Table S1 shows summary statistics for the PPI networks before and after processing.

S2.2 Synthetic lethal interactions We constructed datasets of synthetic lethal interactions (SLI) in *S.c.* and *S.p.* from published epistatic miniarray profiles (E-MAPs). E-MAPs include genetic interactions scores for pairs of genes, where the magnitude of the score reflects the strength of the genetic interaction. We downloaded E-MAPs for *S.c.* from the supplementary information of Collins, et al. (12), and for *S.p.* from the supplementary information of Roguev, et al. (53). We classified each pair of genes in each E-MAP as SLI, non-SLI, or uncertain. We used the thresholds from the Collins, et al. (12) supplementary information to classify pairs in *S.c.*. Given a pair with E-MAP score ϵ, we classified it as SLI if $\epsilon < -3$, uncertain if $-3 \leq \epsilon < -1$, and non-SLI otherwise. Similarly, we used the threshold for synthetic lethality from the Roguev, et al (53) supplementary information and used the same threshold for uncertainty, classifying *S.p.* pairs as SLI if $\epsilon < -2.5$, uncertain if $-2.5 \leq \epsilon < -1$, and non-SLI otherwise. We also remove pairs of genes in which either gene is not found in the corresponding PPI networks described in *Data*. The resulting datasets included 7,165 SLI and 123,507 non-SLI in *S.c.*, and 5,599 SLI and 97,541 non-SLI in *S.p.*

We also constructed datasets of SLI from BioGRID (8) (v3.4.157).[3] For both *S.c.* and *S.p.*, we extracted interactions of type "Synthetic Lethality" only (i.e. ignoring other negative genetic interactions), yielding datasets of 13,645 and 908 SLI, respectively.

We then standardized the datasets by mapping genes names to Uniprot Accession IDs (59). Genes that could not be mapped via UniProt were excluded for this study, as were those that were not found in the processed PPI networks. For the BioGRID SLI dataset, we followed Jacunski, et al. (27), by sampling an equivalent number of non-SLI pairs from genes PPI networks that do not partake in SLI in the BioGRID dataset. Table S2 shows summary statistics of the SLI datasets before and after processing.

S3 Results

S3.1 Associations of MUNK *scores with functional similarity for other pairs of species* We associated MUNK similarity scores and functional similarity for pairs of proteins in additional pairs of species, using the methodology described in Section . Figures S1 and S2 show the results for embedding human into yeast, and mouse into yeast.

[3]`https://thebiogrid.org`

S3.2 Evaluating the generalization of synthetic lethal interaction classifiers to held-out genes Predicting synthetic lethal interactions between gene pairs using features constructed for individual genes is an example of a pair-input classification problem. A challenge with evaluating classifiers trained on pair inputs with held-out data is that, for a given pair (u,v), it is possible that the features for only u, only v, both u and v, or neither u and v, can be found in the training data (48). Thus, information concerning genes found in the held-out data may be "leaked" to the classifier during training. To evaluate the effect of this issue, Park & Marcotte (48) suggest evaluating classifications for gene pairs which contain one, two, or no genes in the training data separately. This is analogous to holding out individual *genes* instead of *gene pairs* at training time and, thus, we evaluate the effect of pair-inputs by repeating the experiments above but hold out genes instead of gene pairs for evaluation. We report the results in Table S7. We find that the classifiers are able to predict SLI for genes *not* found in the training data, but with a significant change in performance compared to genes found in the training data. On the BioGRID dataset, the classifiers achieve an AUROC of 0.872 in *S.c.* (0.823 in *S.p.*), an AUPRC of 0.875 (0.814), and maximum F_1 of 0.797 (0.772). On the chromosome biology dataset, the classifiers achieve an AUROC of 0.701 in *S.c.* (0.691 in *S.p.*), an AUPRC of 0.160 (0.207), and maximum F_1 of 0.202 (0.285). We hypothesize that the larger drop in performance on the chromosome biology data is due to the matched nature of the *S.c.* and *S.p.* datasets. We also find similar drops in performance for SINATRA when holding out genes instead of pairs (also in Table S7).

S4 Supplementary Tables

Table S1. Summary statistics of PPI networks. We processed the graphs to restrict to the two-core of the largest connected component.

Species	Source	Processing	Nodes	Edges
Baker's yeast (*S.c.*)	BioGRID (8)	Before	5,961	99,539
		After	5,609	95,997
Fission yeast (*S.p.*)	BioGRID (8)	Before	2,888	9,433
		After	1,865	7,712
Human	STRING (28)	Before	15,129	155,866
		After	12,872	153,609
Mouse	STRING (28)	Before	6,596	18,697
		After	4,217	16,318

Table S2. Summary statistics of synthetic lethal interaction datasets.

Species	Reference	Processing	Total	SLs	Non-SLs	Uncertain
Baker's yeast (*S.c.*)	Collins, et al. (12)	Before	150,636	7,240	125,927	17,469
		After	129,385	7,112	122,273	0
	BioGRID v3.4.157 (8)	Before	16,630	16,630	0	0
		After	27,050	13,525	13,525	N/A
Fission yeast (*S.p.*)	Roguev, et al. (53)	Before	118,248	5,754	101,595	10,899
		After	24,214	2,556	21,658	0
	BioGRID v3.4.157 (8)	Before	1020	1020	0	0
		After	1,316	658	658	N/A

Table S3. GOC of aligned gene pairs computed from MUNK-similarity scores and network alignment algorithms HUBALIGN and ISORANK. The GOC scores reported are for non-homologs only (pairs with weak sequence similarity). Homolog pairs recovered from the alignments are removed from the analysis above. For these results, the algorithms are restricted to only use the same amount of sequence information (BLAST scores for the 400 "landmark" homolog pairs used by MUNK). The reported results for MUNK are for embedding the smaller of the two networks into the larger one. Note that the ISORANK software did not produce an alignment for human-baker's yeast with restricted sequence information.

Species	Algorithm	Non-homologs matched	GOC
human→ mouse	MUNK	3696	0.126
	ISORANK	3623	**0.145**
	HUBALIGN	4041	0.103
human→ *S.c.*	MUNK	5292	**0.135**
	ISORANK	– – –	– – –
	HUBALIGN	5294	0.098
S.c.→ *S.p.*	MUNK	1457	0.204
	ISORANK	1642	**0.229**
	HUBALIGN	1756	0.171

Table S4. Functional consistency (FC) of aligned gene pairs computed from MUNK similarity scores and network alignment algorithms ISORANK and HUBALIGN. For ISORANK and HUBALIGN, FC scores are computed for alignments computed using the same amount of sequence similarity information as used for MUNK (BLASTscores for only 400 landmark pairs). Following (26), FC scores are computed using the specific gene-to-term GO labels given in the Gene Ontology gene-to-term file.

Species	No. of shared GO terms	MUNK	ISORANK	HUBALIGN
human→ mouse	≥1	55.9	60.6	42.7
	≥2	31.3	37.4	19.1
	≥3	19.2	24.8	8.9
	≥4	14.2	18.8	5.1
	≥5	11.4	14.8	3.6
human→ *S.c.*	≥1	37.1	–	27.4
	≥2	15.8	–	9.9
	≥3	7.9	–	5.0
	≥4	4.3	–	2.9
	≥5	2.4	–	1.8
S.c.→ *S.p.*	≥1	53.8	53.3	38.0
	≥2	29.9	26.4	12.5
	≥3	17.5	13.9	5.7
	≥4	9.3	6.1	2.5
	≥5	4.0	3.0	0.8

Table S5. k-functional similarity results for SINATRAand BLAST. For SINATRA, we compute similarity scores using Euclidean distance following (27). For BLAST, we directly interpret bitscores as similarity scores.

Species	Algorithm	AUPR
human→ mouse	SINATRA	0.043
	BLAST	0.064
human→ *S.c.*	SINATRA	0.018
	BLAST	0.029
S.c.→ *S.p.*	SINATRA	0.041
	BLAST	0.062

Table S6. Results training linear support vector machines to classify synthetic lethal interactions on *S.c.* and *S.p.* data *simultaneously*. We compute performance separately for each species (indicated by "Test species"). For each statistic, we report the average on held-out data from 4-fold cross-validation over *gene pairs*, and bold the highest (best) score.

Dataset	Test species	Algorithm	AUROC	AUPRC	Max F_1
BioGRID (8)	*S.c.*	MUNK	**0.953**	**0.933**	**0.892**
		SINATRA	0.798	0.774	0.750
	S.p.	MUNK	0.714	0.657	0.731
		SINATRA	**0.786**	**0.846**	**0.850**
Chromosome biology (12, 53)	*S.c.*	MUNK	**0.865**	**0.334**	**0.396**
		SINATRA	0.631	0.092	0.156
	S.p.	MUNK	**0.711**	0.123	**0.208**
		SINATRA	0.569	**0.124**	0.214

Table S7. Results training random forests to classify synthetic lethal interactions on *S.c.* and *S.p.* data *simultaneously*. We compute performance separately for each species (indicated by "Test species"). For each statistic, we report the average on held-out data from 4-fold cross-validation over *genes*, and bold the highest (best) score.

Dataset	Test species	Algorithm	AUROC	AUPRC	Max F_1
BioGRID (8)	*S.c.*	MUNK	**0.872**	**0.875**	**0.797**
		SINATRA	0.848	0.852	0.779
	S.p.	MUNK	0.823	0.814	**0.772**
		SINATRA	**0.839**	**0.855**	0.769
Chromosome biology (12, 53)	*S.c.*	MUNK	**0.701**	**0.160**	**0.202**
		SINATRA	0.681	0.115	0.184
	S.p.	MUNK	0.691	0.207	0.285
		SINATRA	**0.723**	**0.239**	**0.314**

Table S8. Improvement in functional prediction using two other species.

Target Species	AUC	F_1 score
Human	0.3%	2.6%
Mouse	1.0%	8.6%
Baker's yeast	0.3%	16.0%

S5 Supplementary Figures

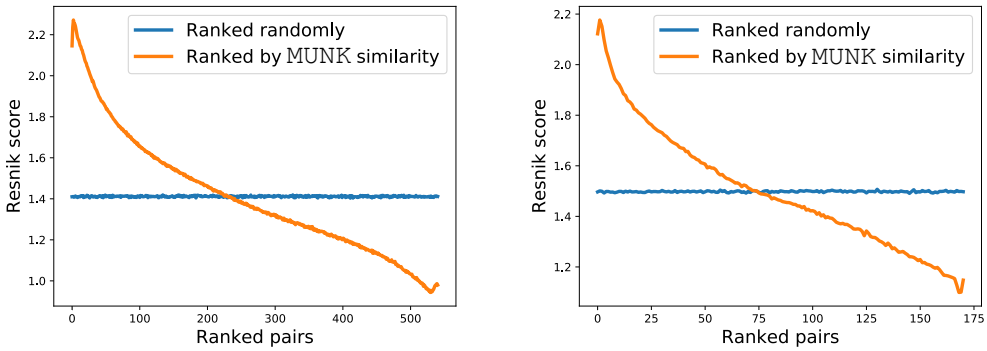


Figure S1. Relationship between cross-species Resnik similarity and MUNK homology score, for (a) human (source) - yeast (target) and (b) mouse (source) - yeast (target) comparison.

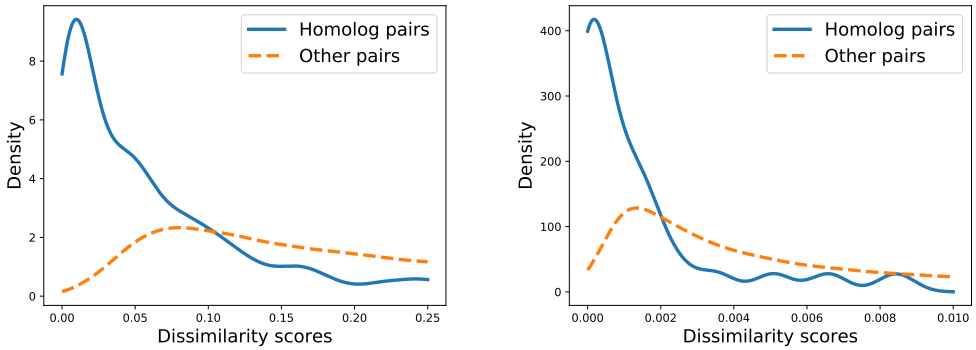


Figure S2. Distribution of MUNK dissimilarity scores for homologs, compared to distribution for all protein pairs, for (a) human (source) - yeast (target) and (b) mouse (source) - yeast (target) comparison.

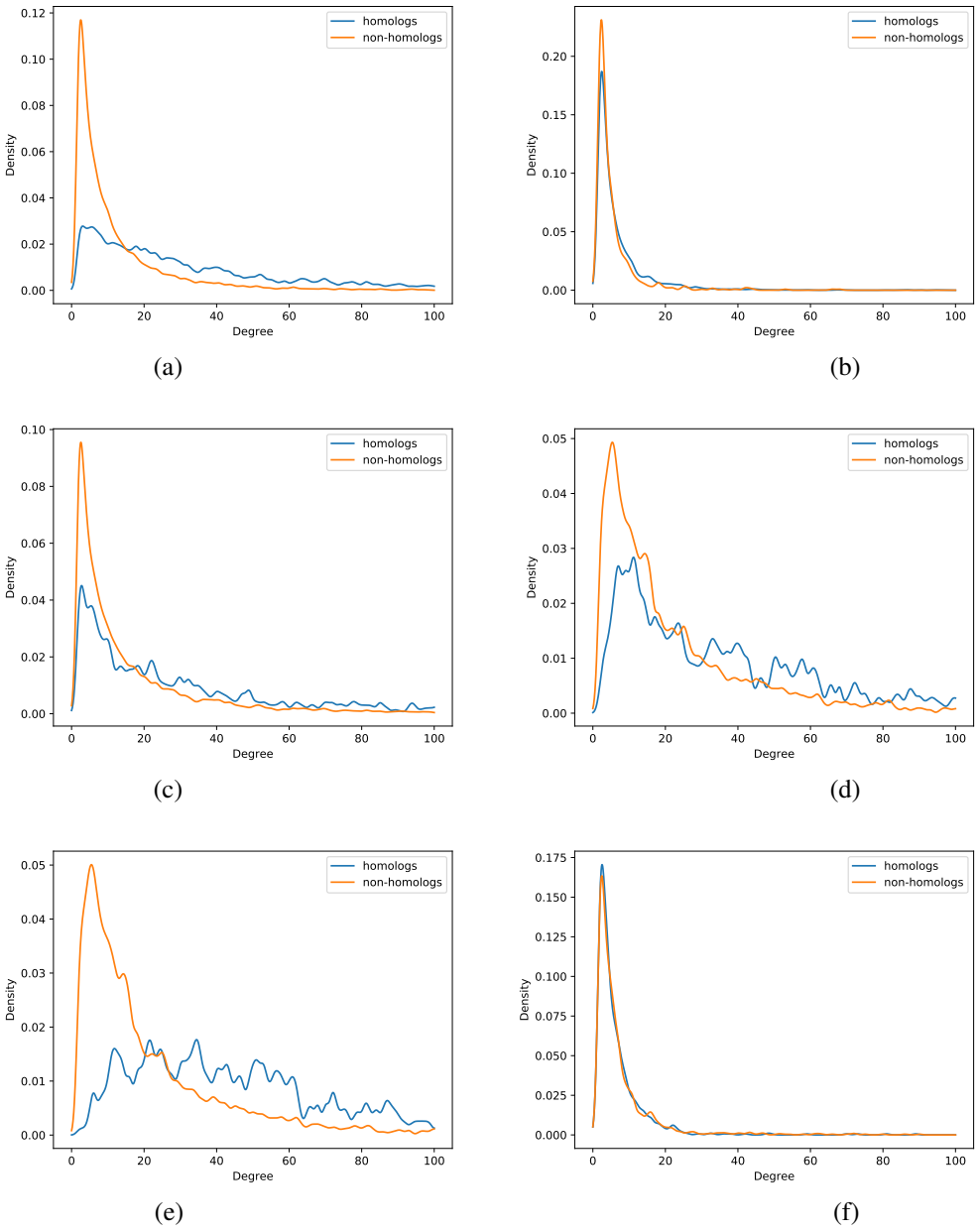


Figure S3. Distribution of node degrees of homologs compared to distribution of non-homologs, for (a) human genes with and without mouse homologs, (b) mouse genes with and without human homologs, (c) human genes with and without baker's yeast homologs, (d) baker's yeast genes with and without human homologs, (e) baker's yeast genes with and without fission yeast homologs, (f) fission yeast genes with and without baker's yeast homologs.

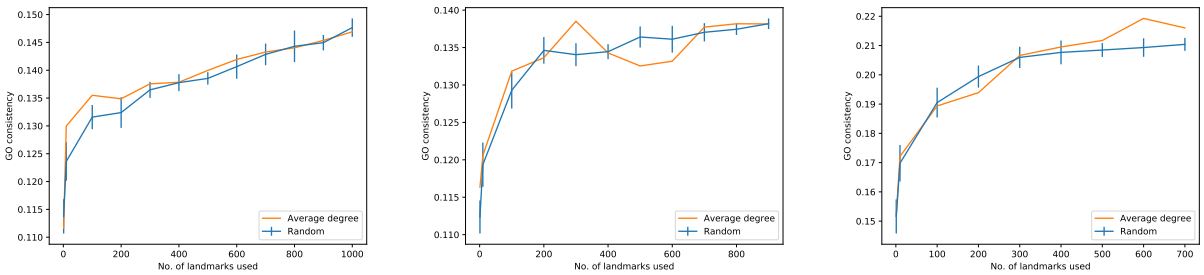


Figure S4. GO consistency scores of network alignments using MUNK-similarity scores (landmark pairs found in the matching are excluded from GOC score computation) using varying number of MUNK-landmark pairs, for (a) human (source) - mouse (target), (b) human (source) - baker's yeast (target), and (c) baker's yeast (source) - fission yeast (target).