Data-driven biological network alignment that uses topological, sequence, and functional information

Shawn Gu and Tijana Milenković

Department of Computer Science and Engineering, Eck Institute for Global Health, and Center for Network and Data Science, University of Notre Dame, Notre Dame, IN 46556

*To whom correspondence should be addressed (email: tmilenko@nd.edu)

Abstract

Many proteins remain functionally unannotated. Sequence alignment (SA) uncovers missing annotations by transferring functional knowledge between species' sequence-conserved regions. Because SA is imperfect, network alignment (NA) complements SA by transferring functional knowledge between conserved biological network, rather than just sequence, regions of different species. Existing NA assumes that it is topological similarity (isomorphic-like matching) between network regions that corresponds to the regions' functional relatedness. However, we recently found that functionally unrelated proteins are almost as topologically similar as functionally related proteins. So, we redefined NA as a data-driven framework, TARA, which learns from network and protein functional data what kind of topological relatedness (rather than similarity) between proteins corresponds to the proteins' functional relatedness. TARA used topological information (within each network) but not sequence information (between proteins across networks). Yet, its alignments yielded higher protein functional prediction accuracy than alignments of existing NA methods, even those that used both topological and sequence information. Here, we propose TARA++ that is also data-driven, like TARA and unlike other existing methods, but that uses across-network sequence information on top of within-network topological information, unlike TARA. To deal with the within-and-across-network analysis, we adapt social network embedding to the problem of biological NA. TARA++ outperforms protein functional prediction accuracy of existing methods.

1 Introduction

1.1 Background

Many proteins remain functionally unannotated [9]. A popular way to uncover missing annotations is to transfer functional knowledge across proteins of different species. This task of across-species protein functional prediction is the focus of this paper. The orthogonal task of within-a-species protein functional prediction, where a function of a protein in a species is predicted from function(s) of other protein(s) in the same species [30, 26], is out of the scope.

Genomic sequence alignment is commonly used for the task of across-species protein functional prediction, by transferring functional knowledge between conserved (aligned) sequence regions of proteins in different species. However, sequence alignment often fails: many sequence-similar proteins do not perform the same function(s), i.e., are functionally unrelated, and many sequence-dissimilar proteins are functionally related [17]; “functionally related” means that, according to current Gene Ontology (GO) annotation data [2], two proteins share a GO term, while “functionally unrelated” means that they share no GO terms. For example, of all yeast-human sequence orthologs from YeastMine [3], ~42% are not functionally related [17]. Such imperfect performance of sequence alignment could be due to sequence alignment failing to consider interactions between the genes, i.e., their protein products. However, a protein does not function alone. Instead, it interacts in complex networked ways with other proteins. So, accounting for protein-protein interactions (PPIs) is important for better protein functional prediction across species.

Luckily, large amounts of PPI network data are publicly available [6,34]. In a PPI network of a species, nodes are the species’ proteins and edges are PPIs between the proteins. Then, network alignment (NA) compares PPI networks of different species to find a "good" mapping between their nodes (proteins), i.e., a node mapping that uncovers regions of high network topological (and often sequence) conservation between the species; conservation typically means similarity. So, analogous to sequence alignment, NA can be used to transfer functional knowledge between conserved
(aligned) PPI network, rather than just sequence, regions of different species \[12,24,11,10,18\]. While the focus of our study is on computational biology, NA is also applicable to many other domains \[11\], such as machine translation in natural language processing, identity matching across different social media platforms, and visual feature matching in computer vision.

NA can be categorized into several broad types, whose high-level input/output/goal differences are as follows (more detailed algorithmic differences between specific NA methods are discussed in Section \[1,4\]).

First, NA can be pairwise (aligns two networks) or multiple (aligns three or more networks) \[12,18\]. We focus on pairwise NA because current multiple NA is more computationally complex \[18\] while also generally less accurate \[36\] than current pairwise NA.

Second, NA can be local or global \[24,18\], just like sequence alignment. Traditionally, local NA aims to find highly conserved network regions but usually results in such regions being small. Global NA aims to maximize overall network similarity, and while it usually results in large aligned network regions, these regions are suboptimally conserved. Each of local and global NA has its own (dis)advantages \[24,18\]. (More on local versus global NA follows shortly. We first need to define one-to-one versus many-to-many NA.)

Third, NA can be one-to-one (each node can be aligned to exactly one distinct node in another network) or many-to-many (a node may be aligned to more than one node in another network).

Traditionally, local NA has meant the same as many-to-many NA – given two networks \(G_1(V_1, E_1)\) and \(G_2(V_2, E_2)\), such an alignment is a relation \(R \subseteq V_1 \times V_2\). Also, global NA has meant the same as one-to-one NA – an injective function \(f : V_1 \to V_2\). Over time, local one-to-one and global many-to-many NA methods have been proposed. So, both local and global NA can now be seen as \(R \subseteq V_1 \times V_2\), with the two differing in how many nodes are covered by the aligned node pairs in the relation – significantly fewer for local than global NA, as the latter often aims to include into its alignment as many nodes from both networks as possible.

Because global NA has received more attention recently than local NA, we focus on the former (i.e., papers that proposed NA methods that we consider described the methods as global). Our considered task of across-species protein functional prediction is applicable to both one-to-one and many-to-many alignments. So, we do not favor one over the other. Yet, it is many-to-many NA methods that are the state-of-the-art in this task, which is why our considered methods happen to be many-to-many.

Fourth, there exist three NA method groups based on how input data are processed. The first group consists of NA methods that, given “only” two PPI networks, calculate each node’s feature using only the topological information within the given node’s own network (in isolation from the other network). As such, we refer to them as within-network-only NA methods. The resulting nodes’ topological features, which aim to summarize the nodes’ extended PPI network neighborhoods, are then used in various alignment processes (Section \[1,4\]). For state-of-the-art NA methods from this group, the topological features are based on graphlets \[25\], which are subgraphs, i.e., small, Lego-like building blocks of networks. The second group consists of NA methods that, given just two PPI networks but also sequence information for nodes across networks, first calculate each node’s topological feature in the same way as within-network-only methods, and only afterwards combine the sequence information with the topological features. Then, the combined data are used in various alignment processes (Section \[1,4\]). Because both within-network topological and across-network sequence information are used, but because the two are initially processed in isolation from each other and are combined only after the fact, we refer to this second group as isolated-within-and-across-network methods. Within-network-only methods can easily be used as isolated-within-and-across-network methods as long as sequence information is available; the latter usually lead to better alignments than their within-network-only versions \[24\]. The third group consists of NA methods that, given two PPI networks and sequence information for nodes across networks, first “integrate” the two networks into one by adding across-network “anchor” links (edges) between the highly sequence-similar proteins and only then proceed with any feature extraction or alignment process. So, the third group use both within-network topological and across-network sequence information. But, they first integrate the two data types and only then process them. This is why we refer to them as integrated-within-and-across-network methods.

1.2 Motivation

Regardless of which NA category they belong to, almost all existing NA methods assume that it is topological similarity between nodes (i.e., a high level of isomorphism-like matching between their extended PPI network neighborhoods as captured by the nodes’ topological features) that corresponds to the nodes’ functional relatedness, and thus they try to align such (topologically similar) nodes and network regions in general. However, it has been observed by multiple NA studies over the last several years that while existing NA methods result in high topological alignment quality (intuitively, many edges are conserved, i.e., the aligned network regions indeed have a high level of isomorphism-like match), their functional alignment quality is far from perfect (i.e., often, the aligned nodes are actually not functionally related \[10,24,18\].

Only recently, an attempt was made to understand this observation, and this was effort done by us \[17\]. Namely, we questioned the key assumption of NA as currently defined – that topologically similar nodes correspond to functionally
related nodes. We found for both synthetic and real PPI networks of different species that the functionally related nodes were only marginally more topologically similar than the functionally unrelated nodes, no matter which topological similarity measure was used [17].

This shocking result – the current NA assumption not holding – led us to redefine the NA problem as a data-driven framework, which aims to learn from the existing PPI network and protein functional data what kind of “topological relatedness” between proteins corresponds to the proteins’ functional relatedness, without assuming that topological relatedness means topological similarity. To understand how this framework operates, we next explain the difference between topological similarity and topological relatedness (see Fig. 1 for an illustration).

Suppose that: (i) PPI networks of yeast and human, the most commonly studied species in the NA field, are being aligned, (ii) the small toy networks in Fig. 1 are only parts of the full PPI networks, (iii) each node performs either the “green” or “yellow” function, and (iv) because of molecular evolutionary events such as gene duplication, deletion, or mutation, the “green” functional module in human (nodes 1, 2, 3, and 9) has an extra protein compared to the “green” module in yeast (nodes a, b, and c), and the “yellow” module in yeast has an extra interaction compared to the “yellow” module in human (Fig. 1). So, when aligning our toy networks, the “green” modules should be aligned to each other because they both perform the “green” function, and similarly the “yellow” modules should be aligned to each other. But, an NA method based on topological similarity will align “yellow” nodes e, f, g, and h in yeast to “green” nodes 1, 2, 3, and 9 in human (Fig. 1(a)), because both node sets form the same subgraph – a square with a diagonal, i.e., because the set of “yellow” nodes in yeast are topologically more similar to the set of “green” nodes in human than to the set of “yellow” nodes in human. However, this alignment is functionally incorrect because “yellow” and “green” nodes perform different functions. On the other hand, our NA framework based on topological relatedness will use the network topological and functional data to try to learn that a triangle in yeast (a, b, and c) should be aligned to a square-with-diagonal in human (1, 2, 3, and 9) because both perform the same function (“green”), and that a square-with-diagonal in yeast (e, f, g, and h) should be aligned to a square in human (5, 6, 7, and 8) because both perform the same function (“yellow”) (Fig. 1(b)). Then, in other parts of the PPI networks, our NA framework will try to align these learned patterns, to transfer the function from a triangle to a square-with-diagonal, and to transfer the function from a square-with-diagonal to a square. Loosely speaking, topological relatedness tries to account for evolutionary or other events that are likely to break the isomorphism-like assumption of the traditional topological similarity-based NA.

We named our topological relatedness-based NA framework TARA [17]. TARA uses supervised classification to learn what topological patterns should be aligned to each other (as described above). Specifically, given (i) a set of node pairs across the networks being aligned, say again yeast and human PPI networks for simplicity of explanation, such that the node in a given pair are functionally related (these node pairs form the “functionally related” class), (ii) a set of yeast-human node pairs across the networks such that nodes in a given pair are functionally unrelated (these node pairs form the “functionally unrelated” class), and (iii) graphlet-based network topological features of each node pair, TARA divides the node pairs (from each of the two classes) into training data and testing data. Then, it uses a classifier to learn from the training data what graphlet features distinguish between the functionally related and functionally unrelated node pairs. Next, given node pairs from the testing data and their graphlet features, TARA predicts whether the nodes in a given pair are functionally related or unrelated. Node pairs predicted as functionally related are added to TARA’s alignment, and this alignment is given to an established across-species protein functional prediction methodology [24] to obtain a list of protein functional annotations (i.e., protein-GO term pairs).

By learning topological relatedness patterns, TARA outperformed, in the task of across-species protein functional prediction between yeast and human, three state-of-the-art NA methods, WAVE [32], SANA [23], and PrimAlign [22]. To better understand the implications of these results, it is important to understand what type, i.e., within-network-only, isolated-within-and-across-network, or integrated-within-and-across-network, each method is. TARA, WAVE, and SANA are all within-network-only methods. They also all use graphlet-based topological node features. Their key difference is that TARA is uses topological relatedness and is supervised, while WAVE and SANA use topological
We introduce TARA-TS (TARA within-network Topology and across-network Sequence information) as our novel framework implementing the above idea. Like TARA, TARA-TS is supervised. However, unlike TARA and like PrimAlign, TARA-TS extracts features from an integrated yeast-human network. Because of the additional anchor links required compared to TARA, we cannot “just” use TARA’s feature extraction method in TARA-TS. Also, while both PrimAlign and TARA-TS are integrated-within-and-across-network methods, for reasons discussed in Section 2.2, we cannot “just” use PrimAlign’s feature extraction method in TARA-TS either. As a solution, we turn elsewhere, leveraging the extensive research on graph representation learning \cite{7,4,40}, whose goal is to embed nodes of a network into a low dimensional space such that network structure is preserved; the low-dimensional node representations are then used as node features. Graph representation learning, i.e., network embedding, has primarily been studied in the social network domain. So, we extend it from that domain to the field of computational biology. Specifically, our proposed TARA-TS generalizes a prominent network embedding method that was proposed for single-network analysis in machine learning tasks such as node classification, clustering, and link prediction, to the multi-network task of biological NA. Given the node features extracted by network embedding, TARA-TS works just as TARA to produce an alignment. Then, we use this alignment for across-species protein functional prediction.

We compare prediction accuracy of TARA-TS (pairwise, global, many-to-many, integrated-within-and-across-network, supervised) with accuracies of TARA and PrimAlign, as they are state-of-the-art NA methods that were already shown to outperform many other existing NA methods on the exact same data as what we use here. So, by transitivity, if TARA-TS is shown to be superior to TARA and PrimAlign, this will mean that TARA-TS is superior to the other existing methods as well. Also, of all existing methods, TARA and PrimAlign are the most similar and thus fairly comparable to TARA-TS. Namely, TARA is pairwise, global, many-to-many, and supervised, like TARA-TS. The difference is that TARA is a within-network-only method while TARA-TS is an integrated-within-and-across-network method. So, we can fairly test the effect of going from within-network-only to integrated-within-and-across-network in the supervised context. PrimAlign is a pairwise, global, many-to-many, and integrated-within-and-across-network method, like TARA-TS. The difference is that PrimAlign is unsupervised while TARA-TS is supervised. So, we can fairly test the effect of going from unsupervised to supervised for integrated-within-and-across-network methods.

When we compare TARA-TS against TARA, we actually compare whether using across-network sequence information on top of within-network topological information leads to more accurate predictions, as we expect. Shockingly, we find that TARA-TS and TARA are almost equally as accurate. Closer examination reveals that their quantitatively similar results are not because the two methods are predicting the same information (which would make one of them redundant). Instead, their predicted protein functional annotations are quite complementary. So, we then look at those predictions (protein-GO term associations) that are made by both methods, only those predictions made by TARA-TS but not TARA, and only those predictions made by TARA but not TARA-TS. We find the former (the overlapping predictions) to be more accurate than the predictions made by any one of TARA-TS or TARA alone. So, we take this overlapping version of TARA-TS and TARA as our final method, TARA++. In a sense, TARA++ is integrating state-of-the-art research knowledge across computational biology and social network domains, by combining TARA’s graphlet-based topology-only features with TARA-TS’s embedding-based topology-and-sequence features, each of which boosts the other’s performance. Very few studies have explored such a promising direction to date \cite{27}. Importantly, we find that TARA++ not only outperforms TARA but also PrimAlign.

1.3 Our contributions

We introduce TARA-TS (TARA within-network Topology and across-network Sequence information) as our novel framework implementing the above idea. Like TARA, TARA-TS is supervised. However, unlike TARA and like PrimAlign, TARA-TS extracts features from an integrated yeast-human network. Because of the additional anchor links required compared to TARA, we cannot “just” use TARA’s feature extraction method in TARA-TS. Also, while both PrimAlign and TARA-TS are integrated-within-and-across-network methods, for reasons discussed in Section 2.2, we cannot “just” use PrimAlign’s feature extraction method in TARA-TS either. As a solution, we turn elsewhere, leveraging the extensive research on graph representation learning \cite{7,4,40}, whose goal is to embed nodes of a network into a low dimensional space such that network structure is preserved; the low-dimensional node representations are then used as node features. Graph representation learning, i.e., network embedding, has primarily been studied in the social network domain. So, we extend it from that domain to the field of computational biology. Specifically, our proposed TARA-TS generalizes a prominent network embedding method that was proposed for single-network analysis in machine learning tasks such as node classification, clustering, and link prediction, to the multi-network task of biological NA. Given the node features extracted by network embedding, TARA-TS works just as TARA to produce an alignment. Then, we use this alignment for across-species protein functional prediction.

1.4 Related work

First, we discuss within-network-only and isolated-within-and-across-network methods. They have two parts. First, similarities are computed for all pairs of nodes across networks. For within-network-only methods, these are topological similarity and are unsupervised. Thus, WAVE and SANA were the most fairly comparable methods to TARA. So, we could fairly evaluate whether moving from WAVE’s and SANA’s topological similarity to TARA’s supervision-based topological relatedness helped. TARA significantly outperformed WAVE and SANA, so we could conclude that it did help. PrimAlign is one of very few existing integrated-within-and-across-network methods. Because PrimAlign was already shown to outperform many isolated-within-and-across-network methods \cite{22} on the exact same data as in TARA’s evaluation \cite{17}, there was no need to evaluate TARA against any methods of that type. Importantly, TARA still outperformed PrimAlign, despite the former not using any sequence information. This already showed how powerful the supervised NA paradigm is. In this study, we push the boundary further. TARA “only” showed that going from unsupervised to supervised for within-network-only methods improved alignment accuracy, but we also already know that going from within-network-only to isolated-within-and-across-network to integrated-within-and-across-network in the unsupervised context also improves accuracy. So, a method that is both supervised and of the integrated-within-and-across-network type should further improve alignment accuracy. Thus, here, we propose the first ever method of this type.
As typically done in NA studies, we analyze PPI networks of yeast (5,926 nodes and 88,779 edges) and human (15,848 nodes and 269,120 edges) from BioGRID [6]. Also, like the PrimAlign study [22], we consider 55,594 yeast-human protein pairs with E-value sequence similarities ≤ 10^{-7} as anchor links.

Our supervised NA framework also requires knowledge about whether two proteins are functionally related. As

2 Methods

2.1 Data

As typically done in NA studies, we analyze PPI networks of yeast (5,926 nodes and 88,779 edges) and human (15,848 nodes and 269,120 edges) from BioGRID [6]. Also, like the PrimAlign study [22], we consider 55,594 yeast-human protein pairs with E-value sequence similarities ≤ 10^{-7} as anchor links.

Our supervised NA framework also requires knowledge about whether two proteins are functionally related. As
a possible reason for this lack is that the node pair feature vectors obtained in this way are not substantially different.

TARA-TS analyzes the integrated yeast-human network, consisting of 21,774 nodes (5,926 yeast + 15,848 human proteins) and 413,493 edges (88,779 yeast PPIs + 269,120 human PPIs + 55,594 anchor links). Then, TARA-TS uses a prominent network embedding method, node2vec [14], on this network to obtain nodes’ feature vectors. Node2vec uses random walks to explore the neighborhood of a node in a network, and then uses a skip-gram model to summarize these random walks into the node’s feature vector. The only way to use node2vec, a single-network method, in the multi-network NA task, is to first integrate the two networks via anchor links, as we do. Otherwise, node2vec fails if applied to the two networks individually [16]. TARA-TS first applies node2vec to the integrated network with the default parameters to obtain a feature vector for each node. Then, as suggested by the node2vec study [14], to get the feature vector of a yeast-human node pair, TARA-TS takes the element-wise average of the nodes’ feature vectors.

There are three reasons why we use node2vec over other network embedding methods. First, even more recent embedding methods, when evaluated in their own respective papers, achieve similar performance as node2vec in many tasks. So, we do not expect those recent methods to perform significantly better than node2vec in our task. Second, the node2vec source code is readily available and well documented, unlike for many other methods. Third, more generally, the goal of this study is not to find the absolute best feature vector for supervised NA, but rather to test how combining topological and sequence information in supervised NA affects protein functional prediction. Importantly to this point, if using node2vec already improves upon current NA methods, then using any more sophisticated ways to extract features will only improve further. We propose a framework such that features from any new extraction method can simply be “swapped” in, allowing flexibility for further advancements.

To understand why TARA-TS does not perform node pair feature extraction in the same way as TARA, we must first understand how TARA does it this way, i.e., why it does not use TARA’s or PrimAlign’s feature extraction method.

To obtain node features that capture both within-network topological and across-network sequence information, TARA-TS analyzes the integrated yeast-human network, consisting of 21,774 nodes (5,926 yeast + 15,848 human proteins) and 413,493 edges (88,779 yeast PPIs + 269,120 human PPIs + 55,594 anchor links). Then, TARA-TS uses the element-wise average of the nodes’ feature vectors.
TARA do not capture heterogeneous information present in the integrated network, i.e., do not distinguish between different types of nodes (yeast and human) or edges (yeast PPIs, human PPIs, and yeast-human sequence-based anchor links). Essentially, for many nodes, their graphlets will likely be capturing more of the within-network topological information than of the across-network sequence information due factors (i) and (ii), and without a way to “boost” this implicit across-network information by explicitly distinguishing between different node/edge types, the nodes’ graphlet counts end up being relatively similar between TARA-TS and TARA’s.

Then, the question arises: why does node2vec on the integrated network (i.e., TARA-TS) not have (as big of a) problem extracting both within-network topological and across-network sequence information compared to graphlets on the very same integrated network? We hypothesize that node2vec is better able to do so because of its “Length of walk per source (\(-1:1\))” parameter, which specifies how many steps a given random walk takes. At a value of 80 (typically used by node2vec as well as its predecessors), this means that even though the number of across-network anchor links is lower than the number of within-network edges, any random walk starting from any node in the yeast PPI network is very likely to contain some nodes from the human PPI network (and vice versa), allowing node2vec-based feature vectors to capture more of the across-network sequence information than graphlet-based feature vectors, which only look a maximum of four nodes away (for 5-node graphlets). Also, because of the increased amount of implicit across-network information captured, it may not be as necessary to explicitly distinguish between node/edge types in node2vec (although this could help further – see below). In other words, factors (i), (ii), and (iii) are not as significant for the node2vec-based feature vectors compared to the graphlet-based ones.

If counting graphlets cannot effectively consider across-network information on the integrated network, a better way to distinguish which graphlets contain across-network nodes may lead to improved results. As referenced in factor (iii) above, the integrated network could be seen as a heterogeneous network containing different node and types. Then, heterogeneous graphlets \([15]\) (as opposed to traditional homogeneous graphlets discussed thus far) could in theory be used to extract both within- and across-network feature vectors of the nodes while distinguishing between node/edge types. However, in practice, heterogeneous graphlet counting is infeasible for as large networks as studied in this paper, due to its exponential computational complexity. The reason this is not an issue for homogeneous graphlet counting is that methods such as Orca \([20]\) rely on combinatorics to infer the counts of some (larger) graphlets from the counts of other (smaller) graphlets, significantly reducing the computational complexity. Such combinatorial relationships have not yet been developed for counting heterogeneous graphlets, so we cannot use them. A similar issue affects a method that can directly extract the feature vector of a node pair \([21]\), as opposed to extracting graphlet features of individual nodes and then combining the features of two nodes, as TARA does: a combinatorial approach for direct node pair graphlet counting does not exist. Instead, current heterogeneous and node pair graphlet counting require exhaustive graphlet enumeration and are thus infeasible.

Note that heterogeneous random walk-based network embedding methods do exist \([8]\), and could be used to better distinguish between node/edge type than the homogeneous node2vec. However, such methods require extensive testing to define “metapaths” (sequences of node types that direct how the random walks move) that perform well for the task at hand. This is outside the scope of this study because, again, our goal is not to find the absolute best feature for supervised NA, but rather to test the effect of combining topological and sequence information in supervised NA. Any more advanced features, when “swapped” into our supervised NA framework, can only improve our framework even further.

Since existing graphlets approaches cannot effectively capture both within-network topological and across-network sequence information, the next logical choice is to use feature vectors based on PrimAlign, which already integrates such information. However, PrimAlign’s algorithmic design does not allow for this. Recall from Section 1.4 that PrimAlign models the integrated network as a Markov chain, which is then repeatedly transitioned until convergence. This means that the weights between every node pair are updated at the same time, based on the weights of every node pair from the previous state of the chain. So, PrimAlign operates on every node pair at once with respect to their weights, rather than on individual nodes or node pairs with respect to any kind of feature vector, meaning that we cannot extract such information. For this reason, we instead had to resort to generalizing social network embedding to biological NA.

### 2.3 TARA-TS’s classification and alignment generation

Here, we describe the supervised NA process, given the yeast and human PPI networks, functional information, and node pair feature vectors. For a given ground truth-rarity dataset (Section \([24]\), the positive class consists of functionally related node pairs, and the negative class consists of functionally unrelated node pairs. Because the latter is much larger, we create a balanced dataset by sampling the negative class to match the size of the positive class, as typically done \([33]\). Then, we train a logistic regression classifier using TARA-TS’s feature vector for a node pair to predict whether the two nodes are functionally related (in the TARA paper we found that the classifier choice had no significant effect on alignment accuracy). Specifically, for a balanced dataset, the classifier is trained on \(y\%\) of the data and tested on the remaining \((100 – y)\%\) of the data, where the data is split such that the class balance is preserved in each of the training
and testing sets. We vary $y$ from 10 to 90 in increments of 10 (called a “$y$ percent training test”). Each value of $y$ can be considered a different version of TARA-TS, i.e., TARA-TS-10 to TARA-TS-90.

To evaluate a given TARA-TS version, we use the trained classifier to predict on the testing set, measuring accuracy and area under receiver operating characteristic curve (AUROC). To ensure the outcomes are not due to how the data is split, for a given $y$ value, we randomly create 10 instances of the training/testing split, resulting in 10 accuracy and 10 AUROC scores, and we average the results. Then, to ensure results are not due to how the negative class is sampled, we create 10 balanced datasets and repeat the classification and evaluation process for each one, averaging the results. Only if the average accuracy and AUROC are high, i.e., a given TARA-TS version accurately predicts functionally related nodes to be functionally related, does it make sense to create an alignment for use in protein functional prediction. If this is the case, then on a given balanced dataset, for a given TARA-TS version, for only one training/testing split, we take every node pair from the testing set that is predicted as functionally related, and add it to the alignment – it is important to only use the testing set for the alignment, to avoid any circular argument. For simplicity, instead of using all 10 training/testing splits, we only use the “first” one, which in our implementation corresponds to a starting seed of 0 for Python’s random number generator; we found that the split choice had no major effect on the prediction accuracy. This results in 10 alignments, corresponding to the 10 balanced datasets, for each of the nine TARA-TS versions. We use all 90 TARA-TS alignments for protein functional prediction, as follows.

2.4 Using an alignment for protein functional prediction

An ultimate goal of biological NA is across-species protein functional prediction, so each NA method must be evaluated in this context. For a given ground truth-rarity dataset (atleast$k$-EXP at the $r$ rarity threshold), we consider GO terms that annotate at least two yeast proteins and at least two human proteins; these minimums are required to be able to make predictions for the GO term. Then, we use a (TARA-TS’s or an existing method’s) alignment in an established protein functional prediction framework [24], as follows. In the alignment, for each protein $u$ that is annotated by at least $k$ GO term(s), such that the GO term(s) annotate $r$ or fewer proteins (i.e., for each protein for which a prediction can actually be made), $u$’s true GO term(s) are hidden. Then, for each GO term $g$, the framework determines if the alignment is significantly “enriched” in $g$. The hypergeometric test is used for this, calculating if the number of aligned node pairs in which the aligned proteins share $g$ is significantly high ($p$-value less than 0.05 [24]). If so, node $u$ is predicted to be annotated by GO term $g$. Repeating for all applicable proteins and GO terms results in the final list of predicted protein-GO term associations. From this prediction list, the framework calculates the precision (percentage of the predictions that are in a given ground truth-rarity dataset) and recall (percentage of the protein-GO term association from a given ground truth-rarity dataset that are among the predictions).

3 Results

3.1 TARA-TS versus TARA in the classification task

Here, we study classification performance of TARA-TS and TARA, i.e., how well they distinguish between functionally related and functionally unrelated yeast-human proteins from the testing data. We would ideally do this on all nine ground truth-rarity datasets. However, two of the datasets, atleast3-EXP at the 50 and 25 thresholds, are too small for TARA-TS and TARA to perform any classification on; inability to learn on small datasets is a drawback of machine learning methods in general, not just TARA-TS and TARA. Thus, we only have seven viable ground truth-rarity datasets.

Due to space constraints, we discuss the effects of various parameters ($k$ in atleast$k$-EXP, GO term rarity threshold, and $y$ percent training test) on classification accuracy and AUROC for TARA-TS in Supplementary Section S1.1. Instead, here we focus on comparing TARA-TS against TARA. Because the two only differ in their features, we can fairly compare a given TARA-TS version (i.e., TARA-TS in the given $y$ percent training test) to the corresponding TARA version. We expect TARA-TS to have higher accuracy and AUROC, as it extracts topology plus sequence features from the integrated yeast-human network. Indeed, this is what we observe for every comparison (Supplementary Figs. S1-S2). Namely, TARA-TS’s relative accuracy increase over TARA’s is 6%-27% depending on the atleast$k$-EXP ground truth dataset, GO term rarity threshold, and $y$ percent training test, with an average increase of 14%; TARA-TS’s relative AUROC increase is 9%-32%, with an average increase of 16%.

Since TARA-TS predicts functional relatedness better than TARA, it makes sense to use TARA-TS in the protein functional prediction task.
3.2 TARA-TS versus TARA in the task of protein functional prediction: toward TARA++

Here, we evaluate protein functional prediction accuracy of TARA-TS’s and TARA’s alignments. For simplicity, we consider a subset of TARA-TS versions and the corresponding TARA versions. Specifically, we focus on the extremes (10 and 90) and the middle (50) because classification accuracy does not vary significantly between the different percent training versions. Recall that classification cannot be performed on two (small) ground truth-rarity datasets, at least 3-EXP at thresholds 50 and 25, so no alignments exist for them, and thus protein functional prediction is not possible. So, we have 21 total TARA-TS versus TARA evaluation tests, resulting from combinations of the three selected versions and the seven viable (as opposed to the nine total) ground truth-rarity datasets.

First, we study the performance of TARA-TS. We expect that as we increase the amount of training data (10 to 50 to 90), precision will increase and recall will decrease. This is because a larger training dataset likely means that the classifier will be better (increasing precision), but will lead to a smaller testing dataset and thus smaller alignments and fewer predictions (decreasing recall). We expect that as \( k \) increases (in our at least \( k \)-EXP ground truth datasets), precision will increase and recall will decrease. This is because at higher \( k \), TARA-TS will be training on higher-quality data (increasing precision), but there will be less data overall, resulting in smaller alignments and fewer predictions (decreasing recall). We expect that as we consider rarer GO terms, precision will increase and recall will decrease. Rarer GO terms may be more meaningful \([19, 17]\), so the data will be of higher quality (increasing precision), but again there will be less of it overall (decreasing recall). Indeed, we observe all of these expected trends (Fig. 2(a) and Supplementary Fig. S3).

Second, we compare the performance of TARA-TS and TARA. Shockingly, while TARA-TS results in higher classification accuracy than TARA (Section 3.1), TARA-TS and TARA are almost equally as accurate in the task of protein functional prediction. This is unexpected because TARA-TS uses sequence information that TARA does not (in addition to both using the same topological information). To better understand why this is happening, we take a closer look at the alignments and predictions made by each method in order to see if the two methods are aligning the same nodes, or predicting the same protein-GO term associations. So, we investigate how much their alignments overlap, and how much their predictions overlap. We find that neither their alignments (Fig. 2(b) and Supplementary Fig. S4) nor their predictions (Fig. 2(c) and Supplementary Fig. S5) are highly overlapping; instead, they are quite complementary.

If they have different alignments and make different predictions, how could they still have similar accuracy? To answer this, we look at the precision and recall of predictions made by both methods, only those predictions made by TARA-TS but not TARA, and only those predictions made by TARA but not TARA-TS (Fig. 2(c) and Supplementary Fig. S5). From this, we highlight two findings. First, graphlets, which only use topological information, can perform as well as network embedding features that use both topological and sequence information. This is supported by the fact that predictions made only by TARA and only by TARA-TS produce similar accuracy in almost all evaluation tests. So, this motivates the need to develop graphlet-based methods for integrated networks as future work. Second, predictions made by both methods are significantly more accurate than predictions made by any one method alone. In a sense, their overlap is integrating state-of-the-art research across the computational biology and social network domains, by
combining TARA’s graphlet-based topology-only features with TARA-TS’s embedding-based topology-and-sequence features. So, the overlapping predictions kind of combine the strengths of both domains.

Because the overlap of TARA-TS and TARA has such high prediction accuracy, we take it as our new TARA++ method, which we consider further. Then, to simplify comparisons between TARA++ and existing NA methods, we choose a representative percent training test (TARA++10, TARA++50, or TARA++90) for each of the seven viable ground truth-rarity datasets. In other words, we go from 21 “TARA++ versus existing methods” evaluation tests to seven. Generally, we try to choose the percent training test that has both high precision (predictions are accurate) and a large number of predictions (we uncover as much of biological knowledge as possible), as these represent TARA++’s best results. So, we choose TARA++90 for all ground truth-rarity datasets except at least 2-EXP at the 50 and 25 rarity thresholds, where we choose TARA++10. Henceforth, we refer to all of the selected TARA++ versions simply as TARA++.

### 3.3 TARA++ versus existing NA methods in the task of protein functional prediction

Per reasoning in Section 1, we compare TARA++’s predictions against those resulting from alignments of two state-of-the-art and most fairly comparable existing methods, TARA and PrimAlign. Also, we consider predictions resulting from an alignment that only uses sequence information, Sequence. For the latter, we consider the 55,594 anchor links by themselves and treat them as the alignment; clearly, no topological information is used in this alignment, so this is not an NA method. With TARA and Sequence we can separately analyze each aspect, i.e., within-network topological information only and across-network sequence information only, and evaluate how each compares to our integrative TARA++. (Clearly, TARA++’s predictions are by definition a subset of TARA’s predictions, and so we expect TARA++ to have higher precision but lower recall.) With PrimAlign, we can evaluate how this integrative but unsupervised method compares to our also integrative but supervised TARA++. Importantly, TARA and PrimAlign were already shown to outperform many previous NA methods (Section 1.4). So, comparing to these two methods is sufficient. Also, keep in mind that like with TARA, a theoretical precision of 1 is not practically possible with TARA++. This is because TARA++ uses a part (up to 90%) of the ground truth functional data for training, and it impossible to make predictions for that portion.

We believe that precision is more important than recall. This is because for potential wet lab validation of predictions, it is more important to have fewer but mostly correct predictions (e.g., 90 correct out of 100 made) than a greater number of mostly incorrect predictions (e.g., 300 correct out of 1000 made). While in the latter example more predictions are correct, leading to higher recall, many more are also incorrect, leading to lower precision. We do not completely discount recall though, as it still brings value.

Our key results are as follows (Fig. 1 and Supplementary Fig. S6). In terms of precision, TARA++ is the best for 6/7 ground truth-rarity datasets. It is only slightly inferior to PrimAlign for 1/7 datasets (at least 1-EXP for ALL GO terms), but TARA++ has much higher recall than PrimAlign on this dataset. Speaking of recall, TARA is expected to always outperform TARA++, and this is what we observe. Of the remaining existing methods, TARA++ is the best for 2/7 datasets – at least 1-EXP at the ALL and 50 rarity thresholds – even though TARA++ makes much fewer predictions than the next best method, Sequence. For the other datasets, TARA++’s recall is lower than that of PrimAlign and Sequence. This is expected, since TARA++ makes fewer predictions than the other methods. Importantly, the difference in recall between TARA++ and every other method is relatively small, for example only 0.06 lower on average compared to TARA, while TARA++ is much better in terms of precision than every other method, for example 0.2 greater on average compared to TARA. As discussed above, such a trade-off between precision and recall is worth it for our task.

We see that the precision of TARA++ is much greater than simply the sum of precision from TARA and Sequence, suggesting that integrating within-network topological and across-network sequence information has compounded effects. This further highlights the need for such approaches.

Finally, we look at the time it takes to obtain an alignment for TARA-TS, TARA, and PrimAlign for the ALL GO term rarity threshold, as this threshold has the most data and thus will be the worst case time-wise out of all thresholds. Because TARA++ comes from the intersection of TARA-TS’s and TARA’s results, one can think of TARA++’s time as either the maximum or sum of TARA-TS’s and TARA’s, if the two are run at the same time or one after the other, respectively. As k (in the at least k-EXP ground truth dataset) increases, we expect the time for TARA-TS to produce an alignment to decrease, as there is less (but higher-quality) data overall, and thus less data to train on. This is what we observe (Table 1 and Supplementary Table S1). We also find that TARA-TS is faster than TARA, and this difference comes from the feature computation time, as both use the same supervised framework. TARA-TS’s computation using node2vec is expectedly faster than TARA’s graphlet counting even when using Orca for two reasons. First, the random walks produced by node2vec can be thought of as samples of the network structure, which loosely correspond to samples of each node’s true graphlet counts. Naturally, sampling is much faster than obtaining the exact counts. Second, node2vec is automatically parallelized over the number of random walks per node (parameter -r), so that...
multiple random walks can be performed at the same time. However, Orca is not parallelized. To comment on this point, in node2vec, the same number of random walks is performed for each node, meaning that parallelization helps significantly as no one node takes much longer than any other. However, for graphlet counting, nodes with e.g., high degrees are the limiting time factor, so parallelization would not help as much. Also, TARA-TS’s (and PrimAlign’s) running time has an unfair advantage over TARA in that TARA-TS (and PrimAlign) requires sequence-based anchor links computed by BLAST [1], and those were obtained precomputed from the PrimAlign study. Therefore, TARA-TS’s (and PrimAlign’s) running time is missing a step compared to TARA’s. Despite this missing step, regardless of how TARA-TS and TARA are combined to form TARA++, PrimAlign will still be faster. However, it is about half as precise as TARA++. And even though TARA++ is slower, it is still practically feasible in terms of speed, so the extra time is definitely worth the almost doubling of precision.

|                | TARA-TS | TARA  | PrimAlign | Sequence |
|----------------|---------|-------|-----------|----------|
| Running time   | 444     | 4634  | 16        | N/A      |

Table 1: Running times for TARA-TS, TARA, PrimAlign, and Sequence, for the atleast3-EXP ground truth dataset at the ALL GO term rarity threshold. TARA++’s running time is a function of TARA-TS’s and TARA’s (see the text). We use a precomputed alignment for Sequence (see the text), hence the “N/A”. Running time results are qualitatively similar for the other ground truth-rarity datasets (Supplementary Table S1).

4 Conclusion

TARA++ pushes the data-driven NA paradigm further. We showed that by integrating research knowledge across the computational biology and social network domains, TARA++ outperforms state-of-the-art NA methods in the task of protein function prediction, an ultimate goal of NA. As TARA++ is the first data-driven NA method to integrate topological and sequence information, it is just a proof-of-concept approach. This work can be taken further. We found that graphlet-based features on the isolated networks (on topological information alone) perform as well as embedding-based features on the integrated network (on topological and sequence information combined), even though the latter (using more data) was expected to be better. So, developing a graphlet feature that would efficiently deal with an integrated network could yield further improvements. This might include novel algorithms for significantly speeding up counting of heterogeneous graphlets in large data. Heterogeneous graphlets, or heterogeneous network embedding features, could distinguish between different node/edge types in an integrated network and thus only improve over the homogeneous features considered in this study. Also, we focused on NA of static networks. However, research in NA of dynamic (e.g., aging- or disease progression-related) networks is becoming popular [35][37]. So, our framework can be adapted to such novel NA categories.
Funding

National Science Foundation (NSF) CAREER CCF-1452795 award.

References

[1] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research, 25(17):3389–3402, 1997.

[2] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, et al. Gene Ontology: tool for the unification of biology. Nature Genetics, 25(1):25, 2000.

[3] R. Balakrishnan, J. Park, K. Karra, B. C. Hitz, G. Binkley, et al. YeastMine: an integrated data warehouse for Saccharomyces cerevisiae data as a multipurpose tool-kit. Database, 2012, 2012.

[4] H. Cai, V. W. Zheng, and K. C.-C. Chang. A comprehensive survey of graph embedding: Problems, techniques, and applications. IEEE Transactions on Knowledge and Data Engineering, 30(9):1616–1637, 2018.

[5] X. Cao, Z. Chen, X. Zhang, and Y. Yu. IMAP: An iterative method for aligning protein-protein interaction networks. In 2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pages 317–324. IEEE, 2017.

[6] A. Chatr-Aryamontri, R. Oughtred, L. Boucher, J. Rust, C. Chang, et al. The BioGRID interaction database: 2017 update. Nucleic Acids Research, 45(D1):D369–D379, 2017.

[7] P. Cui, X. Wang, J. Pei, and W. Zhu. A survey on network embedding. IEEE Transactions on Knowledge and Data Engineering, 31(5):833–852, 2018.

[8] Y. Dong, N. V. Chawla, and A. Swami. metapath2vec: Scalable representation learning for heterogeneous networks. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 135–144. ACM, 2017.

[9] K. W. Ellens, N. Christian, C. Singh, V. P. Satagopam, P. May, and C. L. Linster. Confronting the catalytic dark matter encoded by sequenced genomes. Nucleic Acids Research, 45(20):11495–11514, 2017.

[10] A. Elmsallati, C. Clark, and J. Kalita. Global alignment of protein-protein interaction networks: A survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 13(4):689–705, 2016.

[11] F. Emmert-Streib, M. Dehmer, and Y. Shi. Fifty years of graph matching, network alignment and network comparison. Information Sciences, 346:180–197, 2016.

[12] F. E. Faisal, L. Meng, J. Crawford, and T. Milenković. The post-genomic era of biological network alignment. EURASIP Journal on Bioinformatics and Systems Biology, 2015(1):3, 2015.

[13] J. Fan, A. Cannistra, I. Fried, T. Lim, T. Schaffner, et al. Functional protein representations from biological networks enable diverse cross-species inference. Nucleic Acids Research, 47(9):e51–e51, 2019.

[14] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 855–864. ACM, 2016.

[15] S. Gu, J. Johnson, F. E. Faisal, and T. Milenković. From homogeneous to heterogeneous network alignment via colored graphlets. Scientific Reports, 8(1):12524, 2018.

[16] S. Gu and T. Milenković. Graphlets versus node2vec and struc2vec in the task of network alignment. In Proceedings of the 14th International Workshop on Mining and Learning with Graphs (MLG) at the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018.

[17] S. Gu and T. Milenković. Data-driven network alignment. arXiv preprint arXiv:1902.03277, 2019.

[18] P. H. Guzzi and T. Milenković. 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, 2017.

[19] W. B. Hayes and N. Mamano. SANA NetGO: a combinatorial approach to using Gene Ontology (GO) terms to score network alignments. Bioinformatics, 34(8):1345–1352, 2017.

[20] T. Hočevar and J. Demšar. A combinatorial approach to graphlet counting. Bioinformatics, 30(4):559–565, 2014.

[21] Y. Hulovaty, R. W. Solava, and T. Milenković. Revealing missing parts of the interactome via link prediction. PLOS ONE, 9(3):e90073, 2014.

[22] K. Kalecky and Y.-R. Cho. PrimAlign: PageRank-inspired Markovian alignment for large biological networks. Bioinformatics, 34(13):i537–i546, 2018.

[23] N. Mamano and W. B. Hayes. SANA: simulated annealing far outperforms many other search algorithms for biological network alignment. Bioinformatics, 33(14):2156–2164, 02 2017.
Supplementary material: Data-driven biological network alignment that uses topological, sequence, and functional information

S1 Results

S1.1 TARA-TS versus TARA in the classification context

Here, we comment on the performance of TARA-TS. For a fixed GO term rarity threshold, as \( k \) in our atleast\( k \)-EXP ground truth datasets increases, we expect TARA-TS’s (and TARA’s) accuracy and AUROC to increase, as the condition for proteins to be functionally related becomes more stringent and thus the functional data becomes of higher quality. Also, for a fixed \( k \), as we decrease the GO term rarity threshold (i.e., consider rarer GO terms), we expect accuracy and AUROC to increase, since rarer GO terms may be meaningful (Hayes and Mamano, 2017), again resulting in higher-quality data. We find the former expectation to hold, for all GO term rarity thresholds (Supplementary Figs. S1-S2). However, for the latter expectation, we find that classification accuracy and AUROC somewhat decrease (Supplementary Figs. S1-S2). This may be because as rarer GO terms are considered, the amount of training data decreases, which is what could be causing performance decreases.

As we increase \( y \), the amount of training data, we expect accuracy and AUROC to increase, as more data is used during classification. For accuracy, we observe this for 6/7 ground truth-rarity datasets, although for 4/6 of the datasets, the increase is minimal (~1%). In the remaining case, the accuracy increases until about 60% training data, and then drops. For AUROC, we observe the expected trend for all ground truth-rarity datasets, although for 4/7 of these datasets, the increase is minimal (~1%). These unexpected trends (mostly minor increase of accuracy and AUROC even with large increase of \( y \)) are promising though, because they mean that TARA-TS does not have to use a majority of the functional data for training to still obtain good results; even using only 10% of the data seems to suffice.

S2 Supplementary figures and tables

|           | atleast1-EXP | atleast2-EXP | atleast3-EXP |
|-----------|---------------|---------------|---------------|
| TARA-TS   | 3811          | 480           | 444           |
| TARA      | 8090          | 4676          | 4634          |
| PrimAlign | 16            | 16            | 16            |
| Sequence  | N/A           | N/A           | N/A           |

Supplementary Table S1: Running times (in seconds) of TARA-TS, TARA, PrimAlign, and Sequence, when considering ALL GO terms. TARA++’s running time is a function of TARA-TS’s and TARA’s (see Section 3.3 in the main paper). We use a precomputed alignment for Sequence (see Section 3.3 in the main paper), hence the "N/A"s.
Supplementary Figure S1: Average prediction accuracy of percent training tests for rarity thresholds (a, d, g) ALL, (b, e) 50, and (c, f) 25 using ground truth datasets (a, b, c) atleast1-EXP, (d, e, f) atleast2-EXP, and (g) atleast3-EXP. A dotted black line indicates the accuracy expected if the classifier makes random predictions. Qualitatively similar results for AUROC are shown in Supplementary Figs. S2.
Supplementary Figure S2: Average AUROC of percent training tests for rarity thresholds (a, d, g) ALL, (b, e) 50, and (c, f) 25 using ground truth datasets (a, b, c) at least 1-EXP, (d, e, f) at least 2-EXP, and (g) at least 3-EXP. A dotted black line indicates the AUROC expected if the classifier makes random predictions.
Supplementary Figure S3: Comparison of TARA and TARA-TS for rarity thresholds (a, d, g) ALL, (b, e) 50, and (c, f) 25 using ground truth datasets (a, b, c) atleast1-EXP, (d, e, f) atleast2-EXP, and (g) atleast3-EXP in the task of protein functional prediction. The alignment size (i.e., the number of aligned yeast-protein pairs) and number of functional predictions (i.e., predicted protein-GO term associations) made by each method are shown above. For example, the alignment for TARA-10 in (a) contains 244,433 aligned yeast-human protein pairs, and predicts 538,397 protein-GO term associations. Raw precision and recall values are color-coded inside each panel.
Supplementary Figure S4: Overlap of the alignments made by TARA and TARA-TS for rarity thresholds (a, d, g) ALL, (b, e) 50, and (c, f) 25 using ground truth datasets (a, b, c) at least1-EXP, (d, e, f) at least2-EXP, and (g) at least3-EXP. Percentages are out of the total number of unique aligned node pairs made by both methods combined.
Supplementary Figure S5: Overlap of the predictions made by TARA and TARA-TS for rarity thresholds (a, d, g) ALL, (b, e) 50, and (c, f) 25 using ground truth datasets (a, b, c) at least 1-EXP, (d, e, f) at least 2-EXP, and (g) at least 3-EXP. Percentages are out of the total number of unique predictions made by both methods combined. Precision and recall of each set are shown below.
Supplementary Figure S6: Comparison of four NA methods for rarity thresholds (a, d, g) ALL, (b, e) 50, and (c, f) 25 using ground truth datasets (a, b, c) atleast1-EXP, (d, e, f) atleast2-EXP, and (g) atleast3-EXP in the task of protein functional prediction. The alignment size (i.e., the number of aligned yeast-protein pairs) and number of functional predictions (i.e., predicted protein-GO term associations) made by each method are shown above. For example, the alignment for TARA in (a) contains 27,155 aligned yeast-human protein pairs, and predicts 91,618 protein-GO term associations. Raw precision and recall values are color-coded inside each panel.