Supplementary material: Pairwise versus multiple global network alignment

Vipin Vijayan, Shawn Gu[‡], Eric Krebs[‡], Lei Meng, and Tijana Milenković
Department of Computer Science and Engineering, ECK Institute for Global Health,
Interdisciplinary Center for Network Science and Application,
University of Notre Dame, IN 46556.
[‡]These authors equally contributed to this work.
E-mail: tmilenko@nd.edu

S1 Methods

S1.1 NA methods that we evaluate

The PNA methods that we evaluate are GHOST, MAGNA++, WAVE, and L-GRAAL. The MNA methods that we evaluate are IsoRankN, BEAMS, multiMAGNA++, and ConvexAlign.

PNA methods. Most NA methods are *two-stage* aligners: in the first stage, they calculate the similarities (based on network topology and, optionally, protein sequence information) between nodes in the compared networks, and in the second stage, they use an alignment strategy to find high scoring alignments with respect to the total similarity over all aligned nodes. GHOST is an example of two-stage PNA methods. GHOST calculates the similarity of "spectral signatures" of nodes between the compared networks in its first stage. Then, GHOST uses an alignment strategy consisting of a seed-and-extend global alignment step followed by a local search procedure that aims to improve, with respect to node similarity, upon the seed-and-extend step. An issue with two-stage methods is that while they find high scoring alignments with respect to total node similarity (a.k.a. node conservation), they do not take into account the amount of conserved edges during the alignment construction process. But the quality of a network alignment is often measured in terms of the amount of conserved edges. To address this issue, MAGNA++ directly optimizes both edge and node conservation *while* the alignment is constructed; its node conservation measure typically uses graphlet-based node similarities (Milenković and Pržulj, 2008). MAGNA is a *search-based* (rather than a two-stage) PNA method. Search-based aligners can directly optimize edge conservation or any other alignment quality measure. WAVE was proposed as a two-stage (rather than search-based) PNA method that optimizes both a graphlet-based node conservation measure as well as (weighted) edge conservation by using a seed-and-extend alignment strategy based on the principle of voting. Similarly, L-GRAAL optimizes a graphlet-based node conservation measure and a (weighted) edge conservation measure, but it uses a seed-and-extend strategy based on integer programming.

MNA methods. IsoRankN is a two-stage MNA method. It calculates node similarities between all pairs of compared networks using a PageRank-based spectral method. IsoRankN then creates a graph of the node similarities and partitions the graph using spectral clustering in order to produce a many-to-many alignment. BEAMS is a two-stage method that optimizes both a (protein sequence-based) node conservation measure and an edge conservation measure. BEAMS uses a maximally weighted clique finding algorithm on a graph of node similarities to produce a one-to-one alignment, where node similarity is based only on protein sequence information, without considering any topological node similarity information. BEAMS then creates a many-to-many alignment from the one-to-one alignment using an iterative greedy algorithm that maximizes both node and edge conservation. ConvexAlign is also a two-stage method. It optimizes an objective function that combines topological node similarity, optional sequence-based node similarity, and edge conservation. That is, it optimizes both node and edge conservation. ConvexAlign optimizes its objective function with an optimization strategy that is formulated as an integer program, which is relaxed into a convex optimization problem. This problem is then solved using the alternating direction method of multipliers (ADMM). This allows ConvexAlign to align multiple networks simultaneously. Like MAGNA++, multiMAGNA++ is a search-based method that directly optimizes both edge and node conservation while the alignment is constructed. Of the MNA methods, IsoRankN and BEAMS produce many-to-many alignments, while ConvexAlign and multiMAGNA++ produce one-to-one alignments.

Aligning using network topology only versus using both topology and protein sequences. In our analysis, for each method, we study the effect on output quality when (i) using only network topology while constructing alignments (T alignments) versus (ii) using both network topology and protein sequence information while constructing alignments (T+S alignments). For T alignments, we set method parameters to ignore any sequence information. All methods except BEAMS can produce T alignments and all methods can produce T+S alignments. For T+S alignments, we set method parameters to include sequence information. Supplementary Table S2 shows the specific parameters that we use. Specifically, the methods combine topological information with sequence information in order to optimize $\theta S_T + (1 - \theta)S_P$, where S_T is the (node or edge) cost function based on *topological information*, S_P is the node cost function based on *protein sequence information*, and θ weighs between topological information and sequence information. When $\theta = 1$, only network topology is used in the alignment process, and when $\theta = 0$, only sequence information is used. We set $\theta = 0.5$ in our study due to the following reasons (except for ConvexAlign, see below). First, Meng et al. (2016), who used the same datasets that we use in our study, showed that as long as some amount of topological information and some amount of protein sequence information are used in the alignment process (i.e., as long as θ does not equal 0 or 1), the quality of the resulting alignments is not drastically affected. They showed this for ten PNA methods, including GHOST, MAGNA++, WAVE, and L-GRAAL, which are the PNA methods that we use in this study. Second, it was shown by the original studies which introduced two of the MNA methods used in this study that varying θ between 0.3 and 0.7 has no large effect on the quality of alignments produced by BEAMS and IsoRank (Alkan and Erten, 2014), and that varying θ between 0.2 and 0.8 has no large effect on the quality of alignments produced by FUSE (Gligorijević et al., 2015). Third, the original MAGNA++ paper, which multiMAGNA++ is based on, showed that varying θ between 0.1 and 0.9 has no large effect on the quality of alignments produced by MAGNA++. So, in the original multiMAGNA++ paper, the θ parameter was set to 0.5. We believe that all of this justifies our choice of using θ of 0.5 for all methods considered in our study (except for ConvexAlign, see below). Also, using the same θ value for all methods (except for ConvexAlign, see below) ensures that any potential differences in results of the different methods are not caused

by using different amounts of network topology versus protein sequence information. While in an ideal scenario we would have wanted to use $\theta = 0.5$ for ConvexAlign's T+S alignments as well (just like we do for all other considered methods), the authors of ConvexAlign pre-set this value in ConvexAlign's implementation to a recommended value of 0.343 (see below), thus weighing topological information by 0.343 and sequence information by 0.657. We respect this recommendation and consequently use $\theta = 0.343$ for ConvexAlign.

Next, we clarify how the given method's parameter values from Supplementary Table S2 match the desired θ value.

Recall that the methods combine topological information with sequence information in order to optimize $\theta S_T + (1 - \theta)S_P$, where S_T is the (node or edge) cost function based on *topological information*, S_P is the node cost function based on *protein sequence information*, and θ weighs between topological information and sequence information.

For T alignments, we set parameters such that only topological information is used (i.e., such that $\theta = 1.0$). Namely, setting $\theta = 1.0$ is equivalent to setting the following parameter value(s) for each of the methods, where E_T, N_T, and N_S are the topological edge conservation function, topological node cost function, and sequence-based node cost function, respectively. (That is, E_T and/or N_T form S_T from the above θ-related formula, and N_S is S_P from the above θ-related formula.)

- For GHOST, which optimizes $\alpha N_T + (1 - \alpha)N_S$, setting $\theta = 1.0$ corresponds to setting $\alpha = 1.0$, i.e., alpha=1.0 in the GHOST implementation.

- For L-GRAAL, which optimizes $(1 - \alpha)E_T + \alpha N_S$ (where E_T is edge conservation weighted by topological node similarity), setting $\theta = 1.0$ corresponds to setting $\alpha = 0.0$, i.e., a=0.0 in the L-GRAAL implementation.

- For MAGNA++, which optimizes $\alpha E_T + (1 - \alpha)(\beta N_T + (1 - \beta)N_S)$, setting $\theta = 1.0$ corresponds to setting $\alpha = 0.5$ and $\beta = 1.0$, i.e., setting a=0.5 and inputting only topological node similarity into the MAGNA++ implementation, respectively. Note that we use a=0.5 to give equal weight to edge conservation and node conservation.

- For WAVE, which optimizes $\alpha N_T + (1 - \alpha)N_S$, setting $\theta = 1.0$ corresponds to setting $\alpha = 1.0$, i.e., inputting only topological node similarity to the WAVE implementation. Note that WAVE also optimizes edge conservation, but it does so implicitly, as a part of its alignment strategy. That is, edge conservation is not an input parameter of WAVE or its implementation.

- For IsoRankN, which optimizes $\alpha N_T + (1 - \alpha)N_S$, setting $\theta = 1.0$ corresponds to setting $\alpha = 1.0$, i.e., alpha=1.0 in the IsoRankN implementation.

- For ConvexAlign, which optimizes $\lambda_2 E_T + (1 - \lambda_2)(\lambda_1 N_T + (1 - \lambda_1)N_S)$, setting $\theta = 1.0$ corresponds to setting $\lambda_1 = 1.0$, i.e., inputting no node similarity into the ConvexAlign implementation. Note that we use λ_2 of 0.02, as recommended and pre-set by the authors of the ConvexAlign paper. ConvexAlign authors have recommended all of its parameter values after testing them using cross-validation. So, we did not need to set any parameter values ourselves.

- For multiMAGNA++, which optimizes $\alpha E_T + (1-\alpha)(\beta N_T + (1-\beta)N_S)$, setting $\theta = 1.0$ corresponds to setting $\alpha = 0.5$ and $\beta = 1.0$, i.e., setting a=0.5 and inputting only topological node similarity into the multiMAGNA++ implementation, respectively. Note that we use a=0.5 to give equal weight to edge conservation and node conservation.

For T+S alignments, we set parameters such that both topological and sequence information is used (i.e., such that $\theta = 0.5$, unless recommended otherwise by the authors of the given method). Namely, setting $\theta = 0.5$ is equivalent to setting the following parameter value(s) for each of the methods.

- For GHOST, which optimizes $\alpha N_T + (1-\alpha)N_S$, setting $\theta = 0.5$ corresponds to setting $\alpha = 0.5$, i.e., alpha=0.5 in the GHOST implementation.

- For L-GRAAL, which optimizes $(1-\alpha)E_T + \alpha N_S$ (where E_T is edge conservation weighted by topological node similarity), setting $\theta = 0.5$ corresponds to setting $\alpha = 0.5$, i.e., a=0.5 in the L-GRAAL implementation.

- For MAGNA++, which optimizes $\alpha E_T + (1-\alpha)(\beta N_T + (1-\beta)N_S)$, setting $\theta = 0.5$ corresponds to setting $\alpha = 0.25$ and $\beta = 0.33$, i.e., setting a=0.25 and inputting the combined node similarity information into the MAGNA++ implementation. With these parameter values, topological and sequence-based cost functions are equally weighted. Namely, the optimization formula for MAGNA++ becomes $0.25E_T + 0.75(0.33N_T + 0.67N_S) = 0.25E_T + 0.25N_T + 0.5N_S = 0.5S_T + 0.5S_P$, i.e., $\theta = 0.5$, as desired.

- For WAVE, which optimizes $\alpha N_T + (1-\alpha)N_S$, setting $\theta = 0.5$ corresponds to setting $\alpha = 0.5$, i.e., inputting both topological and sequence-based node similarities to the WAVE implementation. Note that WAVE also optimizes edge conservation, but it does so implicitly, as a part of its alignment strategy. That is, edge conservation is not an input parameter of WAVE or its implementation.

- For IsoRankN, which optimizes $\alpha N_T + (1-\alpha)N_S$, setting $\theta = 0.5$ corresponds to setting $\alpha = 0.5$, i.e., alpha=0.5 in the IsoRankN implementation.

- For ConvexAlign, which optimizes $\lambda_2 E_T + (1-\lambda_2)(\lambda_1 N_T + (1-\lambda_1)N_S)$, we use $\lambda_1 = 0.33$ and $\lambda_2 = 0.02$, as recommended and pre-set by the authors of the ConvexAlign paper. ConvexAlign authors have recommended all of its parameter values after testing them using cross-validation. So, we did not need to set any parameter values ourselves. With these two parameter values, the optimization formula for ConvexAlign becomes $0.02E_T + 0.98(0.33N_T + 0.67N_S) = 0.02E_T + 0.323N_T + 0.657N_S = 0.343S_T + 0.657S_P$, i.e., $\theta = 0.343$. Clearly, ConvexAlign weighs sequence information higher than the other methods (65.7% of the whole objective function for ConvexAlign, as opposed to 50% of the while objective function for the other methods). Again, this is because the authors of ConvexAlign suggested doing 65.7% for their method, while our justification for 50% for the other methods is discussed above.

- For multiMAGNA++, which optimizes $\alpha E_T + (1-\alpha)(\beta N_T + (1-\beta)N_S)$, setting $\theta = 0.5$ corresponds to setting $\alpha = 0.25$ and $\beta = 0.33$ as recommended by the multiMAGNA++ paper, i.e., setting a=0.25 and inputting the combined node similarity information into the multiMAGNA++ implementation. With these parameter

values, topological and sequence-based cost functions are equally weighted. Namely, the optimization formula for multiMAGNA++ becomes $0.25E_T + 0.75(0.33N_T + 0.67N_S) = 0.25E_T + 0.25N_T + 0.5N_S = 0.5S_T + 0.5S_P$, i.e., $\theta = 0.5$.

S1.2 Alignment quality measures

Here, we describe the alignment quality measures that we use to evaluate the NA methods. To do so, we first need to formally define an alignment. Typical PNA methods produce alignments comprising node pairs and typical MNA methods produce alignments comprising node clusters. We introduce the term *aligned node group* to describe either an aligned node pair or an aligned node cluster. With this, we can represent a pairwise or multiple alignment as a set of aligned node groups. Let $G_1(V_1, E_1)$, ..., $G_k(V_k, E_k)$ be k networks with node and edge sets V_l and E_l, respectively, for $l = 1, 2, \ldots, k$. An *alignment* of the k networks is a set of disjoint node groups, where each group is represented as a tuple $(a_1, \ldots, a_k)$ with the following properties: (i) a_l is the set of nodes in the node group from network G_l, i.e., $a_l \subseteq V_l$, for $l = 1, 2, \ldots, k$, (ii) no two node groups have any common nodes, i.e., given two different groups $(a_1, a_2, \ldots, a_k)$ and $(b_1, b_2, \ldots, b_k)$, $a_l \cap b_l = \varnothing$ for $l = 1, 2, \ldots, k$, and (iii) there must be at least two nodes in each node group, i.e., $|\cup_{l=1,\ldots,k} a_l| \geq 2$. If for each node group in the given alignment there is at most one node from each network, i.e., if for each node group $|a_l| \leq 1$ for $l = 1, \ldots, k$, then the alignment is a *one-to-one* alignment. Otherwise, it is a *many-to-many* alignment.

S1.2.1 Topological quality (TQ) measures

A good NA method should produce aligned node groups that have internal consistency with respect to protein labels. If we know the true node mapping between the networks, then we can let the labels be protein names. When the labels are based on the true node mapping, i.e., on protein names, we consider measures that rely on node labels to be capturing topological alignment quality (TQ). If we do not know the true node mapping, we let the labels be GO terms. In this case, since GO terms capture protein functions, we consider measures that rely on GO terms to be capturing functional alignment quality (FQ). We discuss such measures in Supplementary Section S1.2.2.

Also, a good NA method should find a large amount of common network structure across the compared networks, i.e., produce high edge conservation.

Finally, for a good NA method, conserved edges should form large, dense, connected regions (as opposed to small or isolated conserved regions).

Below, first, we discuss how we measure internal consistency of aligned protein groups in a pairwise alignment. Second, we comment on how we do this in a multiple alignment. Third, we discuss how we measure edge conservation in a pairwise alignment. Fourth, we comment on how we do this in a multiple alignment. Fifth, we discuss how we capture the notion of large, dense, and connected conserved network regions (for both pairwise and multiple alignments).

1. Measuring internal node group consistency of a pairwise alignment via precision, recall, and F-score of node correctness (P-NC, R-NC, and F-NC, respectively). These measures (Meng et al., 2016) are a generalization of node correctness (NC) from one-to-one to many-to-many pairwise alignments. NC for one-to-one pairwise alignments is the fraction of node pairs from the alignment that are present in the true node mapping. As such, NC evaluates the *precision* of the alignment. NC is extended to many-to-many pairwise alignments as follows. For

each aligned node group C_i in the alignment, C_i is converted into a set of all possible $\binom{|C_i|}{2}$ node pairs in the group. The union of all resulting node pairs over all groups C_i forms the set X of all aligned node pairs. Then, given the set Y of all node pairs from the true node mapping, P-NC = $\frac{|X \cap Y|}{|X|}$, R-NC = $\frac{|X \cap Y|}{|Y|}$, and F-NC is the harmonic mean of P-NC and R-NC. These three measures work for both one-to-one and many-to-many pairwise alignments.

2. Measuring internal node group consistency of a multiple alignment via adjusted multiple node correctness (NCV-MNC). Multiple node correctness (MNC) (Vijayan and Milenković, 2016) is a generalization of the NC measure to multiple alignments. MNC uses the notion of normalized entropy (NE), which measures, for a given aligned node group, how likely it is to observe the same or higher level of internal node group consistency by chance, i.e., if the group of the same size was formed by randomly assigning to it proteins from the compared networks. The lower the NE, the more consistent the node group. Then, MNC is one minus the mean of NEs across all node groups. We refer to Vijayan and Milenković (2016) for the formal definition of MNC. Since a good NA method should align (or cover) many of the nodes in the compared networks, as was done by Vijayan and Milenković (2016), we adjust the MNC measure to account for node coverage (NCV), which is the fraction of nodes that are in the alignment out of all nodes in the compared networks. Then, MNC-NCV= $\sqrt{(\text{NCV})(\text{MNC})}$. When either NCV or MNC is low, the geometric mean of the two is penalized. The NCV-MNC measure works for both one-to-one and many-to-many multiple alignments.

3. Measuring edge conservation of a pairwise alignment via adjusted generalized S^3 (NCV-GS^3). Given two compared networks, generalized S^3 (GS^3) (Meng et al., 2016) measures the fraction of conserved edges out of both conserved and non-conserved edges, where an edge is conserved if it maps to an edge in the other network and an edge is not conserved if it maps to a non-adjacent node pair (i.e., a non-edge) in the other network. We refer to Meng et al. (2016) for formal definition of GS^3. As was done by Meng et al. (2016), we penalize alignments with low node coverage by combining NCV with GS^3 into the adjusted GS^3 measure, NCV-GS^3, which equals $\sqrt{(\text{NCV})(\text{GS}^3)}$. The NCV-$GS^3$ measure works for both one-to-one and many-to-many pairwise alignments.

4. Measuring edge conservation of a multiple alignment via adjusted cluster interaction quality (NCV-CIQ). CIQ (Alkan and Erten, 2014) is a weighted sum of edge conservation between all pairs of aligned node groups. We refer to Vijayan and Milenković (2016) for the formal definition of CIQ. As was done by Vijayan and Milenković (2016), we penalize alignments with low node coverage by combining NCV with CIQ into the adjusted CIQ, NCV-CIQ, which equals $\sqrt{(\text{NCV})(\text{CIQ})}$. The NCV-CIQ measure works for both one-to-one and many-to-many multiple alignments.

5. Measuring the size of the largest connected region using largest common connected subgraph (LCCS). The LCCS measure, which was recently extended from PNA (Saraph and Milenković, 2014) to MNA (Vijayan and Milenković, 2016), simultaneously captures the size (i.e., the number of nodes) and the density (i.e., the number of edges) of the largest common connected subgraph formed by the conserved edges, penalizing smaller or sparser subgraphs. We refer to Vijayan and Milenković (2016) for the formal definition of LCCS. The LCCS measure works for both one-to-one and many-to-many alignments, and for both pairwise and multiple alignments.

S1.2.2 Functional quality (FQ) measures

Per Supplementary Section S1.2.1, a good alignment should have internally consistent aligned node groups. Instead of protein names as in Supplementary Section S1.2.1, in this section we use GO terms as protein labels to measure

internal consistency.

Having aligned node groups that are internally consistent with respect to protein labels is important for protein function prediction. In addition to measuring internal node group consistency, we directly measure the accuracy of protein function prediction. That is, we first use a protein function prediction approach (Section II-C3 of the main paper) to predict protein-GO term associations, and then we compare the predicted associations to known protein-GO term associations to see how accurate the predicted associations are.

Below, first, we discuss how we measure internal node group consistency with respect to GO terms. Second, we discuss an alternative popular measure for doing the same. Third, we discuss how we measure the accuracy of protein function prediction, i.e., of predicted protein-GO term associations (note that we describe a strategy that we use to make the predictions in Section II-C3 of the main paper).

1. Measuring internal node group consistency using mean normalized entropy (MNE). MNE (Liao et al., 2009) first uses normalized entropy (NE) to measure GO term-based consistency of an individual aligned node group. The lower the NE, the more consistent the given node group. Then, MNE is the mean of the NEs across all node groups. We refer to Vijayan and Milenković (2016) for the formal definition of MNE. The MNE measure works for both one-to-one and many-to-many alignments, and for both pairwise and multiple alignments.

2. Measuring internal node group consistency using GO correctness (GC). GO correctness, which was recently extended from PNA (Kuchaiev et al., 2010) to MNA (Vijayan and Milenković, 2016), measures the internal consistency of aligned node groups with respect to GO terms as follows. For each node group C_i in the alignment, C_i is converted into a set of all possible $\binom{|C_i|}{2}$ node pairs in the group. The union of all resulting node pairs over all groups C_i forms the set X of all aligned node pairs. A subset of X that consists of all node pairs in which each of the two nodes is annotated with at least one GO term is denoted as Y. Then, GO correctness is the fraction of node pairs in Y in which the two nodes are both annotated with the same GO term. In other words, GO correctness is the fraction of all pairs of aligned nodes in which the aligned nodes share a GO term. The GO correctness measure works for both one-to-one and many-to-many alignments, and for both pairwise and multiple alignments.

3. Precision, recall, and F-score of protein function prediction (P-PF, R-PF, and F-PF, respectively). We describe how we predict protein-GO term associations in Section II-C3 of the main paper. Here, we describe how we evaluate accuracy of such predictions. Given predicted protein-GO term associations, we calculate accuracy of the predictions via precision, recall, and F-score measures. Formally, given the set X of predicted protein-GO term associations, and the set Y of known protein-GO term associations, P-PF = $\frac{|X \cap Y|}{|X|}$, R-PF = $\frac{|X \cap Y|}{|Y|}$, and F-PF is the harmonic mean of precision and recall. These three measures work for both one-to-one and many-to-many alignments, and for both pairwise and multiple alignments.

S1.2.3 Protein function prediction approaches

Approach 3. New protein function prediction for multiple alignments. We follow our discussion from Section II-C3 of the main paper regarding approach 3, our new protein function prediction approach for multiple alignments. Formally, given an alignment of k networks, $G_1(V_1, E_1)$, $G_2(V_2, E_2)$, ..., $G_k(V_k, E_k)$, and given node v in the alignment that has at least one annotated GO term, and given GO term g, we hide the protein's true GO term(s) and find the significance of the alignment with respect to GO term g using the hypergeometric test, as follows. For each node group C_i in the alignment, we convert C_i into a set of node pairs F_i by taking all node pairs in the node group,

after which we concatenate the sets of node pairs into a single set F. Then, let $V_i^* \subset V_i$ be such that each node in V_i^* is annotated with at least one GO term. Let S_1 be the set of all possible pairs of proteins in F such that one protein is in V_i^* and the other is in V_j^*, where $i \neq j$. Let $A_i \subset V_i^*$ be such that each node in A_i is annotated with g. Let S_2 be the set of all possible pairs of proteins between A_i and A_j, where $i \neq j$. Let K be the set of pairs of proteins that are in F and in S_1. Let X be the set of pairs of proteins that are in F and in S_2. Then, we use the hypergeometric test to calculate the probability of observing by chance $|X|$ or more pairs of proteins in F with each node annotated with g is $p = 1 - \sum_{i=0}^{|X|-1} \frac{\binom{|K|}{i}\binom{|S_1|-|K|}{|S_2|-i}}{\binom{|S_1|}{|S_2|}}$. We consider the alignment to be significant with respect to g if the p-value is less than 0.05. We predict v to be associated with g if the alignment is significant with respect to g, resulting in predicted protein-GO term associations. If the alignment is significant with respect to g, we predict v to be associated with g. Repeating this process for all nodes and GO terms results in predicted protein-GO term associations X.

S1.2.4 Statistical significance of alignment quality scores

We continue our discussion from Section II-C4 of the main paper on how to compute the p-value of a quality score of an actual alignment. This is done as follows. We construct a set of 1,000 corresponding random alignments (1,000 is what was practically reasonable given our computational resources), under a null model that conserves the following properties of the actual alignment: the number of node groups, the number of nodes in each group, and the network from which each node in each node group originates from. Then, the p-value of the alignment quality score is the fraction of the 1,000 random alignments with equal or better score than the actual alignment. We consider an alignment quality score to be significant if its p-value is less than 0.001. Note that if a given method fails to produce an alignment of a network pair/set, we set the p-values of all quality scores associated with the method and network pair/set to 1 and hence consider all of the associated quality scores to be non-significant.

S1.3 Evaluation framework

S1.3.1 Multiple evaluation (ME) framework

We continue our discussion from Section II-D2 of the main paper on how we combine the pairwise alignments over every network pair in the given set into a multiple alignment, i.e., how we produce alignments from the ME-P-P and ME-M-P categories. This procedure was inspired by Dohrmann et al. (2015). Given pairwise alignments of k networks $G_1(V_1, E_1), \ldots, G_k(V_k, E_k)$, Dohrmann et al. (2015) produce a multiple alignment of the k networks as follows. First, they select a "scaffold" network G_r among the k networks, namely the network whose sum of "topological similarities" to the remaining $k - 1$ networks is maximized; one of the suggested "topological similarity" measures is Graphlet Degree Distribution (GDD) agreement (Pržulj, 2007). Second, they align G_r to each of the remaining $k - 1$ networks. Third, they take the union of all aligned node groups from the resulting $k - 1$ alignments. Let us denote this union as set A. Since the node groups in set A are not necessarily disjoint, Dohrmann et al. (2015) use set A to create a new set A' of aligned node groups that are disjoint. This is done as follows. Let A' be an empty set. First, randomly pick an aligned node group C that is currently in A (initially, all node groups are in A) and remove it from A. Then, remove from A all node groups that have at least one node in common with C, and merge the node groups into C. Repeat this process until there are no more node groups in A that have at least one node in common with C. Then, add C to A'. Repeat this process until A is empty. This results

in a new multiple alignment A'. We illustrate this procedure in Fig. 3(b,c) of the main paper. In our work, instead of choosing one of the k analyzed networks as a scaffold network using BLAST protein similarity information as Dohrmann et al. (2015) does, because the choice of scaffold network affects the quality of the resulting multiple alignment (which we actually validate in Supplementary Fig. S9), we vary each of the k networks as the scaffold network G_r, and we choose the scaffold based on the quality of the resulting multiple alignments. That is, we rank (as explained below) each of the k multiple alignments, in order to select the best (in terms of the rank) of them. We rank the alignments as follows. For each alignment quality measure, we rank the alignments from the best one to the worst one. (In case of ties, we let the ranks of the tied alignments be the tied alignments' average rank.) Then, we compute the total rank of each alignment by taking the average of the given alignment's ranks over all of the alignment quality measures. Finally, we select the best (in terms of the total rank) of all alignments. Note that here, we consider all measures that can deal with multiple alignments, except NCV-MNC, which we leave out because not all network pairs/sets have the true node mapping (and NCV-MNC requires knowing this mapping), and except MNE, which we leave out so that the number of TQ and FQ measures matches (which is required in order to prevent the ranks to be dominated by topological or functional alignment quality). That is, we consider NCV-CIQ and LCCS TQ measures and GC and F-PF FQ measures.

S1.4 T versus T+S alignments

Here, we continue our discussion from Section III-A of the main paper regarding the similarity (overlap) of the alignments produced the different NA methods, each with its T and T+S versions (Supplementary Figs. S1–S3). Surprisingly, over all considered network datasets, in each of the PE and ME frameworks, the T+S versions of the different methods are overall more similar than the T+S and T versions of the same method are (with the exception of IsoRankN in the PE framework and also GHOST in the ME framework). That is, the T+S versions of the different methods cluster together in Supplementary Fig. S1 and are clearly separated from the T versions. In contrast, the T versions do not cluster together. This shows that using protein sequence information overall yields alignment consistency independent of which NA method is used. Similar holds when we break down this analysis for networks with known versus unknown node mapping (Supplementary Figs. S2–S3), with the exception of networks with unknown node mapping under the ME framework, where the T and T+S versions of the same approach are often clustered together.

S1.5 Method comparison in the ME framework: accuracy versus running time

The running time discussion in Section III-C of the main paper deals with empirical running times of the considered PNA and MNA methods, when the methods are run on our considered network sets, the largest one of which contains six networks. Since the PNA methods must align every pair of networks in a network set in order to produce a multiple alignment, and since this results in a quadratically increasing running time with respect to the number of networks k, we next ask whether there is some (larger than six) value of k at which PNA might become less efficient (i.e., slower) than MNA. To answer this, because of the limited sizes (in terms of k) of our considered network sets, we need to analyze the methods' theoretic running time complexities with respect to k (Supplementary Table S3). All of the PNA methods's running times grow quadratically with k due to the required pairwise alignments, per the above discussion. Of the MNA methods, IsoRankN's running time also grows quadratically, ConvexAlign's

running time grows cubically, BEAMS' running time grows exponentially, and multiMAGNA++'s time grows linearly with k. So, when comparing the PNA and MNA methods, only multiMAGNA++ grows slower (i.e., is expected to be faster) with k than the PNA methods, IsoRankN grows at the same rate as the PNA methods, and ConvexAlign and BEAMS grow faster than the PNA methods. Hence, we do not expect that the MNA methods will have advantage over the PNA methods as k increases, with the exception of multiMAGNA++. However, note that the analysis of the methods' theoretic running times is different than the analysis of their empirical running times, and also, note that their theoretic as well as empirical running times depend not just on k but also on the sizes of the considered networks in terms of the numbers of their nodes and edges, and also potentially on some method-specific parameters. For example, while multiMAGNA++ theoretically grows the slowest with k of all considered PNA and MNA methods, as we can see from its empirical running time analysis (Fig. 5, View III, in the main paper), multiMAGNA++ is significantly slower than BEAMS on our considered network sets with up to six networks. So, it is hard to estimate the exact value of k at which multiMAGNA++ would empirically perform faster than the other methods, as this would also depend on e.g., the size of each network in the considered network set.

SUPPLEMENTARY TABLES

Set	Species	No. of proteins	No. of interactions
Yeast+%LC	Yeast+0%LC	1,004	8,323
	Yeast+5%LC	1,004	8,739
	Yeast+10%LC	1,004	9,155
	Yeast+15%LC	1,004	9,571
	Yeast+20%LC	1,004	9,987
	Yeast+25%LC	1,004	10,403
PHY_1	Fly	7,887	36,285
	Worm	3,006	5,506
	Yeast	6,168	82,368
	Human	16,061	157,650
PHY_2	Yeast	768	13,654
	Human	8,283	19,697
$Y2H_1$	Fly	7,097	23,370
	Worm	2,874	5,199
	Yeast	3,427	11,348
	Human	9,996	39,984
$Y2H_2$	Yeast	744	966
	Human	1,191	1,567

Table S1: Details on the PINs that we use in our study. The true node mapping is known for the Yeast+%LC network set, unlike for the PHY_1, PHY_2, $Y2H_1$, and $Y2H_2$ network sets. Since the largest connected components of the fly and worm networks in PHY_2 and $Y2H_2$ are too small, we do not use those networks in our analysis.

Algorithms	Parameters
PNA methods, T alignments	
GHOST	beta=1e10 alpha=1.0
L-GRAAL	a=0.0 node similarity = graphlet degree vector (GDV) similarity
MAGNA++	m=S3 p=15000 n=10000 a=0.5 node similarity = GDV similarity
WAVE	node similarity = GDV similarity
PNA methods, T+S alignments	
GHOST	beta=1e10 alpha=0.5
L-GRAAL	a=0.5 node similarity = GDV and BLAST protein similarity
MAGNA++	m=S3 p=15000 n=10000 a=0.25 node similarity = GDV and BLAST protein similarity
WAVE	node similarity = GDV and BLAST protein similarity
MNA methods, T alignments	
IsoRankN	K=30 thresh=1e-4 maxveclen=5000000 alpha=1.0
ConvexAlign	lambda_edge=3 numOuterIterations=4 flag_fast=1 mu=150 min_val=0.5 lambda_mul=0.5 node similarity = none
multiMAGNA++	m=CIQ p=15000 n=100000 e=0.5 a=0.5 node similarity = GDV similarity
MNA methods, T+S alignments	
IsoRankN	K=30 thresh=1e-4 maxveclen=5000000 alpha=0.5 node similarity = BLAST protein similarity
ConvexAlign	lambda_edge=3 numOuterIterations=4 flag_fast=1 mu=150 min_val=0.5 lambda_mul=0.5 node similarity = BLAST protein similarity
BEAMS	beta=0.4 alpha=0.5 node similarity = BLAST protein similarity
multiMAGNA++	m=CIQ p=15000 n=100000 e=0.5 a=0.25 node similarity = GDV and BLAST protein similarity

Table S2: Method parameters that we use in our study. We use parameters recommended in the methods' original publications. The parameter "node similarity" indicates the node similarity information that is inputted to the NA method. Note that graphlet degree vector (GDV) similarity uses only network topological information, while BLAST protein similarity uses only protein sequence information. Note that sometimes different methods use the same name (e.g., α) for different parameters, or they use different names (e.g., α versus a) for the same parameter. For T alignments, we set parameters such that only topological information is used (i.e., such that $\theta = 1.0$; see Supplementary Section S1.1). For T+S alignments, we set parameters such that topological and sequence information are equally weighted (i.e., such that $\theta = 0.5$; see Supplementary Section S1.1), as recommended by Meng et al. (2016). The only exception is ConvexAlign, for which we use a lower θ value, as recommended and pre-set in its implementation by its authors. See Supplementary Section S1.1 for details.

Algorithms	Time complexity
PNA methods	
GHOST	$O(n(\frac{m}{n})^4) = O(\frac{m^4}{n^3})$
L-GRAAL	$O(n^3 + n^2\frac{m}{n}^3) = O(n^3 + \frac{m^3}{n})$
MAGNA++	$O(n + m)$
WAVE	$O(n^3)$
MNA methods	
IsoRankN	$O(\binom{k}{2}m^2) = O(k^2m^2)$
ConvexAlign	$O(k^3n^3)$
BEAMS	$O(k^2n^2(\frac{m}{n})^{k+1})$
multiMAGNA++	$O(k(n + m))$

Table S3: Theoretic time complexity of the considered PNA methods when they align two networks and of the considered MNA methods when they align k networks, with respect to network size and the number of networks. Regarding network size, n and m is the number of nodes and edges, respectively, averaged over all networks under consideration.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
WAVE (PE-P-P)	1.70 (1.23)	NA	NA	0.00 (0.00)
multiMAGNA++ (PE-M-P)	1.93 (1.32)	2.28e-01	NA	0.00 (0.00)
MAGNA++ (PE-P-P)	3.21 (1.85)	1.39e-04	2.64e-06	0.00 (0.00)
GHOST (PE-P-P)	4.09 (3.66)	1.06e-04	7.27e-04	0.14 (0.14)
LGRAAL (PE-P-P)	4.21 (2.18)	5.13e-08	2.00e-06	0.05 (0.05)
multiMAGNA++ (PE-M-M)	5.09 (1.56)	2.36e-07	6.92e-08	0.00 (0.00)
BEAMS (PE-M-P)	8.74 (1.99)	5.06e-09	6.81e-09	0.02 (0.00)
ConvexAlign (PE-M-M)	9.07 (1.56)	5.23e-09	5.19e-09	0.00 (0.00)
ConvexAlign (PE-M-P)	9.09 (2.27)	5.07e-09	5.02e-09	0.00 (0.00)
BEAMS (PE-M-M)	9.16 (1.90)	5.08e-09	6.59e-09	0.09 (0.00)
IsoRankN (PE-M-P)	9.56 (1.75)	4.80e-09	4.45e-09	0.23 (0.00)
IsoRankN (PE-M-M)	9.63 (2.21)	4.89e-09	4.65e-09	0.33 (0.00)

Table S4: Overall ranking of the NA methods for the **PE framework** over all evaluation tests (where a test is a combination of an NA method, a network pair, and an alignment quality measure) that use **TQ measures**, for T+S alignments, for networks with both known and unknown node mapping. By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D of the main paper). Namely, there are 12 NA methods in the PE framework (four PNA methods associated with the PE-P-P categories and four MNA methods associated with each of the PE-M-M and PE-M-P categories). The alignment categories are color coded. The "Overall rank" column shows the rank of each method averaged over all evaluation tests, along with the corresponding standard deviation (in brackets). Since there are 12 methods in a given framework, the possible ranks range from 1 to 12. The lower the rank, the better the given method. The "p_1-value" column shows the statistical significance of the difference between the ranking of each method and the 1^{st} best ranked method. The "p_2-value" column shows the statistical significance of the difference between the ranking of each method and the 2^{nd} best ranked method. The "Frac. non. sig. (failed)" column shows the fraction of evaluation tests in which the alignment quality score is not statistically significant, and, in brackets, the fraction of evaluation tests in which the given NA method failed to produce an alignment.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
ConvexAlign (PE-M-P)	4.33 (4.25)	NA	NA	0.07 (0.00)
MAGNA++ (PE-P-P)	4.98 (3.43)	3.32e-01	NA	0.11 (0.00)
ConvexAlign (PE-M-M)	5.42 (4.60)	4.29e-02	2.60e-01	0.21 (0.00)
multiMAGNA++ (PE-M-P)	6.14 (4.17)	7.71e-02	4.62e-05	0.19 (0.00)
LGRAAL (PE-P-P)	7.02 (3.82)	2.40e-03	1.10e-03	0.30 (0.05)
WAVE (PE-P-P)	7.21 (4.21)	7.42e-03	6.25e-07	0.25 (0.00)
IsoRankN (PE-M-M)	7.51 (3.48)	2.34e-05	5.87e-04	0.28 (0.00)
multiMAGNA++ (PE-M-M)	7.54 (4.22)	1.55e-03	3.80e-05	0.35 (0.00)
GHOST (PE-P-P)	7.56 (4.33)	2.56e-03	3.61e-06	0.37 (0.16)
IsoRankN (PE-M-P)	7.82 (4.08)	4.04e-05	6.55e-05	0.39 (0.00)
BEAMS (PE-M-P)	8.33 (4.28)	1.50e-05	1.77e-05	0.39 (0.00)
BEAMS (PE-M-M)	8.79 (4.22)	8.96e-06	2.30e-06	0.47 (0.00)

Table S5: Overall ranking of the NA methods for the **PE framework** over all evaluation tests (where a test is a combination of an NA method, a network pair, and an alignment quality measure) that use **FQ measures**, for T+S alignments, for networks with both known and unknown node mapping. By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D of the main paper). Namely, there are 12 NA methods in the PE framework (four PNA methods associated with the PE-P-P categories and four MNA methods associated with each of the PE-M-M and PE-M-P categories). The table can be interpreted the same way as Supplementary Table S4.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
multiMAGNA++ (PE-M-P)	1.03 (0.18)	NA	NA	0.00 (0.00)
MAGNA++ (PE-P-P)	1.27 (0.91)	1.86e-01	NA	0.00 (0.00)
GHOST (PE-P-P)	1.60 (1.16)	1.31e-02	2.32e-01	0.00 (0.00)
WAVE (PE-P-P)	1.60 (1.45)	2.61e-02	9.42e-02	0.00 (0.00)
LGRAAL (PE-P-P)	3.70 (2.00)	3.77e-05	3.69e-05	0.00 (0.00)
multiMAGNA++ (PE-M-M)	4.97 (1.87)	1.80e-06	2.84e-06	0.00 (0.00)
IsoRankN (PE-M-M)	7.13 (1.85)	9.23e-07	9.36e-07	0.00 (0.00)
IsoRankN (PE-M-P)	7.83 (2.44)	1.47e-06	1.49e-06	0.00 (0.00)
ConvexAlign (PE-M-M)	8.37 (2.97)	1.71e-06	2.03e-06	0.00 (0.00)
BEAMS (PE-M-M)	8.53 (3.50)	5.01e-06	5.24e-06	0.00 (0.00)
BEAMS (PE-M-P)	8.60 (3.56)	5.24e-06	5.36e-06	0.00 (0.00)
ConvexAlign (PE-M-P)	10.60 (1.81)	5.04e-07	5.22e-07	0.00 (0.00)

Table S6: Overall ranking of the NA methods for the **PE framework** over all evaluation tests (where a test is a combination of an NA method, a network pair, and an alignment quality measure) that use network pairs with **known node mapping**, for T+S alignments, for both TQ and FQ measures. By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D of the main paper). Namely, there are 12 NA methods in the PE framework (four PNA methods associated with the PE-P-P categories and four MNA methods associated with each of the PE-M-M and PE-M-P categories). The table can be interpreted the same way as Supplementary Table S4.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
ConvexAlign (PE-M-P)	4.57 (3.66)	NA	NA	0.06 (0.00)
MAGNA++ (PE-P-P)	5.49 (2.64)	4.81e-02	NA	0.09 (0.00)
multiMAGNA++ (PE-M-P)	5.74 (3.84)	3.70e-02	4.12e-01	0.16 (0.00)
WAVE (PE-P-P)	6.23 (4.32)	1.15e-02	1.52e-01	0.20 (0.00)
ConvexAlign (PE-M-M)	6.40 (4.31)	9.33e-08	8.63e-02	0.17 (0.00)
LGRAAL (PE-P-P)	6.71 (3.62)	2.85e-03	1.66e-01	0.27 (0.07)
multiMAGNA++ (PE-M-M)	7.14 (3.89)	3.71e-03	1.36e-04	0.29 (0.00)
GHOST (PE-P-P)	7.99 (3.83)	1.08e-05	1.83e-05	0.39 (0.21)
BEAMS (PE-M-P)	8.47 (3.47)	3.04e-08	2.33e-06	0.33 (0.00)
IsoRankN (PE-M-P)	8.89 (3.69)	1.43e-10	1.02e-07	0.46 (0.00)
IsoRankN (PE-M-M)	8.97 (3.46)	4.28e-11	3.50e-07	0.43 (0.00)
BEAMS (PE-M-M)	9.13 (3.38)	1.58e-09	7.46e-08	0.44 (0.00)

Table S7: Overall ranking of the NA methods for the **PE framework** over all evaluation tests (where a test is a combination of an NA method, a network pair, and an alignment quality measure) that use network pairs with **unknown node mapping**, for T+S alignments, for both TQ and FQ measures. By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D of the main paper). Namely, there are 12 NA methods in the PE framework (four PNA methods associated with the PE-P-P categories and four MNA methods associated with each of the PE-M-M and PE-M-P categories). The table can be interpreted the same way as Supplementary Table S4.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
multiMAGNA++ (ME-M-P)	1.71 (1.25)	NA	NA	0.00 (0.00)
WAVE (ME-P-P)	2.14 (1.46)	3.56e-01	NA	0.00 (0.00)
MAGNA++ (ME-P-P)	3.00 (2.31)	1.01e-01	2.05e-01	0.00 (0.00)
GHOST (ME-P-P)	3.86 (4.02)	9.87e-02	1.40e-01	0.14 (0.00)
multiMAGNA++ (ME-M-M)	4.00 (2.00)	4.46e-02	7.47e-02	0.00 (0.00)
LGRAAL (ME-P-P)	5.00 (3.70)	3.66e-02	2.90e-02	0.14 (0.00)
IsoRankN (ME-M-M)	7.57 (1.13)	1.08e-02	1.11e-02	0.00 (0.00)
ConvexAlign (ME-M-M)	8.71 (2.36)	1.08e-02	1.07e-02	0.00 (0.00)
BEAMS (ME-M-M)	9.14 (2.54)	1.10e-02	1.10e-02	0.29 (0.00)
IsoRankN (ME-M-P)	10.43 (1.99)	1.09e-02	1.10e-02	0.57 (0.00)
BEAMS (ME-M-P)	10.71 (1.89)	1.07e-02	1.09e-02	0.57 (0.00)
ConvexAlign (ME-M-P)	11.43 (0.98)	9.95e-03	1.05e-02	0.43 (0.00)

Table S8: Overall ranking of the NA methods for the **ME framework** over all evaluation tests (where a test is a combination of an NA method, a network set, and an alignment quality measure) that use **TQ measures**, for T+S alignments, for networks with both known and unknown node mapping. By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D of the main paper). Namely, there are 12 NA methods in the ME framework (four PNA methods associated with the ME-P-P categories and four MNA methods associated with each of the ME-M-M and ME-M-P categories). The table can be interpreted the same way as Supplementary Table S4.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
MAGNA++ (ME-P-P)	4.22 (2.82)	NA	NA	0.00 (0.00)
ConvexAlign (ME-M-M)	5.11 (3.82)	3.83e-01	NA	0.00 (0.00)
ConvexAlign (ME-M-P)	5.44 (5.27)	3.83e-01	6.12e-01	0.11 (0.00)
LGRAAL (ME-P-P)	5.78 (4.18)	3.67e-01	3.63e-01	0.22 (0.00)
GHOST (ME-P-P)	5.89 (4.59)	1.75e-01	4.06e-01	0.11 (0.00)
multiMAGNA++ (ME-M-P)	5.89 (3.98)	7.13e-02	4.06e-01	0.11 (0.00)
IsoRankN (ME-M-M)	6.00 (4.00)	2.20e-01	2.19e-01	0.22 (0.00)
WAVE (ME-P-P)	7.00 (4.47)	1.07e-02	2.36e-01	0.11 (0.00)
multiMAGNA++ (ME-M-M)	7.33 (4.18)	2.36e-02	2.38e-01	0.11 (0.00)
BEAMS (ME-M-M)	7.56 (4.67)	5.32e-02	9.04e-02	0.33 (0.00)
IsoRankN (ME-M-P)	8.78 (3.90)	2.09e-02	5.98e-02	0.44 (0.00)
BEAMS (ME-M-P)	9.00 (4.39)	2.10e-02	4.13e-02	0.56 (0.00)

Table S9: Overall ranking of the NA methods for the **ME framework** over all evaluation tests (where a test is a combination of an NA method, a network set, and an alignment quality measure) that use **FQ measures**, for T+S alignments, for networks with both known and unknown node mapping. By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D of the main paper). Namely, there are 12 NA methods in the ME framework (four PNA methods associated with the ME-P-P categories and four MNA methods associated with each of the ME-M-M and ME-M-P categories). The table can be interpreted the same way as Supplementary Table S4.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
GHOST (ME-P-P)	1.17 (0.41)	NA	NA	0.00 (0.00)
multiMAGNA++ (ME-M-P)	1.33 (0.82)	5.00e-01	NA	0.00 (0.00)
MAGNA++ (ME-P-P)	1.50 (1.22)	5.00e-01	5.00e-01	0.00 (0.00)
WAVE (ME-P-P)	2.17 (1.83)	1.73e-01	1.86e-01	0.00 (0.00)
LGRAAL (ME-P-P)	3.17 (2.40)	7.45e-02	8.68e-02	0.00 (0.00)
multiMAGNA++ (ME-M-M)	4.17 (2.48)	4.96e-02	4.96e-02	0.00 (0.00)
IsoRankN (ME-M-M)	6.33 (2.66)	2.39e-02	2.72e-02	0.00 (0.00)
IsoRankN (ME-M-P)	7.33 (3.20)	2.67e-02	2.39e-02	0.00 (0.00)
BEAMS (ME-M-M)	8.17 (3.60)	2.67e-02	2.39e-02	0.00 (0.00)
BEAMS (ME-M-P)	8.50 (4.04)	2.84e-02	2.72e-02	0.17 (0.00)
ConvexAlign (ME-M-M)	10.33 (1.03)	1.55e-02	1.55e-02	0.00 (0.00)
ConvexAlign (ME-M-P)	11.17 (2.04)	1.31e-02	1.31e-02	0.17 (0.00)

Table S10: Overall ranking of the NA methods for the **ME framework** over all evaluation tests (where a test is a combination of an NA method, a network set, and an alignment quality measure) that use network pairs with **known node mapping**, for T+S alignments, for both TQ and FQ measures. By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D). Namely, there are 12 NA methods in the ME framework (four PNA methods associated with the ME-P-P categories and four MNA methods associated with each of the ME-M-M and ME-M-P categories). The table can be interpreted the same way as Supplementary Table S4.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
ConvexAlign (ME-M-M)	4.50 (2.76)	NA	NA	0.00 (0.00)
MAGNA++ (ME-P-P)	5.00 (2.31)	3.60e-01	NA	0.00 (0.00)
multiMAGNA++ (ME-M-P)	5.70 (3.80)	2.21e-01	4.39e-01	0.10 (0.00)
ConvexAlign (ME-M-P)	6.20 (5.33)	1.42e-01	3.04e-01	0.30 (0.00)
WAVE (ME-P-P)	6.50 (4.45)	1.92e-01	2.06e-01	0.10 (0.00)
LGRAAL (ME-P-P)	6.80 (4.02)	1.06e-01	3.41e-01	0.30 (0.00)
IsoRankN (ME-M-M)	6.90 (3.48)	3.96e-03	1.92e-01	0.20 (0.00)
multiMAGNA++ (ME-M-M)	6.90 (4.07)	1.10e-01	1.00e-01	0.10 (0.00)
GHOST (ME-P-P)	7.30 (3.95)	6.28e-02	1.20e-01	0.20 (0.00)
BEAMS (ME-M-M)	8.30 (4.19)	2.05e-02	6.30e-02	0.50 (0.00)
BEAMS (ME-M-P)	10.50 (3.17)	2.86e-03	7.06e-03	0.80 (0.00)
IsoRankN (ME-M-P)	10.80 (2.57)	2.82e-03	7.06e-03	0.80 (0.00)

Table S11: Overall ranking of the NA methods for the **ME framework** over all evaluation tests (where a test is a combination of an NA method, a network set, and an alignment quality measure) that use network pairs with **unknown node mapping**, for T+S alignments, for both TQ and FQ measures. By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D of the main paper). Namely, there are 12 NA methods in the ME framework (four PNA methods associated with the ME-P-P categories and four MNA methods associated with each of the ME-M-M and ME-M-P categories). The table can be interpreted the same way as Supplementary Table S4.

NA method	Overall rank
multiMAGNA++ (ME-M-M)	2.25 (1.50)
MAGNA++ (ME-P-P)	3.25 (3.20)
ConvexAlign (ME-M-P)	4.25 (1.71)
GHOST (ME-P-P)	4.25 (2.36)
LGRAAL (ME-P-P)	4.25 (1.50)
WAVE (ME-P-P)	5.00 (2.45)
multiMAGNA++ (ME-M-P)	6.25 (1.50)
IsoRankN (ME-M-M)	7.75 (3.30)
ConvexAlign (ME-M-M)	8.25 (0.96)
IsoRankN (ME-M-P)	9.50 (0.58)

Table S12: Overall ranking of the NA methods for the **ME framework** over all evaluation tests (where a test is a combination of an NA method and a network set) that use the **mean normalized entropy measure**, for T alignments. By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D of the main paper). Namely, there are 12 NA methods in the ME framework (four PNA methods associated with the ME-P-P categories and four MNA methods associated with each of the ME-M-M and ME-M-P categories). The alignment categories are color coded. The "Overall rank" column shows the rank of each method averaged over all evaluation tests, along with the corresponding standard deviation (in brackets). Since there are 12 methods in a given framework, the possible ranks range from 1 to 12. The lower the rank, the better the given method.

NA method	Overall rank
LGRAAL (ME-P-P)	3.5 (1.00)
IsoRankN (ME-M-M)	4.25 (4.72)
multiMAGNA++ (ME-M-P)	5.25 (4.35)
MAGNA++ (ME-P-P)	5.50 (4.43)
ConvexAlign (ME-M-M)	6.75 (2.22)
multiMAGNA++ (ME-M-M)	7.00 (4.05)
WAVE (ME-P-P)	7.00 (2.94)
BEAMS (ME-M-P)	7.25 (4.11)
IsoRankN (ME-M-P)	7.5 (4.20)
GHOST (ME-P-P)	7.5 (4.79)
BEAMS (ME-M-M)	8.25 (1.50)
ConvexAlign (ME-M-P)	8.25 (2.06)

Table S13: Overall ranking of the NA methods for the **ME framework** over all evaluation tests (where a test is a combination of an NA method and a network set) that use the **mean normalized entropy measure**, for T+S alignments. By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D of the main paper). Namely, there are 12 NA methods in the ME framework (four PNA methods associated with the ME-P-P categories and four MNA methods associated with each of the ME-M-M and ME-M-P categories). The table can be interpreted the same way as Supplementary Table S12.

`https://nd.edu/~cone/PNA_MNA/table_pef.csv`

Table S14: Detailed alignment quality scores for the **PE framework**.

`https://nd.edu/~cone/PNA_MNA/table_mef.csv`

Table S15: Detailed alignment quality scores for the **ME framework**.

Method rankings including GEDEVO-M

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
multiMAGNA++ (ME-M-P)	1.71 (1.25)	NA	NA	0.00 (0.00)
WAVE (ME-P-P)	2.29 (1.60)	2.85e-01	NA	0.00 (0.00)
MAGNA++ (ME-P-P)	3.29 (2.75)	9.87e-02	2.05e-01	0.00 (0.00)
GHOST (ME-P-P)	4.00 (4.08)	1.01e-01	1.40e-01	0.14 (0.00)
multiMAGNA++ (ME-M-M)	4.14 (1.86)	3.67e-02	7.47e-02	0.00 (0.00)
LGRAAL (ME-P-P)	5.14 (3.63)	2.92e-02	2.90e-02	0.14 (0.00)
IsoRankN (ME-M-M)	7.86 (1.35)	1.07e-02	1.12e-02	0.00 (0.00)
GEDEVO-M (ME-M-M)	8.80 (4.66)	2.95e-02	5.28e-02	0.00 (0.00)
ConvexAlign (ME-M-M)	9.14 (2.41)	1.10e-02	1.11e-02	0.00 (0.00)
BEAMS (ME-M-M)	9.43 (2.64)	1.09e-02	1.10e-02	0.29 (0.00)
IsoRankN (ME-M-P)	10.71 (2.29)	1.10e-02	1.11e-02	0.57 (0.00)
BEAMS (ME-M-P)	11.00 (2.16)	1.01e-02	1.10e-02	0.57 (0.00)
ConvexAlign (ME-M-P)	12.00 (1.15)	1.07e-02	1.08e-02	0.43 (0.00)

Table S16: Overall ranking of the NA methods for the **ME framework** over all evaluation tests (where a test is a combination of an NA method, a network set, and an alignment quality measure) that use **TQ measures**, for T+S alignments, for networks with both known and unknown node mapping. The table mimics the analyses from Supplementary Table S8 with the inclusion of an additional method, GEDEVO-M associated with the ME-M-M category.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
MAGNA++ (ME-P-P)	4.22 (2.82)	NA	NA	0.00 (0.00)
ConvexAlign (ME-M-M)	5.11 (3.82)	3.83e-01	NA	0.00 (0.00)
ConvexAlign (ME-M-P)	5.56 (5.43)	3.83e-01	6.12e-01	0.11 (0.00)
LGRAAL (ME-P-P)	5.89 (4.37)	3.67e-01	3.37e-01	0.22 (0.00)
GHOST (ME-P-P)	6.00 (4.77)	1.75e-01	3.83e-01	0.11 (0.00)
multiMAGNA++ (ME-M-P)	6.00 (4.18)	7.13e-02	3.83e-01	0.11 (0.00)
IsoRankN (ME-M-M)	6.11 (4.20)	2.20e-01	2.02e-01	0.22 (0.00)
WAVE (ME-P-P)	7.11 (4.62)	1.07e-02	2.20e-01	0.11 (0.00)
multiMAGNA++ (ME-M-M)	7.44 (4.33)	2.36e-02	2.20e-01	0.11 (0.00)
BEAMS (ME-M-M)	7.67 (4.80)	5.32e-02	8.02e-02	0.33 (0.00)
IsoRankN (ME-M-P)	9.00 (4.12)	2.07e-02	4.80e-02	0.44 (0.00)
BEAMS (ME-M-P)	9.33 (4.66)	2.10e-02	3.28e-02	0.56 (0.00)
GEDEVO-M (ME-M-M)	12.50 (0.84)	1.68e-02	1.78e-02	0.33 (0.00)

Table S17: Overall ranking of the NA methods for the **ME framework** over all evaluation tests (where a test is a combination of an NA method, a network set, and an alignment quality measure) that use **FQ measures**, for T+S alignments, for networks with both known and unknown node mapping. The table mimics the analyses from Supplementary Table S9 with the inclusion of an additional method, GEDEVO-M associated with the ME-M-M category.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
GHOST (ME-P-P)	1.17 (0.41)	NA	NA	0.00 (0.00)
multiMAGNA++ (ME-M-P)	1.33 (0.82)	5.00e-01	NA	0.00 (0.00)
MAGNA++ (ME-P-P)	1.50 (1.22)	5.00e-01	5.00e-01	0.00 (0.00)
WAVE (ME-P-P)	2.17 (1.83)	1.73e-01	1.86e-01	0.00 (0.00)
LGRAAL (ME-P-P)	3.17 (2.40)	7.45e-02	8.68e-02	0.00 (0.00)
multiMAGNA++ (ME-M-M)	4.17 (2.48)	4.96e-02	4.96e-02	0.00 (0.00)
IsoRankN (ME-M-M)	6.33 (2.66)	2.39e-02	2.72e-02	0.00 (0.00)
IsoRankN (ME-M-P)	7.33 (3.20)	2.67e-02	2.39e-02	0.00 (0.00)
BEAMS (ME-M-M)	8.17 (3.60)	2.67e-02	2.39e-02	0.00 (0.00)
BEAMS (ME-M-P)	8.67 (4.23)	2.90e-02	2.84e-02	0.17 (0.00)
ConvexAlign (ME-M-M)	10.50 (1.22)	1.70e-02	1.70e-02	0.00 (0.00)
ConvexAlign (ME-M-P)	11.50 (2.26)	1.68e-02	1.68e-02	0.17 (0.00)
GEDEVO-M (ME-M-M)	12.33 (0.82)	1.68e-02	1.70e-02	0.00 (0.00)

Table S18: Overall ranking of the NA methods for the **ME framework** over all evaluation tests (where a test is a combination of an NA method, a network set, and an alignment quality measure) that use network pairs with **known node mapping**, for T+S alignments, for both TQ and FQ measures. The table mimics the analyses from Supplementary Table S10 with the inclusion of an additional method, GEDEVO-M associated with the ME-M-M category.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
ConvexAlign (ME-M-M)	4.70 (3.02)	NA	NA	0.00 (0.00)
MAGNA++ (ME-P-P)	5.20 (2.44)	3.60e-01	NA	0.00 (0.00)
multiMAGNA++ (ME-M-P)	5.80 (3.99)	2.86e-01	4.80e-01	0.10 (0.00)
ConvexAlign (ME-M-P)	6.50 (5.66)	1.42e-01	3.04e-01	0.30 (0.00)
WAVE (ME-P-P)	6.70 (4.52)	2.07e-01	2.06e-01	0.10 (0.00)
LGRAAL (ME-P-P)	7.00 (4.08)	1.17e-01	3.41e-01	0.30 (0.00)
multiMAGNA++ (ME-M-M)	7.10 (4.09)	1.10e-01	1.10e-01	0.10 (0.00)
IsoRankN (ME-M-M)	7.20 (3.74)	3.96e-03	1.92e-01	0.20 (0.00)
GHOST (ME-P-P)	7.50 (4.03)	7.61e-02	1.30e-01	0.20 (0.00)
BEAMS (ME-M-M)	8.60 (4.38)	2.06e-02	6.30e-02	0.50 (0.00)
GEDEVO-M (ME-M-M)	9.00 (4.85)	1.39e-01	2.05e-01	0.40 (0.00)
BEAMS (ME-M-P)	10.90 (3.41)	2.91e-03	7.12e-03	0.80 (0.00)
IsoRankN (ME-M-P)	11.20 (2.82)	2.86e-03	7.12e-03	0.80 (0.00)

Table S19: Overall ranking of the NA methods for the **ME framework** over all evaluation tests (where a test is a combination of an NA method, a network set, and an alignment quality measure) that use networks pairs with **unknown node mapping**, for T+S alignments, for both TQ and FQ measures. The table mimics the analyses from Supplementary Table S11 with the inclusion of an additional method, GEDEVO-M associated with the ME-M-M category.

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
MAGNA++ (ME-P-P)	3.81 (2.74)	NA	NA	0.00 (0.00)
multiMAGNA++ (ME-M-P)	4.12 (3.84)	5.18e-01	NA	0.06 (0.00)
WAVE (ME-P-P)	5.00 (4.31)	1.26e-01	3.91e-02	0.06 (0.00)
GHOST (ME-P-P)	5.12 (4.46)	1.98e-01	1.52e-01	0.12 (0.00)
LGRAAL (ME-P-P)	5.56 (3.95)	1.24e-01	8.38e-02	0.19 (0.00)
multiMAGNA++ (ME-M-M)	6.00 (3.78)	1.87e-02	5.39e-03	0.06 (0.00)
ConvexAlign (ME-M-M)	6.88 (3.79)	3.91e-02	8.88e-02	0.00 (0.00)
IsoRankN (ME-M-M)	6.88 (3.30)	1.32e-02	1.97e-02	0.12 (0.00)
ConvexAlign (ME-M-P)	8.38 (5.21)	1.68e-02	1.42e-02	0.25 (0.00)
BEAMS (ME-M-M)	8.44 (3.98)	3.42e-03	5.35e-03	0.31 (0.00)
IsoRankN (ME-M-P)	9.75 (3.45)	6.25e-04	1.01e-03	0.50 (0.00)
BEAMS (ME-M-P)	10.06 (3.77)	6.50e-04	1.21e-03	0.56 (0.00)
GEDEVO-M (ME-M-M)	10.82 (3.57)	5.36e-03	2.90e-03	0.18 (0.00)

Table S20: Overall ranking of the NA methods for the **ME framework** over all evaluation tests (where a test is a combination of an NA method, a network set, and an alignment quality measure) for T+S alignments, for both TQ and FQ measures, for networks with both known and unknown node mapping. The table mimics the analyses from View I of Figure 5 from the main paper, with the inclusion of an additional method, GEDEVO-M associated with the ME-M-M category.

SUPPLEMENTARY FIGURES

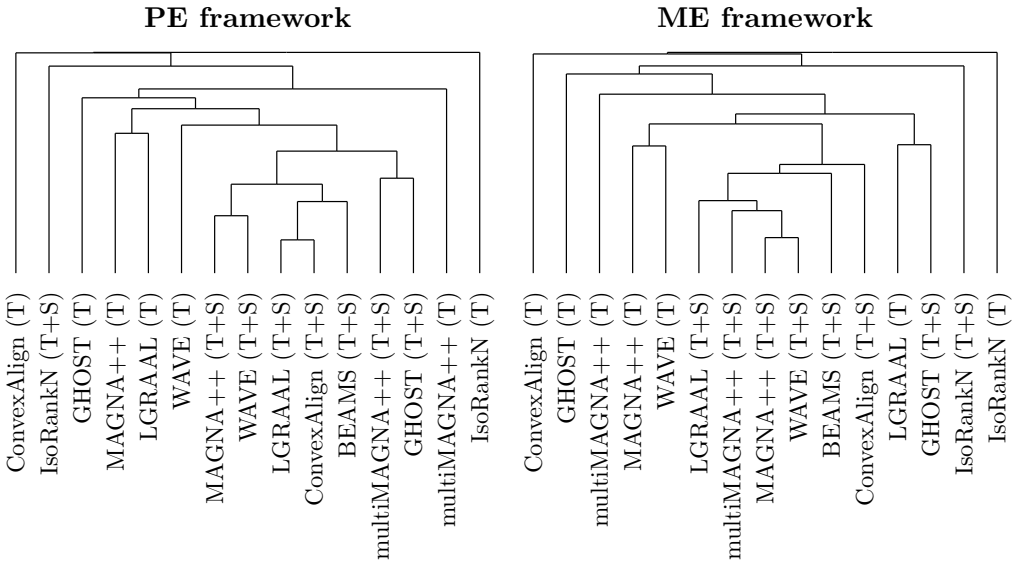


Figure S1: Clustering of NA methods, each with its T and T+S versions, using each of the **PE** and **ME** frameworks. Clustering is based on pairwise method similarities, which we compute as follows. The similarity between two NA methods is the mean of the Adjusted Rand Index (ARI; explained below) of each pair of corresponding alignments produced by the two NA methods, over all network pairs/sets. Each alignment of a network pair/set is a set of node groups, i.e., a partition of the nodes in all of the networks in the network pair/set, and we measure similarity between two alignments by comparing their partitions using ARI. ARI (Vinh et al., 2007) is a widely used measure to calculate the similarity between two partitions. Given the similarities between all pairs of the NA methods, we cluster using complete linkage hierachical clustering (Everitt et al., 2001) and visualize the clustering using a dendrogram. The results shown in this figure rely on all alignments over all network sets (Yeast+%LC, PHY_1, PHY_2, $Y2H_1$, and $Y2H_2$). Equivalent results broken down into results for networks with known node mapping and results for networks with unknown node mapping are shown in Supplementary Figs. S2 and S3, respectively.

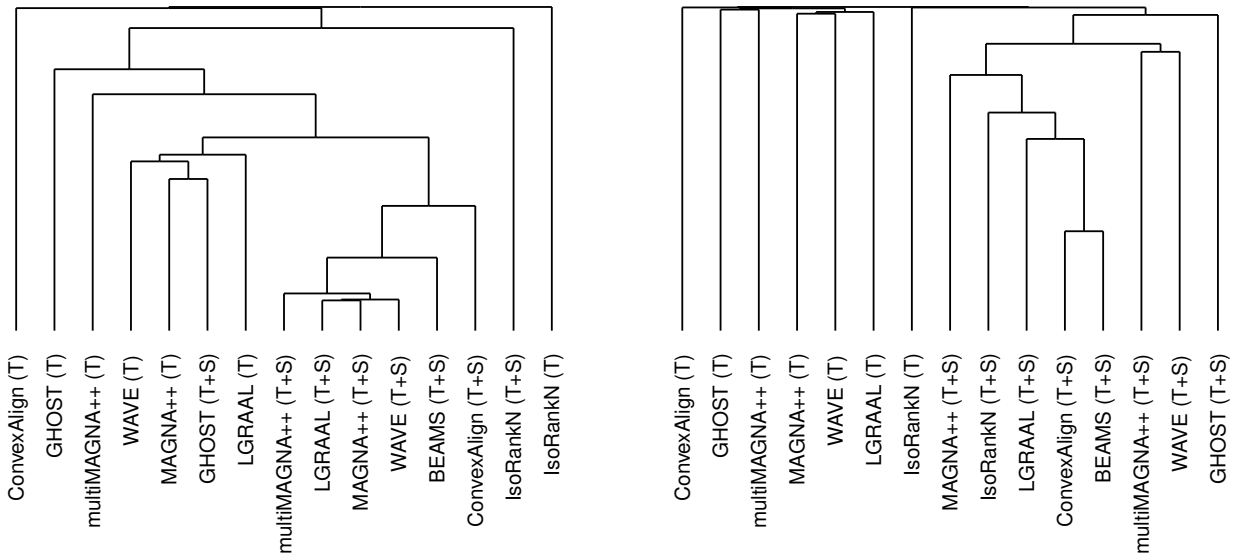


Figure S2: Clustering of NA methods, each with its T and T+S versions, using all network sets with (a) **known node mapping** and (b) **unknown node mapping** in the **PE framework**. The figure can be interpreted the same way as Supplementary Fig. S1.

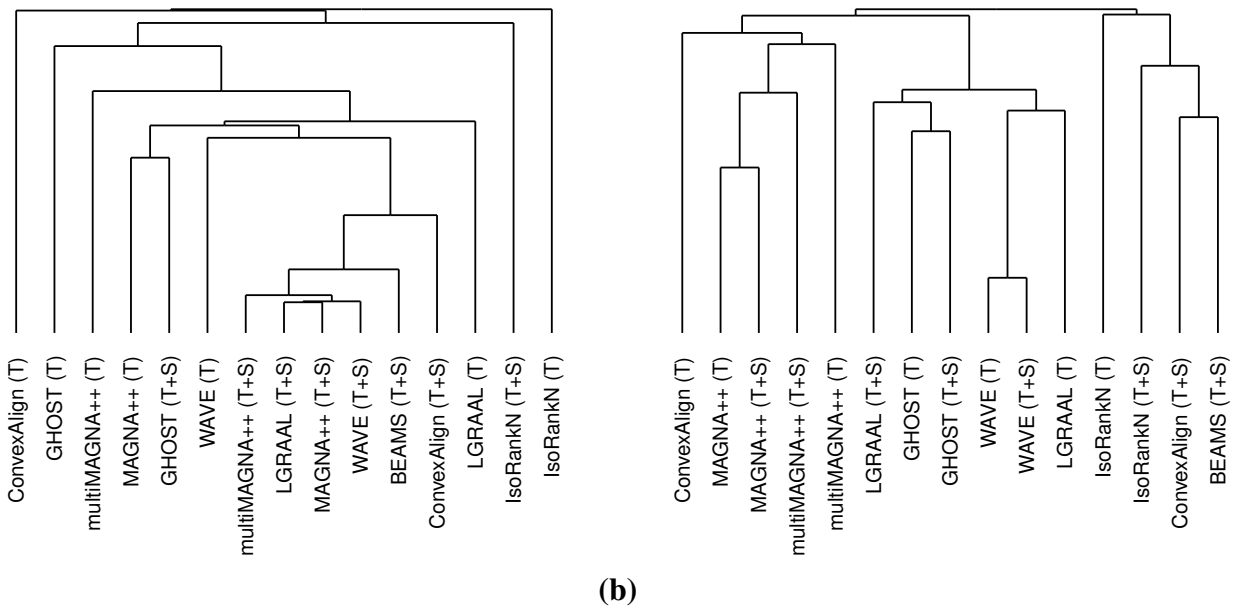


Figure S3: Clustering of NA methods, each with its T and T+S versions, using all network sets with (a) **known node mapping** and (b) **unknown node mapping** in the **ME framework**. The figure can be interpreted the same way as Supplementary Fig. S1.

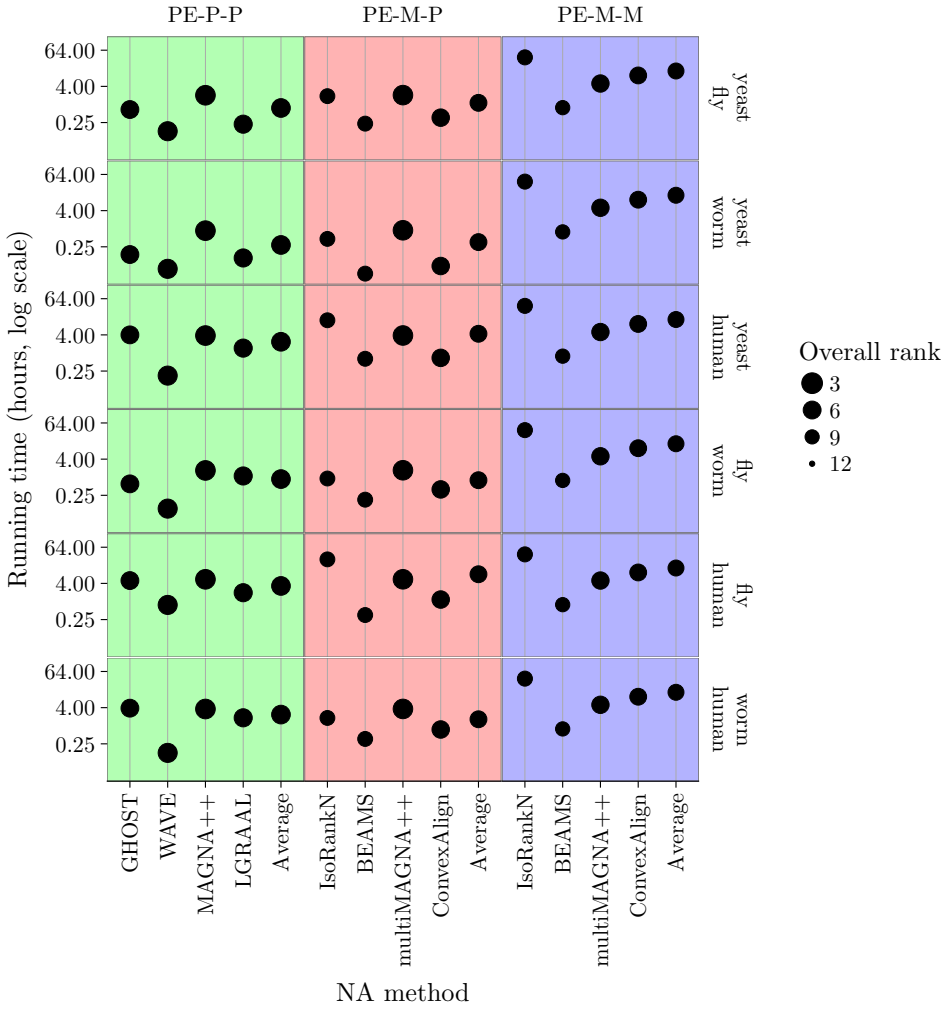


Figure S4: Overall ranking of an NA method versus its running time for the **PE framework** over all evaluation tests (where a test is a combination of an NA method, a network pair, and an alignment quality measure). By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D). Namely, there are 12 NA methods in the PE framework (four PNA methods associated with the PE-P-P categories and four MNA methods associated with each of the PE-M-M and PE-M-P categories). The alignment categories are color coded. The running time results are when aligning all network pairs in the $Y2H_1$ network set, where each method is restricted to use a **single core**. The size of each point visualizes the overall ranking of the corresponding method over all evaluation tests over all network pairs/sets, corresponding to the "Overall rank" column in View I of Fig. 5 in the main paper; the larger the point size, the better the method. In order to allow for easier comparison between the different alignment categories, "Average" shows the average running times and average rankings of the methods in each alignment category.

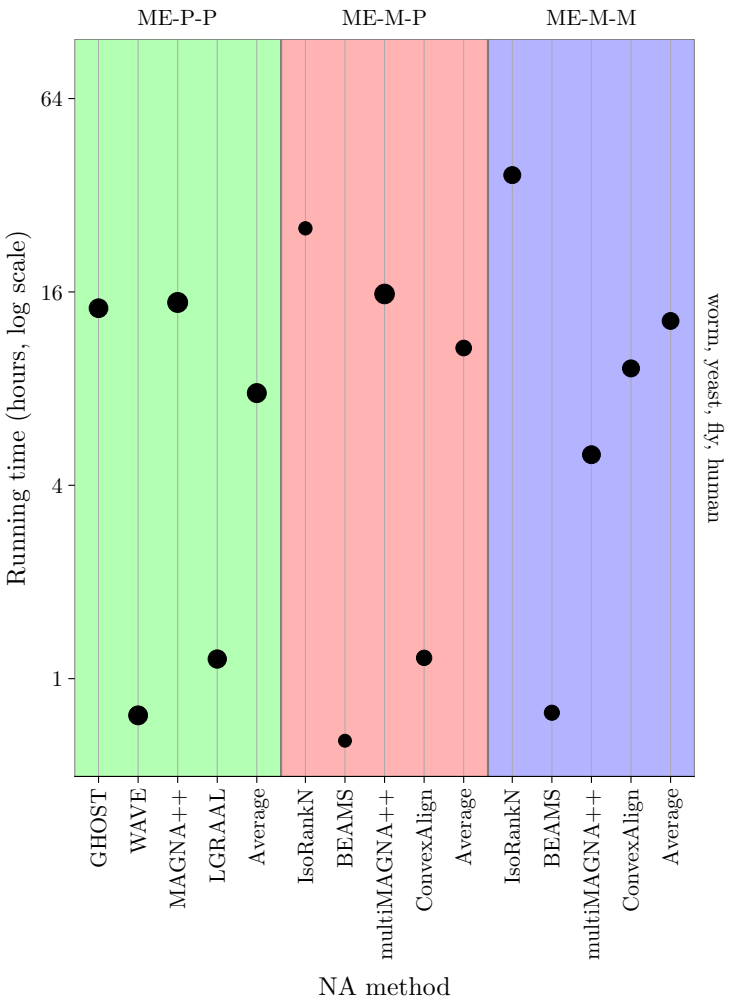
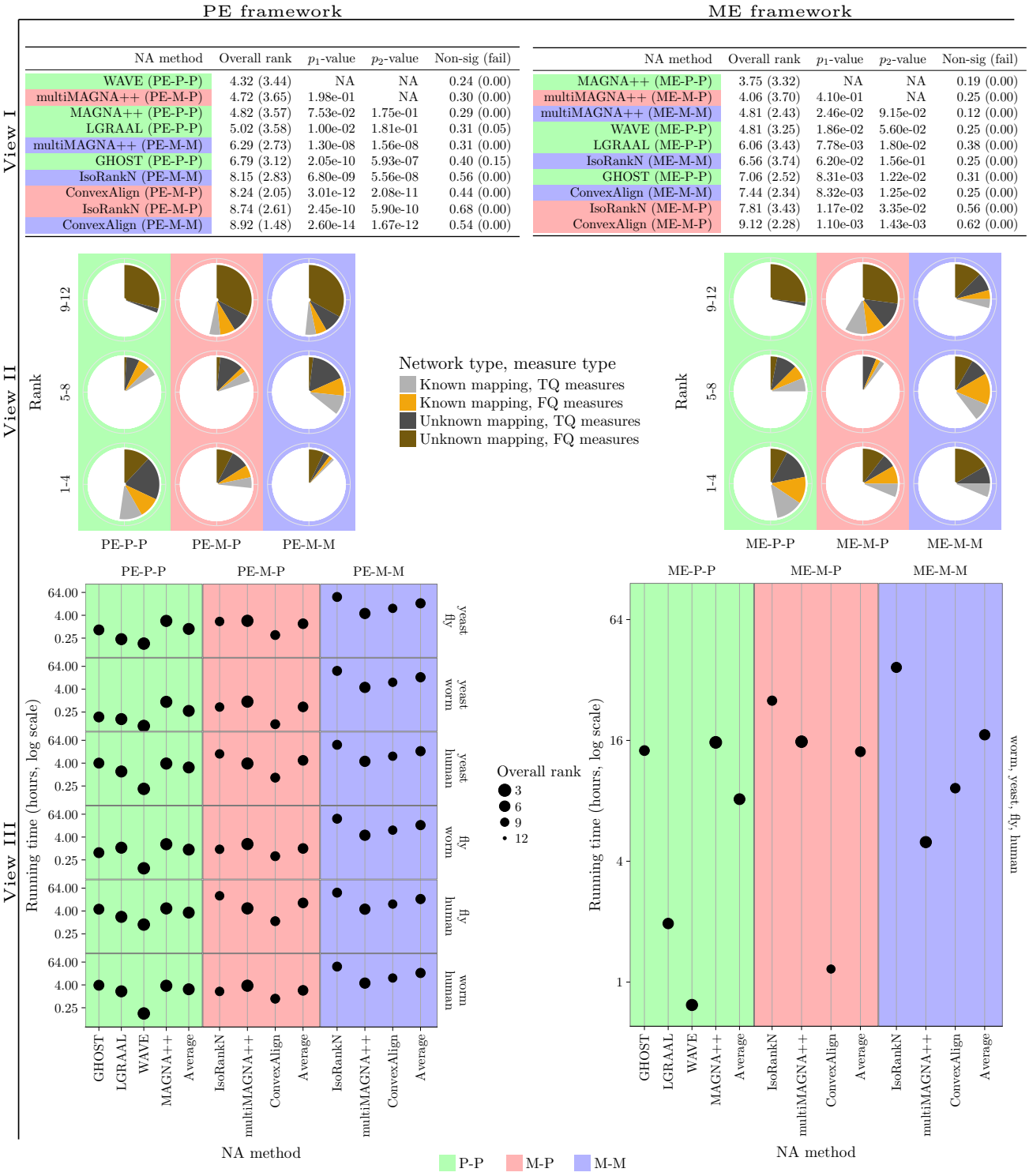


Figure S5: Overall ranking of an NA method versus its running time for the **ME framework** over all evaluation tests (where a test is a combination of an NA method, a network pair, and an alignment quality measure). By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D of the main paper). Namely, there are 12 NA methods in the ME framework (four PNA methods associated with the ME-P-P categories and four MNA methods associated with each of the ME-M-M and ME-M-P categories). The alignment categories are color coded. The running time results are when aligning the $Y2H_1$ network set, where each method is restricted to use a **single core**. The size of each point visualizes the overall ranking of the corresponding method over all evaluation tests over all network pairs/sets, corresponding to the "Overall rank" column in View I of Fig. 5 in the main paper; the larger the point size, the better the method. In order to allow for easier comparison between the different alignment categories, "Average" shows the average running times and average rankings of the methods in each alignment category.



NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
WAVE (PE-P-P)	4.32 (3.44)	NA	NA	0.24 (0.00)
multiMAGNA++ (PE-M-P)	4.72 (3.65)	1.98e-01	NA	0.30 (0.00)
MAGNA++ (PE-P-P)	4.82 (3.57)	7.53e-02	1.75e-01	0.29 (0.00)
LGRAAL (PE-P-P)	5.02 (3.58)	1.00e-02	1.81e-01	0.31 (0.05)
multiMAGNA++ (PE-M-M)	6.29 (2.73)	1.30e-08	1.56e-08	0.31 (0.00)
GHOST (PE-P-P)	6.79 (3.12)	2.05e-10	5.93e-07	0.40 (0.15)
IsoRankN (PE-M-M)	8.15 (2.83)	6.80e-09	5.56e-08	0.56 (0.00)
ConvexAlign (PE-M-P)	8.24 (2.05)	3.01e-12	2.08e-11	0.44 (0.00)
IsoRankN (PE-M-P)	8.74 (2.61)	2.45e-10	5.90e-10	0.68 (0.00)
ConvexAlign (PE-M-M)	8.92 (1.48)	2.60e-14	1.67e-12	0.54 (0.00)

NA method	Overall rank	p_1-value	p_2-value	Non-sig (fail)
MAGNA++ (ME-P-P)	3.75 (3.32)	NA	NA	0.19 (0.00)
multiMAGNA++ (ME-M-P)	4.06 (3.70)	4.10e-01	NA	0.25 (0.00)
multiMAGNA++ (ME-M-M)	4.81 (2.43)	2.46e-02	9.15e-02	0.12 (0.00)
WAVE (ME-P-P)	4.81 (3.25)	1.86e-02	5.60e-02	0.25 (0.00)
LGRAAL (ME-P-P)	6.06 (3.43)	7.78e-03	1.80e-02	0.38 (0.00)
IsoRankN (ME-M-M)	6.56 (3.74)	6.20e-02	1.56e-01	0.25 (0.00)
GHOST (ME-P-P)	7.06 (2.52)	8.31e-03	1.22e-02	0.31 (0.00)
ConvexAlign (ME-M-M)	7.44 (2.34)	8.32e-03	1.25e-02	0.25 (0.00)
IsoRankN (ME-M-P)	7.81 (3.43)	1.17e-02	3.35e-02	0.56 (0.00)
ConvexAlign (ME-M-P)	9.12 (2.28)	1.10e-03	1.43e-03	0.62 (0.00)

Figure S6: Method comparison results for each of the **PE** and **ME** frameworks over all evaluation tests (where a test is a combination of an NA method, a network pair/set, and an alignment quality measure), for T alignments. By NA method, here, we mean the combination of a PNA or MNA method and the alignment category (Section II-D). Namely, there are 12 NA methods in the PE framework (four PNA methods associated with the PE-P-P categories and four MNA methods associated with each of the PE-M-M and PE-M-P categories) and 12 NA methods in the ME framework (four PNA methods associated with the ME-P-P categories and four MNA methods associated with each of the ME-M-M and ME-M-P categories). The alignment categories are color coded. **View I.** Overall ranking of the NA methods. The "Overall rank" column shows the rank of each method averaged over all evaluation tests, along with the corresponding standard deviation (in brackets). Since there are 12 methods in a given framework, the possible ranks range from 1 to 12. The lower the rank, the better the given method. The "p_1-value" column shows the statistical significance of the difference between the ranking of each method and the 1^{st} best ranked method. The "p_2-value" column shows the statistical significance of the difference between the ranking of each method and the 2^{nd} best ranked method. The "Non. sig. (fail)" column shows the fraction of evaluation tests in which the alignment quality score is not statistically significant, and, in brackets, the fraction of evaluation tests in which the given NA method failed to produce an alignment. Equivalent results over all evaluation tests broken down into functional and topological alignment quality measures, as well as over all evaluation tests broken down into network pairs/sets with known and unknown node mapping, are shown in Supplementary Tables S4–S11. **View II.** Alternative view of ranking of the NA methods. Each pie chart shows the fraction of evaluation test ranks that fall into the 1–4, 5–8, and 9–12 rank bins out of all evaluation test ranks in the given alignment category. For example, for the PE framework, in the PE-P-P alignment category, 56%, 26%, and 18% of the evaluation test ranks fall into ranks 1–4, 5–8, and 9–12, respectively, totaling to 100% of the evaluation test ranks in the PE-P-P alignment category. The pie charts allow us to compare the three alignment categories rather than individual NA methods in each category. The larger the pie chart for the better (lower) ranks, and the smaller the pie chart for the worse (higher) ranks, the better the alignment category. For example, in the PE framework, PE-P-P has the most evaluation tests ranked 1–4 and the fewest evaluation tests ranked 9–12, followed by PE-M-P, followed by PE-M-M. This implies that PE-P-P is superior to PE-M-P and PE-M-M. The pie charts are color coded with respect to alignments of network pairs/sets with known and unknown node mapping, and FQ and TQ measures. **View III.** Overall ranking of an NA method versus its running time. The latter are running time results when aligning all network pairs in the $Y2H_1$ network set under the PE framework, and when aligning the $Y2H_1$ network set under the ME framework, where each method is restricted to use a maximum of 64 cores. The size of each point visualizes the overall ranking of the corresponding method over all evaluation tests over all network pairs/sets, corresponding to the "Overall rank" column in View I; the larger the point size, the better the method. In order to allow for easier comparison between the different alignment categories, "Average" shows the average running times and average rankings of the methods in each alignment category.

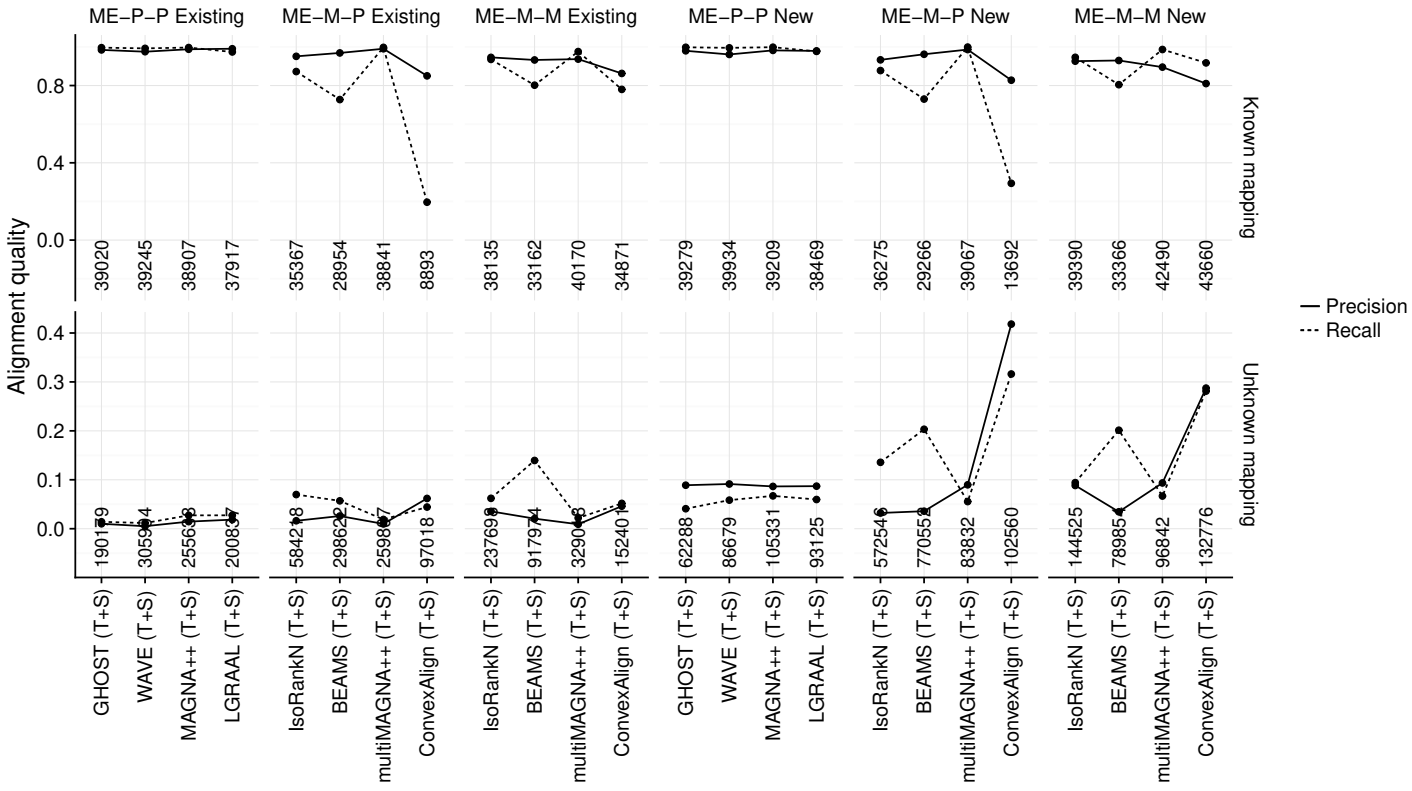


Figure S7: Comparison of protein function prediction accuracy between the **new** (approach 3) versus the **existing** prediction approach for multiple alignments (approach 2), for all alignments from the ME framework (i.e., ME-P-P, ME-M-P, and ME-M-M categories). We calculate the prediction accuracy as described in Fig. 6 in the main paper. Each column shows the precision and recall achieved by the new or existing prediction approach for each NA method, as well as the number of predictions made by the approach. The alignments are separated into networks sets with known and unknown mapping.

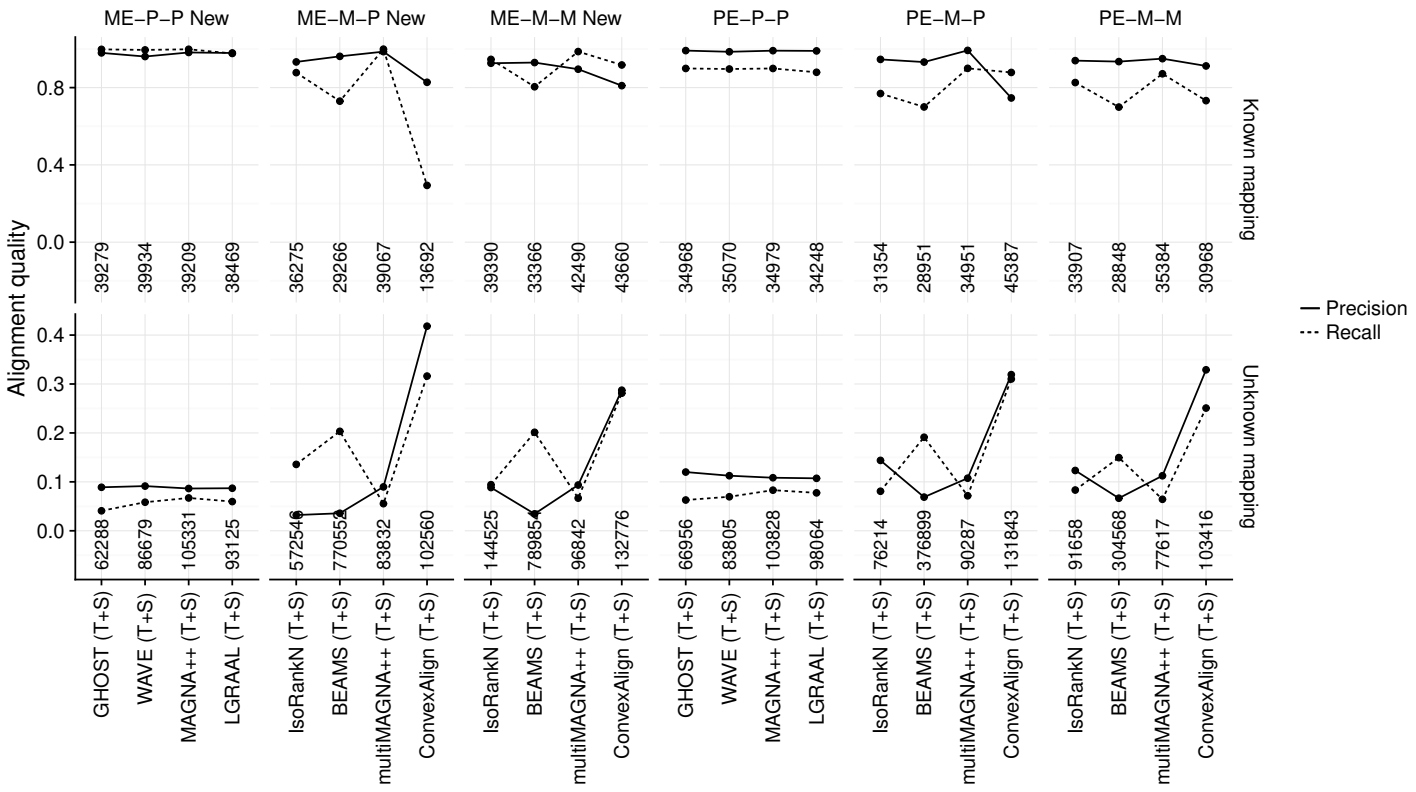


Figure S8: Comparison of protein function prediction accuracy under the **PE framework** (i.e., PE-P-P, PE-M-P, and PE-M-M categories) and **ME framework** (i.e., ME-P-P, ME-M-P, and ME-M-M categories). We calculate the prediction accuracy as described in Fig. 6 in the main paper. Each column shows the precision and recall achieved by the new or existing prediction approach for each NA method, as well as the number of predictions made by the approach. The alignments are separated into networks sets with known and unknown mapping.

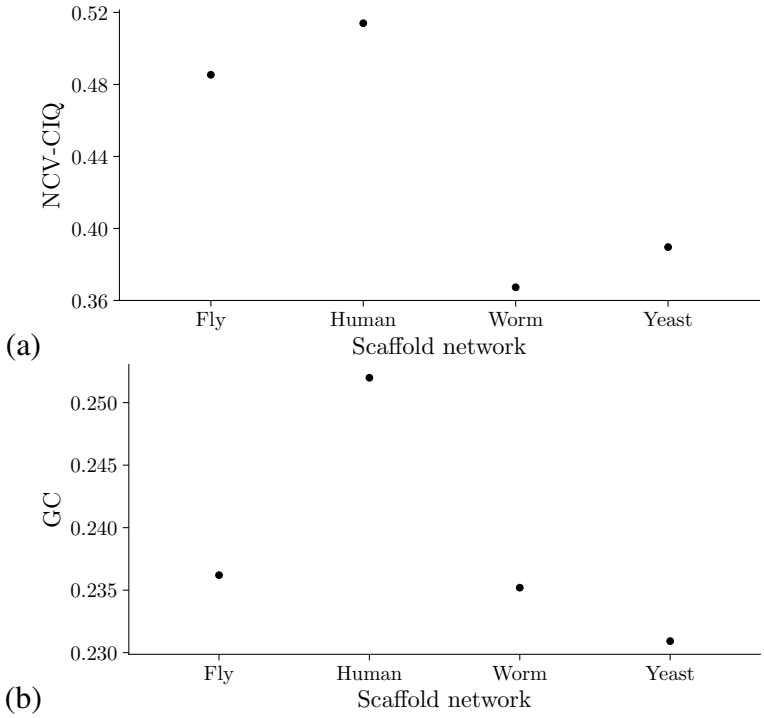
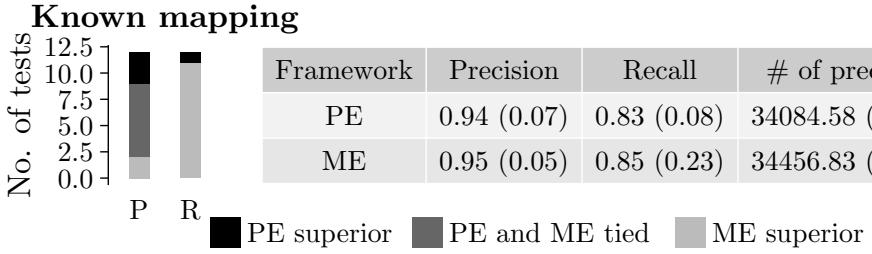
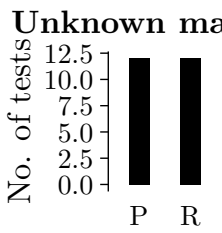


Figure S9: Illustration of the effect of the choice of scaffold network on alignment quality when combining pairwise alignments into a multiple alignment. These are representative results for one of the analyzed TQ measures (NCV-CIQ; panel (a)), one of the analyzed FQ measures (GO correctness – GC; panel (b)), one of the analyzed network sets (Y2H1), and one of the analyzed NA methods (WAVE). Clearly, different choices of scaffold network (x-axis) yield different alignment quality scores (y-axis). The same holds for other combinations of alignment quality measures, network sets, and NA methods. In our evaluation, of all scaffold network choices, the one that yields the best multiple alignment is chosen. In this particular representative scenario, it is the human network that was chosen as the scaffold, since this scaffold choice clearly yields significantly better alignment quality than any other scaffold choice.



Framework	Precision	Recall	# of predictions
PE	0.94 (0.07)	0.83 (0.08)	34084.58 (4303.68)
ME	0.95 (0.05)	0.85 (0.23)	34456.83 (8672.75)



Framework	Precision	Recall	# of predictions
PE	0.14 (0.09)	0.12 (0.08)	133762.92 (99302.50)
ME	0.02 (0.02)	0.05 (0.04)	319117.00 (223913.99)

Figure S10: Comparison of protein function prediction accuracy under the the PE and ME frameworks, where we use approach 2 for the ME framework (rather than using approach 3 for the ME framework like we do in Fig. 7 of the main paper). The figure can be interpreted the same way as Fig. 6 in the main paper.

Funding

This work was supported by the Air Force Office of Scientific Research (AFOSR) YIP FA9550-16-1-0147 grant.

References

Alkan, F. and Erten, C. (2014). BEAMS: backbone extraction and merge strategy for the global many-to-many alignment of multiple PPI networks. *Bioinformatics*, 30(4):531–539.

Dohrmann, J., Puchin, J., and Singh, R. (2015). Global multiple protein-protein interaction network alignment by combining pairwise network alignments. *BMC Bioinformatics*, 16(Suppl 13):S11.

Everitt, B. S., Landau, S., and Leese, M. (2001). *Cluster Analysis*. Wiley.

Gligorijević, V., Malod-Dognin, N., and Pržulj, N. (2015). FUSE: Multiple Network Alignment via Data Fusion. *Bioinformatics*, 32(8):1195–1203.

Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., and Pržulj, N. (2010). Topological network alignment uncovers biological function and phylogeny. *Journal of The Royal Society Interface*, 7(50):1341–1354.

Liao, C., Lu, K., Baym, M., Singh, R., and Berger, B. (2009). IsoRankN: Spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–258.

Meng, L., Striegel, A., and Milenković, T. (2016). Local versus global biological network alignment. *Bioinformatics*, 32(20):3155–3164.

Milenković, T. and Pržulj, N. (2008). Uncovering biological network function via graphlet degree signatures. *Cancer Informatics*, 6:257–273.

Pržulj, N. (2007). Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183.

Saraph, V. and Milenković, T. (2014). MAGNA: Maximizing accuracy in global network alignment. *Bioinformatics*, 30(20):2931–2940.

Vijayan, V. and Milenković, T. (2016). Multiple network alignment via multiMAGNA++. In *Proc. of Workshop on Data Mining in Bioinformatics (BIOKDD) at the Conference on Knowledge Discovery and Data Mining (KDD)*.

Vinh, N. X., Epps, J., and Bailey, J. (2007). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *The Journal of Machine Learning Research*, 11:410–420.