SUPPLEMENTARY INFORMATION FOR: Local versus Global Biological Network Alignment

Lei Meng [1,2,3] **Aaron Striegel** [1,3] **and Tijana Milenković** [1,2,*]
[1]Department of Computer Science and Engineering
[2]ECK Institute of Global Health and Interdisciplinary Center for Network Science and Applications
[3]Wireless Institute
University of Notre Dame, Notre Dame, IN 46556
[*]To whom correspondence should be addressed

Supplementary Sections

S1 Data description

Networks with unknown true node mapping. We have obtained PPI data for the following four species: *S. cerevisiae* (yeast), *D. melanogaster* (fly), *C. elegans* (worm), and *H. sapiens* (human). For each species, we extract four different PPI networks containing different interaction types and confidence levels: PHY_1, PHY_2, Y2H_1, and Y2H_2. Each of the four PPI network sets (PHY_1, PHY_2, Y2H_1, and Y2H_2) contains four networks (i.e., the largest connected components (LCCs)) corresponding to the four species (yeast, fly, worm, and human). Given four PPI network sets and six pairs of species per data set, we would ideally align (with a given NA method) $4 \times 6 = 24$ pairs of networks. Indeed, we align all of these network pairs, except those pairs involving PHY_2 and Y2H_2 networks of worm and fly, because these four networks are extremely small and sparse (Supplementary Table S1). Hence, we exclude these four networks from our analysis.

S2 Network aligners evaluated in our study

To evaluate LNA against GNA, we evaluate four LNA methods against six GNA methods. Each of the evaluated NA method is described as follows.

LNA methods. 1) NetworkBLAST identifies not only simpler pathways (as its predecessors do) but also more complex structures, such as dense functional modules or protein complexes. The method first constructs an alignment graph, in which a node represents a pair of putative orthologs from different networks, and an edge represents a conserved interaction. Next, it identifies the highest-scoring seeds and then searches for highly conserved network structures by extending around the seeds in a greedy fashion. **2)** NetAligner features pathway to interactome, complex to interactome, and interactome to interactome alignment. The method can identify conserved network regions of arbitrary topological structure by performing both inter- and intra-species alignment. During the alignment construction process, NetAligner first constructs an alignment graph, and then it finds connected components within the alignment graph. The method further extends the alignment graph: if the nodes in different connected components are connected directly or indirectly in one of the compared networks, they are then connected in the alignment graph. The connected components are searched for again in the extended alignment graph, and they represent the final alignment. **3)** AlignNemo is a recent local aligner, which is able to handle sparse networks. Unlike the previous LNA methods, AlignNemo builds a *weighted* alignment graph, in which a node represents a pair of orthologs from different species but edges are weighted via a scoring strategy accounting for direct and indirect interactions between their end nodes in the compared networks. So, the more paths go through both of the nodes, the greater the edge weight. Then, AlignNemo applies a seed-and-extend strategy to find relatively dense groups of nodes, i.e., nodes that have more interactions among themselves than with the rest of the networks. Note that the above notion of node orthology is based on sequence similarity between the corresponding proteins. When we want to use an alternative node similarity measure, such as topological (rather than sequence) similarity between nodes, we provide to AlignNemo as "new orthologs" the top K most similar node pairs, where node similarity is computed with respect to the alternative node similarity measure (e.g., topological similarity). We set the value of K to match the number of sequence-based orthologs across the compared networks. **4)** AlignMCL is another recent local aligner, and it is robust to the choice of networks to be aligned. The method first builds a *weighted* alignment graph in the same manner as AlignNemo and then applies Markov Clustering (MCL), a well-known graph clustering method that is robust to noise and graph alterations, to the alignment graph to identify conserved protein modules. When we use alternative node similarity measures, e.g., topological (rather than sequence) similarity between nodes, to build the alignment graph, we provide "new orthologs" to AlignMCL just as we do it for AlignNemo (see above).

GNA methods. **1)** GHOST first uses "spectral signatures" to compute pairwise topological similarities between nodes from different networks (also called *node cost function*). Next, GHOST uses the following *alignment* *strategy* on top of its node cost function from the previous step to construct injective node mapping between the networks: It searches for high scoring nodes (with respect to the node cost function) from the different networks, uses these seed nodes as the current alignment, and iteratively extends around the current alignment while aiming to approximate the solution to the quadratic assignment problem (again with respect to the node cost function). **2)** Unlike GHOST, which computes node cost function scores only once, prior to running its alignment strategy step, NETAL recomputes its node cost function scores during each iteration of its alignment strategy. Namely, at each step, NETAL recomputes the similarity of unaligned nodes based on the current alignment and the expected number of conserved interactions incident to the nodes if the nodes were to be aligned. **3)** Unlike GHOST and NETAL that try to maximize node similarity in hope to conserve as many edges as possible, GEDEVO uses an evolutionary algorithm aiming to optimize graph edit distance, i.e., the minimal number of modifications (edge insertions and deletions) required to convert one network to the other. The evolutionary algorithm iteratively performs three steps including individual evaluation, offspring generation, and survival function application until certain condition is met. **4)** While most of the existing methods aim to optimize either node conservation in hope to conserve more edges or optimize edge conservation directly during alignment construction process, MAGNA++ simultaneously optimizes both node and edge conservation while constructing an alignment. The intuitive idea behind MAGNA++ is to use a genetic algorithm to crossover two parent alignments into a superior child alignment. **5)** WAVE is a recent aligner that employs the two-step node cost function and alignment strategy idea, just as GHOST and NETAL, but that also aims to optimize both node and edge conservation, just as MAGNA++ does. Further, unlike MAGNA++, WAVE *weighs* conserved edges to favor aligning edges whose end nodes are similar over aligning edges whose end nodes are dissimilar. **6)** Similar to MAGNA++, L-GRAAL directly optimizes both (sequence-based) node conservation and (graphlet-based) edge conservation, and it does so using a heuristic seed-and-extend strategy based on integer programming and Lagrangian relaxation.

Note that some methods are unable to complete successfully on the largest network pairs (Supplementary Table S2) due to high computational complexity. Namely, for these networks, AlignNemo and AlignMCL crash mostly due to memory errors, while NetworkBLAST, GHOST, GEDEVO, and L-GRAAL cannot complete even when run for several weeks. We leave out from consideration such combinations of NA methods and network pairs when we report our results.

S3 Aligners' node cost functions

Topology- and sequence-based NCF. Regarding topology-based NCF, each NA method considered in our study uses one of the following measures: graphlet degree vector similarity (GDV-similarity), neighborhood similarity, or spectral signature similarity. For methods that use GDV-similarity to compute topological node similarities within NCF, we vary the considered graphlet size. Namely, for Y2H$_1$ and Y2H$_2$ network sets that are of reasonable size, we use all up to 5-node graphlets. For PHY$_1$ and PHY$_2$ network sets that are relatively large, we use all up to 4-node graphlets, due to high computational complexity of counting 5-node graphlets in these networks. Regarding sequence-based NCF, each of the NA methods considered in our study uses either *E-value* scores or normalized *E-value* scores. We compute normalized *E-value* scores by first taking the negative logarithm of the corresponding *E-value* scores and then dividing the results by the maximum value over all node pairs between the compared networks. Topology- and sequence-based NCFs that we use within the different NA methods are shown in Supplementary Table S4.

S4 Evaluation of alignment quality

GO correctness (GC). This measure quantifies the extent to which protein pairs that are aligned under f are annotated with the same GO terms. *GC* is defined as the percentage of protein pairs that are aligned under f in which both proteins share at least k GO terms, out of protein pairs that are aligned under f in which each protein is annotated with at least k GO term ($k = 1, 2, \dots$). In our study, we choose $k = 1$. GC is directly related to an alternative measure of biological alignment quality – GO semantic similarity [1].

Precision, recall, and F-score of known protein function prediction (P-PF, R-PF, and F-PF, respectively). We make GO term prediction(s) for each protein from G_1 and G_2 that is annotated with at least one GO term through a multi-step process. First, we hide the protein's true GO terms. Second, we find statistically significant alignments with respect to each of those GO terms. To compute the statistical significance of an alignment with respect to GO term g, we use *hypergeometric test* as follows. Let V_1^* and V_2^* be the sets of nodes from G_1 and G_2, respectively, such that each node is annotated with at least one GO term. Let S_1 be the set of possible node pairs that could be aligned between V_1^* and V_2^*. Let A_1 and A_2 be the sets of nodes from V_1^* and V_2^*, respectively, such that each node is annotated with g. Let S_2 be the set of possible node pairs that could be aligned between A_1 and A_2. Let K be the set of aligned node pairs that are also in S_1. Let X be the set of aligned node pairs that are also in S_2. The statistical significance of the alignment with respect to a GO term g

(i.e., the probability p of observing by chance $|X|$ or more aligned node pairs in which each node is annotated with g) is:

$$p = 1 - \sum_{i=0}^{|X|-1} \frac{\binom{|K|}{i}\binom{|S_1|-|K|}{|S_2|-i}}{\binom{|S_1|}{|S_2|}} \qquad (1)$$

We use p-value threshold of 0.05. Third, we predict from statistically significant alignments the protein's GO terms based on the GO terms of its aligned counterpart(s) under f. After we make predictions for all proteins, we evaluate the precision, recall, and F-score of the prediction results with respect to the true GO terms of the proteins. Formally, let X be the set of all predicted protein-GO term associations where each association contains a protein and its predicted GO term. Let Y be the set of all protein-GO term associations where each association contains a protein and its true GO term. P-PF is defined as $\frac{|X \cap Y|}{|X|}$. R-PF is defined as $\frac{|X \cap Y|}{|Y|}$. F-PF, combining P-PF and R-PF, is the harmonic mean of precision and recall.

S5 Application to novel protein function prediction

We use LNA and GNA to find statistically significant alignments and make novel protein function predictions from such alignments (as described in Supplementary Section S4). We use p-value threshold of 0.05. In an alignment that is statistically significant with respect to GO term g, for each aligned protein pair, if only one protein is annotated with g, we predict the other protein to be annotated with g as well. We refer to this as a novel prediction.

S6 Validation of the alignment quality measures

Recall that in the main paper we evaluate whether the analyzed measures, as well as NA methods, are insensitive to the edge order of the input networks (as good measures/methods should be). Namely, throughout this study, it has come to our attention that some measures or NA methods might be biased when aligning certain input networks. For example, when aligning a network to itself, some measures or methods may produce higher-quality alignment scores when the edge orders of the two input networks match than when their edge orders are different. For example, if both networks are triangles with the same nodes a, b, and c, giving both input networks as edge list $((a, b), (a, c), (b, c))$ may result in higher alignment quality than giving one input network as edge list $((a, b), (a, c), (b, c))$ and the other input network as edge list $((b, c), (a, c), (a, b))$, even though in both scenarios the input networks are the same. Hence, we analyze whether any of our considered measures or NA methods are sensitive to the edge order of input networks. We find that most of the measures and NA methods are insensitive to this parameter (Supplementary Figure S3), which further confirms their meaningfulness. The only exceptions are related to NetworkBLAST, NETAL, and WAVE, for which some measures show sensitivity to edge order (Supplementary Figure S3).

S7 Networks with known true node mapping

S7.1 Relationships between different alignment quality measures

We first study relationships and potential redundancies of different alignment quality measures, in order to select only non-redundant measures to fairly evaluate LNA against GNA. This is needed for the following reason. Let us assume that we are considering three measures, of which the first two are highly redundant to each other (in the sense that their alignment quality scores across different network pairs or different NA methods are very "correlated"), while the third one is very different from the first two. Also, let us assume that we are considering two NA methods, of which the first one is found to be superior over the second one with respect to the first two measures, while the second NA method is found to be superior over the first one with respect to the third measure. If we compare the performance of the two methods simply by considering all three measures independently of each other, clearly, the first method would mistakenly be overall superior in 2/3 of all cases. However, if we account for the redundancy of the first two measures and remove one of them from consideration, then the two methods would fairly be identified as comparable, as each would be superior in 1/2 of all cases. Thus, our goal is to first understand potential redundancies of the considered alignment measures and choose the best and most representative of all redundant measures. Only then, we aim to use the selected non-redundant measures to fairly evaluate LNA against GNA.

For networks with known true node mapping, we use the six topological measures: P-NC, R-NC, F-NC, NCV, GS^3, and NCV-GS^3 (Section 2.4.1 in the main paper). For a given alignment quality measure, we compute the score of aligning each of the five pairs of networks with known true node mapping by each of the four LNA and six GNA methods. Then, for each pair of measures, we compute the Pearson correlation coefficients across all of the $5 \times 4 = 20$ alignment quality scores for LNA and all of the $5 \times 6 = 30$ alignment quality scores for GNA. We do this for each type of information used within NCF during the alignment construction process, namely T, T&S, and S.

Since all six measures are topological, we expect them to be overall highly correlated with each other. Indeed, most of the measures are significantly correlated with each other for both LNA and GNA, with respect to each of T, T&S, and S (Figure 4 (a) and (b) in the main paper and Supplementary Figure S4). Specifically, for LNA, with respect to each of T, T&S, and S, all measures are significantly correlated (p-values < 0.05), with the following exceptions with respect to S: the pair of P-NC and R-NC are not correlated, and NCV is significantly correlated only with R-NC but not with any of the other measures. For GNA, with respect to T, all measures are significantly correlated (p-values $< 10^{-15}$) except NCV, which is not correlated with any of the other measures; for T&S, all measures are significantly correlated (p-values $< 10^{-5}$), with the exception of NCV, which is only significantly correlated with R-NC and F-NC (p-values < 0.05) but not with any of the other measures; with respect to S, all measures are significantly correlated with each other (p-values < 0.05), except the pair of P-NC and NCV-GS^3, which are not correlated, and except GS^3, which is only significantly correlated with P-NC (p-value $< 10^{-3}$) but not with any of the other measures.

S7.2 Comparison of LNA and GNA

We zoom into the results from Figure 6 in the main paper and Supplementary Figures S6 and S7 to identify the best of the NA methods considered in our study. Recall that for LNA, NetworkBLAST and NetAligner do not allow for using topological information in NCF. Thus, we cannot consider these methods for T and T&S. Given this, the results for LNA are as follows. For T, the remaining methods, AlignNemo and AlignMCL, are comparable (Figure 6 (a) and (b) in the main paper). For T&S, S and B, of all four methods, AlignMCL is superior (Figure 6 (c) and (d) in the main paper). Hence, we conclude AlignMCL to be the best LNA method among our selections. Recall that for GNA, NETAL and GEDEVO do not allow for using sequence information in NCF. So, we cannot consider these methods for T&S and S. (For these methods, B is the same as T). Given this, the results for GNA are as follows. For T, all GNA methods are comparable, except GHOST, which is inferior. For T&S, all methods are comparable. For S, all methods are comparable, except L-GRAAL, which is inferior. For B, all methods are comparable, except NETAL and GEDEVO, which are inferior. Hence, since MAGNA++ and WAVE are never inferior for any of T, T&S, S, or B, we conclude these two methods to be the best GNA methods among all methods considered in our study.

S8 Networks with unknown true node mapping

S8.1 Relationships of different alignment quality measures

For the networks with unknown true node mapping, we use all seven measures: three topological (NCV, GS^3, and NCV-GS^3; Section 2.4.1 in the main paper) and four biological (GC, P-PF, R-PF, and F-PF; Section 2.4.2 in the main paper). For a given measure, we compute the score of aligning each of the 14 pairs of networks with unknown true node mapping by each of the four LNA and six GNA methods. Then, for each pair of measures, we compute the Pearson correlation coefficients across all of the $14 \times 4 = 56$ alignment quality scores for LNA and $14 \times 6 = 84$ alignment quality scores for GNA. We do this for each type of information used within NCF during the alignment construction process, namely T, T&S, and S.

Since by definition all seven measures naturally cluster into two groups (one group consisting of the three topological measures that capture the size of the alignment in terms of the number of nodes or edges, and the other group consisting of the four biological measures that quantify the extent of functional similarity of the aligned nodes), we expect within-group correlations to be higher than across-group correlations. Indeed, this is what we observe overall for both LNA and GNA with respect to each of T, T&S, and S (Figure 4 (c) and (d) in the main paper and Supplementary Figure S10).

Specifically, for LNA, with respect to each of T, T&S, and S, all measures within the first group are significantly correlated (p-values < 0.01), all measures within the second group are significantly correlated (p-values < 0.05), but no measures across the two groups are significantly correlated, with the following exceptions. Within the first group, with respect to T&S, none of GC and P-PF are significantly correlated with any of R-PF and F-PF; also, with respect to S, none of P-PF, R-PF, F-PF, and GC are significantly correlated with each other except the pair of R-PF and F-PF. Within the second group, with respect to T&S, NCV is not significantly correlated with any of GS^3 and NCV-GS^3; also, with respect to S, NCV is not significantly correlated with NCV-GS^3. Across the two groups, with respect to S only, the following pairs are significantly correlated: P-PF and GS^3, P-PF and NCV-GS^3, R-PF and NCV, and F-PF and NCV.

For GNA, with respect to each of T, T&S, and S, all measures within the first group are significantly correlated (p-values $< 10^4$), all measures within the second group are significantly correlated (p-values < 0.01), but no measures across the two groups are significantly correlated, with the following exceptions. Within the first group, with respect to T&S only, P-PF is not significantly correlated with any of R-PF and F-PF. Within the second group, with respect to each of T, T&S, and S, NCV and GS^3 are not significantly correlated; also, with respect to S, NCV and NCV-GS^3 are not significantly correlated. Across the two groups, with respect to T&S, GS^3 and GC are significantly correlated.

S8.2 Comparison of LNA and GNA

We zoom into the results from Figure 8 in the main paper and Supplementary Figures S12-S16 in order to identify the best of the NA methods considered in our study. Recall that for LNA, NetworkBLAST and NetAligner do not allow for using topological information in NCF. So, we cannot consider these methods for T and T&S. Given this, the results for LNA are as follows. With respect to topological alignment quality: for T, T&S, and B, AlignNemo is the best method among our selections; for S, AlignMCL is the best method among our selections. With respect to biological alignment quality: for T, T&S, S and B, AlignNemo and AlignMCL are the best methods among our selections and are comparable. Hence, we conclude AlignNemo and AlignMCL to be the best of all analyzed LNA methods among our selections. Recall that for GNA, NETAL and GEDEVO do not allow for using sequence information in NCF. So, we cannot consider these methods for T&S and S. (Clearly, for these methods, B is the same as T). Given this, the results for GNA are as follows. With respect to topological alignment quality: for T, NETAL is the best method among all methods considered in our study; for T&S, MAGNA++, WAVE, and L-GRAAL are the best methods and are comparable; for S, GHOST and MAGNA++ are the best methods considered in our study; for B, NETAL and MAGNA++ are superior. With respect to biological alignment quality: for T, GHOST, NETAL, MAGNA++, and WAVE are the best of all considered methods; for T&S and S, L-GRAAL is superior; for B, WAVE and L-GRAAL are the best methods among our selections. Hence, we conclude that for GNA, in this analysis, the best of all considered methods varies depending on whether we are measuring topological versus biological alignment quality and depending on the type of information used in NCF.

S9 Running time method comparison

Single-core analysis. For the entire running time, overall, for T, most of the GNA methods run faster than the LNA methods; for T&S, the GNA methods run similarly to the LNA methods. For S, LNA is faster than GNA. A given NA method typically runs faster for T than for S. This is because producing topological similarity scores takes less time than producing sequence similarity scores for relatively small and sparse networks. However, this may not be the case for large and dense networks. Furthermore, a given method typically runs slower for T&S than for T or S. This is because for T&S, a method needs to compute both topological and sequence similarity scores, rather than just one of the two, as is the case for T and S. For only the time needed to construct alignments, overall, LNA methods run faster than GNA methods for each of T, T&S, and S. This could be because given the node similarity scores, GNA typically searches for larger alignments (conserved subgraphs) compared to LNA. Also, a given LNA method runs faster for T than S, while a given GNA method runs similarly for T, T&S, and S.

Multi-core analysis. For the entire running time, for T, GNA remains faster than LNA. However, for T&S and S, unlike in the above single-core analysis where LNA is comparable or superior to GNA, GNA is now always comparable (if not even superior) to LNA. Within the LNA method group, all methods are comparable in terms of their running times, for each of T, T&S, and S. Within the GNA group, for T, NETAL is the fastest, followed by GHOST, WAVE, MAGNA++, GEDEVO, and L-GRAAL, respectively, while for T&S and S, GHOST and L-GRAAL are faster than MAGNA++ and WAVE. For only the time needed to construct alignments, LNA mostly remains faster than GNA. Within the LNA method group, for T, AlignMCL is the fastest, while for T&S and S, AlignNemo is the fastest, in terms of their running times. Within the GNA group, for T, NETAL is the fastest, followed by WAVE, GHOST, MAGNA++, L-GRAAL, and GEDEVO, respectively, while for T&S and S, WAVE is the fastest followed by GHOST, L-GRAAL, and MAGNA++.

S10 User-friendly GUI plus source code

We make publicly available our software for fairly evaluating a NA (LNA or GNA) method (`http://www.nd.edu/~cone/LNA_GNA`). The software provides a friendly and intuitive GUI (Supplementary Figure S20) and python source code for any platform.

Intuitively, the input to our software is an alignment and the output is the quality of the alignment. Formally, the input varies based on the alignment quality measure of interest. One required input to our software is the alignment to be evaluated (in the form of aligned node pairs), which can be produced by aligning any two networks via any NA method (this is not limited to the networks and NA methods discussed in our paper). For P-NC, R-NC, and F-NC topological evaluation, the true node mapping (in the form of aligned node pairs) between the aligned networks is required as additional input. For the remaining NCV, GS^3 and NCV-GS^3 topological evaluation, the two aligned networks are required as additional input. For GC, P-PF, R-PF, and F-PF biological evaluation, GO annotations of the nodes in both aligned networks are required as input. The output consists of the alignment quality scores of the user's interest.

In addition to a user-friendly GUI that is targeted primarily at domain scientists, we also provide the source code for easily extensibility by computational scientists. For example, additional topological and biological alignment quality measures can be easily incorporated into our evaluation framework.

Our software is intended to allow for analyzing an alignment that is provided as input into the software. It does not implement the existing NA methods that produce the input alignment in the first place. The latter is certainly of interest, but this is an extremely challenging task. This is because the different NA methods are typically developed by different research groups and are often required to run in different system environments. Thus, it would be very hard (perhaps even not permitted) to integrate them into our software.

Supplementary Figures

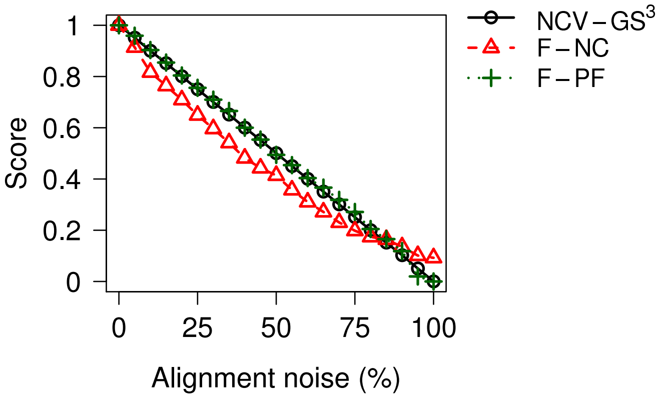
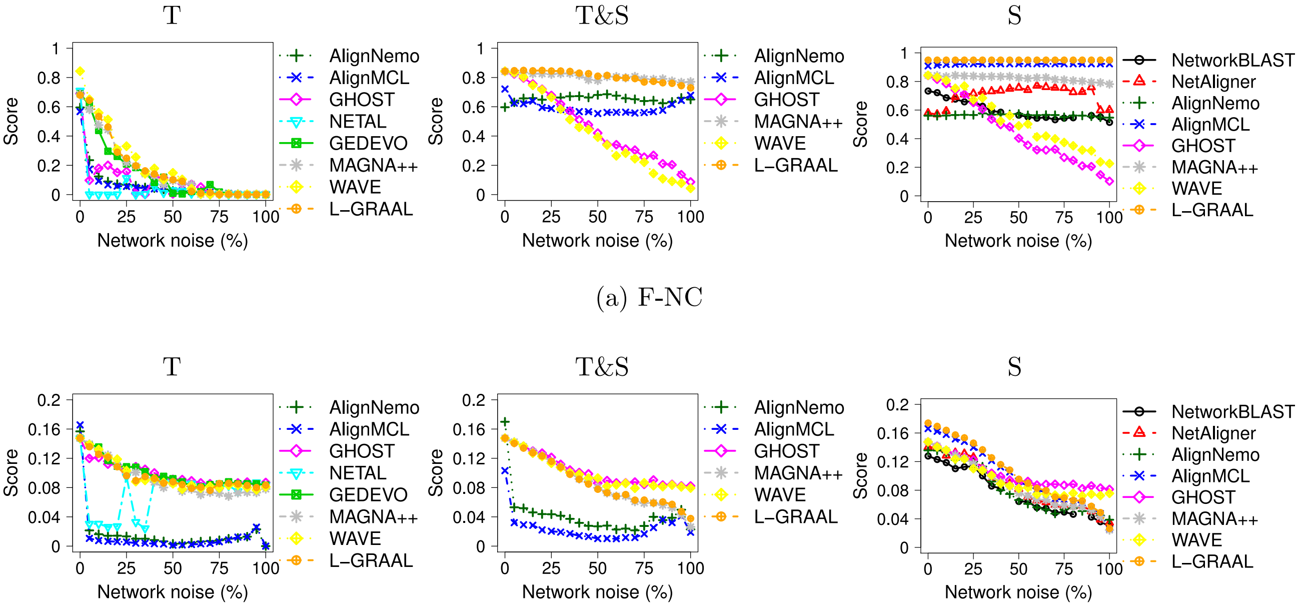


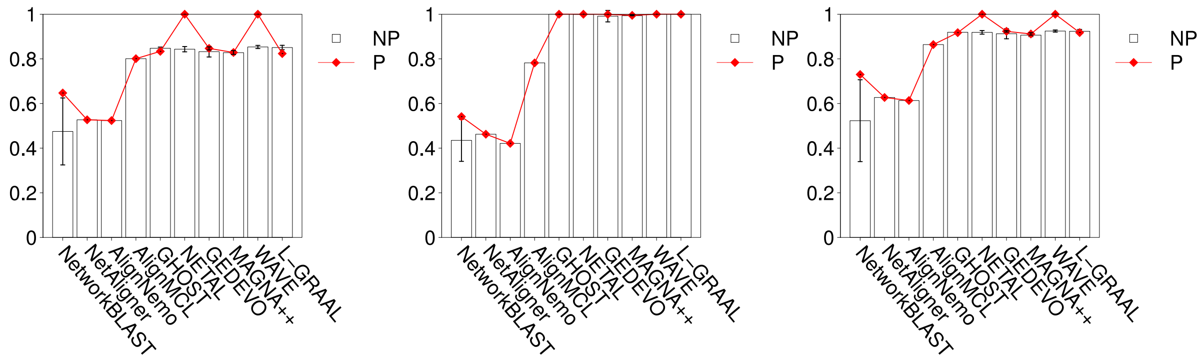
Figure S1: Validation of the three alignment quality measures when increasing the noise level from 0% to 100% in the alignment of the high-confidence yeast network (from the set of networks with known true node mapping) with itself. Note that the noise is not introduced into the network structure prior to the alignment process. Instead, the noise is introduced into the alignment directly. A good measure should show decreasing alignment quality with increase in the noise level. Since all of the measures show this trend, we conclude that the measures are meaningful.



(a) F-NC

(b) NCV-GS^3

Figure S2: Validation of the representative newly proposed alignment quality measures, **(a)** F-NC and **(b)** NCV-GS^3, when introducing the increasing noise level from 0% to 100% into the high-confidence yeast network (from the set of networks with known true node mapping) prior to aligning the high-confidence network with its noisy versions, for each of the aligners, with respect to T, T&S, and S. Results for F-PF closely match those for F-NC and are thus not reported. If some of the analyzed four LNA and six GNA methods are missing in the given panel, that means that the given method cannot be run with the corresponding type of information used in NCF (T, T&S, or S).



(a) F-NC (b) NCV-GS^3 (c) F-PF

Figure S3: Validation of our the three alignment quality measures, **(a)** F-NC, **(b)** NCV-GS^3, and **(c)** F-PF, when the edge *order* (but *not* the actual edges, i.e., the network structure!) in the high-confidence yeast network (from the network set with known true node mapping) is randomly permuted prior to aligning this network with itself. For each NA method, when we align this network with itself, we: 1) keep the edge order of the two input networks unchanged, so that the edge order matches between the two networks (corresponding to a non-permuted test, or "NP", in the figure) and 2) randomly permute the edge order in each of the two input networks, so that the edge order mismatches between the two networks (corresponding to a permuted test, or "P", in the figure). We do this 10 times to account for randomness in the edge order permutation process. The average alignment quality scores (and the corresponding standard deviations) across the 10 runs are reported for each NA method and each measure, for T. If for a given NA method, a measure is insensitive to the edge order of the input networks (as we want it to be), the NP and P tests would result in comparable alignment quality and small standard deviations. Overall, all measures are insensitive to the edge order of the input networks, for all NA methods, with the exception of NetworkBLAST, the oldest of the considered NA methods, as well as NETAL and WAVE, for which some of the measures are somewhat sensitive to the edge order. Therefore, we conclude that most of the studied methods are insensitive to the edge order of the input networks with respect to most of the measures.

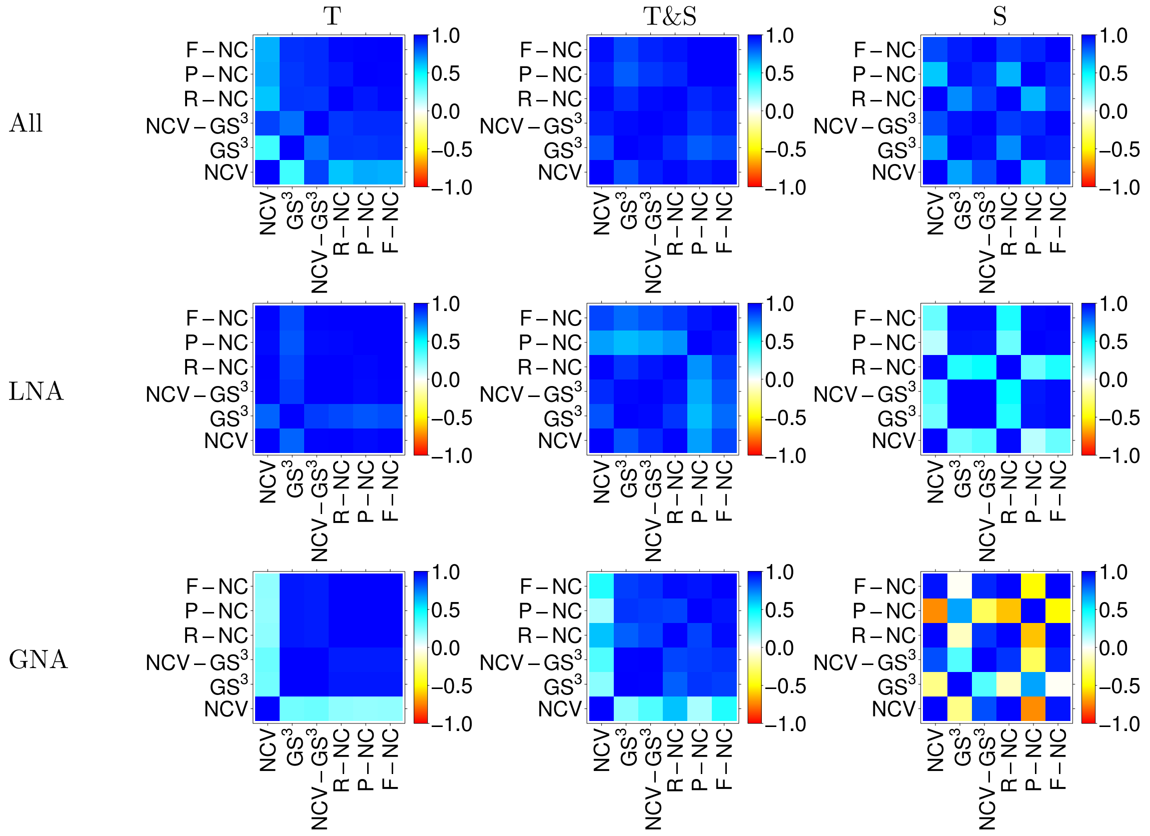


Figure S4: Pairwise relationships (Pearson correlations) between the six topological alignment quality measures over all network alignments, i.e., both LNA and GNA combined (top), all LNAs (horizontal middle), and all GNAs (bottom), for networks with known true node mapping, for T (left), T&S (vertical middle), and S (right).

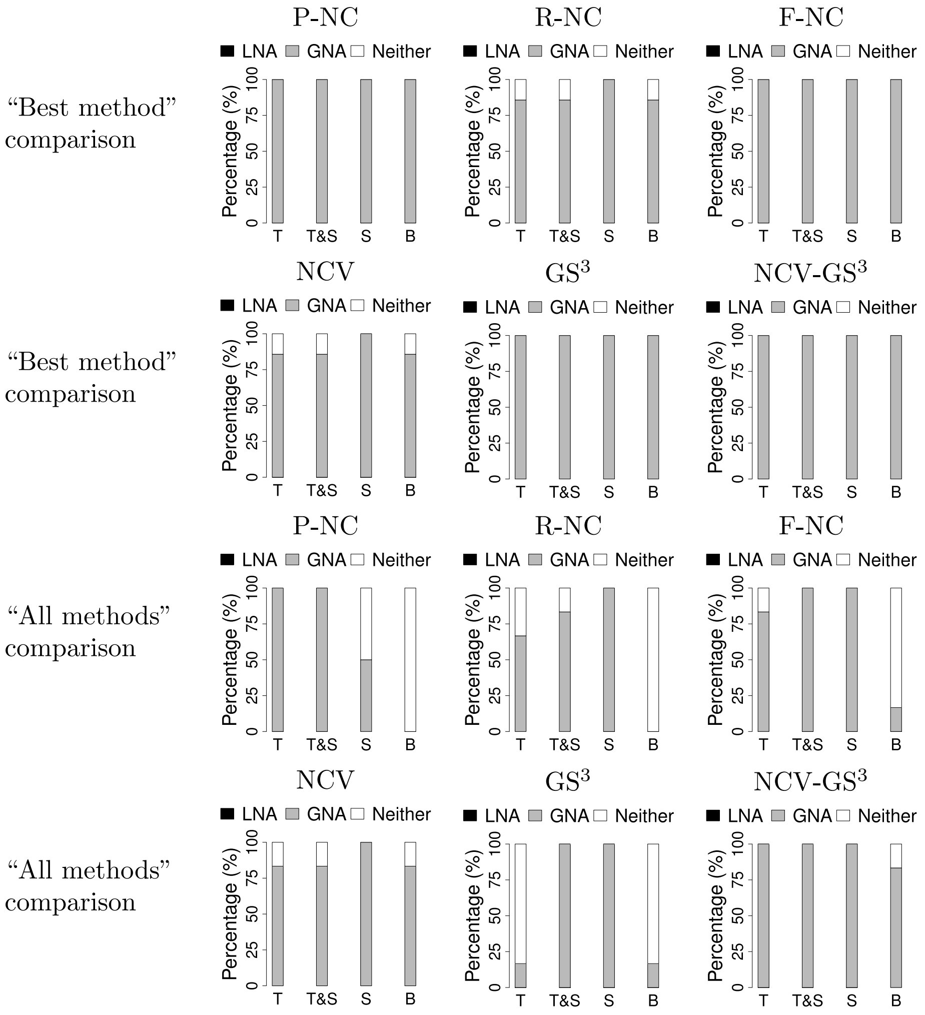


Figure S5: Overall comparison of LNA and GNA for networks with known true node mapping with respect to "best method" and "all methods" comparisons, for T, T&S, S, and B. The results are broken down for each of the six topological alignment quality measures (i.e. P-NC, R-NC, F-NC, NCV, GS^3, and NCV-GS^3). Each bar shows the percentage of the aligned network pairs for which LNA is superior (black), GNA is superior (grey), or neither LNA nor GNA is superior (white).

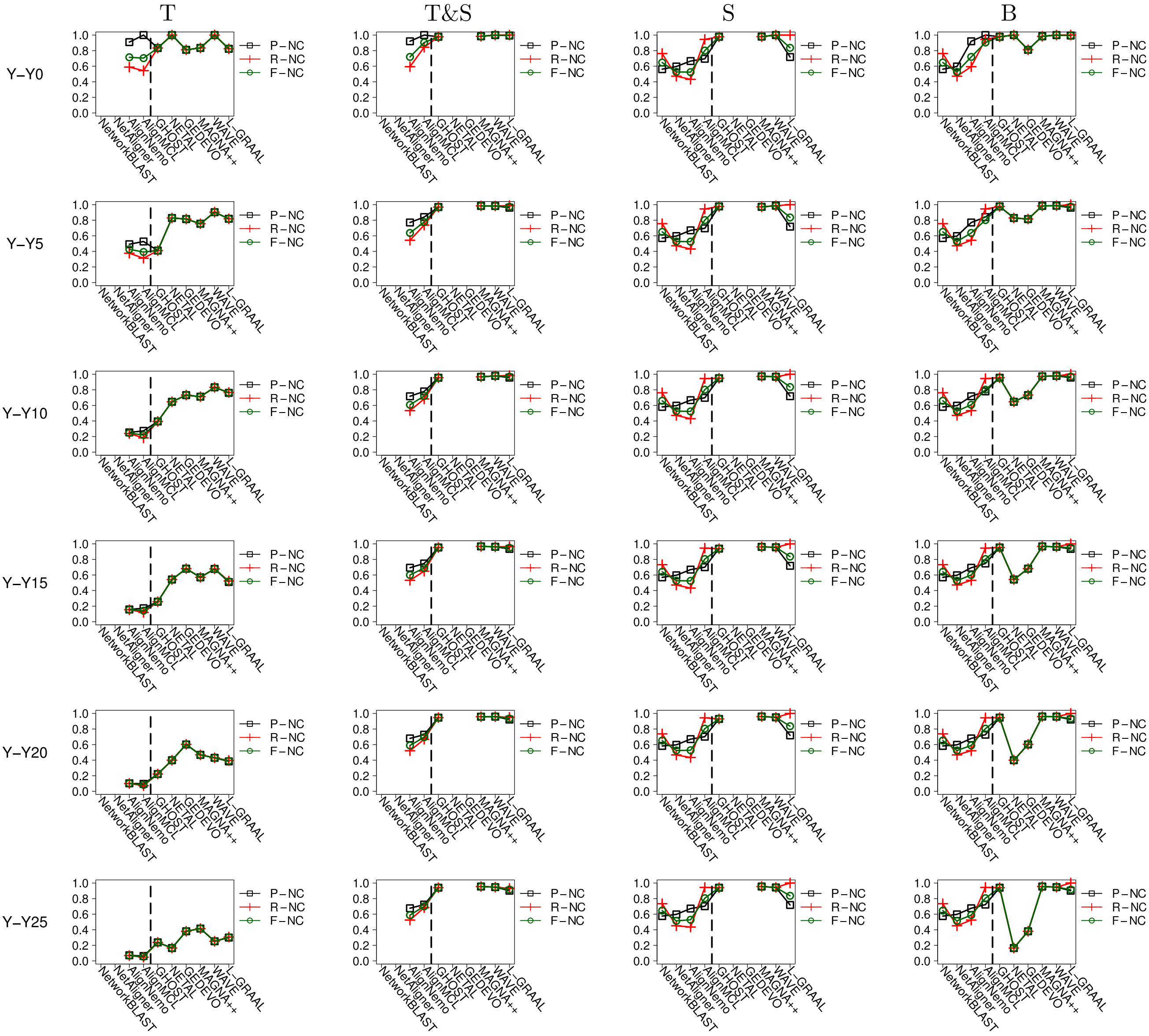


Figure S6: Topological alignment quality with respect to P-NC, R-NC, and F-NC, for each NA method considered in our study, each pair of networks with known true node mapping (Y denotes the high-confidence yeast network, and Yx denotes the noisy yeast network constructed by adding to Y x% of lower-confidence PPIs), and each of T, T&S, S, and B.

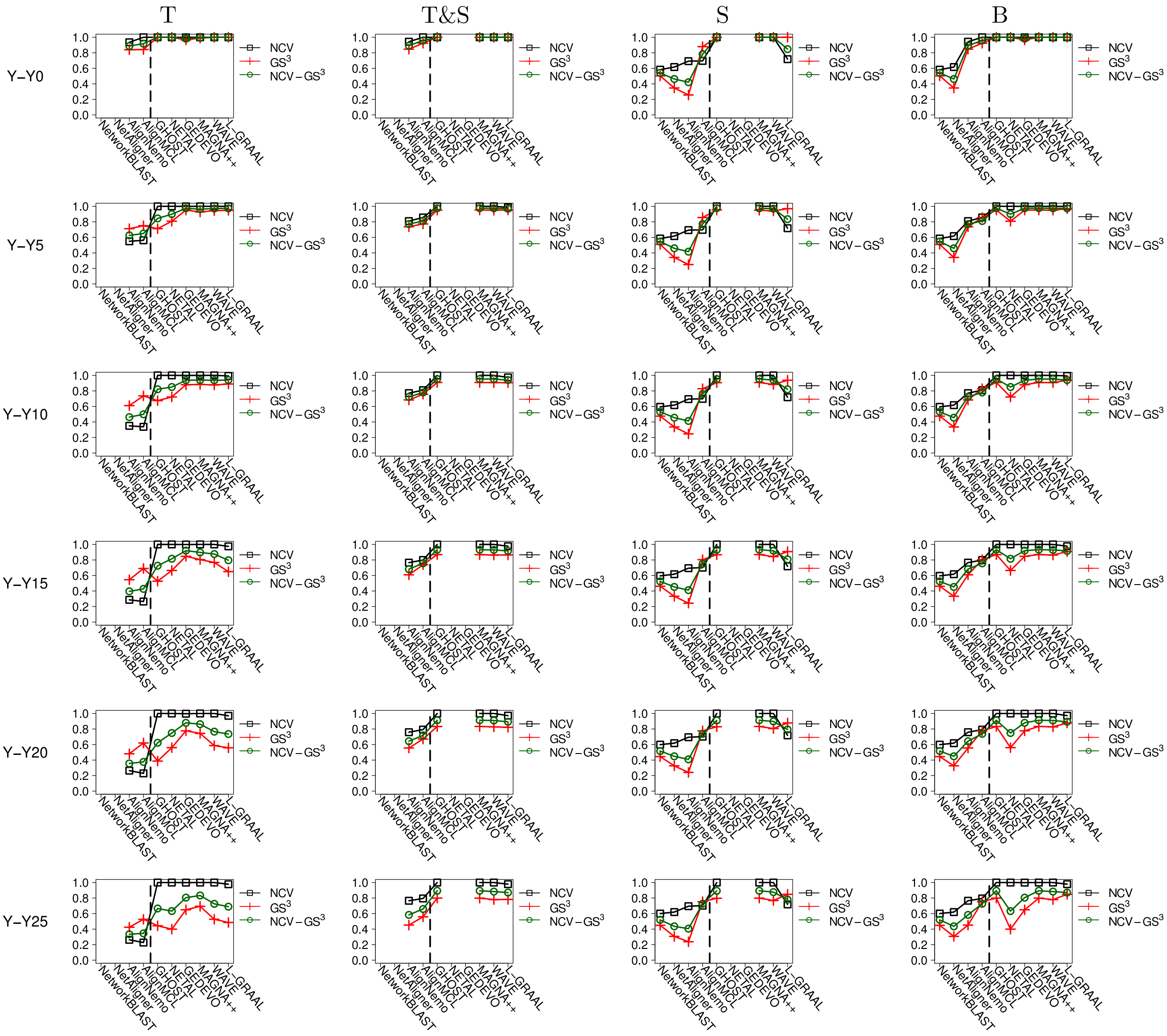


Figure S7: Topological alignment quality with respect to NCV, GS^3, and NCV-GS^3, for each NA method considered in our study, each pair of networks with known true node mapping (Y denotes the high-confidence yeast network, and Yx denotes the noisy yeast network constructed by adding to Y x% of lower-confidence PPIs), and each of T, T&S, S, and B.

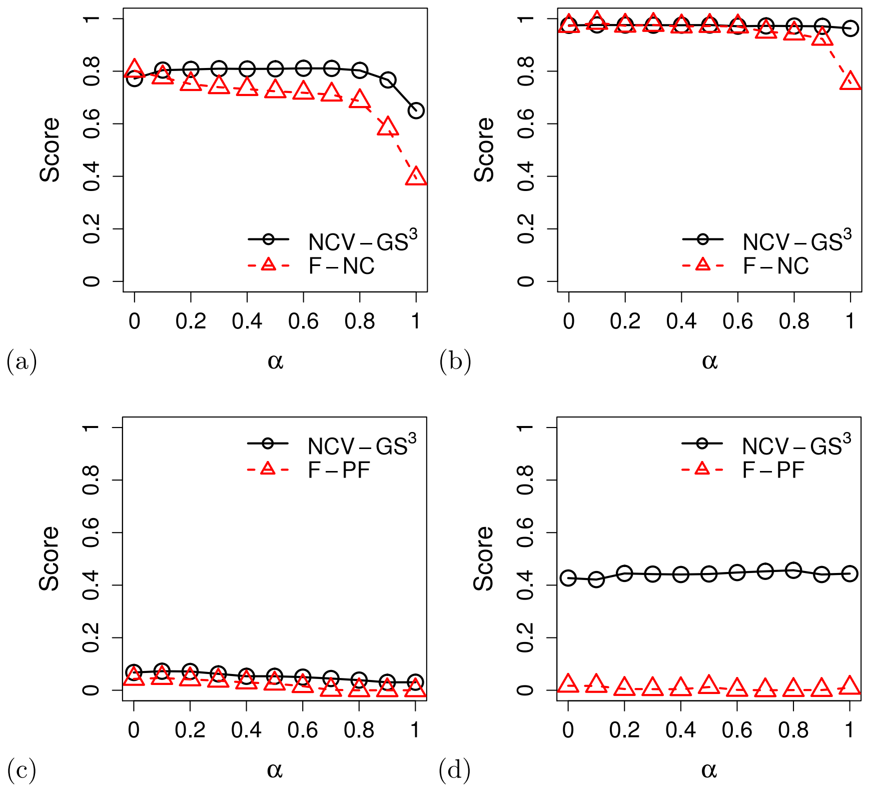


Figure S8: The effect of using different values of α (the amount of topological information) in NCF with respect to the representative LNA method (AlignMCL; panels **(a)** and **(c)**) and the representative GNA method (MAGNA++; panels **(b)** and **(d)**), for a network pair with known true node mapping (panels **(a)** and **(b)**) and a network pair with unknown true node mapping. The network pair with known node mapping contains the high-confidence yeast network and a noisy yeast network constructed by adding to the high-confidence yeast network 5% of lower-confidence PPIs. The network pair with unknown node mapping contains yeast and worm networks from the Y2H$_1$ set. For both network pairs, the choice of α between 0.1 and 0.9 (corresponding to T&S) has no major effect on alignment quality, as in this α range, the scores remain mostly consistent, which demonstrates the robustness of our results to the choice of the α parameter value. Similar results hold for other LNA and GNA methods and other network pairs.

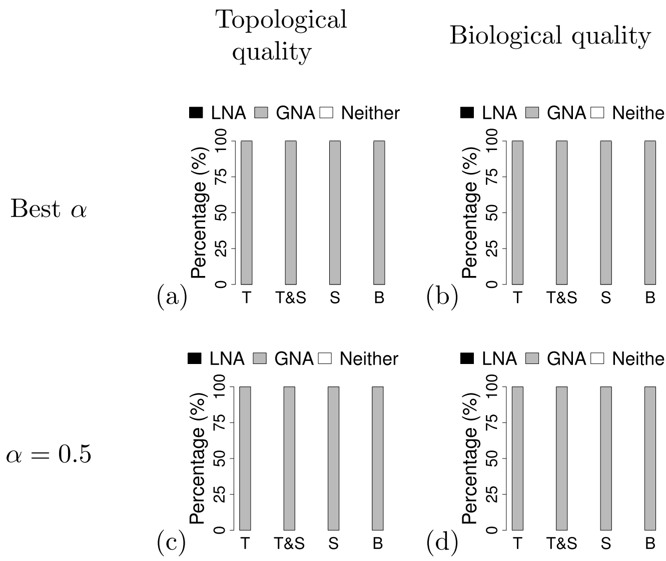


Figure S9: Overall comparison of LNA and GNA when using the best α value between 0.1 and 0.9 (panels **(a)** and **(b)**) or the α value of 0.5 (panels **(c)** and **(d)**), demonstrating the robustness of the results to the choice of the α parameter value. These representative results are for the *"best method"* comparison, for networks with known true node mapping, with respect to either the topological NCV-GS^3 measure (panels **(a)** and **(c)**) or the biological F-NC measure (panels **(b)** and **(d)**). Each bar shows the percentage of the aligned network pairs for which LNA is superior (black), GNA is superior (grey), or neither LNA nor GNA is superior (white).

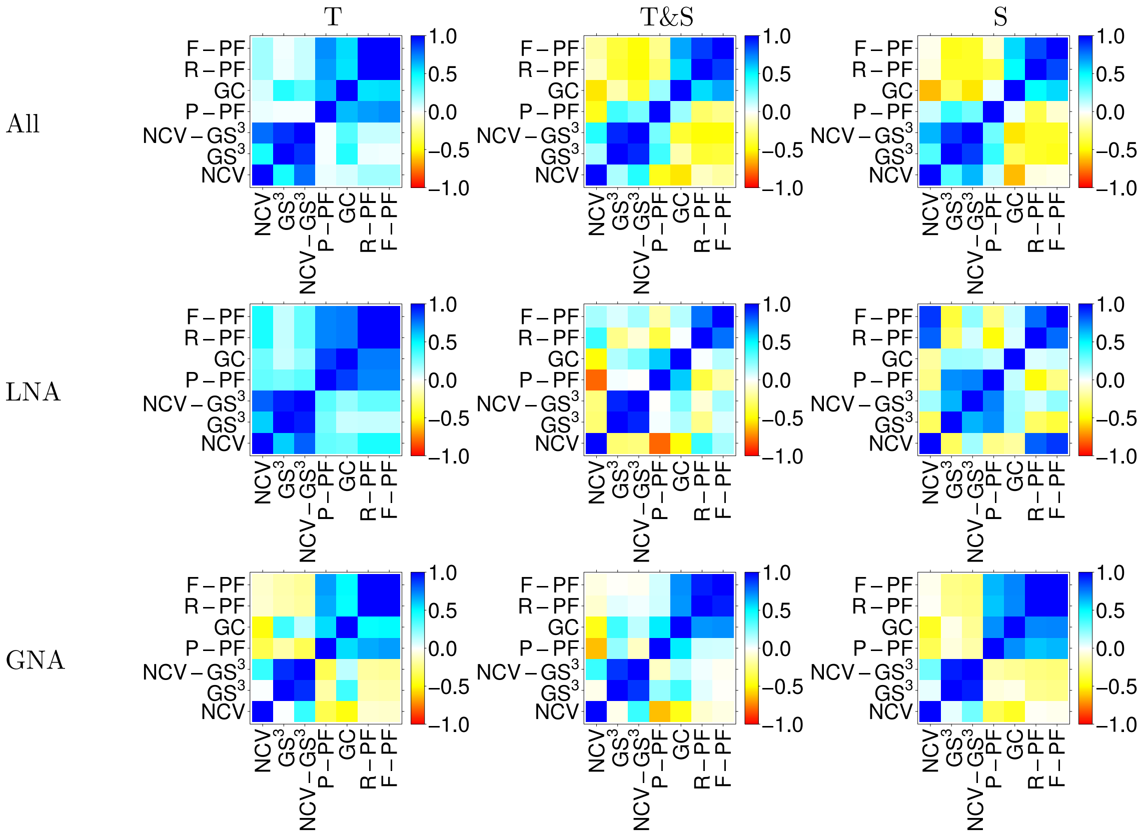


Figure S10: Pairwise relationships (Pearson correlations) between the three topological and four biological alignment quality measures over all network alignments, i.e., both LNA and GNA combined (top), all LNAs (horizontal middle), and all GNAs (bottom), for networks with unknown true node mapping from four different species (i.e., yeast, fly, worm and human) containing four different types of PPIs (ie., $Y2H_1$, $Y2H_2$, PHY_1, and PHY_2), for T (left), T&S (vertical middle), and S (right).

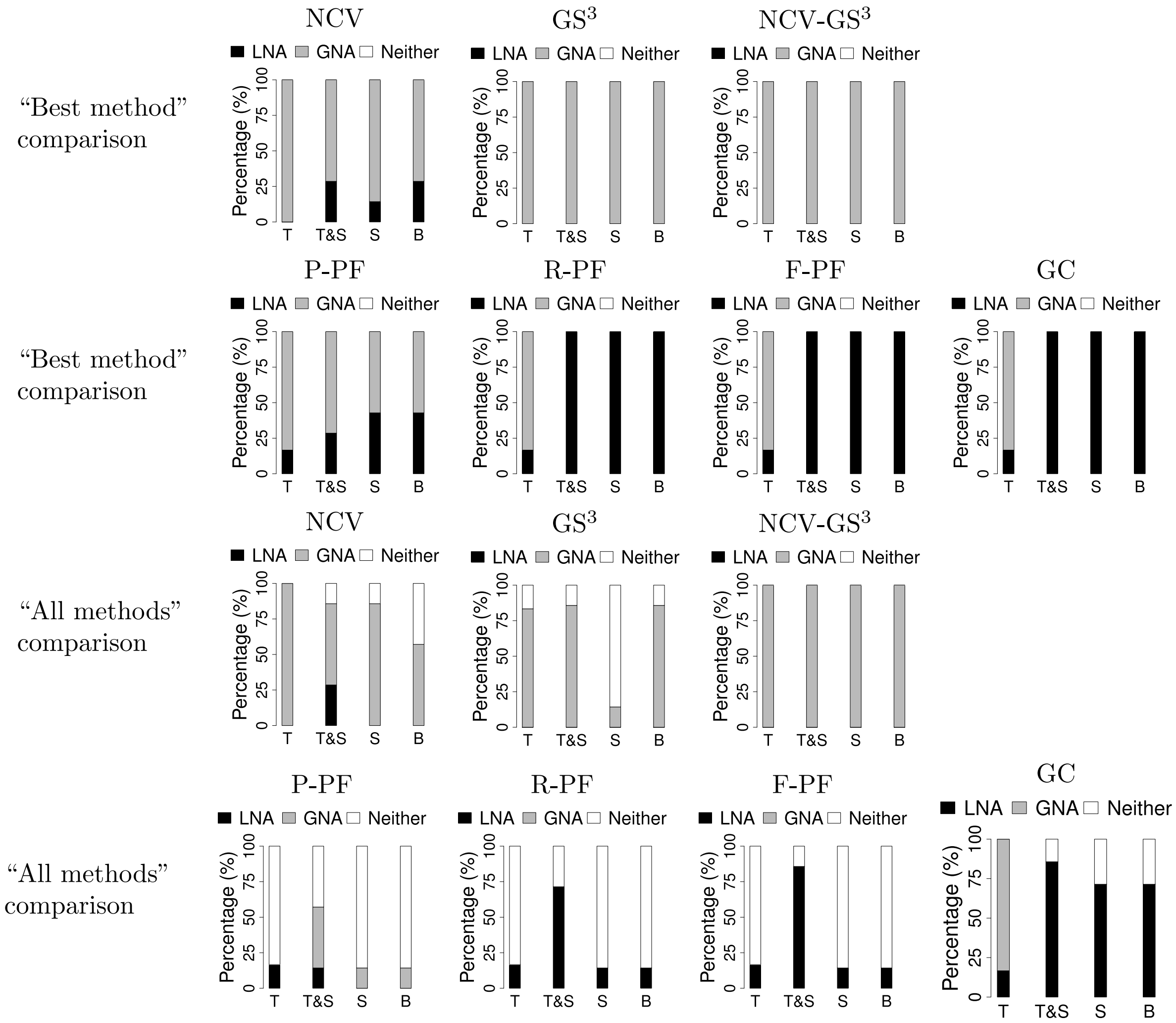


Figure S11: Overall comparison of LNA and GNA for networks with unknown true node mapping with respect to "best method" and "all methods" comparisons, for T, T&S, S, and B. The results are broken down for each of the seven topological (i.e., NCV, GS^3, and NCV-GS^3) and biological (i.e., P-PF, R-PF, F-PF, and GC) alignment quality measures. Each bar shows the percentage of the aligned network pairs for which LNA is superior (black), GNA is superior (grey), or neither LNA nor GNA is superior (white).

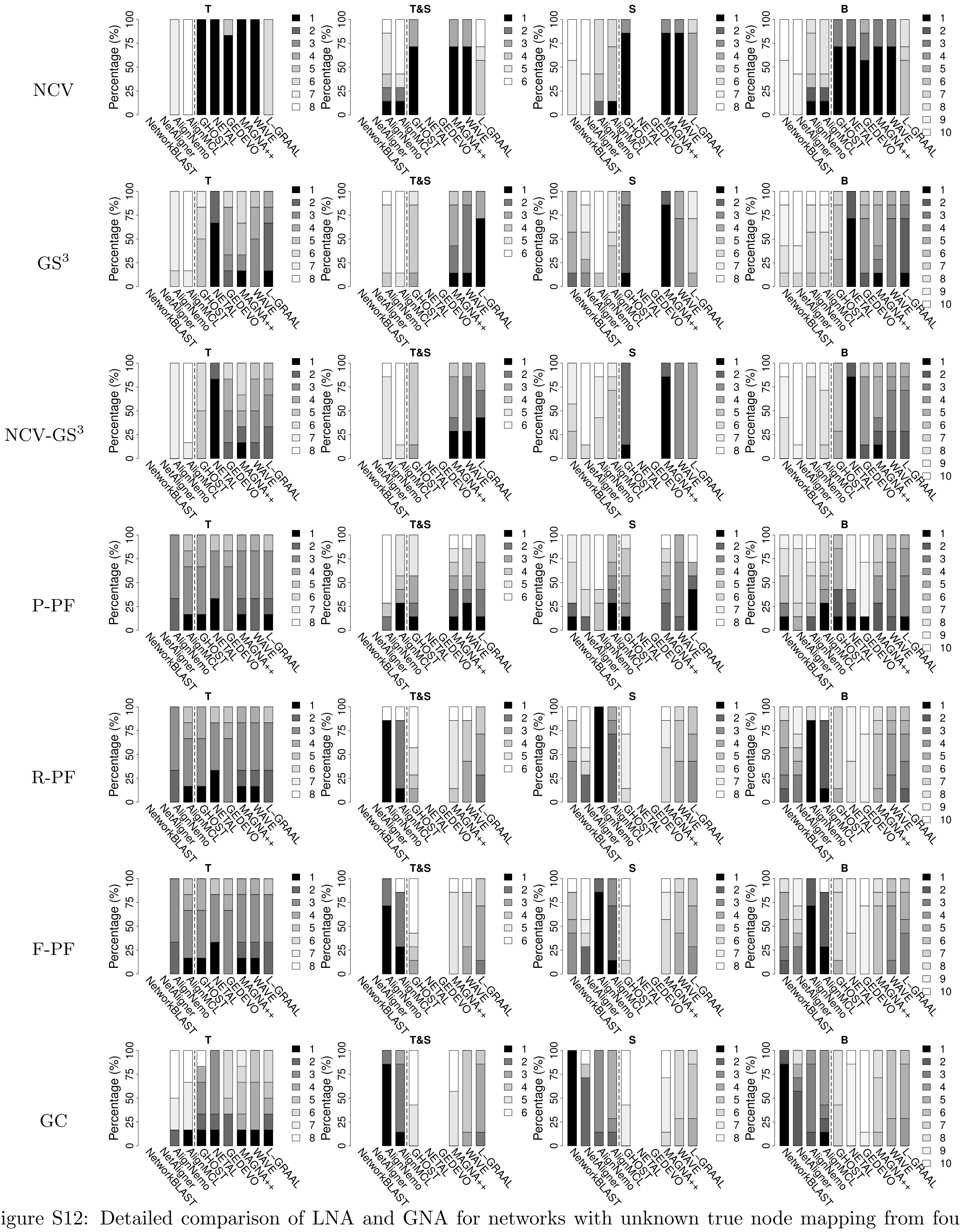


Figure S12: Detailed comparison of LNA and GNA for networks with unknown true node mapping from four different species (i.e., yeast, fly, worm and human) containing four different types of PPIs (ie., $Y2H_1$, $Y2H_2$, PHY_1, and PHY_2) with respect to the seven topological and biological measures, for T, T&S, S, and B. Each bar shows the percentage of the aligned network pairs for which the given NA method performs as the 1^{st} best (1), 2^{nd} best (2), 3^{rd} best (3), etc. among all of the NA (LNA or GNA) methods considered in our study with respect to the corresponding type of information used in NCF and the corresponding alignment quality measure. A missing bar indicates that the given NA method cannot use the corresponding type of information in NCF and thus no result is produced.

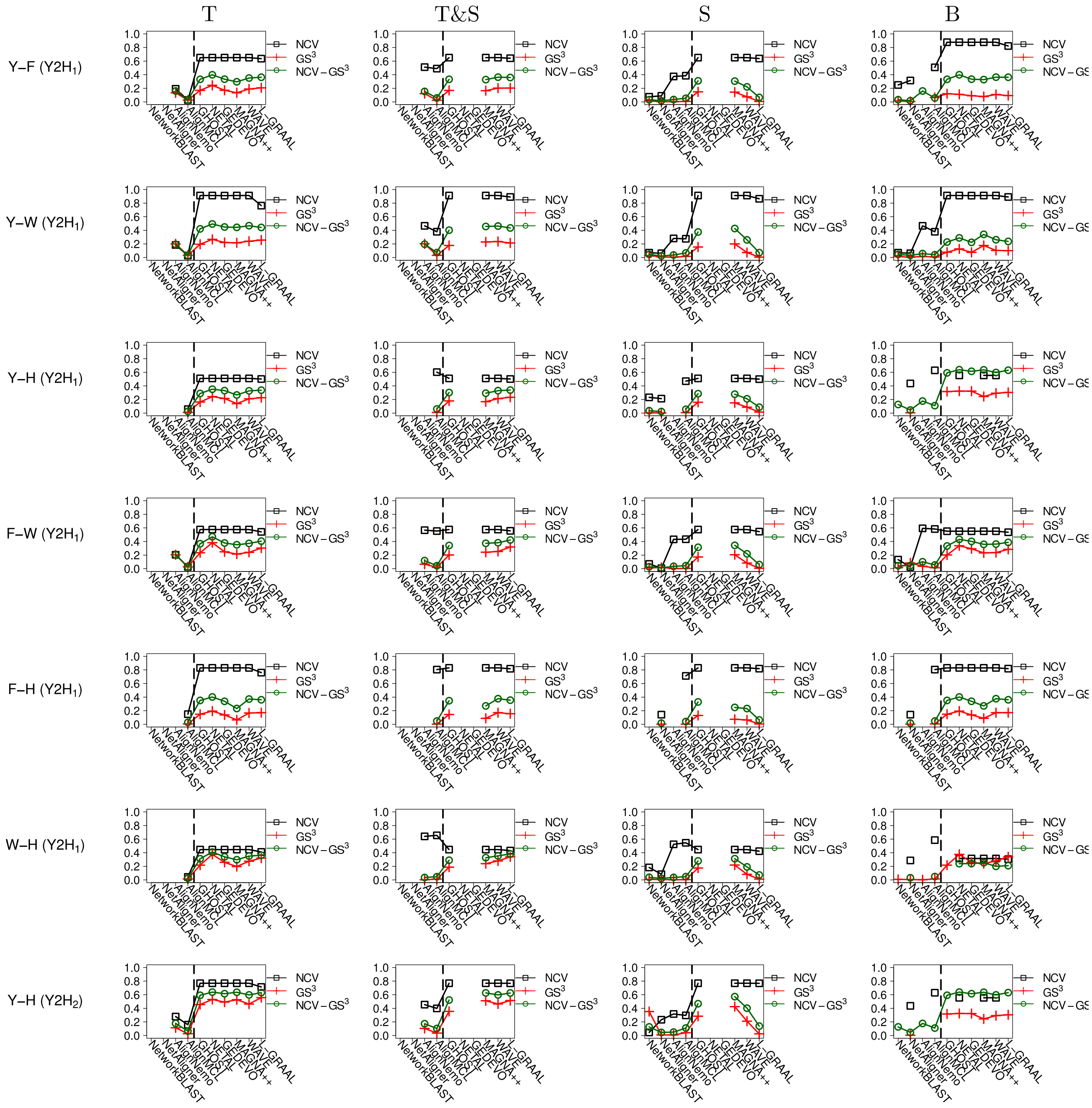


Figure S13: Topological alignment quality for each NA method considered in our study, each pair of networks with unknown true node mapping (denoted by X-Y (Z), where X and Y represent two aligned species and Z represents the PPI network type), and each of T, T&S, S, and B, respectively (part 1 of 2). Y, F, W, and H denote yeast, fly, worm, and human, respectively.

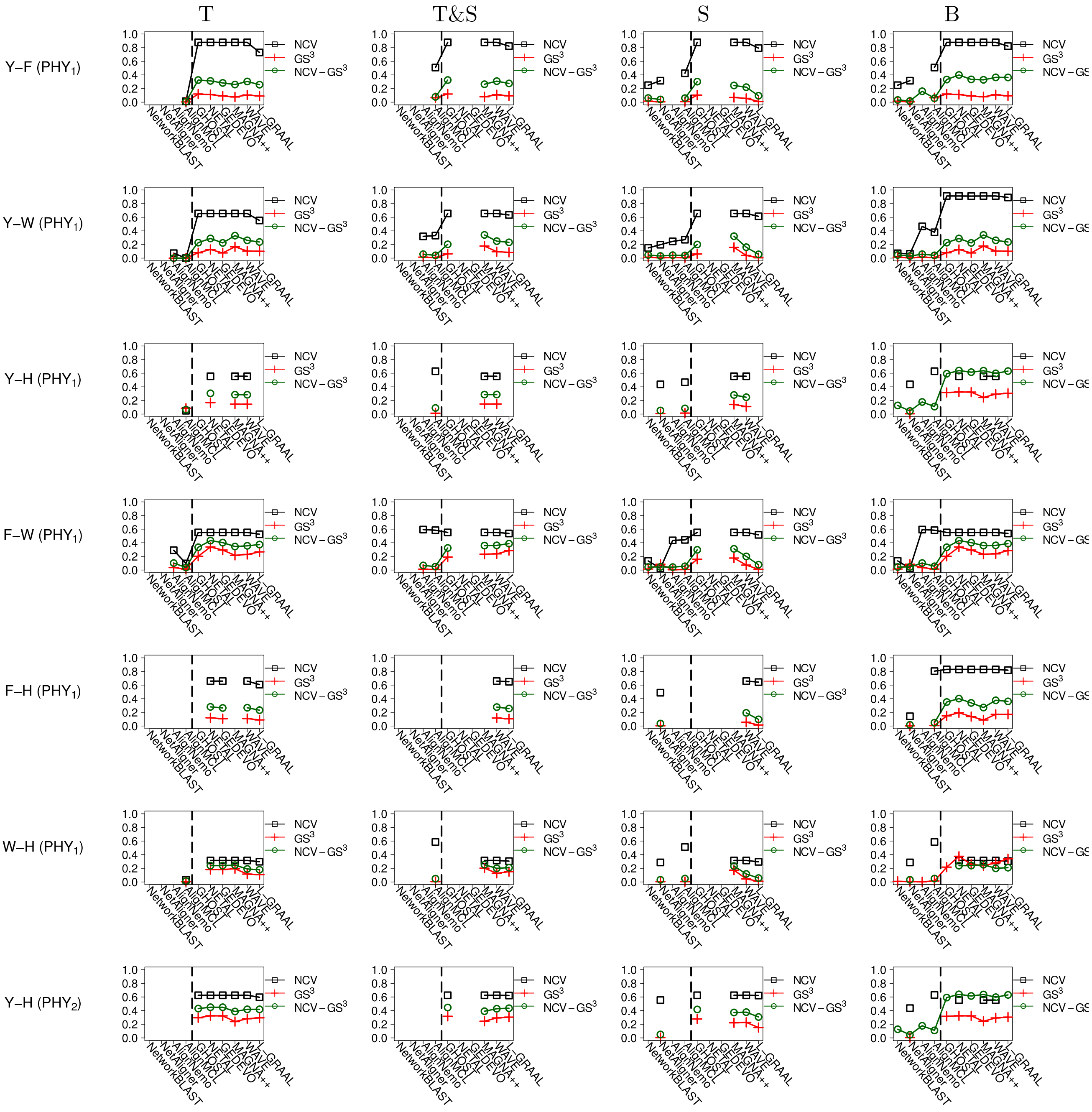


Figure S14: Topological alignment quality for each NA method considered in our study, each pair of networks with unknown true node mapping (denoted by X-Y (Z), where X and Y represent two aligned species and Z represents the PPI network type), and each of T, T&S, S, and B, respectively (part 2 of 2). Y, F, W, and H denote yeast, fly, worm, and human, respectively.

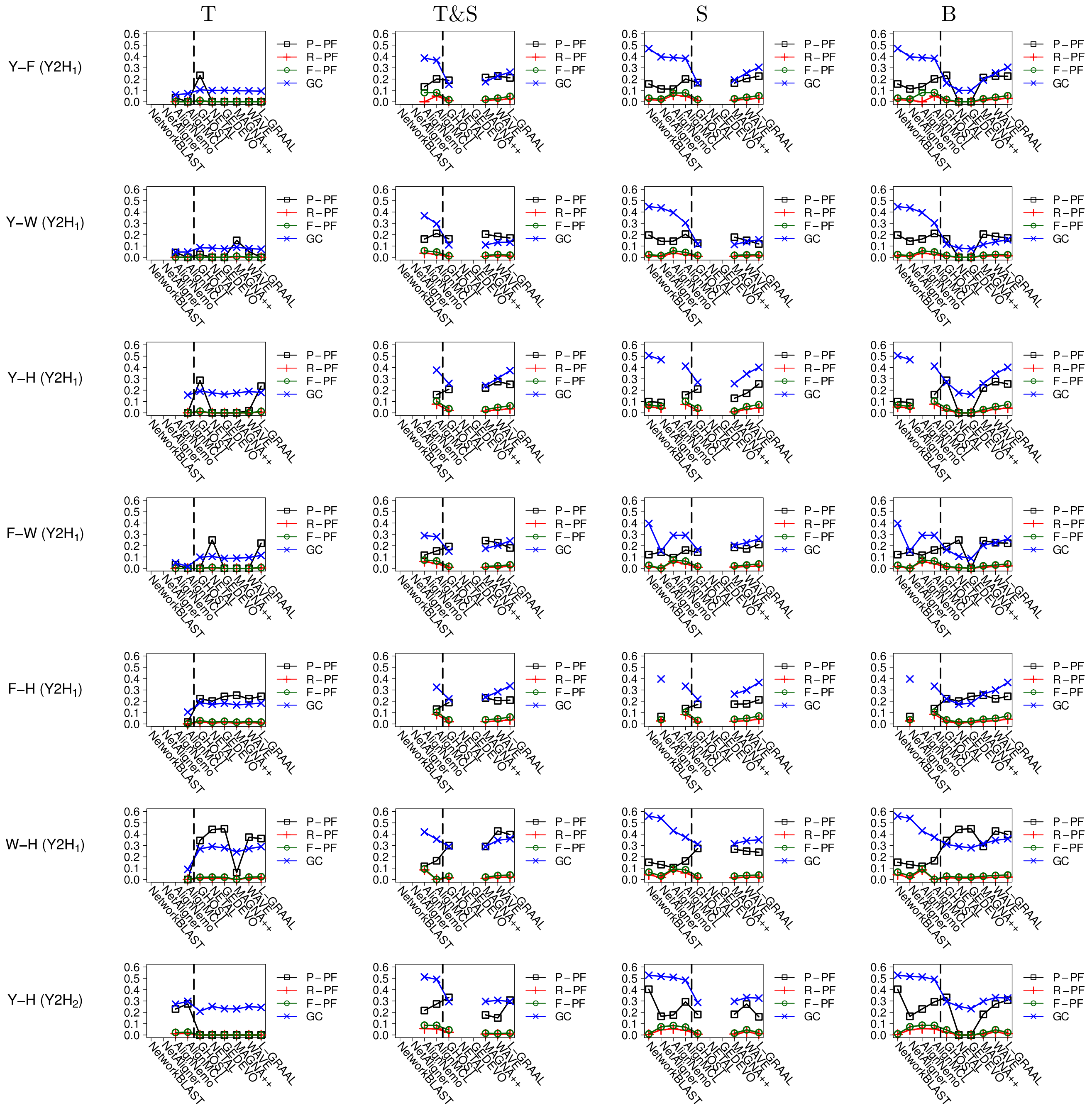


Figure S15: Biological alignment quality for each NA method considered in our study, each pair of networks with unknown true node mapping (denoted by X-Y (Z), where X and Y represent two aligned species and Z represents the PPI network type), and each of T, T&S, S, and B, respectively (part 1 of 2). Y, F, W, and H denote yeast, fly, worm, and human, respectively.

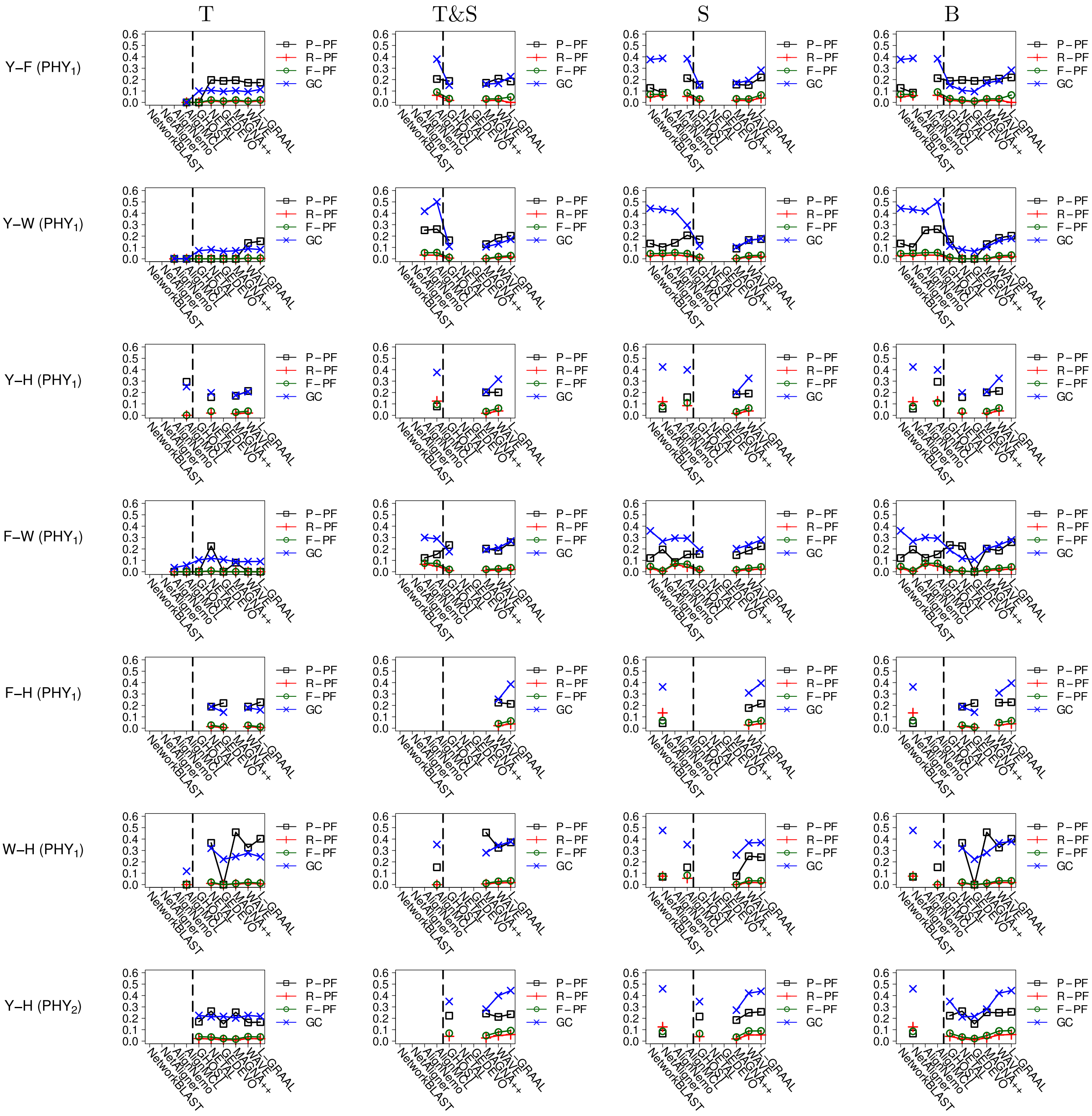


Figure S16: Biological alignment quality for each NA method considered in our study, each pair of networks with unknown true node mapping (denoted by X-Y (Z), where X and Y represent two aligned species and Z represents the PPI network type), and each of T, T&S, S, and B, respectively (part 2 of 2). Y, F, W, and H denote yeast, fly, worm, and human, respectively.

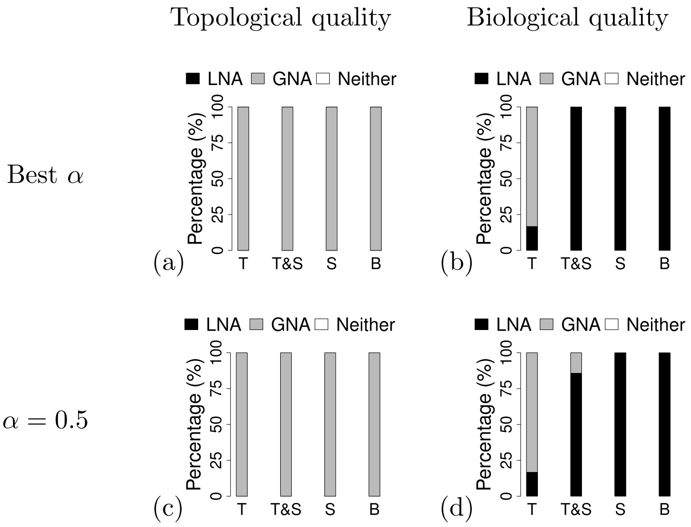


Figure S17: Overall comparison of LNA and GNA when using the best α value between 0.1 and 0.9 (panels **(a)** and **(b)**) or the α value of 0.5 (panels **(c)** and **(d)**), demonstrating the robustness of the results to the choice of the α parameter value. These representative results are for the *"best method"* comparison, for networks with unknown true node mapping, with respect to either the topological NCV-GS^3 measure (panels **(a)** and **(c)**) or the biological F-PF measure (panels **(b)** and **(d)**). Each bar shows the percentage of the aligned network pairs for which LNA is superior (black), GNA is superior (grey), or neither LNA nor GNA is superior (white).

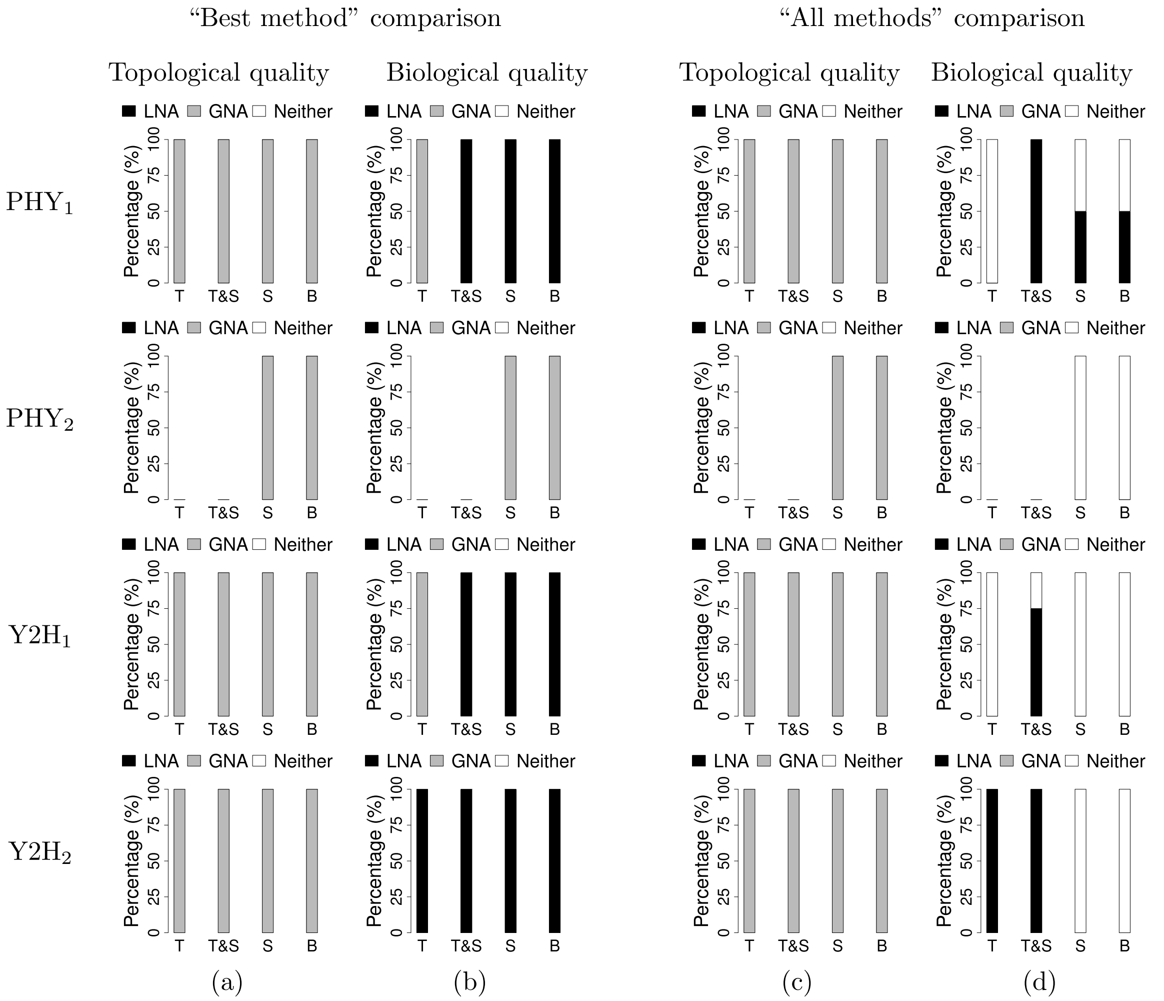


Figure S18: Overall comparison of LNA and GNA for networks with unknown true node mapping from four different species (i.e., yeast, fly, worm and human) containing different types of PPIs (i.e., Y2H$_1$, Y2H$_2$, PHY$_1$, and PHY$_2$) with respect to **(a)** the *"best method"* comparison and the topological NCV-GS^3 measure, **(b)** the *"best method"* comparison and the biological F-PF measure, **(c)** the *"all methods"* comparison and the topological NCV-GS^3 measure, and **(d)** the *"all methods"* comparison and the F-PF biological measure. Results are shown for T, T&S, S, and B. Each bar shows the percentage of the aligned network pairs for which LNA is superior (black), GNA is superior (grey), or neither LNA nor GNA is superior (white).

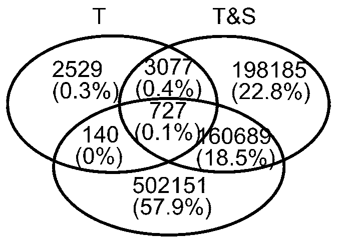
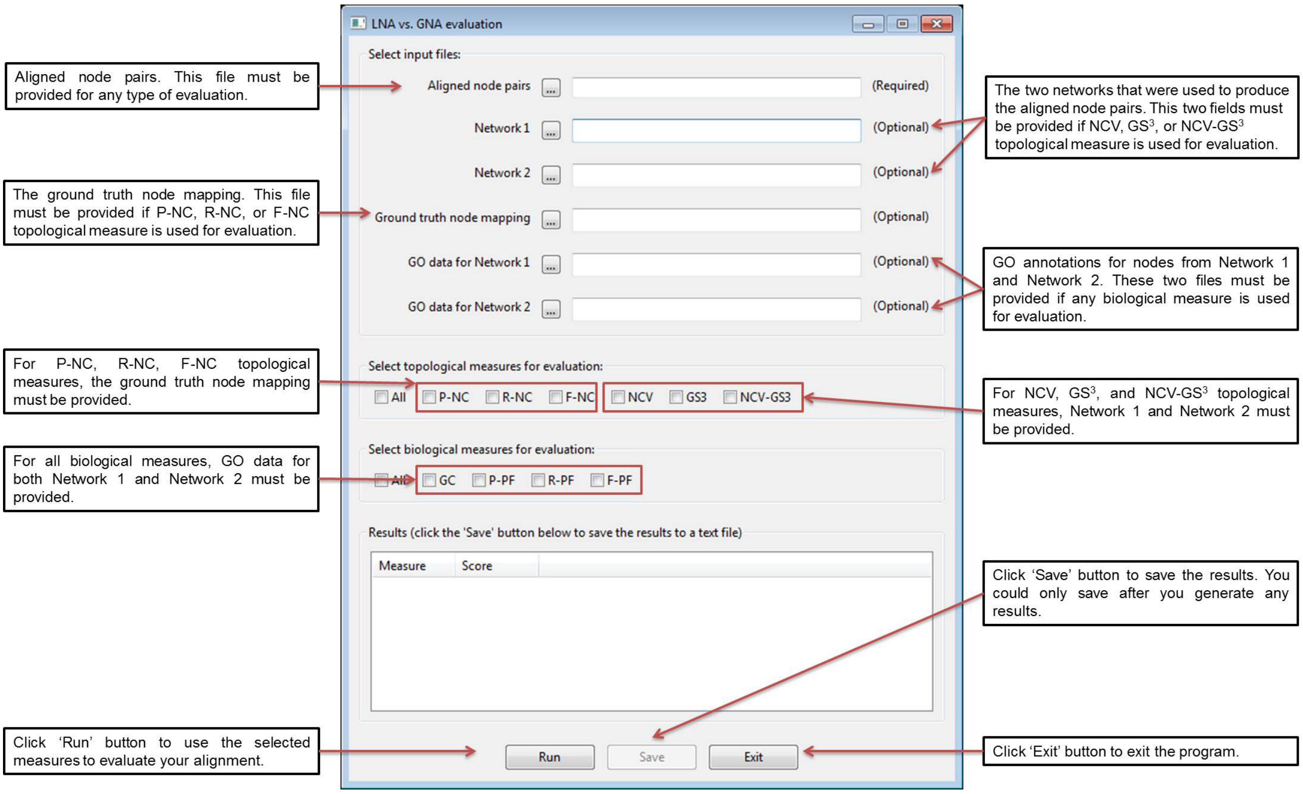


Figure S19: Overlap of unique novel protein function predictions between T, T&S, and S for LNA.


LNA vs. GNA evaluation
Select input files:
Aligned node pairs
(Required)
Network 1
(Optional)
Network 2
(Optional)
Ground truth node mapping
(Optional)
GO data for Network 1
(Optional)
GO data for Network 2
(Optional)
Select topological measures for evaluation:
□ All □ P-NC □ R-NC □ F-NC □ NCV □ GS3 □ NCV-GS3
Select biological measures for evaluation:
□ All □ GC □ P-PF □ R-PF □ F-PF
Results (click the 'Save' button below to save the results to a text file)
Measure
Score
Run
Save
Exit

Aligned node pairs. This file must be provided for any type of evaluation.

The ground truth node mapping. This file must be provided if P-NC, R-NC, or F-NC topological measure is used for evaluation.

For P-NC, R-NC, F-NC topological measures, the ground truth node mapping must be provided.

For all biological measures, GO data for both Network 1 and Network 2 must be provided.

Click 'Run' button to use the selected measures to evaluate your alignment.

The two networks that were used to produce the aligned node pairs. This two fields must be provided if NCV, GS^3, or NCV-GS^3 topological measure is used for evaluation.

GO annotations for nodes from Network 1 and Network 2. These two files must be provided if any biological measure is used for evaluation.

For NCV, GS^3, and NCV-GS^3 topological measures, Network 1 and Network 2 must be provided.

Click 'Save' button to save the results. You could only save after you generate any results.

Click 'Exit' button to exit the program.

Figure S20: A screen shot of the GUI interface of our NA evaluation framework.

Supplementary Tables

Species	PPI type/confidence	# of nodes in LCC	# of edges in LCC
Yeast	PHY$_1$	6,168	82,368
	PHY$_2$	3,768	13,654
	Y2H$_1$	3,427	11,348
	Y2H$_2$	744	966
Fly	PHY$_1$	7,887	36,285
	PHY$_2$	25	25
	Y2H$_1$	7,097	23,370
	Y2H$_2$	11	10
Worm	PHY$_1$	3,273	5,655
	PHY$_2$	223	161
	Y2H$_1$	3,110	5,330
	Y2H$_2$	195	138
Human	PHY$_1$	16,061	157,650
	PHY$_2$	8,283	19,697
	Y2H$_1$	9,996	39,984
	Y2H$_2$	1,191	1,567

Table S1: The sizes of the largest connected components (LCCs) of the networks with unknown true node mapping. The four networks shown in grey are not further considered in our study due to their limited size.

	PHY$_1$						PHY$_2$	Y2H$_1$						Y2H$_2$
NA method	Y-F	Y-W	Y-H	F-W	F-H	W-H	Y-H	Y-F	Y-W	Y-H	F-W	F-H	W-H	Y-H
NetworkBLAST	-	-	-	-	-	-	-	-	-	-	-	-	-	-
NetAligner	-	-	-	-	-	-	-	-	-	-	-	-	-	-
AlignNemo	X		X		X	X	X			X		X	X	
AlignMCL					X		X							
GHOST			X		X	X								
NETAL														
GEDEVO			X											
MAGNA++														
WAVE														
L-GRAAL			X											

(a)

	PHY$_1$						PHY$_2$	Y2H$_1$						Y2H$_2$
NA method	Y-F	Y-W	Y-H	F-W	F-H	W-H	Y-H	Y-F	Y-W	Y-H	F-W	F-H	W-H	Y-H
NetworkBLAST	-	-	-	-	-	-	-	-	-	-	-	-	-	-
NetAligner	-	-	-	-	-	-	-	-	-	-	-	-	-	-
AlignNemo	X		X		X	X	X			X		X		
AlignMCL					X		X							
GHOST			X		X	X								
NETAL	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GEDEVO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MAGNA++														
WAVE														
L-GRAAL			X											

(b)

	PHY$_1$						PHY$_2$	Y2H$_1$						Y2H$_2$
NA method	Y-F	Y-W	Y-H	F-W	F-H	W-H	Y-H	Y-F	Y-W	Y-H	F-W	F-H	W-H	Y-H
NetworkBLAST			X		X	X	X					X		
NetAligner														
AlignNemo	X		X		X	X	X			X		X		
AlignMCL					X		X							
GHOST			X		X	X								
NETAL	-	-	-	-	-	-	-	-	-	-	-	-	-	-
GEDEVO	-	-	-	-	-	-	-	-	-	-	-	-	-	-
MAGNA++														
WAVE														
L-GRAAL			X											

(c)

Table S2: The completion status of each NA method considered in our study for each network pair, each PPI type, and each of **a)** T, **b)** T&S, and **c)** S. The 'X' character indicates that the given NA method is unable to successfully finish due to high computational complexity or memory errors. The '-' character indicates that the given NA method cannot use the corresponding type of information in NCF. An empty cell indicates that the given NA method is able to complete successfully. Y, F, W, and H denote yeast, fly, worm, and human, respectively.

Aligners	Parameters

NetworkBLAST	no_species=2
	beta=0.9
	blast_th=1e-10
	true_factor0=0.5
	true_factor1=0.5

NetAligner	#main config file
	predict_likely_conserved_interactions=1
	vertex_probability_threshold=0.0
	edge_probability_threshold=0.3
	num_randomizations=10000
	#vertex/interaction conservation probability config file
	cvp_max_posterior_probability=1
	cvp_create_plots=0
	cicp_default_interaction_reliability=0.9
	cicp_default_interaction_conservation_pro-bability=0.0
	cicp_max_posterior_probability=0.9
	cicp_convergence_threshold=1e-10
	cicp_num_bins=100
	cicp_null_model_size=1000000
	cicp_create_plots=0

AlignNemo	No user-provided parameters
AlignMCL	No user-provided parameters
GHOST	beta=1e^{10} alpha=the values of the α parameter reported in Section 2.3 in the main paper

NETAL	a=0.0001
	b=0
	c=1
	i=2

GEDEVO	pop=500
	maxsame=3000
	no-prematch undirected

MAGNA++	m=S3
	p=15000
	n=2000
	f=1
	a=0

WAVE	No user-provided parameters
L-GRAAL	Default parameter values

Table S3: Parameter values used in each NA method considered in our study. We report parameter values that we use for all ten LNA and GNA methods. We set all methods' parameters to their recommended values according to the original publications or documentation of the methods, unless otherwise noted. For the meanings of the reported parameters, refer to the corresponding publications and documentations.

Type	NA method	NCF(T)	NCF(S)
LNA	NetworkBLAST	-	E-value scores
	NetAligner	-	E-value scores
	AlignNemo	GDV-similarity	Normalized E-value scores
	AlignMCL	GDV-similarity	Normalized E-value scores
GNA	GHOST	Spectral signature similarity	Normalized E-value scores
	NETAL	Neighborhood similarity	-
	GEDEVO	GDV-similarity	-
	MAGNA++	GDV-similarity	Normalized E-value scores
	WAVE	GDV-similarity	Normalized E-value scores
	L-GRAAL	GDV-similarity	Normalized E-value scores

Table S4: The topology-based (T) and sequence-based (S) NCFs used by each NA method considered in our study. The '-' character indicates that the given information cannot be used by the given method in its NCF.

References

[1] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, IJCAI'95, pages 448–453, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.