IGLOO: Integrating global and local biological network alignment

Lei Meng
Department of Computer Science and Engineering, ECK Institute for Global Health, Interdisciplinary Center for Network Science and Applications, and Wireless Institute
University of Notre Dame

Joseph Crawford
Department of Computer Science and Engineering, ECK Institute for Global Health, and Interdisciplinary Center for Network Science and Applications,
University of Notre Dame

Aaron Striegel
Department of Computer Science and Engineering and Wireless Institute
University of Notre Dame

Tijana Milenković*
Department of Computer Science and Engineering, ECK Institute for Global Health, and Interdisciplinary Center for Network Science and Applications,
University of Notre Dame

ABSTRACT

Analogous to genomic sequence alignment, biological network alignment (NA) aims to find regions of similarities between molecular networks (rather than sequences) of different species. NA can be either local (LNA) or global (GNA). LNA aims to identify highly conserved common subnetworks, which are typically small, while GNA aims to identify large common subnetworks, which are typically suboptimally conserved. We recently showed that LNA and GNA yield complementary results: LNA has high functional but low topological alignment quality, while GNA has high topological but low functional alignment quality. Thus, we propose IGLOO, a new approach that integrates GNA and LNA in hope to reconcile the two. We evaluate IGLOO against state-of-the-art LNA (NetworkBLAST, NetAligner, AlignNemo, and AlignMCL) and GNA (GHOST, NETAL, GEDEVO, MAGNA++, WAVE, and L-GRAAL) methods. We show that IGLOO allows for a trade-off between topological and functional alignment quality better than the existing LNA and GNA methods considered in our study.

1. INTRODUCTION

Large amounts of protein-protein interaction (PPI) data have become available due to advancements of high throughput biotechnologies for data collection [2, 3]. In PPI networks, nodes are proteins and edges correspond to physical interactions between the proteins. Network alignment (NA) of PPI data across species is gaining importance. This is because NA aims to find a good node mapping between PPI networks of different species that identifies topologically and functionally similar (i.e., conserved) network regions [9]. As such, NA can guide the transfer of biological knowledge from well- to poorly-studied species between conserved network regions. Consequently, NA is expected to lead to new discoveries in evolutionary biology. While we focus on NA in the domain of computational biology, NA and thus our work has applications in other domains, such as online social networks [18], pattern recognition [6, 29], and language processing [1].

The challenge arises from the fact that NA is computationally intractable, since the underlying subgraph isomorphism problem, which determines if a network is an exact subgraph of another network, is NP-complete. Therefore, efficient heuristic approaches are needed to solve the NA problem approximately.

There exists two types of NA methods: local network alignment (LNA) and global network alignment (GNA). LNA aims to find a many-to-many node mapping (i.e., a node can be mapped to one or more nodes from the other network) between networks of different species that identifies small but highly conserved subnetworks (Figure 1 (a)) [4, 11, 17, 20, 24]. On the other hand, GNA aims to find a one-to-one (injective) node mapping (i.e., every node in the smaller network is mapped to exactly one unique node in the larger network) that maximizes overall similarity of the compared networks, which often results in suboptimal conservation in local network regions (Figure 1 (b)) [5, 7, 12–14, 19, 21–23, 25–28].

We recently showed that LNA and GNA produce complementary results, especially when gene or protein sequence information is included on top of PPI network topological information during the alignment construction process [15]. This is because LNA and GNA are designed to "optimize" different types of alignment quality: LNA typically aims to "optimize" functional alignment quality, while GNA aims to "optimize" topological alignment quality. Intuitively, an alignment is of good functional quality if the aligned nodes perform similar biological functions, and it is of good topo-

In proceedings of the 12th International Workshop on Mining and Learning with Graphs (MLG), August 14, 2016, San Francisco, CA, USA.

*Corresponding author (e-mail: tmilenko@nd.edu)

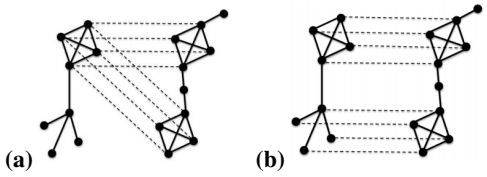


Figure 1: Illustration of (a) LNA and (b) GNA, taken from [15].

logical quality if large and dense network regions are aligned to each other. It is very challenging to design an NA (LNA or GNA) method that is of high topological as well as functional quality, since the topological versus functional fit between aligned networks conflict to a larger extent than previously realized [5, 15, 21].

Thus, we propose IGLOO, a new NA method that integrates algorithmic components from both LNA and GNA in the hope of reconciling the two NA types. That is, IGLOO aims to "inherit" the high functional quality of LNA and the high topological quality of GNA. IGLOO's input are two networks and pairwise similarity scores between their nodes, where the scores are computed via some node cost function. This is the same as the input of existing LNA and GNA methods. Then, IGLOO produces an alignment in a similar way as GNA does, by first identifying a high-scoring seed alignment and then expanding around the seed via an alignment strategy. However, IGLOO differs from GNA as follows. While GNA uses as its seed just a single pair of nodes from the compared networks, IGLOO's seed is a local alignment (or a part of it) of high functional quality generated by an existing LNA method. Given the seed alignment, IGLOO expands around it via an existing alignment strategy to increase topological quality of the alignment. The difference between IGLOO and LNA is that IGLOO builds on top of the given local alignment to improve its topological quality. The difference between IGLOO and GNA is that IGLOO uses as the seed a local alignment of high functional quality (or a part of it) rather than just a single node pair, in order to improve functional quality of GNA. As a result, IGLOO's alignment is local in the sense that it allows for many-to-many mapping between nodes of the two compared networks, just as LNA does. Yet, its alignment is global in the sense that it allows for mapping large conserved subgraphs across the compared networks, just as GNA does.

In this paper, we comprehensively evaluate IGLOO against existing NA methods both topologically and functionally. We study the following state-of-the-art LNA methods: NetworkBLAST [24], NetAligner [20], AlignNemo [4], and AlignMCL [17]. We study the following state-of-the-art GNA methods: GHOST [21], NETAL [19], GEDEVO [12], MAGNA++ [28], WAVE [26], and L-GRAAL [14]. We evaluate all methods on four sets of PPI networks of varying interaction types and confidence levels, with respect to proven measures of topological and functional alignment quality (node coverage combined with edge conservation, and precision and recall of protein function prediction combined into F-score, respectively) [15]. We show that IGLOO produces a better trade-off between topological and functional alignment quality than the existing LNA and GNA methods. Namely, across all NA methods and network pairs, IGLOO is comparable or superior to the existing methods both functionally and topologically in 62% of all cases.

2. METHODS

2.1 IGLOO algorithm

IGLOO aligns two networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where V_i is the set of vertices in graph i, and E_i the set of edges in graph i. IGLOO's alignment is a list of aligned node pairs, where a node can appear in multiple aligned pairs. When producing its alignment, IGLOO aims to "inherit" the advantages of both LNA and GNA, i.e., the high functional quality of LNA and the high topological quality of GNA. IGLOO achieves this by using the output of an LNA method (i.e., its local alignment that is of high functional quality, or a part of this alignment) as the seed within a GNA alignment strategy that will then expand the alignment around the seed (as GNA typically does) to improve its topological quality. Typical GNA uses as the seed only a single highly similar pair of nodes from the compared networks. Instead, we vary the size of the seed from the entire local alignment (i.e., 100% of it) on one extreme (we expect this version of IGLOO to resemble LNA the most) to only a single node pair (i.e., 0% of the local alignment) as the other extreme (we expect this version of IGLOO to resemble GNA the most), with several in-between-the-extremes versions of IGLOO that use as the seed a certain portion (between 100% and 0%) of the local alignment (which we expect will balance between high functional quality of LNA and high biological quality of GNA). To achieve this, IGLOO requires four algorithmic steps:

1. Use a state-of-the-art LNA approach to find a local alignment, i.e., a set of small conserved subnetworks of high functional quality, which becomes the initial alignment.
2. Compute a new node cost function (as described below) that will be used in the following steps to modify (decrease or increase) the current alignment.
3. Iteratively decrease the size of the current alignment by greedily removing (as explained below) the aligned node pair with the lowest similarity score one at a time, until the user-specified alignment size is reached. This step is used to balance the contributions of the LNA and the GNA during the alignment process. Namely, the fewer node pairs are removed in step 3 from the local alignment resulting from step 1 (i.e., the larger the seed size, per our discussion above), the more similar IGLOO is to LNA (this corresponds to the first extreme discussed above); the more node pairs are removed (i.e., the smaller the seed size), the more similar IGLOO is to GNA (this corresponds to the second extreme discussed above). The resulting alignment becomes the new alignment.
4. Greedily expand (as explained below) around the current alignment by iteratively adding to the alignment node pairs with the highest similarity that have remained unaligned up to this point, until no more node pairs can be added, meaning that each node in at least one of the two compared networks has been aligned to some node(s). This step is performed in order to find large conserved subnetworks of high topological quality, just as GNA does. The resulting alignment becomes the final alignment and the aligned node pairs (i.e., a many-to-many node mapping) are returned as IGLOO's output.

Next, we detail each step.

Step 1: Searching for small conserved subnetworks of high functional quality

Since IGLOO must first generate a local alignment that has high functional quality, IGLOO begins by using an LNA method. In this step, unless otherwise noted, all default parameters (including node cost function) of the given existing LNA method are used. In order to evaluate the robustness of IGLOO to the choice of LNA methods, we use within IGLOO each of AlignMCL and AlignNemo, two of the best LNA methods [15]. These methods rely on parameter α, which balances between the amount of topological versus sequence information used in node cost function during the alignment construction process. For each method, we use the α value that results

in the highest functional quality of its local alignment, when varying α from 0 (corresponding to using only sequence information to compute node cost function) to 1 (corresponding to using only topological information to compute node cost function) in increments of 0.1 [15]. For IGLOO under AlignMCL, this results in $\alpha = 0.1$ for any network pair. For IGLOO under AlignNemo, this results in $\alpha = 0$ for any network pair except yeast-worm ($Y2H_1$) and yeast-fly ($Y2H_1$), for which the best value of α is 0.1. Note that we describe the network data that we use in Section 2.2.

Step 2: Computing node cost function to modify (decrease or increase) the current alignment in steps 3 and 4

We compute node cost function for steps 3 and 4 in the same way as NETAL [19] does, by combining node topological similarity (denoted as TS), node sequence similarity (denoted as SS), and interaction score (denoted as IS): $S = \beta(\alpha \cdot TS + (1-\alpha) \cdot SS) + (1-\beta) \cdot IS$. The first measure (i.e., TS) quantifies topological similarities between nodes from different networks using graphlet degree vector similarity (GDV-similarity) [15, 16]. The second measure (i.e., SS) quantifies sequence similarities between nodes from different networks using normalized E-value [15]. The third measure (i.e., IS) quantifies the similarity between two nodes as the number of edges that would be conserved if the two nodes were to be added next to the current alignment. The reason that we mimic NETAL's node cost function is that compared to the existing GNA methods, NETAL results in the highest topological alignment quality, especially when using only topological information in node cost function (corresponding to $\alpha = 1$) [15]. Note that NETAL's original implementation can use only topological information in node cost function. For the purpose of IGLOO's development, we re-implement NETAL's node cost function to also allow for using sequence similarity in node cost function. Consequently, in terms of the α parameter, we tested both $\alpha = 0$ and $\alpha = 1$, and the results are similar. Hence, we simply choose $\alpha = 1$ as was done in the original NETAL study. In terms of the β parameter above, we use $\beta = 0.001$ because IGLOO relies on NETAL's node cost function and this value was suggested in the NETAL study [19].

Unlike SS and TS, which do not get updated throughout steps 3 and 4, IS needs to be updated in each iteration of each of these steps as IGLOO shrinks or expands the current alignment. When expanding the current alignment, IGLOO mimics NETAL to update IS. When shrinking the current alignment, IGLOO cannot mimic NETAL, since NETAL only expands but never shrinks the current alignment. Thus, we first discuss NETAL's strategy of updating IS and then comment on how IGLOO generalizes this strategy to update IS when both expanding and shrinking the current alignment.

For a node pair $i \in V_1$ and $j \in V_2$, NETAL computes its interaction score, $IS(i,j)$, as follows. Under the assumption that nodes i and j are to be aligned next and thus added to the current alignment, NETAL computes: 1) d_{ij}, the number of conserved edges that are incident to the node pair, 2) p_i, the expected value of the number of *candidate edges* (i.e., edges that are not aligned) that are incident to i and that will be conserved, and 3) p_j, the expected value of the number of candidate edges that are incident to j and that will be conserved. The three values are computed as follows. Let $N(x)$ be the neighbors of a node x. Initially, d_{ij} is set to zero since no conserved edges have formed yet, and all edges that are incident to i and j are candidate edges. Based on NETAL study, NETAL assumes that all candidate edges that are incident to a node x will be chosen to be conserved with equal probability, and the probability that each candidate edge (x,x') where $x' \in N(x)$ will be conserved if x is aligned to a random node is approximately $\frac{1}{|N(x')|}$. Let i' and j' be two nodes from $N(i)$ and $N(j)$, respectively. Since NETAL's assumption is that i and j are aligned, the probability that edge (i,i') will be conserved is $\frac{1}{|N(i')|}$. Similarly, the probability that edge (j,j') will be conserved is $\frac{1}{|N(j')|}$. Therefore, p_i can be measured by summing up the probabilities of all edges that are incident to i and that will be conserved: $p_i = \sum_{i' \in N(i)} \frac{1}{|N(i')|}$. Similarly, $p_j = \sum_{j' \in N(j)} \frac{1}{|N(j')|}$. After computing d_{ij}, p_i, and p_j, $IS(i,j)$ is computed using Equation 1. Since p_i and p_j are not greater than $|N(i)|$ and $|N(j)|$, respectively, $IS(i,j)$ is normalized by the maximum node degree over all nodes from any of the two compared networks (i.e., by $\max_{k \in V_1 \cup V_2}\{|N(k)|\}$),

$$IS(i,j) = \frac{\min\{\sum_{i' \in N(i)} \frac{1}{|N(i')|}, \sum_{j' \in N(j)} \frac{1}{|N(j')|}\}}{\max_{k \in V_1 \cup V_2}\{|N(k)|\}}. \quad (1)$$

After $IS(i,j)$ is computed for the first time, NETAL updates its value during each iteration of the alignment process from steps 3 and 4. Whenever two nodes $x \in V_1$ and $y \in V_2$ are aligned, NETAL updates $IS(i,j)$ as follows: 1) if $x \in N(i)$ and $y \in N(j)$, increase d_{ij} by one; otherwise, do not update d_{ij}; 2) if x is not aligned to any node from V_2 and $x \in N(i)$, decrease p_i by $\frac{1}{|N(x)|}$; otherwise do not update p_i; 3) if y is not aligned to any node from V_1 and $y \in N(j)$, decrease p_j by $\frac{1}{|N(y)|}$; otherwise, do not update p_j; and 4) recompute $IS(I,j)$ using Equation 2. For more details on how NETAL computes and updates IS, see [19].

$$IS(i,j) = \frac{d_{ij} + \min\{p_i, p_j\}}{\max_{k \in V_1 \cup V_2}\{|N(k)|\}} \quad (2)$$

Now, we go back to explaining how IGLOO computes and updates IS. When expanding the current alignment (steps 4) by adding a node pair to it, IGLOO initially computes IS based on the current alignment using Equation 1 and updates IS just as NETAL does. When shrinking the current alignment (step 3) by removing a node pair $x \in V_1$ and $y \in V_2$ from it, IGLOO performs the following modifications: 1) if $x \in N(i)$ and $y \in N(j)$, *decrease* d_{ij} by one; otherwise, do not update d_{ij}; 2) if x is aligned to y only and $x \in N(i)$, *increase* p_i by $\frac{1}{|N(x)|}$; otherwise do not update p_i; 3) if y is aligned to x only and $y \in N(j)$, *increase* p_j by $\frac{1}{|N(y)|}$; otherwise do not update p_j; and 4) recompute $IS(i,j)$ using Equation 2.

Step 3: Decreasing the size of the current alignment to balance between LNA and GNA

IGLOO shrinks the current alignment greedily. Specifically, in each iteration, the node pair with the lowest similarity is removed from the current alignment, and the IS is updated. The removal process terminates when the number of remaining aligned node pairs in the current alignment equals the user-specified alignment size t. We test five values of t to study its effect on alignment quality: 100%, 75%, 50%, 25%, and 0%. These five values of t will produce five alignments that IGLOO (i.e., its five versions) that will be expanded on in step 4. The five versions of IGLOO are: IGLOO 4, IGLOO 3, IGLOO 2, IGLOO 1 and IGLOO 0, respectively. IGLOO 4 takes the exact alignment produced by the given LNA method as the current alignment and expands around it, while IGLOO 0 does not use any of the local alignment and aligns the two networks from scratch, just as GNA does (Step 4). Therefore, we expect IGLOO 4 to be the most similar to LNA and IGLOO 0 the be the most similar to GNA, while the remaining versions of IGLOO will balance between high functional quality of LNA and high topological quality of GNA.

Step 4: Searching for large conserved subnetworks of high topological quality

IGLOO expands around the current alignment greedily to find large conserved subnetworks of high topological quality, similar to how GNA works. In each iteration, IGLOO adds to the current alignment the node pair from different networks that has remained unaligned up to that point and that has the highest node similarity score, and then IGLOO updates node cost function scores accordingly. Each of the nodes that are aligned cannot be used again in this expansion process. The expansion process stops when no more node pairs can be added to the alignment. IGLOO returns the latest current alignment as its final alignment. Note that any expansion (i.e., alignment) strategy [8, 10]) can be used in IGLOO's step 4, including our recent alignment strategy called WAVE [26]. We verified that using WAVE yields qualitatively identical results as does using the above described expansion strategy. Consequently, for brevity and simplicity, we leave out discussion of WAVE's results and instead focus on results of the above described strategy.

2.2 Data

We evaluate each NA (LNA, GNA, and IGLOO) method on four real-world PPI network sets from our recent study [15] containing interactions of different types and confidence levels: 1) only yeast two-hybrid physical PPIs, where each PPI is supported by at least one publication ($Y2H_1$), 2) only yeast two-hybrid physical PPIs, where each PPI is supported by at least two publications ($Y2H_2$), 3) all physical PPIs, where each PPI is supported by at least one publication (PHY_1), and 4) all physical PPIs, where each PPI is supported by at least two publications (PHY_2). Each network set contains four PPI networks of different species: *S. cerevisiae* (yeast), *D. melanogaster* (fly), *C. elegans* (worm), and *H. sapiens* (human). For each network, we use its largest connected component, just as in [15]. We do not include those pairs involving $Y2H_2$ and PHY_2 networks of worm and yeast, since these four networks are extremely small and sparse, with random-like topology.

AlignNemo is able to produce an alignment for six of the aforementioned network pairs (it cannot run for the other network pairs, for reasons discussed in [15]). Thus, since IGLOO is partly based on AlignNemo, in order to fairly evaluate IGLOO against the existing NA methods, we focus on the six network pairs that AlignNemo is able to run on. The network pairs are: yeast-fly ($Y2H_1$), yeast-worm ($Y2H_1$), worm-fly ($Y2H_1$), yeast-human ($Y2H_2$), yeast-worm (PHY_1), and fly-worm (PHY_1). The size of each network is shown in Table 1. For more details on each data set, see [15].

Table 1: Sizes of networks used in this study.

Network	# of nodes	# of edges
yeast ($Y2H_1$)	3,427	11,348
fly ($Y2H_1$)	7,094	23,356
worm ($Y2H_1$)	2,871	5,194
yeast ($Y2H_2$)	744	966
human ($Y2H_2$)	1,191	1,567
yeast (PHY_1)	6,168	82,368
fly (PHY_1)	7,885	36,271
worm (PHY_1)	3,003	5,501

2.3 Alignment quality measures

We evaluate each NA (LNA, GNA, and IGLOO) method in terms of both topological and functional alignment quality. We focus on node coverage (NCV) combined with the generalized symmetric substructure score (GS^3) measure of edge conservation as a measure of topological alignment quality, where the combined topological measure is denoted as NCV-GS^3. Also, we focus on precision (P-FP) and recall (R-PF) of protein function prediction combined into F-score as a measure of functional alignment quality, where the combined functional measure is denoted as F-PF. We use these measures because they are already proven evaluation criteria for both LNA and GNA that can compare the two fairly [15].

Intuitively, NCV-GS^3 quantifies the size of the given alignment in terms of the amount of both conserved nodes (NCV) and conserved edges (GS^3). Let f be an alignment between two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, and let $G_1'(V_1', E_1')$ and $G_2'(V_2', E_2')$ be subgraphs of G_1 and G_2 that are induced on node sets $f(V_2)$ and $f(V_1)$. NCV is the percentage of nodes from G_1 and G_2 that are also in G_1' and G_2' (i.e., $\frac{|V_1'|+|V_2'|}{|V_1|+|V_2|}$). GS^3 is the percentage of conserved edges out of the total of both conserved and non-conserved edges. NCV-GS^3 is the geometric mean of NCV and GS^3.

Before we define F-PF, we note that this measure is computed with respect to Gene Ontology (GO) gene-function annotation data [13]. We only use gene-GO term annotations that have been obtained experimentally. That is, we discard those functional annotations that have been obtained e.g., computationally via sequence alignment. We do this because the NA methods that we evaluate already use sequence information within node cost function when producing their alignments, and thus evaluating such alignments with respect to sequence-based functional annotations would lead to a circular argument, which is undesirable [13].

Now, we go back to defining F-PF. This measure quantifies how similar the aligned nodes are in terms of their functions [15]. To compute F-PF, we first hide proteins' true GO terms and then predict the proteins' GO terms based on GO terms of their aligned counterpart(s) [15]. Next we compute the P-PF and R-PF of the resulting predicted GO terms with respect to the true GO terms. Finally, F-PF is the harmonic mean of P-PF and R-PF.

For more details on the NCV-GS^3 and F-PF measures, see [15].

2.4 Parameters of the existing methods

Different α values (i.e., where parameter α balances between the amount of topological versus sequence information in the given method's node cost function) might result in different alignment quality. Thus, for each existing NA method and each network pair, we choose the α value that results in the best trade-off between topological and functional quality when varying α from 0 to 1 in increments of 0.1 [15]. We measure the trade-off between the two quality types (i.e., between NCV-GS^3 and F-PF) by computing their geometric mean. We report results only for the α value that results in the maximum geometric mean.

3. RESULTS AND DISCUSSION

We evaluate the considered NA methods in terms of both alignment quality (Section 3.1) and running time (Section 3.2).

3.1 Comparison in terms of alignment quality

Setup. Here, we show results for IGLOO when considering the total of 10 IGLOO versions: IGLOO 0–4 when each of AlignMCL and AlignNemo are used in step 1 of the IGLOO algorithm. When we compare the different methods (the existing LNA and GNA methods, and the 10 IGLOO versions), for a given network pair and a given existing NA method, we obtain results for four possible cases: 1) IGLOO is comparable or superior both topologically and functionally, meaning that at least one version of IGLOO is comparable or superior both topologically and functionally; 2) IGLOO

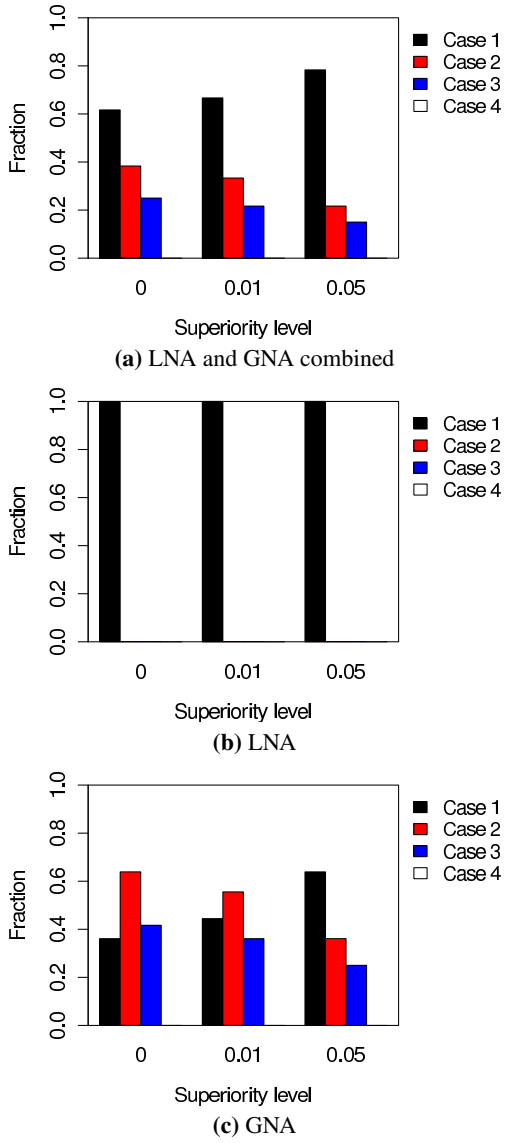
is comparable or superior only functionally but not topologically, meaning that no version of IGLOO is comparable or superior both topologically and functionally, and at least one version of IGLOO is comparable or superior only functionally but not topologically; 3) IGLOO is comparable or superior only topologically but not functionally, meaning that none of the versions of IGLOO are comparable or superior both topologically and functionally, and at least one version of IGLOO is comparable or superior only topologically but not functionally; and 4) IGLOO is inferior both topologically and functionally, meaning that all versions of IGLOO are inferior. Note that cases 2 and 3 could occur at the same time, since it is possible that some version of IGLOO is comparable or superior only topologically but not functionally, while another version is comparable or superior only functionally but not topologically. However, none of cases 1, 2, and 4, or cases 1, 3, and 4, can occur at the same time.

Overall comparison. Our findings are as follows. Overall, IGLOO is comparable or superior to the existing NA methods considered in our study (Figures 2 (a) and 3). Specifically, when considering all combinations of the existing NA methods and network pairs, in 62% of them, IGLOO is comparable or superior both topologically and functionally (case 1). In 38% of the combinations, IGLOO is comparable or superior only functionally but not topologically (case 2). In 25% of the combinations, IGLOO is comparable or superior only topologically but not functionally (case 3). IGLOO is never inferior both topologically and functionally (case 4). Note that 25% of the combinations are in the overlap of cases 2 and 3. When considering two given methods to be comparable if their alignment quality scores are within 1% or 5% of each other, IGLOO is even more comparable or superior both topologically and functionally, in up to 78% of all cases (Figure 2 (a)). That is, often, when the existing methods are superior to IGLOO, their superiority is only within 1% or 5% of IGLOO alignment quality. Equivalent results when considering only the five AlignMCL-based IGLOO versions and only the five AlignNemo-based IGLOO versions (as opposed to all 10 versions of IGLOO) are shown in Supplementary Figures S1 and S2, respectively.

Importantly, for case 2, whenever IGLOO is comparable or superior to the existing methods functionally but not topologically, or in other words whenever the existing methods outperform IGLOO topologically but not functionally, the topological superiority of the existing methods (in terms of NCV-GS^3) comes only from GS^3 but not NCV (Figure 4 (a)). Similarly, for case 3, whenever IGLOO is comparable or superior to the existing methods topologically but not functionally, or in other words whenever the existing methods outperform IGLOO functionally but not topologically, the functional superiority of the existing methods (in terms of F-PF) comes only from recall (R-PF) but not precision (P-PF) in 20-33.4% of all cases (Figure 4 (b)); for biological scientists, precision of protein function prediction (making as accurate predictions as possible, even if few of them) is likely more important than R-PF (making as many predictions as possible, even if less accurate).

Next, we zoom into these results to compare IGLOO to each of LNA and GNA individually (Figure 2 (b) and (c), respectively).

Comparison to LNA. The comparison results against LNA are as follows. IGLOO is comparable or superior to all of the existing LNA methods considered in our study both topologically and functionally for all network pairs. When measuring the within 1% or within 5% accuracy (as described above), IGLOO remains comparable or superior both topologically and functionally. That is, IGLOO is at least 5% better than any of the existing LNA methods, both functionally and topologically. Thus, since IGLOO improves both topological and functional alignment quality of the existing



(a) LNA and GNA combined

(b) LNA

(c) GNA

Figure 2: Overall comparison of IGLOO (the best of its versions) and (a) LNA and GNA combined, (b) LNA, and (c) GNA, when considering 10 different IGLOO versions: IGLOO 0-4 for each of AlignMCL and AlignNemo used in step 1 of the algorithm. The comparison is shown for three different method "superiority levels" (denoted as p): 0%, 1%, and 5%. By a "superiority level", we mean the following. Given two methods A and B with alignment quality scores x and y, respectively, if $\frac{|x-y|}{max(x,y)} \leq p$, we say that A and B are comparable; otherwise, if x is greater/less than y, we say that A is superior/inferior to B. For a given network pair and a given existing method, only the best version of IGLOO is considered. The four cases are: 1) IGLOO is comparable or superior both topologically and functionally; 2) IGLOO is comparable or superior only functionally but not topologically; 3) IGLOO is comparable or superior only topologically but not functionally; and 4) IGLOO is inferior both topologically and functionally. The y-axes indicate the percentage of the combinations of the existing NA methods and network pairs for which the given case occurs.

LNA methods, at the minimum, IGLOO's contribution is the new best LNA method.

Comparison to GNA. The comparison results against GNA are as follows. When considering all combinations of the existing GNA methods and network pairs, in 36% of them, IGLOO is comparable or superior both topologically and functionally (case 1). In 64% of the combinations, IGLOO is comparable or superior only functionally but not topologically (case 2). In 42% of the combinations, IGLOO is comparable or superior only topologically but not functionally (case 3). IGLOO is never inferior both topologically and functionally (case 4). Note that 42% of the combinations are in the overlap of cases 2 and 3. When measuring the within 1% or within 5% accuracy, similar trends hold, except that now IGLOO is comparable or superior in up to 64% of all cases both topologically and functionally. That is, often, when the existing GNA methods are comparable or superior to IGLOO, their superiority is only within 1% or 5% of IGLOO's alignment quality.

Further, over all combinations of the existing GNA methods and network pairs in which IGLOO improves functional quality of the GNA methods but lowers their topological quality (case 2), the average improvement in functional quality is 331% (standard deviation of 276%), while the average decrease in topological alignment quality is only 40% (standard deviation of 11%). Thus, IGLOO gains more than it loses. Over all combinations of the existing GNA methods and network pairs in which IGLOO improves topological quality of the GNA methods but lowers their functional quality (case 3), the average improvement in topological quality is 18% (standard deviation of 12%), while the average decrease in functional alignment quality is 73% (standard deviation of 31%). Therefore, IGLOO overall beats the existing GNA methods for case 2, while the existing GNA methods beat IGLOO for case 3.

Statistical significance of IGLOO's improvement. Next, we summarize the performance of each method over all analyzed network pairs (Figure 3) and compute the statistical significance of the improvement of one method over another (where we use the paired t-test to compare alignment scores of two methods of interest over all network pairs). Based on these results, we comment on which version of IGLOO (out of IGLOO 0-4) is the best.

For LNA, there is a version of IGLOO, in particular IGLOO 4 under AlignNemo, which is superior in a statistically significantly manner (p-value < 0.05) to each considered LNA method in terms of both topological and functional alignment quality (Figure 3). In addition, IGLOO 3-4 under AlignMCL and IGLOO 2-4 under AlignNemo are statistically significantly superior to two of the four considered LNA methods (NetAligner and NetworkBlast).

For GNA, no version of IGLOO is superior in a statistically significantly manner to any existing GNA method in terms of both topological and functional alignment quality. However, importantly: 1) IGLOO is still superior to the existing GNA methods in many cases, as shown in Figure 2, it is just that its superiority is not statistically significant, and 2) none of the existing GNA methods is statistically significantly superior to any version of IGLOO in terms of both topological and functional alignment quality. Clearly, each of IGLOO and an existing GNA method is at best statistically significantly superior either functionally or topologically, but not both. So, for GNA, we split the discussion into two cases: 1) when IGLOO is statistically significantly better than the given existing GNA method in terms of only functional alignment quality, and 2) when IGLOO is statistically significantly better than the given existing GNA method in terms of only topological alignment quality.

For the first case above (IGLOO is statistically significantly superior to GNA only functionally), IGLOO 3-4 under any of AlignMCL or AlignNemo are statistically significantly superior to each considered GNA method. In addition, IGLOO 1-2 under AlignNemo are statistically significantly superior to each considered GNA method except L-GRAAL. The remaining versions out of IGLOO 1-4 under either AlignMCL or AlignNemo are statistically significantly superior to at least one of the considered GNA methods. Of all IGLOO versions, only IGLOO 0 under any of AlignMCL and AlignNemo is never statistically significantly superior to any of the

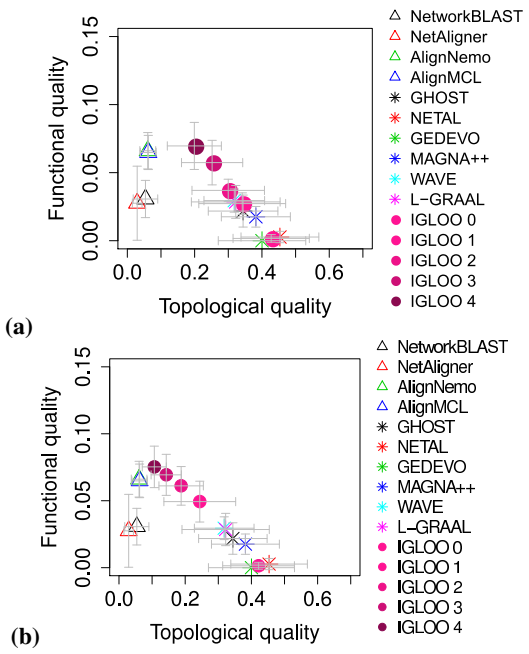


Figure 3: Topological (NCV-GS^3; x-axis) and functional (F-PF; y-axis) alignment quality for the existing LNA methods (triangles), existing GNA methods (stars), and IGLOO versions (circles), averaged over all aligned network pairs, when considering (a) AlignMCL and (b) AlignNemo in the first step of the IGLOO algorithm. For detailed results for each network pair individually, see Supplementary Figures S3 and S4.

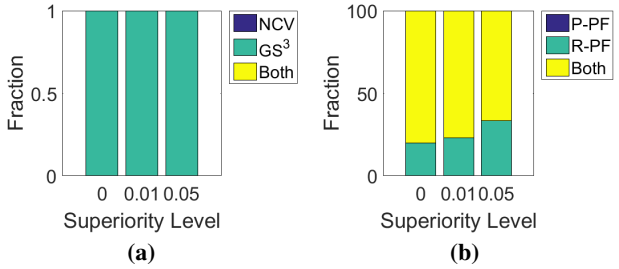


Figure 4: Reasons behind the superiority of the existing methods over IGLOO in (a) case 2 and (b) case 3. For case 2 (whenever the existing methods outperform IGLOO topologically but not functionally), we show the percentage of all instances in which the topological superiority of the existing methods is with respect to NCV only, GS^3 only, or both. Similarly, for case 3 (whenever the existing methods outperform IGLOO functionally but not topologically), we show the percentage of all instances in which the functional superiority of the existing methods is with respect to P-PF only, R-PF only, or both.

existing GNA methods in terms of functional alignment quality.

For the second case above (IGLOO is statistically significantly superior to GNA only topologically), no version of IGLOO beats NETAL or MAGNA++. This is not surprising, because these are among the best GNA methods in terms of topological alignment quality [15]. For the remaining four GNA methods, IGLOO 0 under AlignMCL is statistically significantly superior to each of the four methods. Also, IGLOO 0 under AlignNemo is statistically significantly superior to each of the four methods except GEDEVO. Finally, IGLOO 1 under AlignNemo is statistically significantly superior to L-GRAAL. No other version of IGLOO is statistically significantly superior to any GNA method topologically.

In summary, in terms of functional alignment quality, IGLOO 4 is the strongest compared to both LNA and GNA, and it is followed by IGLOO 3 and IGLOO 2. In terms of topological alignment quality, IGLOO 4 is the strongest compared to LNA, and it is followed by IGLOO 3 and IGLOO 2, while IGLOO 0 is the strongest compared to GNA, and it is followed by IGLOO 1.

Robustness to the choice of LNA method in step 1 of the IGLOO algorithm. The results in terms of superiority of IGLOO over the existing LNA and GNA methods are qualitatively the same independent of whether AlignMCL or AlignNemo is used as IGLOO's input (Supplementary Figures S1 and S2). Hence, qualitatively, IGLOO is robust to the choice of LNA method. Yet, we note that using AlignMCL results in slightly better topological alignment quality compared to using AlignNemo, while using AlignNemo results in slightly better functional alignment quality compared to using AlignMCL (Figure 3 and Supplementary Figures S3-S4).

3.2 Comparison in terms of running time

Here, we compare IGLOO when using AlignMCL in the first step of the algorithm against each of the existing LNA and GNA methods in terms of computational complexity. We run all NA methods on the same Linux machine with 64 CPU cores (AMD Opteron (tm) Processor 6378) and 512 GB of RAM. All methods can run on a single core with the exception of GHOST, which can run on at least two cores. Three of the existing GNA methods (GHOST, GEDEVO, and MAGNA++) can run on multiple cores. The maximum number of cores that the parallelizable methods can use is bounded by the number of cores that our machine has. We analyze the methods' entire running times, which encompass both computing node similarities and constructing alignments. Also, we measure only running times needed to construct alignments, ignoring the time needed to precompute node similarities. We show the results for worm and yeast PPI networks of $Y2H_1$ type, since both networks are relatively small, and even the slowest NA method could finish aligning the two networks on a single core within a reasonable time (within one day). For any other network pair, it could take much longer time for the slowest method to finish.

Regarding the entire running times, the findings are as follows (Figure 5 (a)). Since IGLOO uses AlignMCL and NETAL within its algorithm, it is not surprising that IGLOO is (slightly) slower than these two methods. Of the remaining methods, IGLOO is faster than two methods (i.e., serial GEDEVO and serial MAGNA++), it is relatively comparable to three methods (i.e., AlignNemo, serial GHOST, and WAVE), and it is slower than six methods (i.e., NetworkBLAST, NetAligner, parallelized GHOST, parallelized GEDEVO, parallelized MAGNA++, and L-GRAAL).

Regarding only the times for computing alignments, the findings are as follows (Figure 5 (b)). Again, IGLOO is (slightly) slower than AlignMCL and NETAL. Of the remaining methods, IGLOO is faster than eight methods (NetworkBLAST, serial GHOST, parallelized GHOST, serial GEDEVO, parallelized GEDEVO, serial MAGNA++, parallelized MAGNA++, and L-GRAAL), it is relatively comparable to two methods (NetAligner and WAVE), and it is slower than one method (AlignNemo). Importantly, all methods except perhaps serial and parallelized GEDEVO and serial MAGNA++ have reasonably low running times.

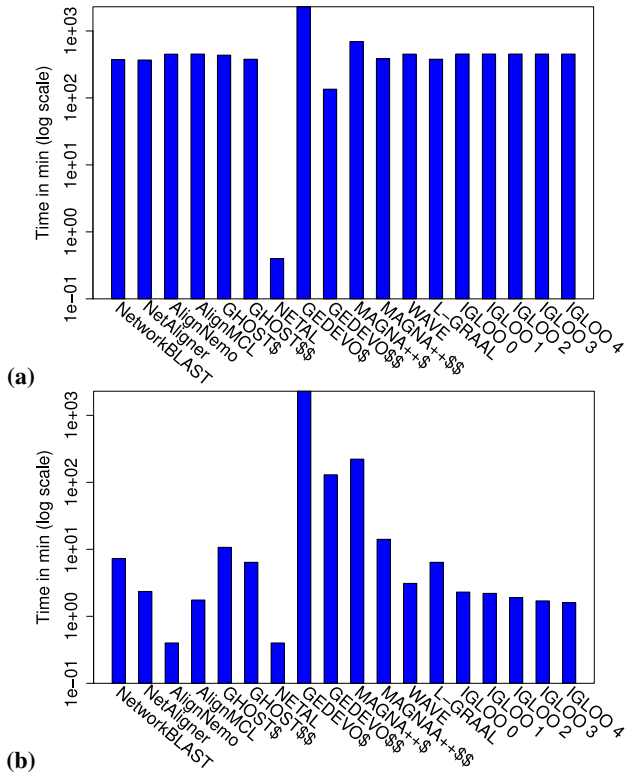


Figure 5: Representative running time comparison of the different NA methods, for (a) the entire running times and (b) only the times for constructing alignments. For each method that is parallelizable, its single-core version is marked with a '$' symbol, and its 64-core version is marked with a '$$' symbol. All other methods are run on a single core. Results are shown for using AlignMCL in the first step of IGLOO algorithm.

4. CONCLUSION

We propose a new NA method, IGLOO, which aims to combine the advantages of both LNA and GNA in order to better balance between functional and topological alignment quality. We demonstrate that IGLOO outperforms all considered LNA methods with respect to both alignment quality types. Also, it outperforms the considered GNA methods in many cases.

IGLOO is generalizable as it can include any existing LNA and GNA methods into its algorithm. (The existing methods we test are simply a proof of concept of combining LNA with GNA.) As the field of NA evolves, including newer and more sophisticated methods could further improve the alignment quality of IGLOO.

5. ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant No. CAREER CCF-1452795 and CCF-1319469.

6. REFERENCES

[1] M. Bayati, M. Gerritsen, D. F. Gleich, A. Saberi, and Y. Wang. Algorithms for large, sparse network alignment problems. In *Data Mining, 2009. ICDM'09. Ninth IEEE International Conference on*, pages 705–710. IEEE, 2009.

[2] B. Breitkreutz, C. Stark, T. Reguly, L. Boucher, A. Breitkreutz, M. Livstone, R. Oughtred, D. H. Lackner, J. Bähler, and V. Wood. The BioGRID Interaction Database: 2008 update. *Nucleic Acids Research*, 36:D637–D640, 2008.

[3] K. R. Brown and I. Jurisica. Unequal evolutionary conservation of human protein interactions in interologous networks. *Genome Biology*, 8(5):R95, 2007.

[4] G. Ciriello, M. Mina, P. H Guzzi, M. Cannataro, and C. Guerra. AlignNemo: a local network alignment method to integrate homology and topology. *PLOS ONE*, 7(6):e38107, 2012.

[5] C. Clark and J. Kalita. A multiobjective memetic algorithm for PPI network alignment. *Bioinformatics*, 31(12):1988–1998, 2015.

[6] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.

[7] J. Crawford and T. Milenković. GREAT: GRaphlet Edge-based network AlignmenT. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2015.

[8] J. Crawford, Y. Sun, and T. Milenković. Fair evaluation of global network aligners. *Algorithms for Molecular Biology*, 10(19), 2014.

[9] F. Faisal, L. Meng, J. Crawford, and T. Milenković. The post-genomic era of biological network alignment. *EURASIP Journal on Bioinformatics and Systems Biology*, 2015(1):1–19, 2015.

[10] F. Faisal, H. Zhao, and T. Milenković. Global network alignment in the context of aging. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 12(1):40–52, 2015.

[11] J. Hu and K. Reinert. LocalAli: an evolutionary-based local alignment approach to identify functionally conserved modules in multiple networks. *Bioinformatics*, 31(3):363–372, 2015.

[12] R. Ibragimov, M. Malek, J. Guo, and J. Baumbach. GEDEVO: An evolutionary graph edit distance algorithm for biological network alignment. *German Conference on Bioinformatics (GCB)*, 34:68–79, 2013.

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

[14] N. Malod-Dognin and N. Pržulj. L-GRAAL: Lagrangian graphlet-based network aligner. *Bioinformatics*, 31:2182–2189, 2015.

[15] L. Meng, A. Striegel, and T. Milenković. Local versus global biological network alignment. *Bioinformatics*, in press, 2016.

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

[17] M. Mina and P. H. Guzzi. AlignMCL: Comparative analysis of protein interaction networks through markov clustering. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference on*, pages 174–181. IEEE, 2012.

[18] A. Narayanan, E. Shi, and B. I. P. Rubinstein. Link prediction by de-anonymization: How we won the Kaggle social network challenge. In *Neural Networks (IJCNN), The 2011 International Joint Conference on*, pages 1825–1834. IEEE, 2011.

[19] B. Neyshabur, A. Khadem, S. Hashemifar, and S. S. Arab. NETAL: a new graph-based method for global alignment of protein-protein interaction networks. *Bioinformatics*, 29(13):1654–1662, 2013.

[20] R. A. Pache and P. Aloy. A novel framework for the comparative analysis of biological networks. *PLOS ONE*, 7(2):e31220, 2012.

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

[22] V. Saraph and T. Milenković. MAGNA: Maximizing Accuracy in Global Network Alignment. *Bioinformatics*, 30(20):2931–2940, 2014.

[23] B. Seah, S. S. Bhowmick, and C. F. Dewey. DualAligner: a dual alignment-based strategy to align protein interaction networks. *Bioinformatics*, 30:2619–2626, 2014.

[24] R. Sharan, S. Suthram, R. M. Kelley, T. Kuhn, S. McCuine, P. Uetz, T. Sittler, R.M. Karp, and T. Ideker. Conserved patterns of protein interaction in multiple species. *Proceedings of the National Academy of Sciences*, 102(6):1974–1979, 2005.

[25] R. Singh, J. Xu, and B. Berger. Pairwise global alignment of protein interaction networks by matching neighborhood topology. In *Research in Computational Molecular Biology*, pages 16–31, 2007.

[26] Y. Sun, J. Crawford, J. Tang, and T. Milenković. Simultaneous optimization of both node and edge conservation in network alignment via WAVE. In *Workshop on Algorithms in Bioinformatics (WABI)*, pages 16–39, 2015.

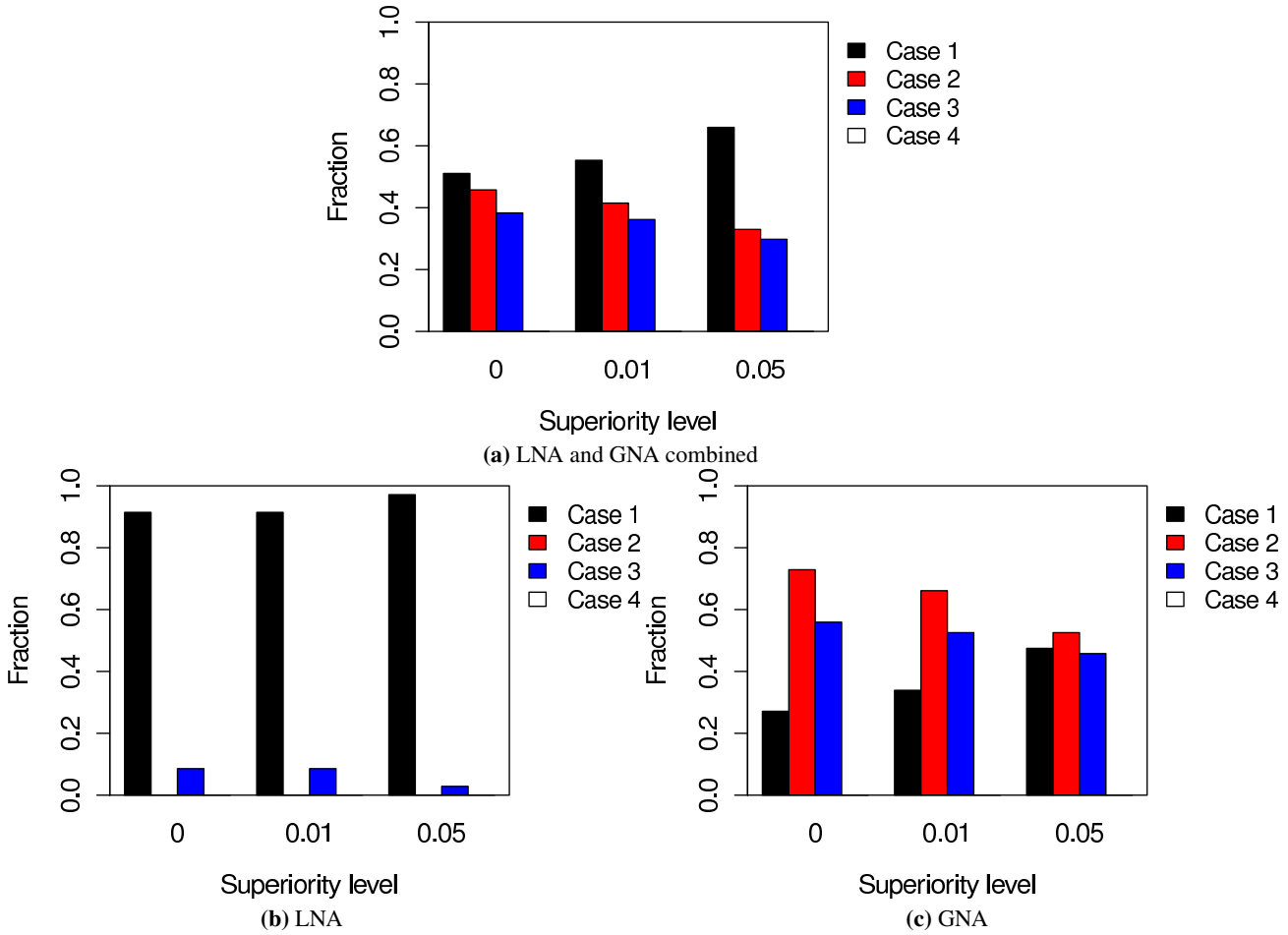
[27] A. Todor, A. Dobra, and T. Kahveci. Probabilistic biological network alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 10(1):109–121, 2013.

[28] V. Vijayan, V. Saraph, and T. Milenković. MAGNA++: Maximizing Accuracy in Global Network Alignment via both node and edge conservation. *Bioinformatics*, 31:2409–2411.

[29] M. Zaslavskiy, F. Bach, and J-P Vert. A path following algorithm for the graph matching problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(12):2227–2242, 2009.

SUPPLEMENTARY INFORMATION FOR: IGLOO: Integrating global and local biological network alignment

Lei Meng [1,2,3,4], **Joseph Crawford** [1,2,3], **Aaron Striegel** [1,4], **and Tijana Milenković** [1,2,3,*]
[1]Department of Computer Science and Engineering
[2]ECK Institute of Global Health
[3]Interdisciplinary Center for Network Science and Applications (iCeNSA)
[4]Wireless Institute
University of Notre Dame, Notre Dame, IN 46556
[*]To whom correspondence should be addressed

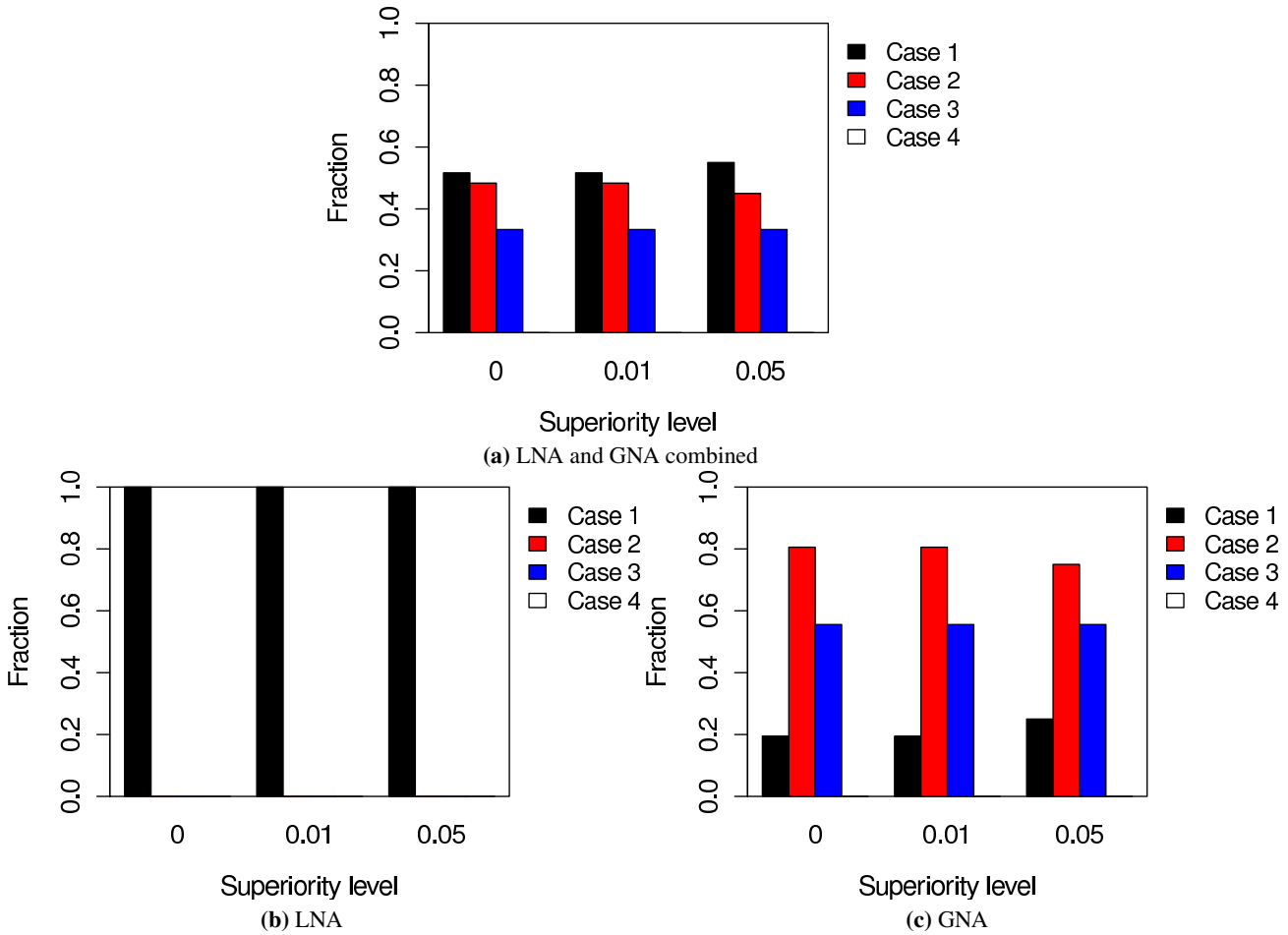


(a) LNA and GNA combined

(b) LNA

(c) GNA

Figure S1: Overall comparison of IGLOO (the best of its versions) and (a) LNA and GNA combined, (b) LNA, and (c) GNA, when considering five different versions of IGLOO (IGLOO 0-4) under AlignMCL in the first step of the algorithm. The comparison is shown for three different method "superiority levels" (denoted as p): 0%, 1%, and 5%. By a "superiority level", we mean the following. Given two methods A and B with alignment quality scores x and y, respectively, if $\frac{|x-y|}{max(x,y)} \leq p$, we say that A and B are comparable; otherwise, if x is greater/less than y, we say that A is superior/inferior to B. For a given network pair and a given existing method, only the best version of IGLOO is considered. The four cases are as follows: 1) IGLOO is comparable or superior both topologically and functionally; 2) IGLOO is comparable or superior only functionally but not topologically; 3) IGLOO is comparable or superior only topologically but not functionally; and 4) IGLOO is inferior both topologically and functionally. The y-axes indicate the percentage of the combinations of the existing NA methods and network pairs for which the given case occurs.



(a) LNA and GNA combined

(b) LNA

(c) GNA

Figure S2: Overall comparison of IGLOO (the best of its versions) and (a) LNA and GNA combined, (b) LNA, and (c) GNA, when considering five different versions of IGLOO (IGLOO 0-4) under AlignNemo in the first step of the algorithm. The comparison is shown for three different method "superiority levels" (denoted as p): 0%, 1%, and 5%. By a "superiority level", we mean the following. Given two methods A and B with alignment quality scores x and y, respectively, if $\frac{|x-y|}{max(x,y)} \leq p$, we say that A and B are comparable; otherwise, if x is greater/less than y, we say that A is superior/inferior to B. For a given network pair and a given existing method, only the best version of IGLOO is considered. The four cases are as follows: 1) IGLOO is comparable or superior both topologically and functionally; 2) IGLOO is comparable or superior only functionally but not topologically; 3) IGLOO is comparable or superior only topologically but not functionally; and 4) IGLOO is inferior both topologically and functionally. The y-axes indicate the percentage of the combinations of the existing NA methods and network pairs for which the given case occurs.

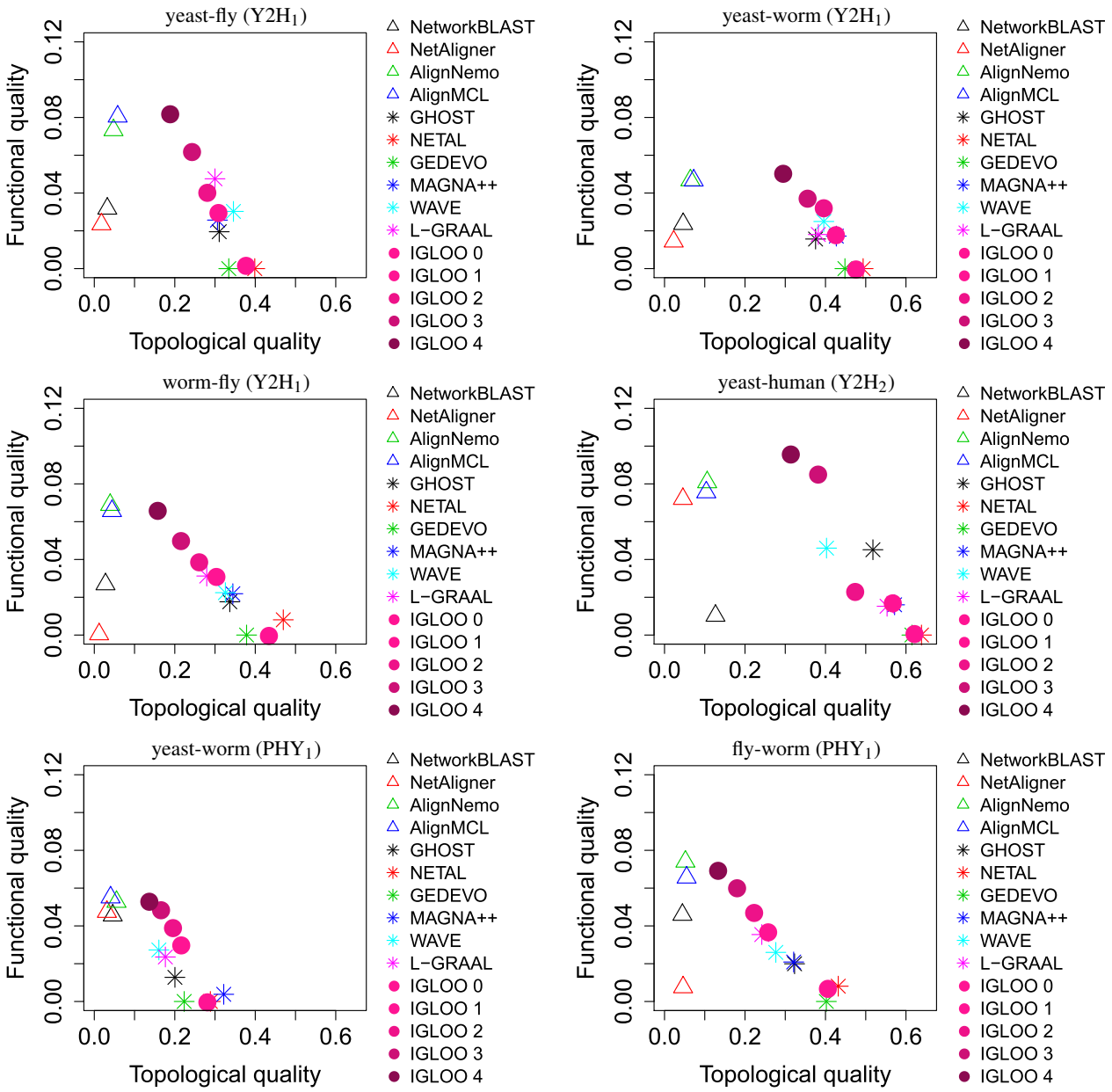


Figure S3: Topological (NCV-GS^3; x-axis) and functional (F-PF; y-axis) alignment quality for the existing LNA methods (triangles), existing GNA methods (stars) and IGLOO versions (circles), for each aligned network pair, when considering AlignMCL in the first step of IGLOO algorithm.

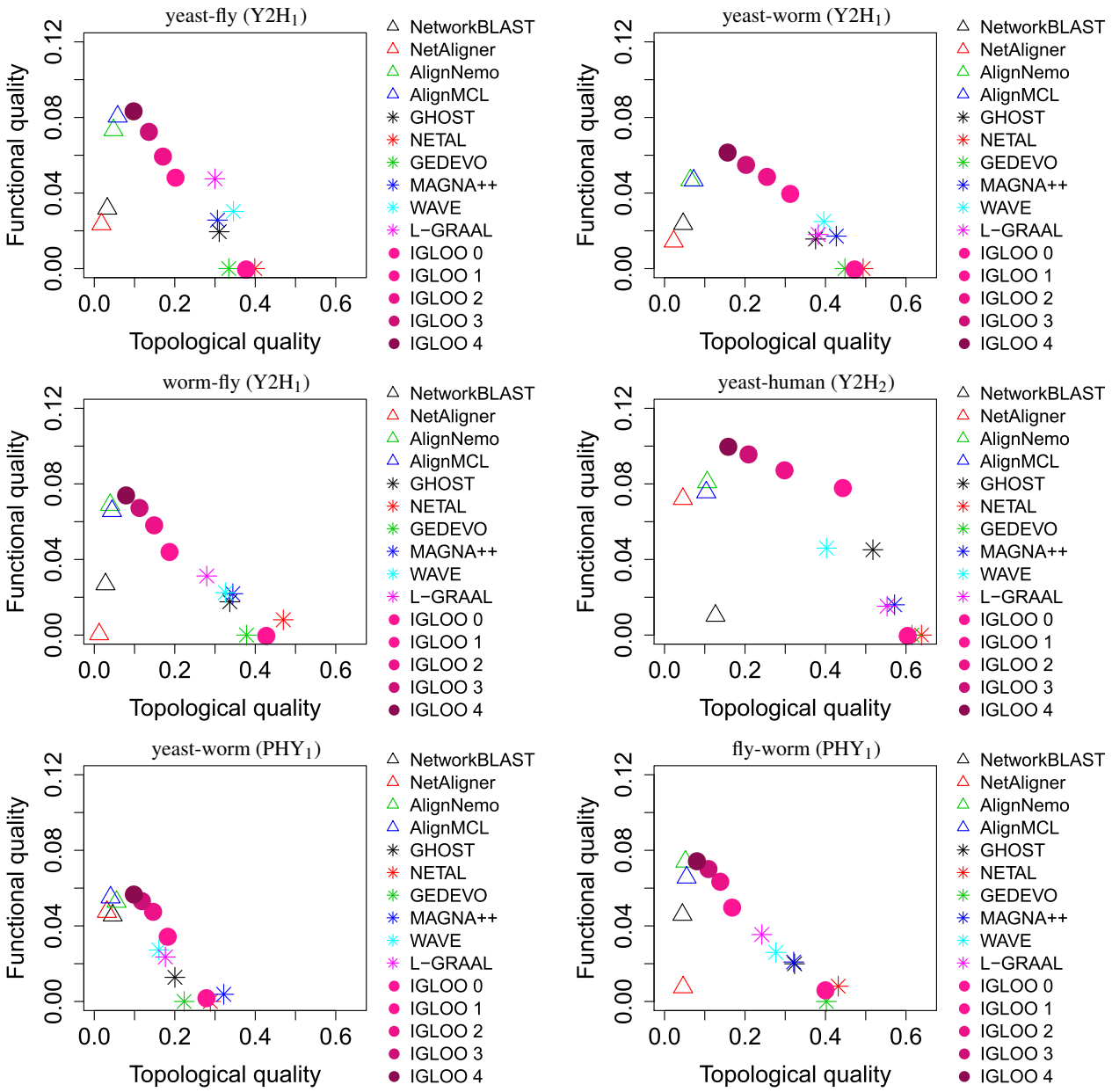


Figure S4: Topological (NCV-GS^3; x-axis) and functional (F-PF; y-axis) alignment quality for the existing LNA methods (triangles), existing GNA methods (stars) and IGLOO versions (circles), for each aligned network pair, when considering AlignNemo in the first step of IGLOO algorithm.