Supplementary Information for: Topological network alignment uncovers biological function and phylogeny

Oleksii Kuchaiev[1,‡], Tijana Milenković[1,‡], Vesna Memišević[1], Wayne Hayes[1,3], Nataša Pržulj[2,*]

[1]Department of Computer Science, University of California, Irvine, CA 92697-3435, USA

[2]Department of Computing, Imperial College London SW7 2AZ, UK

[3]Department of Mathematics, Imperial College London SW7 2AZ, UK

[‡]These authors contributed equally to this work

[*]To whom correspondence should be addressed; E-mail: natasha@imperial.ac.uk .

1 Materials and methods

1.1 Network Alignment Problem

We formulate the network alignment problem as follows. We denote the two networks that we are aligning by $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where V_i is the set of nodes and E_i is the set of edges in network G_i. Without loss of generality, we assume that $|V_1| \leq |V_2|$. As a simplification to possible multi-valued alignments, for now we find an *injective* mapping, i.e., the mapping that does not map two nodes into one, to align each node in G_1 to exactly one node in G_2 with a similar topological neighborhood. Formally, we define "topological similarity" of nodes by using the notion of 73-dimensional "graphlet degree signatures" and the "signature similarity" measure [1] (see Methods for details). Our method produces a *global* network alignment, since our node mapping is defined for all nodes in V_{G_1}. Thus, we do not allow "gaps", i.e., nodes without a match, in G_1, but we do allow them in the larger network, G_2.

The network alignment problem is closely related to the NP-complete subgraph isomorphism problem [2]. An *isomorphism* is a bijection between nodes of two networks that preserves

edge adjacency. The subgraph isomorphism problem asks if network G_2 contains an isomorphic copy of network G_1. Thus, isomorphism fully preserves the topologies of the graphs and the network alignment problem includes the subgraph isomorphism problem as a special case. This makes the network alignment problem computationally hard and therefore, heuristics must be used.

1.2 GRAAL (GRAph ALigner) Algorithm

GRAAL is a greedy algorithm for aligning two networks. First, it computes a matrix C of costs of aligning each node in G_1 with each node in G_2. Rows of C correspond to nodes in G_1 and columns correspond to nodes in G_2. When computing the cost of aligning a node u from G_1 with a node v from G_2, GRAAL takes into account their signature similarity as well as their degrees (see Methods section of the manuscript). It takes into account node degrees to produce a lower cost for aligning two high-degree nodes than two low-degree nodes, if their signature similarities are the same. By doing so, GRAAL assigns higher "weights" to highly connected nodes, and thus, it aligns the densest parts of networks first. Therefore, if $deg(u)$ is the degree of node u, $max\_deg(G)$ is the maximum degree of nodes in network G, $S(u, v)$ is the signature similarity of nodes u and v, and α is a parameter in [0, 1] that controls the contribution of the node signature similarity to the cost function, then the cost of aligning nodes u and v is computed as:

$$C(u, v) = 2 - ((1 - \alpha) \times \frac{deg(u) + deg(v)}{max\_deg(G_1) + max\_deg(G_2)} + \alpha \times S(u, v)).$$

A cost of 0 corresponds to a pair of topologically similar nodes, while a cost close to 2 corresponds to a pair of topologically very different nodes.

After the matrix C is created, GRAAL searches for an initial seed. A seed is a pair of nodes that have the lowest alignment cost. GRAAL finds such a pair of nodes by searching for the lowest value in matrix C. If there exist multiple seed candidates with the same cost, the seed is chosen uniformly at random.

Next, after seed (u, v) is found, GRAAL builds spheres of all possible radii around nodes u and v in graphs G_1 and G_2, respectively. A sphere of radius r around node u in a network G is a set of nodes which are exactly at the distance r from node u: $S_G(u, r) = \{v \in G : d(u, v) = r\}$. Spheres of the same radius in two networks are then greedily aligned together by searching for the pairs $(u', v') : u' \in S_{G_1}(u, r)$ and $v' \in S_{G_2}(v, r)$ that are not already aligned and that can be aligned with the minimal cost. Ties are broken randomly.

When all spheres around seed (u, v) have been aligned as described above, it is possible for some nodes in both networks to remain unaligned. For example, spheres of the same radius in different networks can have different number of nodes. Additionally, biological networks commonly consist of many connected components. For this reason, GRAAL searches for a new seed again and repeats the same algorithm on a pair of networks (G_1^p, G_2^p) for $p = 1, 2,$ and 3. We define network G^p as a new network $G^p = (V, E^p)$ with the same set of nodes as

G, and with $(u, v) \in E^p$ if and only if the distance between nodes u and v in G is less than or equal to p, i.e., $d_G(u, v) \leq p$. Note that $G^1 = G$. Thus, since biological networks commonly consist of many connected components, GRAAL repeats itself by searching for a new seed and aligning the remaining components of two networks. The algorithm raises networks to the next power only if, in the previous step, both of old seed's nodes u and v had a non-empty spheres of radius at least 3 around them. We impose this restriction because real world networks usually have many small connected components with the largest possible sphere of radius less then 3. GRAAL raises networks to the powers of 2 and 3 to allow up to 2 node insertions or deletions. We do not allow more insertions or deletions because of the small-world nature of biological networks, meaning that they have small diameters. GRAAL algorithm stops when each node from G_1 is aligned to exactly one node in G_2.

Having graphlet degree vectors for nodes in both networks, the time complexity of GRAAL is $O(|V_2| \times (|V_2| + \max\{|E_1|, |E_2|\}))$, which makes it applicable to large networks. The algorithm's pseudo code and details about the complexity analysis are presented below.

Note that potential randomness in the process of alignment construction, caused by the existence of node pairs with the same alignment cost, could lead to different alignments. Therefore, we test how alignments change over different runs for a given α. By running multiple experiments, we empirically find that the changes in the edge correctness are minor and that a subset of all aligned pairs is preserved across different runs. The number of pairs present in the intersection of multiple alignments depends both on the choice of α and on the networks being aligned. In our paper, we focus on α of 0.8. Supplementary Figure 3 illustrates that if we intersect k alignments obtained from k runs for $\alpha = 0.8$, starting from $k = 6$, the number of pairs common to all alignments barely changes. We refer to this intersection of aligned pairs across different runs as the "core" alignment.

1.2.1 Detailed description of GRAAL algorithm

The detailed description and the pseudo code of GRAAL algorithm is presented in Box 1. Procedure $findSeed(G_1, G_2)$ takes as an input two graphs and finds the best alignment seed which is a pair of nodes (u, v) such that $C(u, v)$ is minimal. Ties are broken uniformly at random. Procedure $makeSphere(u, radius, G)$ returns a set of nodes from graph G which form $S(u, radius)$ in G. Procedure $alignSpheres(S_1, S_2)$ takes two sets of nodes S_1 and S_2 from graphs G_1 and G_2, respectively, and greedily aligns the two sets by finding node pairs $(u, v) : u \in S_1$, $v \in S_2$ with minimal cost $C(u, v)$. Ties are broken uniformly at random. S_1 and S_2 are not necessarily of the same size. If, for example $sizeof(S_1) > sizeof(S_2)$, then $sizeof(S_2) - sizeof(S_1)$ will not be aligned. Note that in each iteration, after the seed was found, spheres S^1_{radius} and S^2_{radius} can be aligned in parallel for all values of $radius$, because their alignments are independent of each other. Pseudo code for aligning spheres is given in Box 2.

Algorithm 1 Box 1

Compute Matrix C;
int $power \leftarrow 1$;
while $\exists$ node $\in G_1$ which is not aligned **do**
  $(u, v) \leftarrow findSeed(G_1^p, G_2^p)$;
  Align u and v;
  int $size \leftarrow 1$;
  int $radius \leftarrow 1$;
  while $size \neq 0$ **do**
    $S^1_{radius} \leftarrow makeSphere(u, radius, G_1^p)$;
    $S^2_{radius} \leftarrow makeSphere(v, radius, G_2^p)$;
    $size \leftarrow \min\{sizeof(S^1_{radius}), sizeof(S^2_{radius})\}$;
    if $size \neq 0$ **then**
      $alignSpheres(S^1_{radius}, S^2_{radius})$;
    end if
    radius++;
  end while
  if $(radius \geq 3)$and $(p < 3)$ **then**
    p++;
  end if
end while

Algorithm 2 Box2 $alignSpheres(S_1, S_2)$

Set $pairs = \emptyset$;
$cost \leftarrow \infty$;
for all node $n_1 \in S_1$ **do**
  for all node $n_2 \in S_2$ **do**
    $pair_cost = C(n_1, n_2)$;
    if $pair_cost < cost$ **then**
      $cost \leftarrow pair_cost$;
      Clear pairs;
      Add (n_1, n_2) to pairs;
      Delete n_1 and n_2 from S_1 and S_2;
    else if $pair_cost = cost$ **then**
      Add (n_1, n_2) to pairs;
      Delete n_1 and n_2 from S_1 and S_2;
    end if
  end for
end for
Return random pair (n_1, n_2) from pairs as a result;

1.2.2 Complexity

To compute cost matrix C, the algorithm needs graphlet degree signature vectors of all nodes from graphs G_1 and G_2. If G_2 is the larger graph, then the running time of computing signatures of all nodes from both graphs is $O(|V_{G_2}|^5)$. However, since we can parallelize over a cluster of cores, the high computational cost is not prohibitive [3].

Constructing matrix C from signature vectors of all nodes in both graphs requires $O(|V_{G_1}| \times |V_{G_2}|)$ time and space. Also, while computing this matrix, GRAAL creates priority queue with pairs of nodes as items and costs of aligning them as keys. This priority queue allows GRAAL to extract the pair with the smallest alignment cost, i.e., the new seed, in $O(1)$ time, while adding new elements to it requires $O(\log n)$ time. In total, constructing both the cost matrix and the priority queue requires $O(|V_{G_1}| \times |V_{G_2}| \times (\log |V_{G_1}| + \log |V_{G_2}|))$ time.

Having a seed (u, v), constructing all spheres $S(u, radius)$ and $S(v, radius)$ can be done efficiently in $O(|V_{G_1}| + |V_{G_2}| + |E_{G_1}| + |E_{G_2}|)$ time using standard Breadth First Search starting at u and v in graphs G_1 and G_2, respectively. Next, raising graph G to every possible power p takes $O(|V_G| \times (|V_G| + |E_G)|)$ time by running Breadth First Search for each node. Since GRAAL performs at most G^2 and G^3, this is very rough estimation.

Procedure $alignSpheres(S_1, S_2)$ (see Box 2) takes $O(\min\{|S_1|, |S_2|\}^2)$ time. Theoretically, it is possible that the first *while* loop will be executed for every node in G_1; in this case, the entire alignment will be built by $findSeed(G_1^p, G_2^p)$ procedure in $O(|V_{G_1}|^2)$ time using priority queue of pairs. In practice, the first *while* loop will be executed only few times (especially if graphs really share similar structure); in this case, it is the execution of $alignSpheres(S_1, S_2)$ procedure that will take most of the computational time, which is in total $O(|V_{G_1}|^2)$. Therefore, GRAAL requires $O(|V_{G_1}| \times |V_{G_2}| \times (\log |V_{G_1}| + \log |V_{G_2}|))$+$O(|V_{G_2}| \times (|V_{G_2}| + \max\{|E_{G^1}|, |E_{G^2}|\})$ + $O(|V_{G_1}|^2)$. Without loss of generality, we can assume that $|V(G_1)| \leq |V(G_2)|$. Then, since $|E_G| = O(|V_G|^2)$, the complexity of GRAAL algorithm is $O(|V_{G_2}| \times (|V_{G_2}| + \max\{|E_{G_1}|, |E_{G_2}|\}))$.

1.3 Scoring Alignments

To score GRAAL's alignments, we define the following three scores.

Edge Correctness. If we denote by $g : V_1 \to V_2$ an alignment produced by GRAAL, then the edge correctness, henceforth also denoted by "EC", is defined as:

$$EC = \frac{|\{(u, v) \in E_1 : (g(u), g(v)) \in E_2\}|}{|E_1|} \times 100\%.$$

Edge correctness is the percentage of edges in the first graph that are aligned to edges in the second graph. High edge correctness means that networks G_1 and G_2 share similar topologies. In particular, if $EC = 100\%$, then the second network, G_2, contains a subnetwork isomorphic to G_1. Note that this isomorphic subnetwork is not necessarily *induced*, since there can exist additional edges in G_2 that do not exist in G_1 among the corresponding sets of nodes. Intuitively, edge correctness shows us to what extent $g(G_1)$ is topologically similar to G_1.

Node correctness. If we denote by $f : V_1 \to V_2$ the correct node mapping of G_1 to G_2 and by $g : V_1 \to V_2$ an alignment produced by GRAAL, then the node correctness, henceforth also denoted by "NC", is defined as:

$$NC = \frac{|\{u \in V_1 : f(u) = g(u)\}|}{|V_1|} \times 100\%.$$

Thus, NC is the percentage of nodes in network G_1 that are correctly aligned (with respect to "true alignment" f) to nodes in G_2. Clearly, to measure NC, we need to know the correct node mapping.

Interaction correctness. If we denote by $f : V_1 \to V_2$ the correct node mapping of G_1 to G_2 and by $g : V_1 \to V_2$ an alignment produced by GRAAL, then the interaction correctness, henceforth also denoted by "IC", is defined as:

$$IC = \frac{|\{(u,v) \in E_1 : (f(u), f(v)) \in E_2, f(u) = g(u), f(v) = g(v)\}|}{|E_1|} \times 100\%.$$

Thus, IC is the percentage of interactions that are aligned correctly. We say that an interaction A–B is correctly aligned if two connected nodes A and B from G_1 are correctly aligned to their partners in G_2 (with respect to the correct node mapping) and if their partners interact in G_2. Thus, the interaction correctness is a more strict measure than the edge correctness. Clearly, to measure IC, we need to know the correct node mapping.

Note that in real applications such as alignments of biological networks we usually do not know the correct node mappings, and therefore, we can only measure EC.

1.4 Statistical significance

1.4.1 Statistical significance of alignments produced by GRAAL

Given GRAAL's alignment of two networks, $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, with the edge correctness (EC) of $x\%$, the probability P of successfully aligning $k = [m_1 \times EC] = [m_1 \times x]$ or more edges by chance is the tail of the hypergeometric distribution:

$$P = 1 - \sum_{i=k-1}^{m_2} \frac{\binom{m_2}{i}\binom{p-m_2}{m_1-i}}{\binom{p}{m_1}}, \tag{1}$$

where $n_1 = |V_1|$, $n_2 = |V_2|$, $m_1 = |E_1|$, $m_2 = |E_2|$, and $p = \frac{n_2(n_2-1)}{2}$ (i.e., p is the number of pairs of nodes in G_2). Thus, the p-value of an alignment depends on the densities of the two networks being aligned and on the edge correctness. For GRAAL's yeast2-human1 alignment, $x = EC = 11.72\%$, $p = \frac{9{,}141(9{,}141-1)}{2} = 41,774,370$, $m_1 = 16,127$, and $m_2 = 41,456$ (for the description of yeast2 and human1 networks, see Supplementary Table 5). Plugging these numbers into formula (1) results in the p-value of 7×10^{-8}.

1.4.2 Statistical significance of yeast2-human1 alignment

We evaluate the statistical significance of GRAAL's yeast2-human1 alignment as follows. We compute the probability of obtaining EC of yeast2-human1 alignment in GRAAL's alignment of model networks (belonging to an appropriate network model) of the same size the data. For this, we use the following form of the Chebyshev's inequality:

$$P(|X - \mu| \geq \alpha) \leq \frac{\sigma^2}{\alpha^2}. \tag{2}$$

Since model networks that are aligned with GRAAL have *the same* number of nodes and edges as the data, it is reasonable to assume that the distribution of their alignment scores is unimodal. That is, μ is an estimate for the only mode in the distribution. Also, the variance σ^2 is clearly finite. Therefore, we can further refine the p-value using Vysochanskij–Petunin inequality[4], which is more precise than Chebyshev's inequality for unimodal distributions:

$$P(|X - \mu| \geq \lambda\sigma) \leq \frac{4}{9\lambda^2}. \tag{3}$$

Using this inequality, we obtain the p-value of 8.4×10^{-3} for GRAAL's yeast2-human1 alignment.

1.4.3 Statistical significance for sharing common GO terms

In GRAAL's yeast2-human1 alignment, 45.1% of aligned protein pairs share at least one GO term (Supplementary Table 6). In this analysis, we take into account only those protein pairs in which both proteins are annotated with at least one *known* GO term. We excluded from the analysis all "root" GO terms (GO:0008150 for biological process, GO:0003674 for molecular function, and GO:0005575 for cellular component). To examine the statistical significance of our result, we compute the probability that $\geq 45.1\%$ of protein pairs would share at least one GO term in a random alignment of yeast 2 and human1 PPI networks.

Again, we use standard model of sampling without replacement. In human1 network, $8,447$ out of $9,141$ proteins are annotated with at least one known GO term. In yeast2 network, $2,336$ out of $2,390$ proteins are annotated with at least one known GO term. Therefore, there are $p = 2,336 \times 8,447 = 19,732,192$ pairs in which both proteins have a GO term. In $m_2 = 7,825,804$ out of these p pairs, both proteins share at least one common GO term. In our alignment, in $m_1 = 2,173$ protein pairs both proteins are annotated with at least one known GO term and $k = m_1 \times 45.1\%$ of them share at least one common GO term. Then, we compute the statistical significance of our result using formula (1) to be 1.82×10^{-7}.

1.4.4 Statistical significance of Protists and Fungi phylogenetic trees

We construct a phylogenetic tree of different species (within a family of species) based on EC scores of GRAAL's pairwise alignments of their metabolic networks. We focus on two

families: protists and fungi. Our phylogenetic tree for protists is presented and discussed in the main paper. Equivalent to our analyses for protists, to create our phylogenetic tree for fungi, we extract the union of all metabolic pathways from KEGG for each of the organisms within fungi family to form their metabolic networks. Then, we use GRALL to find all-to-all pairwise alignments between the metabolic networks. Supplementary Table 7 describes the metabolic networks that we used in our study. Next, we create a phylogenetic tree for fungi using the average distance algorithm and the pairwise EC scores (ranging from 30% to 77%; Supplementary Table 8) as a distance measure. The resulting tree is presented in Supplementary Figure 4. As shown in the figure, Encephalitozoon cuniculi (Microsporidian) groups together with Schizosaccharomyces pombe (Ascomycetes). This result is encouraging since it has been shown that microsporidia consistently falls not only within fungal diversification but also close to Ascomycetes [5], even though for a long time it has been difficult to resolve the evolutionary relationship between the microsporidia and other eukaryotes.

Below, we examine the statistical significance of the phylogenetic trees that we construct for fungi and protists. We generate phylogenetic trees for a family of species based on EC scores of GRAAL's alignments of their metabolic networks. To measure the probability of obtaining our phylogenetic trees by chance, we repeat the same procedure 30 times to generate "random trees" (defined below). Then, we compare the "distance" (defined below) between "random trees" (defined below) and between our phylogenetic and "random trees." Clearly, we expect to obtain a significantly higher distance between our phylogenetic trees and the corresponding "random trees" than between "random trees" alone. A "random tree" is created in the same way we generate a real phylogenetic tree: it is based on EC scores produced by GRAAL when aligning random equivalents of the analyzed metabolic networks. A random equivalent of a network is a model network with the same number of nodes and edges as the original network, drawn from a given network model. The choice of an appropriate network model used to generate model networks is crucial for this and many other graph-based analyses of biological networks. Since all analyzed metabolic networks are bipartite graphs with enzymes in one bipartition and proteins in another, we use the bipartite random graph model with the same degree distribution as the data. Once model networks are generated for all metabolic networks, we use GRAAL to align all pairs of model equivalents of metabolic networks and we generate the phylogenetic tree from the obtained alignment scores in the same way we generate the phylogenetic trees for real metabolic networks; we call the resulting tree the "random tree." To take into account the randomness of the model, we repeat this procedure 30 times, by regenerating model equivalents of metabolic networks and thus regenerating the "random tree." This results in the total of 30 instances of "random trees."

Next, we compare each pair of "random trees." Additionally, we compare our phylogenetic trees with all 30 corresponding "random trees." To compare two phylogenetic trees, we use the *patristic distance*[1]. Typically, this distance is used to compute evolutionary distances between species in real phylogenetic trees. Note that this distance takes into account both

[1]http://www.mathworks.com/access/helpdesk/help/toolbox/bioinfo/index.html

the similarities between species (which are measured by EC scores of the GRAAL alignments between their metabolic networks) and the topology of the phylogenetic tree. By using the patristic distance, we map each tree into a vector containing pairwise distances between its leaves; note that leaves correspond to species in non-random trees and we always use consistent numbering/ordering of leaves. To measure similarities between two trees, we calculate distances between their corresponding vectors x and y containing pairwise patristic distances as: $d(x, y) = norm(x - y)$. When computing distances between "random trees," we use the "infinity norm", $norm_{inf}(x_1, ..., x_n) = \max_i(x_i)$. However, when computing distances between our real trees and "random trees", we use "$-$infinity norm" ($norm_{-inf}(x_1, ..., x_n) = \min_i(x_i)$). We do this in such a way to find reliable upper bounds for p-values.

For protists, the average pairwise distance between all pairs of 30 "random trees" is $\mu = 0.0494$ and their standard deviation is $\sigma^2 = 2.5 \times 10^{-4}$. The average distance between the real phylogenetic tree for protist constructed from GRAAL metabolic network alignments and the 30 "random trees" is $X = 0.4931$. For fungi, the average pairwise distance between all pairs of 30 "random trees" is $\mu = 0.0791$ and standard deviation is $\sigma^2 = 9 \times 10^{-4}$. The average distance between the real phylogenetic tree for fungi constructed from GRAAL metabolic network alignments and 30 "random trees" is $X = 0.3941$. To compute the upper bounds of p-values for the above presented results, i.e., to find how likely it is to generate our real trees from the random model, we use Chebyshev's inequality (2). By choosing an appropriate value for α, we calculate that the probability of generating at random the phylogenetic tree for protists is $\leq 1.3 \times 10^{-3}$, and the probability of generating at random the phylogenetic tree for fungi is $\leq 9.1 \times 10^{-3}$.

To compare GRAAL with IsoRank[6], we construct (in the same way as described above) phylogenetic trees for protists and fungi based on EC scores of IsoRank's alignments of their metabolic networks. (To perform a fair comparison, we use only network topological informaton in IsoRank's cost function; no protein sequence information is used.) Trees for protists and fungi resulting from IsoRank's alignments are presented in Supplementary Figures 5 and 6, respectively. These phylogenetic trees have decent p-values ($\leq 6.4 \times 10^{-2}$ for protists and $\leq 2.9 \times 10^{-2}$ for fungi) but are quite different than those created from genetic sequence alignments; this is especially true for fungi. GRAAL's maximum EC is 67.96% for fungi and 76.72% for protists, while IsoRank's maximum EC is only 11.9% for both fungi and protists. Similarly, GRAAL's minimum EC is 36.50% for fungi and 29.55% for protists, while IsoRank's minimum EC is only 2.06% for fungi and 2.52% for protists. (Note that the maximum (minimum) EC scores produced by the two algorithms, GRAAL and IsoRank, are not obtained for aligning the same pair of organisms.) Clearly, GRAAL's alignments of protists' and fungi's metabolic networks are of higher topological quality than IsoRank's alignments.

2 Evaluation of GRAAL Algorithm

To measure the performance of GRAAL, we align the largest connected component of the high-confidence yeast *S. cerevisiae* PPI network [7], consisting of 8,323 interactions amongst 1,004 proteins, with its synthetic (or "noisy") counterparts obtained by: (1) random removal of nodes from the network; (2) random removal of edges from the network; (3) random addition of edges to the network; (4) addition of low-confidence PPIs to the network (thus, this is the addition of real data to the network, but the data is of low confidence). For each of these four "noise types," we run experiments with different percentage of added noise: 5%, 10%, 15%, 20% and 25%. Due to randomness, we run each experiment 30 times and average results over the 30 runs. E.g., for experiment of type (1) above and the 5% noise level, we randomize the data 30 times by random removal of 5% of nodes from the data network; this results in 30 randomized networks and we align each of them with the original data network to obtain 30 edge correctness (EC) scores that we average. The only exception is noise type (4) above, the additions of low-confidence PPIs: no randomness exists for this noise type, since top k% most confident PPIs ($k = 5, 10, 15, 20, 25$) from the lower-confidence data set are added. Since we know the true node mappings between the two networks (since the networks have the same node labels), we are able to report all three alignment statistics: node correctness (NC), edge correctness (EC), and interaction correctness (IC) (Section 1.3). The results are presented in Supplementary Figures 7 and 8.

With these tests, we demonstrate that, given two very similar networks, our algorithm is capable of discovering high-quality alignments. For 5% random node removal, we obtain NC, EC, and IC of about 70%, 90%, and 55%, respectively (Supplementary Figure 7 (A)). For 5% random edge removal, we obtain NC, EC, and IC of about 65%, 75%, and 45%, respectively (Supplementary Figure 7 (B)). For 5% of random edge addition, we obtain NC, EC, and IC of about 50%, 70%, and 60%, respectively (Supplementary Figure 8). For 5% of low-confidence edge addition, we obtain NC, EC, and IC of about 70%, 90%, and 65%, respectively (Supplementary Figure 8). Hence, if the networks that we are aligning are similar (i.e, they differ in only 5% of nodes or edges), our algorithm can align them well. For all four noise types, as the percentage of noise increases, NC, EC, and IC decrease, as expected (Supplementary Figures 7 and 8). Also, for all four noise types, we obtain higher EC than NC and IC; this is expected, since EC is a less strict measure of alignment quality than NC and IC, as it does not require us to know the correct node mapping (Section 1.3).

References

[1] Milenković, T. and Pržulj, N. *Cancer Informatics* **6**, 257–273 (2008).

[2] Cook, S. In *Proc. 3rd Ann. ACM Symp. on Theory of Computing: 1971; New York*, 151–158. Assosiation for Computing Machinery, (1971).

[3] Pržulj, N. *Analyzing Large Biological Networks: Protein-Protein Interactions Example*. PhD thesis, University of Toronto, Canada, (2005).

[4] Vysochanskij, D. F. and Petunin, Y. I. *Theory of Probability and Mathematical Statistics* **21**, 25–36 (1980).

[5] Keeling, P., Luker, M., and Palmer, J. *Mol. Biol. Evol.* **17**(1), 23–31 (2000).

[6] Singh, R., Xu, J., and Berger, B. In *Research in Computational Molecular Biology*, 16–31. Springer (2007).

[7] Collins, S., Kemmeren, P., Zhao, X., Greenblatt, J., Spencer, F., Holstege, F., Weissman, J., and Krogan, N. *Molecular and Cellular Proteomics* **6**(3), 439–450 (2008).

[8] Pennisi, E. *Science* **300**(5626), 1692 – 1697 (2003).

Figure 1: Edge correctness scores for aligning all six possible pairs of three yeast and two human PPI networks analyzed in this study.

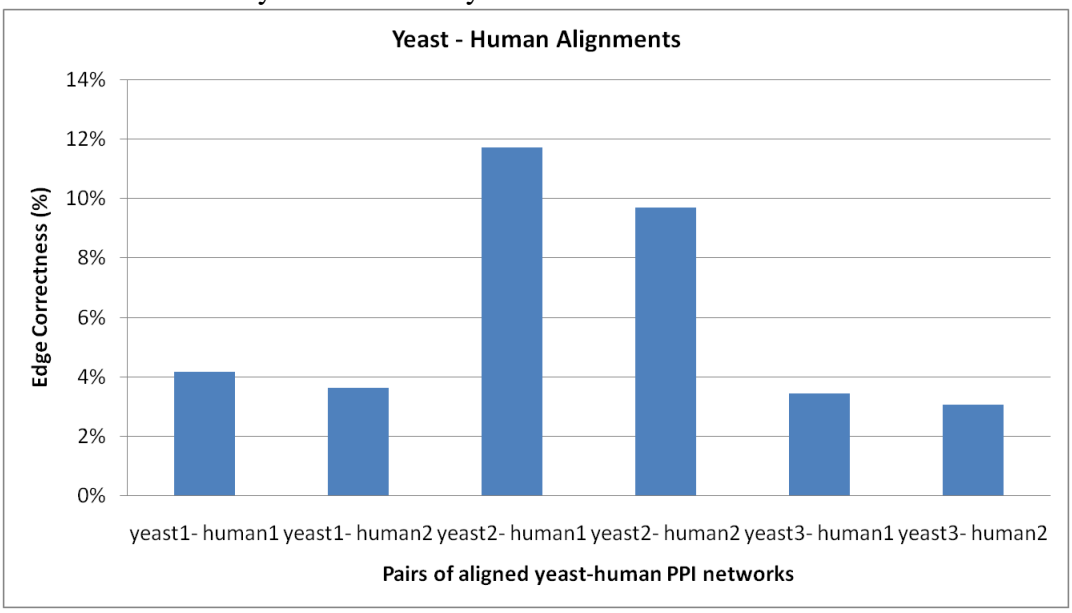


Figure 2: The common subgraph in yeast2 and human1 PPI networks that is identified by GRAAL. The list of connected components, represented by (number of interactions, number of nodes) is: (900,267), (286,52), (127,21), (45,10), (33,24), (30,14), (21,7), (21,17), (19,7), (16,7), (13,12), and (13,6); there exist additional components with fewer than 10 interactions, the majority of which are 2-4-node components.

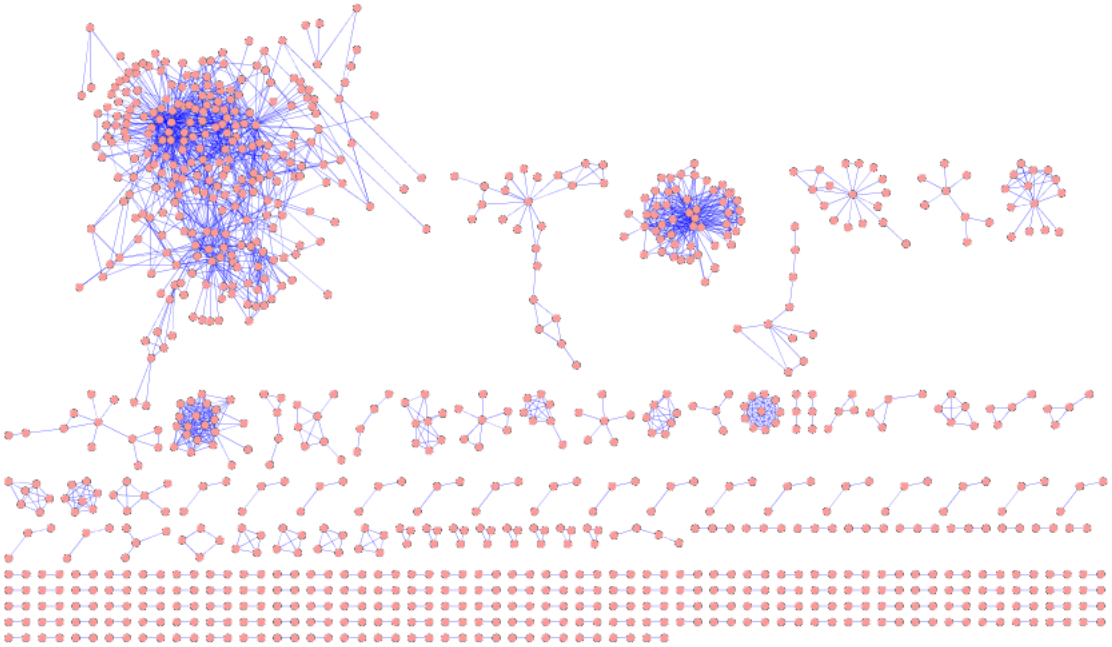
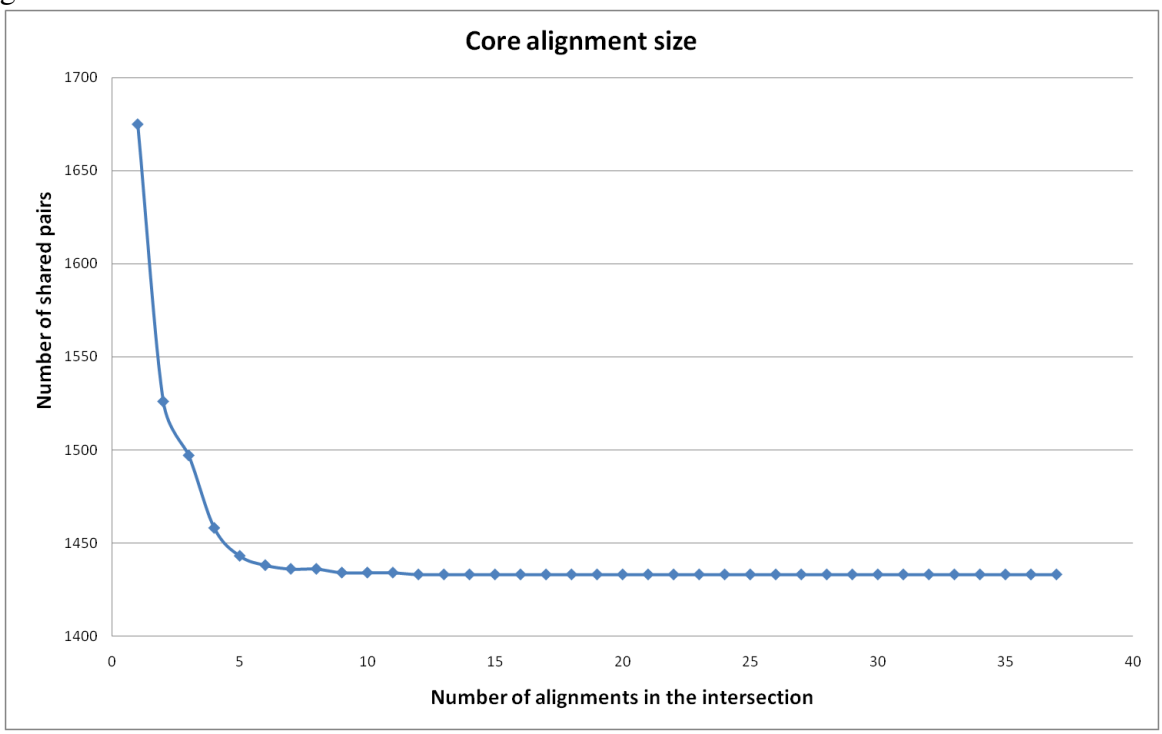


Figure 3: Number of aligned yeast-human pairs in the intersection of several alignments corresponding to different runs of the algorithm for the same α of 0.8. The horizontal axis denotes the number of alignments being intersected and the vertical axis denotes the number of pairs in a given intersection.



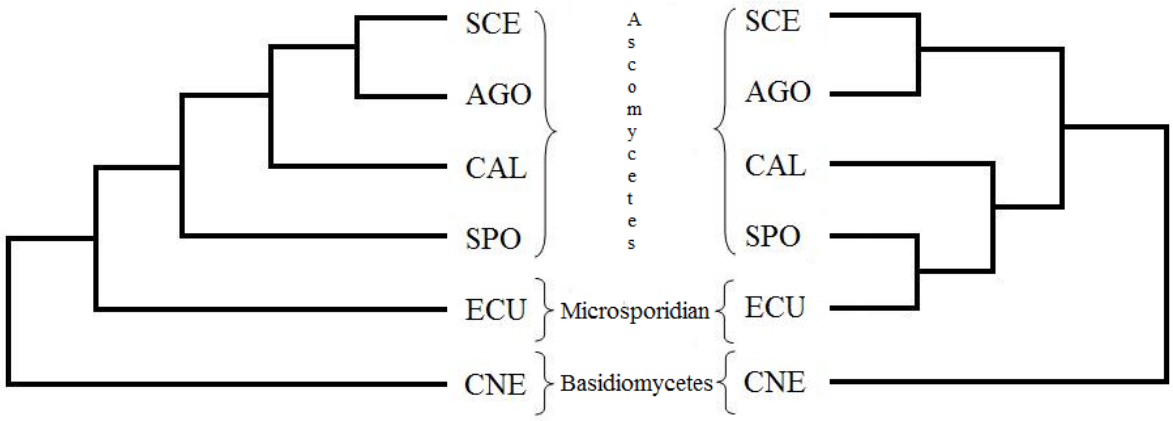


Figure 4: Comparison of the phylogenetic trees for fungi obtained by genetic sequence alignments and by GRAAL's metabolic network alignments. Left: The tree obtained from genetic sequence comparison[8]. Right: The tree obtained from GRAAL. The following abbreviations are used for species: AGO - Ashbya gossypii (Eremothecium gossypii), CAL - Candida albicans, CNE - Cryptococcus neoformans, ECU - Encephalitozoon cuniculi, SCE - Saccharomyces cerevisiae, SPO - Schizosaccharomyces pombe. The species are grouped into the following classes: "Ascomycetes," "Microsporidian," and "Basidiomycetes."

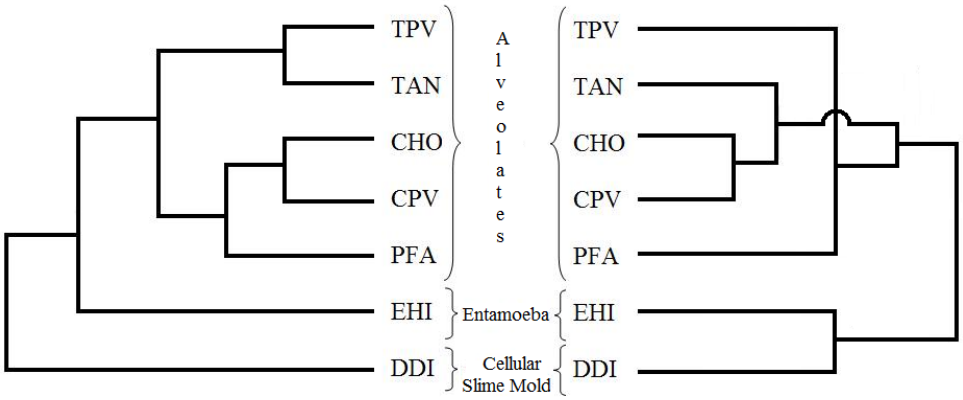


Figure 5: Comparison of the phylogenetic trees for protists obtained by genetic sequence alignments and by IsoRank's metabolic network alignments. Left: The tree obtained from genetic sequence comparison[8]. Right: The tree obtained from IsoRank. The following abbreviations are used for species: CHO - Cryptosporidium hominis, DDI - Dictyostelium discoideum, CPV - Cryptosporidium parvum, PFA - Plasmodium falciparum, EHI - Entamoeba histolytica, TAN - Theileria annulata, TPV - Theileria parva. The species are grouped into the following classes: "Alveolates," "Entamoeba," and "Cellular Slime mold."

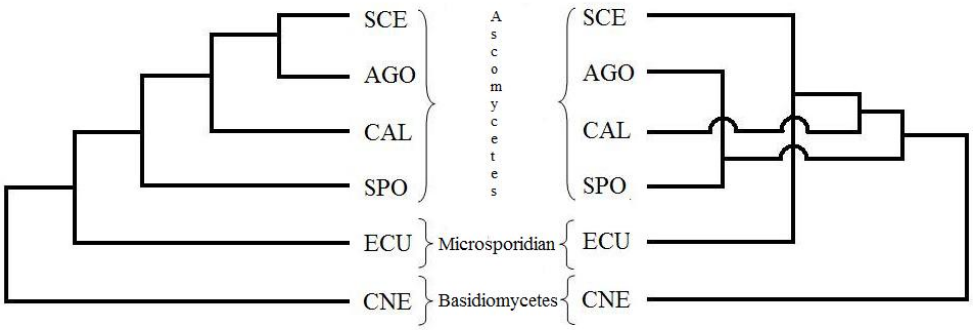


Figure 6: Comparison of the phylogenetic trees for fungi obtained by genetic sequence alignments and by IsoRank's metabolic network alignments. Left: The tree obtained from genetic sequence comparison[8]. Right: The tree obtained from IsoRank. The following abbreviations are used for species: AGO - Ashbya gossypii (Eremothecium gossypii), CAL - Candida albicans, CNE - Cryptococcus neoformans, ECU - Encephalitozoon cuniculi, SCE - Saccharomyces cerevisiae, SPO - Schizosaccharomyces pombe. The species are grouped into the following classes: "Ascomycetes," "Microsporidian," and "Basidiomycetes."

Figure 7: Node, edge, and interaction correctness scores for aligning the yeast PPI network with its synthetic networks obtained by: (A) random deletion of a percentage of nodes from the yeast PPI network; (B) random deletion of a percentage of edges from the yeast PPI network. These percentages of noise are presented on x-axes.

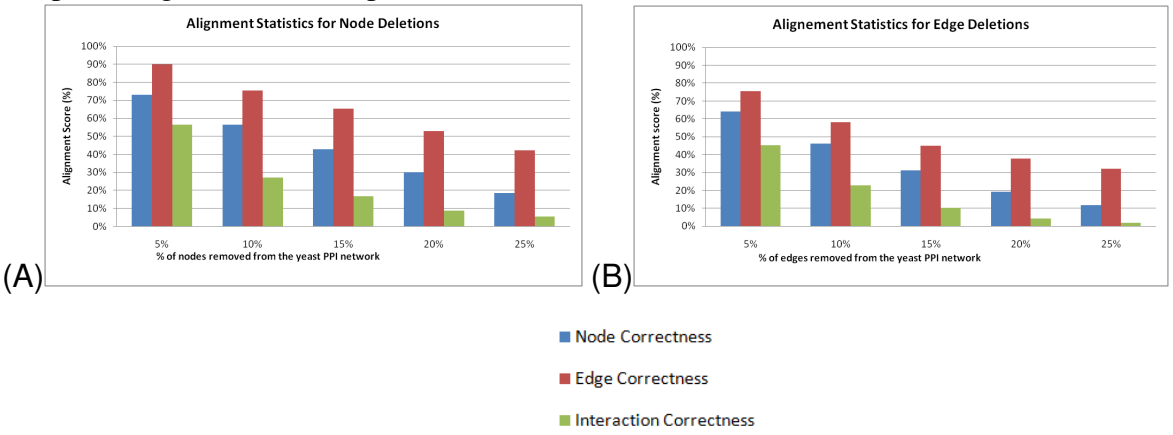


Figure 8: Node, edge, and interaction correctness scores for aligning the yeast PPI network with syntetic networks obtained by: addition of low-confidence interactions to the yeast PPI network (blue); and random addition of edges to the yeast PPI network (red). The percentages of noise are presented on x-axes.