BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 13 2013, pages 1654–1662
doi:10.1093/bioinformatics/btt202

Data and text mining

Advance Access publication May 21, 2013

NETAL: a new graph-based method for global alignment of protein–protein interaction networks

Behnam Neyshabur[1], Ahmadreza Khadem[1], Somaye Hashemifar[2] and Seyed Shahriar Arab[3,4,*]

[1]Department of Computer Engineering, Sharif University of Technology, Tehran, Iran, [2]School of Computer Science, University of Tehran, Tehran, Iran, [3]Department of Biophysics, Faculty of Biological Sciences, Tarbiat Modares University, Tehran, Iran and [4]Bioinformatics Department, School of Computer Science, Institute for Research in Fundamental Science (IPM), Tehran, Iran

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: The interactions among proteins and the resulting networks of such interactions have a central role in cell biology. Aligning these networks gives us important information, such as conserved complexes and evolutionary relationships. Although there have been several publications on the global alignment of protein networks; however, none of proposed methods are able to produce a highly conserved and meaningful alignment. Moreover, time complexity of current algorithms makes them impossible to use for multiple alignment of several large networks together.

Results: We present a novel algorithm for the global alignment of protein–protein interaction networks. It uses a greedy method, based on the alignment scoring matrix, which is derived from both biological and topological information of input networks to find the best global network alignment. NETAL outperforms other global alignment methods in terms of several measurements, such as Edge Correctness, Largest Common Connected Subgraphs and the number of common Gene Ontology terms between aligned proteins. As the running time of NETAL is much less than other available methods, NETAL can be easily expanded to multiple alignment algorithm. Furthermore, NETAL overpowers all other existing algorithms in term of performance so that the short running time of NETAL allowed us to implement it as the first server for global alignment of protein–protein interaction networks.

Availability: Binaries supported on linux are freely available for download at http://www.bioinf.cs.ipm.ir/software/netal.

Contact: sh.arab@modares.ac.ir

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on November 12, 2012; revised on April 11, 2013; accepted on April 23, 2013

1 INTRODUCTION

The interactions among proteins are of high importance within the majority of biological functions. A protein may interact with another protein for the sake of modification; it also may interact to form part of a protein complex or it may carry another protein (Kayarkar *et al.*, 2009). There are numerous experimental techniques including yeast two-hybrid (Fossum *et al.*, 2009; Parrish *et al.*, 2007; Stelzl *et al.*, 2005) and protein co-immunoprecipitation (Aebersold and Mann, 2003) that demonstrate large-scale protein–protein interactions (PPIs) for organisms, including *Saccharomyces cerevisiae* (Collins *et al.*, 2007), *Homo sapiens* (Radivojac *et al.*, 2008) and *Drosophila melanogaster* (Giot *et al.*, 2003). As the data from these experiments have a lot of false-positive interactions, putting several studies containing other data types, such as gene expression into account, they have been used to reduce the number of such interactions (von Mering *et al.*, 2002).

Such an analysis is the alignment between networks that identify functional or structural conserved components in species. The purpose of network alignment is to align nodes of the input networks to maximize the overall match between them. This alignment not only gives the global similarities between the networks of species but also yields knowledge about the topology of the proteins in the networks that is valuable in evolutionary biology. Given a group of such PPI networks, the scores of the pairwise alignment between them can be used to construct their phylogenetic tree.

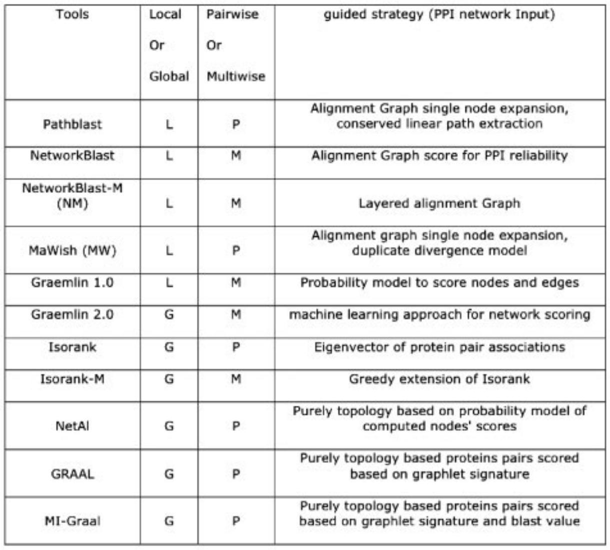
Most of the previous approaches focused on local network alignment (LNA). LNAs find local similar subnetworks that are probably conserved components either functional or structural. PathBLAST (Kelley *et al.*, 2003), NetworkBLAST (Sharan *et al.*, 2005), Mawish (Koyuturk *et al.*, 2006) and Graemlin 1.0 are examples of local network alignment. PathBLAST uses both BLAST similarities of the proteins and the probabilities of interactions to find the biological pathways. These probabilities indicate how much an interaction is true and is not false positive (Kuchaiev *et al.*, 2010). NetworkBLAST generates a network alignment graph based on the sequence similarities and performs a search over the network to identify conserved pathways and clusters. Mawish models the biological deletion and duplication and uses the weighted edge of the network to find the maximum weight induced subgraph. Graemlin1.0 (Flannick *et al.*, 2006) uses a scoring scheme based on two models, in one of which the module is subject to evolutionary constraints; however, in other proteins, the modules are under no constraints.

As mapping of LNA is categorized as one-to-many mapping, it may be ambiguous. In other words, in this kind of alignment, one node can be aligned with different nodes in different local

*To whom correspondence should be addressed.

conserved subnetworks. Although this may indicate gene duplication, in some cases, it may cause implausibly numerous matches for a protein (Singh *et al.*, 2007). Previous LNA algorithms have not generally been able to detect large connected subgraphs that are conserved during the evolution (Kelley *et al.*, 2003). In contrast, the purpose of global alignment is to find a unique correspondence between all nodes of the input networks. Such an overall alignment can result in detection of functional orthologs and cross-species variations (Singh *et al.*, 2008). The pioneering work in this area was carried out by IsoRank (Singh *et al.*, 2007) that mapped the nodes of two input networks based on the similarity of their neighborhood topology. It uses a preprocessed matrix that indicates the efficiency of aligning each node of the network to all others. Afterward, IsoRankN was proposed for aligning multiple networks (Singh *et al.*, 2008). This algorithm uses original IsoRank to find the alignment scores between any pair of networks. It uses a developed PageRank-Nibble algorithm (Andersen *et al.*, 2006) to infer the alignment clusters based on these scores. Graemlin 2.0 (Fossum *et al.*, 2009) is a parameter-learning algorithm that finds the alignment between multiple networks by relying on their phylogenetic relationships. GA and PATH are based on one and two relaxations over the set of doubly stochastic matrices (Zaslavskiy *et al.*, 2009). They use the same objective function that balances the matching similar pairs with increasing the number of conserved edges. PISwap (Chindelevitch *et al.*, 2010) uses a local optimization to find the optimal alignment. This algorithm initially finds an alignment using only sequence data and then adjusts it by incorporating topological information. Natalie (Klau, 2009) is a lagrangian relaxation approach that uses homology information with a branch-and-bound method to compute the global alignment. HopeMap (Tian and Samatova, 2009) is a parameter-free algorithm that iteratively refines the conserved regions by applying a connected-component–based process. GRAAL (Kuchaiev *et al.*, 2010) and H-GRAAL (Milenkovic *et al.*, 2010) rely solely on topological network similarities based on the graphlet degree vector. Although GRAAL is a greedy seed and extend approach, H-GRAAL uses Hungarian algorithm (Mills-Tettey *et al.*, 2007) to find the optimal alignment. MI-GRAAL (Kuchaiev and Przulj, 2011) uses both node similarity and topological network similarity measures to break the ties and find a more stable alignment.

In Figure 1 one can see the categorization of different tools, including NETAL, which shows how the strategies are different. Broadly, network alignment algorithms (local or global) can be categorized into two groups: pairwise alignment and multiple alignment. Pairwise alignment algorithms align two PPI networks. Mawish, Natalie, HopeMap, PISwap, IsoRank, GRAAL, H-GRAAL and MI-GRAAL are examples of pairwise alignment algorithms. However, multiple alignment algorithms try to create an alignment between more than two PPI networks. IsoRankN, NetworkBLAST and Graemlin are in this group. Previous algorithms for the problem of PPI network alignment use two main approaches for scoring the alignments. Some of the algorithms such as GRAAL use solely the topology of networks, whereas most of the alignment methods like Mawish, NetworkBLAST, IsoRankN and Graemlin incorporate previous biological information about nodes, such as the similarities of protein sequences and phylogenetic relationships in PPI networks. In this article, we propose a novel algorithm, NETAL (network aligner), as a solution to the pairwise global PPI network alignment. This problem can be formulated as follows (Kuchaiev and Przulj, 2011): given two networks, find an injective mapping so each node in the smaller network is mapped to one node in the larger network. All the details of our algorithm are discussed on pairwise alignment, but it can be easily promoted to multiple alignment. This algorithm uses a scoring function based on both biological and topological information. The topological information between each pair of nodes is updated gradually during execution. The main advantages of our proposed method are finding the best global PPI alignment and the ability to find big connected common subgraph that is so useful for extracting the biological information. We compare the results of our program with MI-GRAAL, GRAAL and IsoRank to demonstrate the effectiveness and usability of our method. The reason for choosing these tools for comparison is because of the accuracy of their results in comparison with other existing tools that have been used so far.



Tools	Local Or Global	Pairwise Or Multiwise	guided strategy (PPI network Input)
Pathblast	L	P	Alignment Graph single node expansion, conserved linear path extraction
NetworkBlast	L	M	Alignment Graph score for PPI reliability
NetworkBlast-M (NM)	L	M	Layered alignment Graph
MaWish (MW)	L	P	Alignment graph single node expansion, duplicate divergence model
Graemlin 1.0	L	M	Probability model to score nodes and edges
Graemlin 2.0	G	M	machine learning approach for network scoring
Isorank	G	P	Eigenvector of protein pair associations
Isorank-M	G	M	Greedy extension of Isorank
NetAl	G	P	Purely topology based on probability model of computed nodes' scores
GRAAL	G	P	Purely topology based proteins pairs scored based on graphlet signature
MI-Graal	G	P	Purely topology based proteins pairs scored based on graphlet signature and blast value

Fig. 1. A synopsis on network alignment tools

2 METHODOLOGY

2.1 Definitions and notations

In this article, each PPI network is represented as a simple undirected graph $G = (V, E)$ such that each $u \in V$ is a protein v and for each $e \in E$, $e = (i, j)$ is a notation for an interaction between proteins i and j. Apparently, $|V|$ and $|E|$ are the number of nodes and edges in the graph. In addition, the edge weight $w(e)$ is associated with the edge e where $0 \leq w(e) \leq 1$. $N(i)$ demonstrates the set of neighbors of node i; therefore, $|N(i)|$ indicates the degree of node i. A subnetwork of G is a graph $H = (V', E')$ in which $V' \subseteq V$ and $E' \subseteq E$. Let $G_b = (V_b, E_b)$ be a complete bipartite graph, i.e. $V_b = V_1 \cup V_2$, $V_1 \cap V_2 = \emptyset$, $E_b = \{(u, v) \mid u \in V_1, v \in V_2\}$ and suppose that $|V_1| \leq |V_2|$. A *bipartite matching* $M \subseteq E_b$ is a set of edges so that every $u \in V_1$ is incident to at most one edge of M.

Now, suppose we have two networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $|V_1| \leq |V_2|$. Global alignment of these two networks is an injective function $f : V_1 \rightarrow V_2$. There are several measurement methods for assessing the quality of an alignment. We will discuss them in Section 3.

Let M be a bipartite matching for the complete graph $G = (V_1 \cup V_2, E)$ where V_1 is the set of nodes of network A, and V_2 is the set of nodes of network B and let (i, j) be an interaction in network A. We call (i, j) a *conserved interaction* or *conserved edge* when the end points of (i, j) are matched with the end points of an interaction (i', j') of network B.

2.2 NETAL approach

Our computational tool, NETAL, takes two networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ in addition to various configuration parameters as inputs and returns global alignment of them. Without loss of generality, we assume that $|V_1| \leq |V_2|$. This approach includes two main phases. First phase is the construction of alignment score matrix, and the second one is the greedy approach of updating of scores until we get to the final result.

At the first phase *Alignment Score Matrix* is generated. This matrix is constructed based on two other matrices named *Similarity Score Matrix* and *Interaction Score Matrix*. Similarity score indicates both topological and biological similarities between every two nodes $i \in V_1$ and $j \in V_2$. This matrix is a weighted sum of *Topological Score Matrix* and *Biological Score Matrix*. *Topological Score Matrix* and *Biological Score Matrix* indicate topological and biological similarities between every two nodes of input networks, respectively. Interaction score of two nodes $i \in V_1$ and $j \in V_2$ is an estimate on the conditional expected value of the number of conserved interactions that are incident to the node i given the assumption that nodes i and j are aligned to each other. Similarity score matrix that is calculated in phase one will not change during the process of finding alignments, and they remain fixed until the end. Therefore, we can consider this part as an offline pre-process for the continuation of the algorithm. The reason is that the values of the similarity matrix are computed based on structure of the networks and biological properties of proteins in networks, which both are available at the beginning. However, interaction score matrix should be updated iteratively during the next phase of the algorithm because the expected value of the number of conserved interactions changes after aligning each two nodes. Clearly, when interaction scores are updated, the values of alignment score matrix should also be renewed, respectively.

At the second phase, a greedy search is used to find the global alignment based on the values of the alignment score matrix. In each iteration of the greedy search, we find the two nodes with maximum score in alignment score matrix and then align them. This greedy search is repeated until all nodes of the first network are aligned with the nodes of the second network.

Algorithm 1 presents general steps of NETAL. For more specific pseudocodes, see the Supplementary Data.

Algorithm 1. NETAL(G_1, G_2)

```
Require: $|V_1| \leq |V_2|$
  Construct topological score matrix $T$.
  Given biological data, construct biological score matrix $B$.
  Given matrices $T$ and $B$, construct similarity score matrix $S$.
  Construct interaction score matrix $I$.
  Given $S$ and $I$, construct alignment score matrix $A$.
  while there is an unaligned node in $\in V_1$ do
    Find unaligned nodes $i \in V_1, j \in V_2$ with maximum $A(i,j)$.
    $f(i) \leftarrow j$
    Update interaction score matrix $I$.
    Given $S$ and $I$, update Alignment score matrix $A$.
  end while
  return f
```

2.2.1 Topological score matrix

Topological score matrix T consists of $|V_1|$ rows and $|V_2|$ columns, whereas $T(i,j)$ indicates topological similarity between the nodes $i \in V_1$ and $j \in V_2$. Our major assumption is that two nodes are topologically similar if and only if their neighbors are topologically similar.

As you can see in the example provided in Figure 2a, we initiate $T^0(i,j) = 1$, and in each iteration, we update T based on its values on the last iterate. To compute $T^{t+1}(i,j)$, we construct a complete weighted bipartite graph $G_b = (V_b, E_b)$ where V_b is made of two disjoint sets of nodes $N(i)$ and $N(j)$ and every edge $(i', j') \in E_b$ connects node $i' \in N(i)$ and node $j' \in N(j)$. Furthermore, for each edge $(i', j') \in E_b$ we set $w(i', j') = T^t(i', j')$.

After constructing G_b, a matching M will be found using a greedy algorithm. Initially, we select an edge $e = (u, v)$ so that for every $(i', j') \in E_b$ we have $w(u, v) \geq w(i', j')$. Then, we add e to M and remove u, v, and all edges that are incident to u or v in G_b. This process is repeatedly done while traversing all the graphs until all the edges between the two parts of graph are removed. Now we can calculate $T^{t+1}(i,j)$ using formula 1:

$$T^{t+1}(i,j) = \frac{\sum_{(u,v)\in M} T^t(u,v)}{\max\{|N(i)|, |N(j)|\}} \tag{1}$$

where t is the counter for the iteration. Looking at the right-hand side of Equation (1), the numerator is sum over similarity between the matched neighbors. The Figure 2b illustrates how the topological matrix updates for four iterations in our simple graph. If for each $u \in N(i)$ and $v \in N(j)$, $T(u, v) \leq 1$, we can conclude that the enumerator is less than or equal to $min\{|N(i)||N(j)|\}$. Inverse impact of $\max\{|N(i)|, |N(j)|\}$ in the denominator not only keeps T^{t+1} between 0 and 1 but also assigns higher similarity score to the vertices with similar degrees. For instance, if $N(i) = 3$ and $N(j) = 5$, then $T(i,j)$ is no more than 3/5 in any iteration.

After a few iterations, the final values of matrix T are considered as topological similarities between every pairs of nodes (the number of iterations is an input parameter and usually two or three iterations are enough).

2.2.2 Biological score matrix

In this work, we have not used biological score impact in evaluation, which we implement in our future works. But in a nutshell, the biological score matrix B is defined such that $B(i,j)$ is referred to as biological similarity between two nodes $i \in V_1$ and $j \in V_2$. We say two nodes i and j are biologically similar, if and only if the two following conditions hold:

(i) The actual proteins represented by i and j are biologically similar.
(ii) The actual proteins represented by the neighbors of i and j are biologically similar.

$$B^0(i,j) = \frac{P(i', j')}{\max_{u\in V_1,\, v\in V_2}\{P(u, v)\}} \tag{2}$$

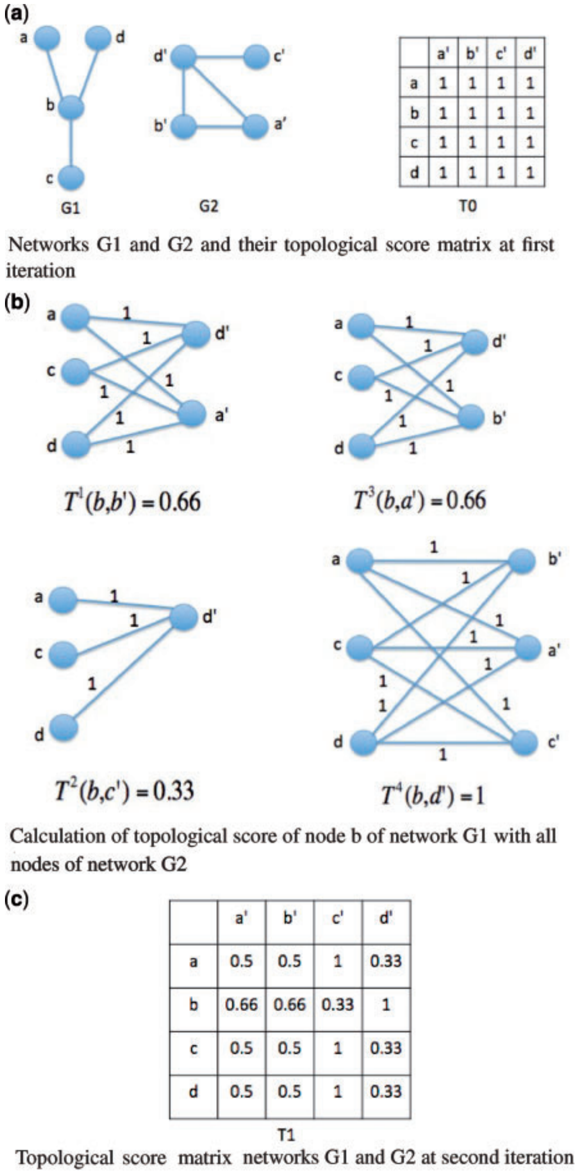
$$B^{t+1}(i,j) = \beta(B^0(i,j)) + (1 - \beta)\frac{\sum_{(i',j')\in M} B^t(i', j')}{\max\{|N(i)||N(j)|\}} \tag{3}$$

2.2.3 Similarity score matrix

Similarity score matrix S with $|V_1|$ rows and $|V_2|$ columns, indicates the similarity between nodes of two networks, i.e. $S(i,j)$ is the similarity of nodes i and j where $i \in V_1$ and $j \in V_2$. After computing topological score matrix and biological score matrix, matrix S is computed as follows:

$$S(i,j) = \alpha T(i,j) + (1 - \alpha)B(i,j) \tag{4}$$

where $T(i,j)$ and $B(i,j)$ are the values of topological score matrix and biological score matrix, whereas $0 \leq \alpha \leq 1$ is a parameter that controls the balance between them. As aforementioned, we do not need to update



	a'	b'	c'	d'
a	1	1	1	1
b	1	1	1	1
c	1	1	1	1
d	1	1	1	1

Networks G1 and G2 and their topological score matrix at first iteration

Calculation of topological score of node b of network G1 with all nodes of network G2

	a'	b'	c'	d'
a	0.5	0.5	1	0.33
b	0.66	0.66	0.33	1
c	0.5	0.5	1	0.33
d	0.5	0.5	1	0.33

Topological score matrix networks G1 and G2 at second iteration

Fig. 2. Example of how Topological Score Matrix is calculated

matrix S, and it never changes during the greedy search. The reason is that we compute the similarity score based on the structural similarity of networks and biological similarity between proteins. As the structural and biological similarity of nodes are constant and do not change during the algorithm, we do not need to update similarity score matrix. As we explained in the previous section, we have not specified and implemented the computation of biological score matrix. Therefore, in our current version of software, the parameter α is equal to 1.

2.2.4 Interaction score matrix If $I_{|V_1|\times|V_2|}$ be the interaction score matrix with $|V_1|$ rows and $|V_2|$ columns, $I(i,j)$ indicates an approximation of the expected value for the number of conserved interactions incident to i in the final alignment of the two nodes i and j. In other words, if the two nodes $i \in V_1$ and $j \in V_2$ are matched, $I(i,j)$ interactions will approximately be conserved. Next, we introduce the concept of *dependency*.

Each node i has a dependency to any of its neighbors equal to $\frac{1}{|N(i)|}$. This value indicates the probability that interaction (i, i') will be conserved, if a node $i' \in N(i)$ is aligned with a random node of the other network. For example, consider a node i with three neighbors i', i'' and i'''. Every interaction (i, i'), (i, i'') and (i, i''') will be conserved with the probability $\frac{1}{|N(i)|}$, e.g. $\frac{1}{3}$. Thus, *dependency* of i is equal to $\frac{1}{3}$.

For each node i summing dependencies of its neighbors will give an approximate expected value for the number of conserved interactions that are incident to i, when i is matched to a random node in the other network. In the aforementioned example, let i', i'' and i''' have degrees 4, 2 and 4, respectively. Summing dependencies of neighbors of node i is equal to $\frac{1}{4} + \frac{1}{2} + \frac{1}{4} = 1$. It means that by aligning i with a node of the other network, the expected value of conserved interactions will be one.

Considering the concept of dependency, $I(i,j)$ can be computed as follows:

$$I(i,j) = \frac{\min\left\{\sum_{i' \in N(i)} \frac{1}{|N(i')|}, \sum_{j' \in N(j)} \frac{1}{|N(j')|}\right\}}{\max_{k \in V_1 \cup V_2}\{|N(k)|\}} \qquad (5)$$

The idea of this formula is as follows. In aligning two nodes i and j, the expected value for the number of conserved edges will be equal to the minimum expected value for the number of conserved interactions between the two nodes i and j. As the expected value of conserved interactions for node i is no more than $|N(i)|$, expected values can be scaled by dividing them by the maximum degree among all the nodes of the two input networks.

As two nodes are matched, the expected value of conserved interactions in alignment will change. Therefore, the values of matrix I should be updated. For this purpose, two new matrices are defined:

- *Definite interaction matrix*: Suppose two nodes $i \in V_1$ and $j \in V_2$ are matched together. $D(i,j)$, is the number of definite conserved interactions of the first network that are incident to i. Let two optional nodes u and v be aligned with each other. Because of binding of these two nodes, two situations may happen. In the first situation, if we align $i \in N(u)$ to $j \in N(v)$, the interaction (u, i) will be conserved because it is aligned with (v, j). Thus, the number of definite conserved interactions resulted from this matching will be increased by one, i.e. $D(i,j) = D(i,j) + 1$. In fact, when two nodes u and v are matched, the number of definite conserved interactions resulted from aligning of any pair of nodes i and j where $i \in N(u)$ and $j \in N(v)$, will be increased by one. In the other situation, if we align i to j where $i \notin N(u)$ or $j \notin N(j)$, the number of definite conserved interactions resulted from this matching will not change.

  Therefore, after the matching of u and v, the values of matrix D get updated based on the formula 6. As a reminder, $D(i,j)$ is the number of interactions that will definitely be conserved, if two nodes i and j are matched. Note that the matrix D is zero at first and k is iteration number.

$$D(i,j)^{k+1} = \begin{cases} D^k(i,j) + 1 & i \in N(u), j \in N(v) \\ D(i,j)^k & \text{otherwise} \end{cases} \qquad (6)$$

- *Canceled interaction matrix*: This is a 1D matrix. As aforementioned, every node u has a dependency equal to $\frac{1}{|N(u)|}$ to any of its neighbors. But, after aligning node u to a node v of the other network, this dependency should be removed from the interaction score of any neighbor i of u, similarly to any neighbor j of v. For this purpose, two canceled interaction arrays C_1 and C_2 are used for the networks G_1 and G_2, respectively. Each value $C_1(i)$ or $C_2(j)$ indicates the number of conserved interactions that should be removed from expected value of number of conserved interaction for node i and j. If two nodes u and v are matched, $C_1(i)$ is updated based on the Equation (7). $C_2(j)$ updates similarly. Note that C_1 and C_2 are set to zero at the beginning, and k is iteration number.

$$C_1(i)^{k+1} = \begin{cases} C_1(i)^k + \frac{1}{|N(u)|} & i \in N(u) \\ C_1(i)^k & \text{otherwise} \end{cases} \tag{7}$$

$C_2(j)$ also updates similar to $C_1(i)$. After updating the matrices D, C_1 and C_2, matrix I will acquire its new value using this following formula:

$$I(i,j) = \frac{\min\{d_1(i), d_2(j)\}}{\max_{k \in V_1 \cup V_2}\{|N(k)|\}} \tag{8}$$

where

$$\begin{aligned} d_1(i) &= \sum_{i' \in N(i)} \frac{1}{|N(i')|} - C_1(i) \\ d_2(j) &= \sum_{j' \in N(j)} \frac{1}{|N(j')|} - C_2(j) \end{aligned} \tag{9}$$

2.2.5 Alignment score matrix After calculating matrices S and I, alignment score matrix A is computed based on Equation (10).

$$A(i,j) = \lambda S(i,j) + (1 - \lambda) I(i,j) \tag{10}$$

where $A(i,j)$ is the scores of aligning two nodes $i \in V_1$ and $j \in V_2$. Although computing alignment score matrix, NETAL simultaneously constructs a priority queue of paired nodes in decreasing order of their alignment scores. This priority queue is used in the second phase to quickly identify the pair nodes with the maximum alignment score.

2.2.6 Greedy search In the second phase, our algorithm uses a greedy search to find the best global alignment between two input networks G_1 and G_2. At first, the pair nodes with maximum alignment score are chosen and aligned to each other. Then interaction score matrix is updated, and the alignment score matrix is changed based on the new values of I. Finally, two cited nodes and their corresponding rows and columns are removed from the matrix. Now, two other nodes with the maximum alignment scores are chosen, and similar steps are performed repeatedly. This process will continue until all the nodes of the smaller network are aligned to some nodes of the other network. At this time, the resulted alignment will be considered as the final alignment of the two networks G_1 and G_2.

2.3 Time complexity

Let $n = \max\{|V_1|, |V_2|\}$ and $m = \max\{|E_1||E_2|\}$. Calculation of topological score matrix and biological score matrix takes $O(m^2)$. Computing each of similarity score matrix, interaction score matrix and alignment score matrix can be done in $O(n^2)$. As we use priority queue to find highest scores in the alignment score matrix, we also need to build this priority queue that takes $O(n^2 \log n)$. Therefore, the total time complexity of the first phase of the algorithm is $O(m^2 + n^2 \log n)$. In the second phase, extracting the pair with highest score takes constant time. However, the major time consuming part is updating matrices and the priority queue, which is $O(nm \log n)$. Therefore, the total time complexity of the algorithm is $O(n^2 \log n + m^2 + nm \log n)$. For a simpler time complexity, if we assume that $m \simeq n \log n$ (which is an acceptable assumption in biological networks), then the time complexity of NETAL will be $O(m^2) = O(n^2 \log^2 n)$. For more details about time complexity of the algorithm, please see the Supplementary Data.

3 RESULTS

In this section, our algorithm is compared with other global network alignment algorithms. The comparisons are done based on five criteria. At first, these criteria for the comparison are introduced (Kuchaiev *et al.*, 2010).

- *Edge Correctness* (*EC*) is the percentage of edges (interactions) of the first network that are aligned to edges in the second network. Higher values of the EC indicates that the two input networks are topologically more similar. EC is computed by the following equation (Milenkovic *et al.*, 2010; Singh *et al.*, 2007):

$$EC = \frac{|\{(u,v) \in E_1 : (g(u), g(v)) \in E_2\}|}{|E|} \times 100\% \tag{11}$$

Obviously, if the second network has a subnetwork that is isomorphic to the first network, then EC can be ideally one.

- *Node Correctness* (*NC*) is the percentage of nodes (proteins) of the first network that are aligned to the *correct* nodes of the second network. Let *f* be the correct node mapping and *g* be the alignment mapping; then NC is defined as

$$NC = \frac{|\{u \in V_1 : f(u) = g(u)\}|}{|V_1|} \times 100\% \tag{12}$$

Obviously, to calculate NC, a correct node mapping like *f* should be known.

- *Interaction Correctness* (*IC*) is the percentage of interactions of the first network that are aligned with a correct interaction in the second alignment. IC is defined as

$$IC = \frac{|\{(u,v) \in E_1 : (f(u), f(v)) \in E_1, (u,v) \in A\}|}{|E|} \times 100\% \tag{13}$$

where $A = \{(u,v) \in E_1 | f(u) = g(u), f(v) = g(v)\}$. Like NC, for calculating IC, a correct node mapping like *f* should be known.

- *Largest Common Connected Subgraph* (*LCCS*) is largest connected subgraph of the first network that is isomorphic to a subgraph of the second network. This common subgraph is not necessarily induced. The larger and denser connected subgraphs are biologically more valuable.
- *NF* is the number of aligned protein pairs that their functional similarity is >0.5. Such protein pairs are considered to be functionally related. Higher values for NF indicate that the alignment is functionally meaningful. Functional similarity between proteins is defined based on their Gene Ontology (GO) terms. The functional similarity of two proteins is calculated by the method defined by (Schlicker *et al.*, 2006). This method only considers biological process and molecular function terms.
- *GO* terms that exist in different databases, describe some biological characteristics of the proteins, such as molecular function, biological process and cellular component. If both proteins corresponding to the aligned pair of nodes share common GO terms, then it means that the aligned proteins are functionally similar. More common GO terms express that two proteins of the aligned pair are more similar to each other. A large number of such pairs in the resulted alignment supports that the alignment is biologically credible.

In the next section, the NETAL algorithm is compared with the other global alignment algorithms, such as IsoRank, GRAAL and MI-GRAAL, on the basis of the mentioned criteria. The reason for choosing these tools is their availability and the accuracy of their results in comparison with other existing tools.

Although IsoRankN and Graemlin are also popular and renowned algorithms for global network alignment, they are not included in our assessments. Output of IsoRankN algorithm is not compatible with ours for which we excluded from our comparisons. It outputs disjoint sets of aligned proteins that each set may contain a node of one network that is aligned to several nodes of the other network. In other words, it gives a one-to-many mapping between the nodes. Moreover, the input information of Graemlin is somehow different with other alignment methods. For example, it takes the phylogenetic relationships among the species corresponding to input networks. Some other algorithms, such as GA and PATH, are not able to align huge networks to each other, including yeast and human. Furthermore, some other methods like Natalie and HopeMap need homology information to align networks (Kuchaiev and Przulj, 2011).

3.1 Alignment of two PPI networks human and yeast

NETAL is used to align two PPI networks of yeast and human. We extracted the needed information and datasets of PPI networks of yeast and human from (Collins *et al.*, 2007) and (Radivojac *et al.*, 2008), respectively. The yeast network has 2390 nodes and 16 127 edges, and the human network consists of 9141 nodes and 41 456 edges. As we omitted the impact of biological information in our algorithm, to keep our assessments fair, we considered the α equal to one. We experimentally found that the inverse of the number of the nodes of the larger network is a good estimation for parameter λ. Thus, we set λ to 0.0001 for alignment of networks of yeast and human. Moreover, the recursion process of the computing topological score matrix was repeated two times. As we used the same network as (Kuchaiev and Przulj, 2011), we compared our result with the results in this article.

The edge correctness (EC) of the alignment by NETAL is equal to 36.10%. But, the edge correctness of the alignment of PPI networks of human and yeast by IsoRank, GRAAL and MI-GRAAL are equal to 3.89, 11.72 and 23.26%, respectively. This means NETAL is able to find fairly more conserved interactions compared with other algorithms. Moreover, the largest common connected subgraph (LCCS) of our algorithm has 5370 edges (interactions). But the LCCSs resulted from IsoRank, GRAAL and MI-GRAAL have 261, 900 and 3467 edges, respectively. Thus, the NETALs LCCS is 20.57, 6.0 and 1.55 times larger than the LCCS of IsoRank, GRAAL and MI-GRAAL, respectively. These comparisons are shown in Table 1.

As aforementioned, for measuring the biological significance of our algorithm, the number of aligned pair nodes that share common GO is considered. GO annotation data that are used in this article were extracted from the GO database (The Gene Ontology Consortium, 2000). In case of GO terms, the global alignment resulted from NETAL consists of 50.41, 20.27, 7.80, 3.30, 1.47 and 0.96 aligned pairs that have at least one, two, three, four, five and six common GO terms, respectively. Note that in this study, we only consider the pairs that have at least one known GO term. These results are compared with the results of IsoRank, GRAAL and MI-GRAAL in Table 2. It is clear that our algorithm has aligned considerably more pair nodes having common GO terms. These results emphasize that our algorithm efficiently aligns the homologue proteins. On the other hand, as the number of common GO terms increases, NETALs results get significantly higher in comparison with other algorithms. This means that NETAL is more powerful in aligning the proteins with high similarity.

3.2 Additional measurements on alignment of different networks with each other

We align different pairs of protein networks of human, fly, yeast, worm and mouse. Protein interaction networks of these species are extracted from IntAct database (Kerrien *et al.*, 2012). We report EC, LCCS and NF in of the alignment obtained by different algorithms. The results of the alignments using different tools are reported in Figures 3–5, respectively. In the alignments of human–fly, human–yeast, human–worm and fly–yeast, NETALs NF is higher than IsoRank, MI-GRAAL and GRAAL. It means that using just topological information, NETAL is able to align the proteins that are functionally related. In the alignment of human–mouse NETALs, NF is still close to IsoRanks NF. Moreover, the EC and LCCS of NETAL are much better than the other algorithms. These results show that our algorithm is able to find large conserved complexes while preserving the biological similarities as much as possible.

3.3 Statistical significance

To demonstrate the quality of an alignment algorithm, two similar networks are aligned with each other. For this purpose, a *noisy network* is constructed from a real PPI network. In this article, for obtaining the noisy network of a real PPI network, a fraction ρ of edges of the network is removed randomly, and instead, the same number of edges is randomly added. This removal and addition causes the real network and the resulted noisy network not to be subgraphs of each other, as a result the comparison between them will be meaningful.

As other results of MI-GRAAL are fairly better than previous algorithms, in this section, we just compare our results with MI-GRAAL. To measure our algorithms performance and comparing it with algorithm MI-GRAAL, PPI network of yeast is aligned to its noisy model. The experiments are over different values of ρ: 5, 10, 15, 20 and 25%. For each value of ρ, the experiment runs 30 times.

The ECs of the resulted alignment of NETAL for these values of ρ are 86.7, 74.05, 63.31, 51.82 and 47.11%, respectively. These measures are as follows in MI-GRAAL: 51.62, 40.84, 35.21, 31.31 and 27.67%. NETALs NCs are equal to 55.68, 40.23, 25.91, 10.57 and 5.27%, respectively, whereas MI-GRAALs percentages are 20.19, 8.19, 4.02, 2.29 and 1.49%, respectively. Also, NETALs ICs are 54.05, 29.88, 13.75, 4.27 and 1.47%, respectively, comparing with MI-GRAALs percentages of 10.1, 3.19, 1.11, 0.51 and 0.25%. Obviously, the values of EC, NC and IC of NETAL are so higher than the corresponding values of MI-GRAAL.

Thus, it is clear that for different values of ρ, either low or high, NETALs performance is much better than MI-GRAAL. Moreover, this indicates that our algorithm has the capability to discover the best global alignment, where two input networks are

Table 1. EC and LCCS of different global alignment algorithms for yeast and human networks

Method	EC	LCCS
IsoRank	3.89	261
GRAAL	11.72	900
MI-GRAAL	23.26	3467
NETAL	36.10	5370

Table 2. The number of aligned pairs with the minimum specified common GO terms

Method	≥ 1	≥ 2	≥ 3	≥ 4
IsoRank	44.2	14.1	4.1	1.5
GRAAL	45.1	15.6	5.1	2
MI-GRAAL	46.67	14	3.58	1.01
NETAL	50.41	20.27	7.8	3.3

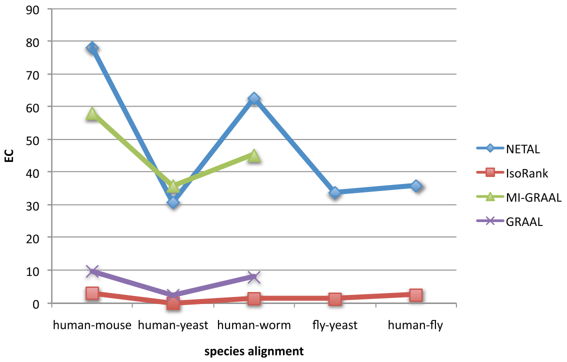


Fig. 3. EC of different alignment tools for different species

so similar. The results are presented in Figures 6–8. As we expected, as the fraction ρ increases, EC, NC and IC decrease.

3.4 Decreasing running time efficiently

Most of the presented algorithms for network alignment are too slow. Although recent algorithms, such as GRAAL and MI-GRAAL, have improved the time complexity, NETAL is still much faster than all of them, as it can be promoted to be used for multiple alignment of large networks. To compare GRAAL, MI-GRAAL and NETAL, we ran them for aligning PPI networks of yeast and human on a 2.66 GHz Linux system with 2 GB random access memory. As it is shown in Table 3, running time of NETAL is much less than the other algorithms. In other words, it is 40.70 and 54.41 times faster than GRAAL and MI-GRAAL, respectively, and this was predictable considering their time complexity.

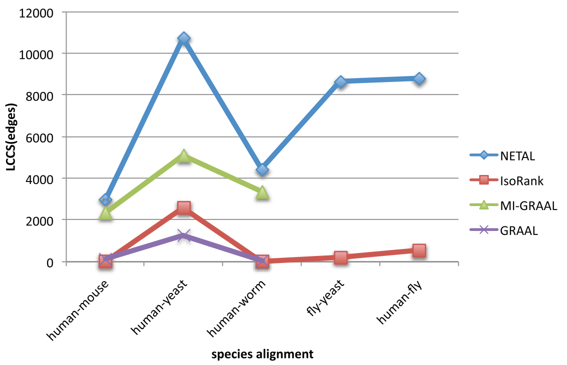


Fig. 4. LCCS of the different alignment tools for different species

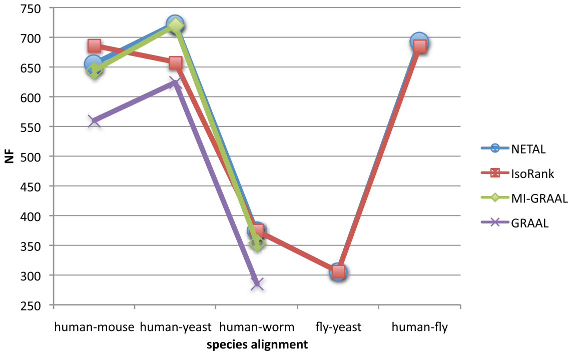


Fig. 5. NF different alignment tools for different species

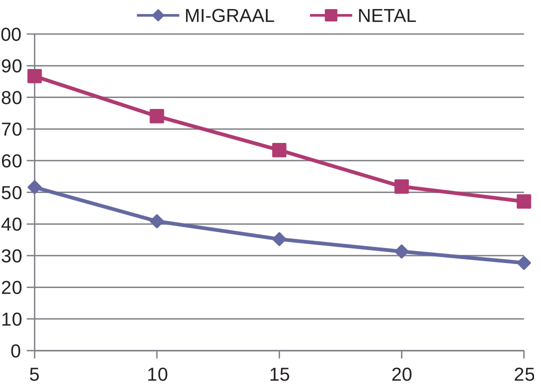


Fig. 6. Edge Correctness (EC) in the noisy for NETAL in comparison with MI-GRAAL

4 DISCUSSION AND CONCLUSIONS

The previous approaches in global alignment of PPI networks use different kinds of topological information of nodes that are calculated based on their neighbors at a pre-processing step, and these scores remain fixed during the algorithm execution. Using stable topological information can not result the best conserved subnetworks. For example, consider that two hubs of two networks are matched together in one step; now, it is reasonable that some neighbors of them are matched together in the next step rather than two nodes that are more similar topologically.

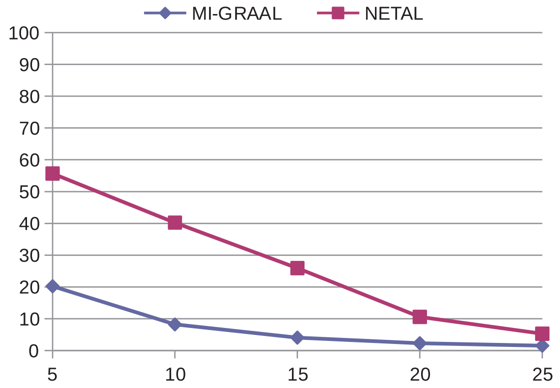


Fig. 7. Node Correctness (NC) in the noisy for NETAL in comparison with MI-GRAAL

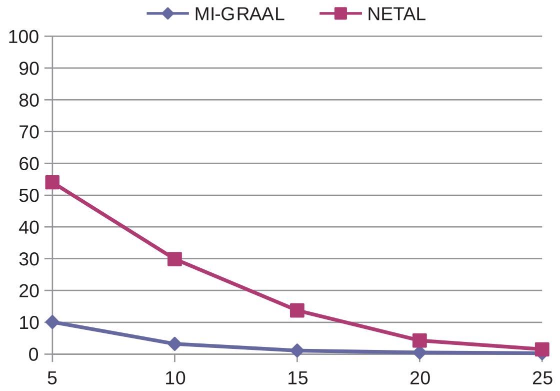


Fig. 8. Interaction Correctness (IC) in the noisy for NETAL in comparison with MI-GRAAL

Table 3. EC and LCCS of different global alignment algorithms for yeast and human networks

Method	Running time (s)	Time complexity
GRAAL	3813.3	$O(n^5)$
MI-GRAAL	5098.3	$O(n^5)$
NETAL	93.7	$O(n^2 \log^2 n)$

Because one of the aims of network alignment is to find the largest and densest connected common subnetworks and clearly by choosing the neighbors of a hub, the probability of finding these subnetworks increases. For this and other reasons, our algorithm uses topological information that is updated during the algorithm. In other words, after aligning each pair of proteins of two input networks, topological scores of remained proteins updates.

In this article, a greedy method is presented for global alignment of PPI networks based on the alignment score matrix. Our method is implemented and tested for aligning PPI networks of yeast and human. The results are compared with three well-known global network alignment algorithms IsoRank, GRAAL and MI-GRAAL. The performance of our method is depicted by comparing our result with the other algorithms. To compare our algorithm with other algorithms, the EC, LCCS and running time measures are used, and common GO terms are applied for comparing biological significance. Moreover, to measure the ability of our algorithm in aligning similar networks, PPI network of yeast is aligned by its noisy model. In our experimental tests, we observe that our algorithm is successful in finding large connected subgraphs and conserved edges in an efficient time comparing with other algorithms. It was also able to find optimal global alignment of yeast and its noisy model.

Funding: This research was in part supported by a grant from IPM (No. CS1385-102).

Conflict of Interest: none declared.

REFERENCES

Aebersold,R. and Mann,M. (2003) Mass spectrometry-based proteomics. *Nature*, **422**, 198–207.

Andersen,R. *et al.* (2006) Local graph partitioning using page rank vectors. In: *Foundations of Computer Science*. IEEE Computer Society, Los Alamitos, CA, USA, pp. 475–486.

Berg,J. *et al.* (2004) Structure and evolution of protein interaction networks: a statistical model for link dynamics and gene duplications. *BMC Evol. Biol.*, **4**, 51.

Chindelevitch,L. *et al.* (2010) Local optimization for global alignment of protein interaction networks. *Pac. Symp. Biocomput.*, **2010**, 123–132.

Collins,S.R. *et al.* (2007) Toward a comprehensive atlas of the physical interactome of *Saccharomyces cerevisiae*. *Mol. Cell Proteomics*, **6**, 439–450.

Flannick,J. *et al.* (2006) Graemlin: general and robust alignment of multiple large interaction networks. *Genome Res.*, **16**, 1169–1181.

Flannick,J. *et al.* (2008) Automatic parameter learning for multiple network alignment. In: *Research in Computational Molecular Biology, Lecture Notes in Computer Science*. Vol. 4955, Springer, Berlin/Heidelberg, pp. 214–231.

Fossum,E. *et al.* (2009) Evolutionarily conserved herpes viral protein interaction networks. *PLoS Pathog.*, **5**, e1000570.

Giot,L. *et al.* (2003) A protein interaction map of *Drosophila melanogaster*. *Science*, **302**, 1727–1736.

Kayarkar,N.A. *et al.* (2009) Protein network in diseases. *Int. J. Drug Discov.*, **1**, 10–17.

Kelley,B.P. *et al.* (2003) Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc. Natl Acad. Sci. USA*, **100**, 11394–11399.

Kelley,B. *et al.* (2004) PathBLAST: a tool for alignment of protein interaction networks. *Nucl. Acids Res.*, **32**, 83–88.

Kerrien,S. *et al.* (2012) The IntAct molecular interaction database in 2012. *BMC Bioinformatics*, **40**, D841–D846.

Klau,G. (2009) A new graph-based method for pairwise global network alignment. *BMC Bioinformatics*, **10**, S59.

Koyuturk,M. *et al.* (2006) Pairwise alignment of protein interaction networks. *J. Comput. Biol.*, **13**, 182–199.

Kuchaiev,O. *et al.* (2010) Topological network alignment uncovers biological function and phylogeny. *J. R. Soc. Interface*, **7**, 1341–1354.

Kuchaiev,O. and Przulj,N. (2011) Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, **27**, 1390–1396.

Milenkovic,T. *et al.* (2010) Optimal network alignment with graphlet degree vectors. *Cancer Informat.*, **9**, 121–137.

Mills-Tettey,G.A. *et al.* (2007) The dynamic Hungarian algorithm for the assignment problem with changing costs. Robotics Institute, Technical Report CMU-RI-TR-07-2.

Parrish,J.R. *et al.* (2007) A proteome-wide protein interaction map for Campylobacter jejuni. *Genome Biol.*, **8**, R130.

Radivojac,P. *et al.* (2008) An integrated approach to inferring gene-disease associations in humans. *Proteins*, **72**, 1030–1037.

Rzhetsky,A. and Gomez,S.M. (2001) Birth of scale-free molecular networks and the number of distinct DNA and protein domains per genome. *Bioinformatics*, **17**, 988–996.

Sharan,R. *et al.* (2005) Conserved patterns of protein interaction in multiple species. *PNAS*, **102**, 1974–1979.

Singh,R. *et al.* (2007) Pairwise global alignment of protein interaction networks by matching neighborhood topology. In: *Research in Computational Molecular Biology*, Vol. 4453, *Lecture Notes in Computer Science*. Springer, Berlin/ Heidelberg, p. 1631.

Singh,R. *et al.* (2008) Global alignment of multiple protein interaction networks. *Proc. Paci. Symposium on Biocomputing*, **13**, 303–314.

Singh,R. *et al.* (2008) Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proc. Natl Acad. of Sci. USA*, **105**, 12763–12768.

Stelzl,U. *et al.* (2005) A human protein-protein interaction network: a resource for annotating the proteome. *Cell*, **122**, 957–968.

The Gene Ontology Consortium. (2000) Gene ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.

Tian,W. and Samatova,N. (2009) Pairwise alignment of interaction networks by fast identication of maximal conserved patterns. In: *Pacific Symposium on Biocomputing*. pp. 99–110.

von Mering,C. *et al.* (2002) Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, **417**, 399–403.

Wagner,A. (2001) The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Molecular Biology and Evolution*, **18**, 1283–1292.

Zaslavskiy,M. *et al.* (2009) Global alignment of protein-protein interaction networks by graph matching methods. *Bioinformatics*, **25**, i259–i267.

Schlicker *et al.* (2006) A new measure for functional similarity of gene products based on Gene Ontology. *BMC Bioinformatics*, **7**, 302.