Aligning Biomolecular Networks Using Modular Graph Kernels

Fadi Towfic[1,2,⋆], M. Heather West Greenlee[1,3], and Vasant Honavar[1,2]

[1]Bioinformatics and Computational Biology Graduate Program
[2]Department of Computer Science
[3]Department of Biomedical Sciences
Iowa State University, Ames, IA
{ftowfic,mheather,honavar}@iastate.edu

Abstract. Comparative analysis of biomolecular networks constructed using measurements from different conditions, tissues, and organisms offer a powerful approach to understanding the structure, function, dynamics, and evolution of complex biological systems. We explore a class of algorithms for aligning large biomolecular networks by breaking down such networks into subgraphs and computing the alignment of the networks based on the alignment of their subgraphs. The resulting subnetworks are compared using graph kernels as scoring functions. We provide implementations of the resulting algorithms as part of BiNA, an open source biomolecular network alignment toolkit. Our experiments using *Drosophila melanogaster*, *Saccharomyces cerevisiae*, *Mus musculus* and *Homo sapiens* protein-protein interaction networks extracted from the DIP repository of protein-protein interaction data demonstrate that the performance of the proposed algorithms (as measured by % GO term enrichment of subnetworks identified by the alignment) is competitive with some of the state-of-the-art algorithms for pair-wise alignment of large protein-protein interaction networks. Our results also show that the inter-species similarity scores computed based on graph kernels can be used to cluster the species into a species tree that is consistent with the known phylogenetic relationships among the species.

1 Introduction

The rapidly advancing field of systems biology aims to understand the structure, function, dynamics, and evolution of complex biological systems [9]. Such an understanding may be gained in terms of the underlying networks of interactions among the large number of molecular participants involved including genes, proteins, and metabolites [47,16]. Of particular interest in this context is the problem of comparing and aligning multiple networks e.g., those generated from measurements taken under different conditions, different tissues, or different organisms [40]. Network alignment methods present a powerful approach for detecting conserved modules across several networks constructed from different

⋆ Corresponding author.

S.L. Salzberg and T. Warnow (Eds.): WABI 2009, LNBI 5724, pp. 345–361, 2009.

species, conditions or timepoints. The detection of conserved network modules may allow the discovery of disease pathways, proteins/genes critical to basic biological functions, and the prediction of protein functions.

The problem of aligning two networks, in the absence of the knowledge of how each node in one network maps to one or more nodes in the other network, requires solving the subgraph isomorphism problem, which is known to be computationally intractable (NP-Hard) [15]. However, in practice, it is possible to establish correspondence between nodes in the two networks to be aligned and to design heuristics that strike a balance between the speed, accuracy and robustness of the alignment of large biological networks. For instance, MaWISh [29] is a pairwise network alignment algorithm with a runtime complexity of $O(mn)$ (where m and n are the number of vertices in the two networks being compared) that relies on a scoring function that takes into account protein duplication events as well as interaction loss/gain events between pairs of proteins to detect conserved protein clusters. Hopemap [44] is an iterative clustering-based alignment algorithm for Protein-Protein Interaction networks. HopeMap starts by clustering homologs based on their sequence similarity and already known KEGG/InParanoid Orthology status. The algorithm then proceeds to search for strongly connected components and outputs the conserved components that statisfy a predefined user threshold [44]. Graemlin 2.0 is a linear time algorithm that relies on a feature-based scoring function to perform an approximate global alignment of multiple networks. The scoring function for Graemlin 2.0 takes into account protein deletion, duplication, mutation, presence and count as well as edge/paralog deletion across the different networks being aligned [13]. NetworkBLAST-M [23] is a progressive multiple network alignment algorithm that constructs a layered alignment graph, where each layer corresponds to a network and edges between layers connect homologs across different networks. Highly conserved subnetworks from networks from different species are first aligned based on highly conserved orthologous clusters, then the clusters are expanded using an iterative greedy local search algorithm [23].

Against this background, we explore a class of algorithms for aligning large biomolecular networks using a *divide and conquer* strategy that takes advantage of the *modular* substructure of biological networks [17,36,19]. The basic idea behind our approach is to align a pair of networks based on the optimal alignments of the subnetworks of one network with the subnetworks of the other. Different ways of decomposing a network into subnetworks in combination with different choices of measures of *similarity* between a pair of subnetworks yield different algorithms for aligning biomolecular networks.

We utilize variants of state-of-the-art *graph kernels* [6,7], first developed for use in training support vector machines for classification of graph-structured patterns, to compute the *similarity* between two subgraphs. The use of graph kernels to align networks offers several advantages: It is easy to substitute one graph kernel for another (to incorporate different application-specific criteria) without changing the overall approach to aligning networks; it is possible to combine multiple graph kernels to create more complex kernels [7] as needed. Our

experiments with the fly, yeast, mouse and human protein-protein interaction networks extracted from DIP (Database of Interacting Proteins) [38] demonstrate the feasibility of the proposed approach for aligning large biomolecular networks.

The rest of the paper is organized as follows: Section 2 precisely formulates the problem of aligning two biomolecular networks and describes the key elements of our proposed solution. Section 3 describes the experimental setup and experimental results. Section 4 concludes with a summary of the main contributions of the paper in the broader context of related literature and a brief outline of some directions for further research.

2 Aligning Protein-Protein Interaction Networks

2.1 Problem Formulation

We consider the problem of pair-wise alignment of protein-protein interaction networks. We model protein-protein interaction networks as undirected and unweighted graphs. In a protein-protein interaction network, the vertices in the graph correspond to proteins and the edges denote interactions between the two proteins. Let the graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ denote two protein-protein interaction networks where $V_1 = \{v_1^1, v_2^1, v_3^1, ...v_n^1\}$ and $V_2 = \{v_1^2, v_2^2, v_3^2, ...v_m^2\}$, respectively, denote the vertices of G_1 and G_2; and E_1 and E_2 denote the edges of G_1 and G_2 respectively. Let a matrix $\mathbf{P}$ with $|V_1|$ rows and $|V_2|$ columns (i.e, $n \times m$ matrix) denote a set of matches between the vertices of G_1 and G_2. The mapping matrix $\mathbf{P}$ is defined such that for any two vertices v_x^1 and v_y^2 (where $1 \leq x \leq n$ and $1 \leq y \leq m$) from graphs G_1 and G_2, respectively, $P_{v_x^1 v_y^2} = 1$ if v_x^1 from G_1 is matched to v_y^2 from G_2 and $P_{v_x^1 v_y^2} = 0$ if v_x^1 in G_1 is not a match to v_y^2 in G_2. For example, the matches between nodes may be based on homology between the sequences of the corresponding proteins. Thus, each node in G_1 is matched to 0 or more nodes of G_2 and vice versa. Note that the number of such matches for any node in G_1 is much smaller than the total number of nodes in G_2 and vice versa.

$C_1(L_1, O_1)$ is said to be a subgraph of $G_1(V_1, E_1)$ if $L_1 \subset V_1$ and $O_1 \subset E_1$ where O_1 consists only of edges whose end points are in L_1. We associate with the graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ sets of subgraphs $S_1 = \{C_1, C_2, C_3, ...C_l\}$ and $S_2 = \{Z_1, Z_2, Z_3, .., Z_w\}$ (respectively), where $C_i(L_i, O_i)$ $1 \leq i \leq l$ is a subgraph of G_1 and $Z_j(W_j, Q_j)$ $1 \leq j \leq w$ is a subgraph of G_2. Our basic strategy is to find a best match for each subgraph in S_1 from S_2 by optimizing a scoring function, $K(C_i, Z_j)$, such that we obtain: (i) a set of vertices that satisfy $P_{v_x^1 v_y^2} = 1$, where $v_x^1 \in L_i$ and $v_y^2 \in W_j$ and (ii) a set of edges where: if (v_x^1, v_d^1) is an edge in O_i, then (v_y^2, v_g^2) is an edge in Q_j where $P_{v_x^1 v_y^2} = 1$ and $P_{v_d^1 v_g^2} = 1$. The resulting solution to the network alignment problem satisfies the condition that each subgraph in S_1 has at most one matching subgraph in S_2. Thus, a pairwise alignment of the networks $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$ is expressed in terms of an optimal alignment among the sets of the corresponding sets of subgraphs in S_1 and S_2.

2.2 Divide-and-Conquer Approach to Aligning Protein-Protein Interaction Networks

As noted earlier, our basic approach to aligning a pair of protein-protein interaction networks involves (a) decomposing each network into a collection of smaller subnetworks; (b) compute the alignment of the two networks in terms of the optimal alignments of the subnetworks of one network with the subnetworks of the other. Different choices of methods for decomposing a network into subnetworks in combination with different choices of measures of *similarity* between a pair of subnetworks yield different algorithms for aligning protein-protein interaction networks. In our current implementation, we establish the matches between nodes in the two protein-protein interaction networks to be aligned based on reciprocal BLASTp [2] hits between the corresponding protein sequences. Thus, $P_{v_x^1 v_y^2} = 1$ if and only if the correponding protein sequences of v_x^1 and v_y^2 are reciprocal BLASTp hits [21] for each other (at some chosen user-specified threshold). Alternatively, the mapping can be established based on known homologies (e.g between the human WNT1 and mouse Wnt1 proteins) [28,10].

Decomposing Networks into *k*-hop Neighborhoods. A k-hop neighborhood-based approach to alignment uses the notion of k-hop neighborhood. The k-hop neighborhood of a vertex $v_x^1 \in V_1$ of the graph $G_1(V_1, E_1)$ is simply a subgraph of G_1 that connects v_x^1 with the vertices in V_1 that are reachable in k hops from v_x^1 using the edges in E_1. Given two graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, a mapping matrix $\mathbf{P}$ that associates each vertex in V_1 with zero or more vertices in V_2 and a user-specified parameter k, we construct for each vertex $v_x^1 \in V_1$ its corresponding k-hop neighborhood C_x in G_1. We then use the mapping matrix $\mathbf{P}$ to obtain the set of matches for vertex v_x^1 among the vertices in V_2; and construct the k-hop neighborhood Z_y for each matching vertex v_y^2 in G_2 and $P_{v_x^1 v_y^2} = 1$. Let $S(v_x^1, G_2)$ be the resulting collection of k-hop neighborhoods in G_2 associated with the vertex v_x^1 in G_1. We compare each k-hop subgraph C_x in G_1 with each member of the corresponding collection $S(v_x^1, G_2)$ to identify the k-hop subgraph of G_2 that is the best match for C_x (based on a chosen similarity measure). This process is illustrated in figure 1. The runtime complexity of the k-hop neighborhood based network alignment algorithm is $O(bmg)$ where m is the number of nodes in the query network G_1, b is the maximum number of matches in the target network G_2 for any node in the query network, and g is the running time of the similarity measure or scoring function used to compare a pair of k-hop subnetworks.

Decomposing Networks Into Clusters. A graph clustering based alignment algorithm works as follows: Given two node-labeled graphs $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, and a mapping matrix $\mathbf{P}$ that associates each vertex in V_1 with zero or more vertices in V_2, we first extract collections of subgraphs $H_1 = \{C_1, C_2, C_3, ...C_l\}$ and $H_2 = \{Z_1, Z_2, Z_3, ...Z_w\}$ from G_1 and G_2 respectively. In principle, any graph clustering algorithm may be used to construct the subgraph sets H_1 and H_2. In our experiments, we used the bicomponent clusterer as implemented in the JUNG (Java Universal Network/Graph) framework

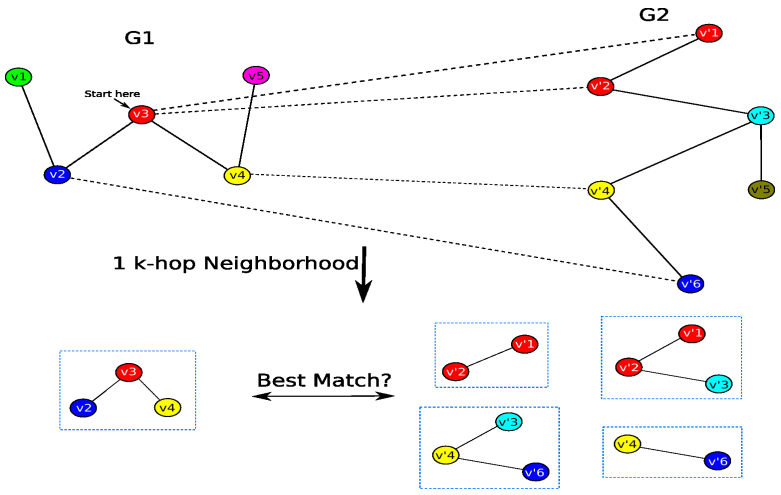

Fig. 1. General schematic of the k-hop neighborhood alignment algorithm. The input to the algorithm are two graphs (G_1 and G_2) with corresponding relationships among their nodes using mapping matrix $\mathbf{P}$ (similarly colored nodes are sequence homologous according to a BLAST search, for example $P_{v_2 v'_6} = 1$). The algorithm starts at an arbitrary vertex in G_1 (red vertex in the figure) and constructs a k-hop neighborhood around the starting vertex (1-hop neighborhood in the figure). The algorithm then matches each of the nodes in the 1-hop neighborhood subgraph from G_1 to nodes in G_2 using mapping matrix $\mathbf{P}$. 1-hop subgraphs are then constructed around each of the matching vertices. The 1-hop subgraphs from G_2 are then compared using a scoring function (e.g. a graph kernel) to the 1-hop subgraph from G_1 and the maximum scoring match is returned.

[35,46] to extract H_1 and H_2. Briefly, the bicomponent clusterer searches for all biconnected components (graphs that cannot be disconnected by removing a single node/vertex [18]) by traversing a graph in a depth-first manner (please see [32] for more details). Once the subgraph sets H_1 and H_2 of the biconnected subgraphs of G_1 and G_2 (respectively) are extracted, an all vs. all comparison is conducted to identify for each subgraph in H_1, the best matching subgraph in H_2 using a scoring function (e.g. a graph kernel, see figure 2). The running time complexity of this algorithm is $O(lwg)$ where l is the number of clusters extracted from the query network G_1, w is the number of clusters extracted from the target network G_2 , and g is the running time of the scoring function used to compare a pair of clusters (subgraphs).

2.3 Scoring Functions

We now proceed to describe the similarity measures or scoring functions used to compare a pair of subgraphs (e.g., a pair of k-hop subgraphs or a pair of bi-component clusters described above).

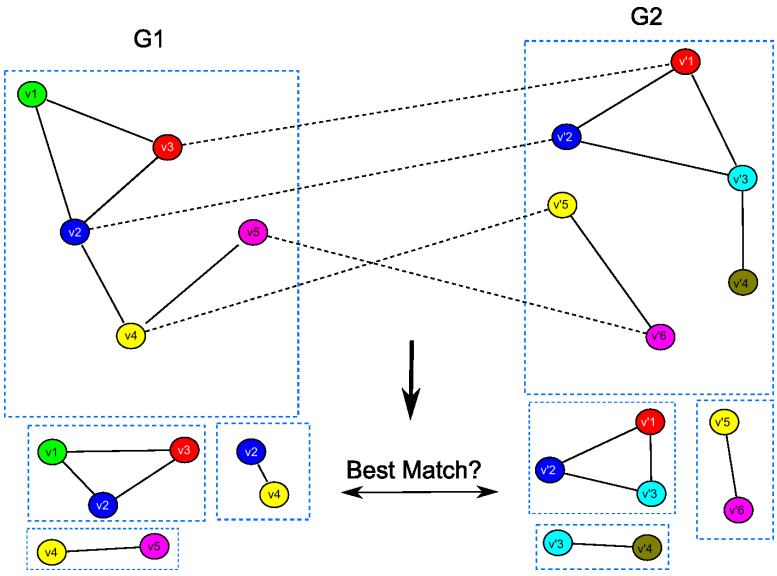

Fig. 2. Schematic for the cluster-based alignment algorithm. The input to the algorithm are two graphs (G_1 and G_2) with corresponding relationships among their nodes using mapping matrix $\mathbf{P}$ (similarly colored nodes are sequence homologous according to a BLAST search, for example $P_{v_2 v'_2} = 1$). Subgraphs are generated from G_1 and G_2 using a graph clustering algorithm (e.g. bicomponent clusterer that finds biconnected subgraphs) and the subgraphs from G_1 are compared against the subgraphs from G_2 to find the best matching subgraphs using an appropriate scoring function.

Modified Shortest Path Distance Graph Kernel. The shortest path graph kernel was first described by Borgwardt and Kriegel [6]. As the name implies, the kernel compares the length of the shortest paths between any two nodes in a graph based on a pre-computed shortest-path distance. The shortest path distances for each graph may be computed using the Floyd-Warshall algorithm as implemented in the CDK (Chemistry Development Kit) package [41]. We modified the Shortest-Path Graph Kernel to take into account the sequence homology of nodes being compared as computed by BLAST [2]. The shortest path graph kernel for subgraphs Z_{G_1} and Z_{G_2} (e.g., k-hop subgraphs, bicomponent clusters extracted from G_1 and G_2 respectively) is given by:

$$K(Z_{G_1}, Z_{G_2}) = \log \left[\sum_{v_i^1, v_j^1 \in Z_{G_1}} \sum_{v_k^2, v_p^2 \in Z_{G_2}} \delta(v_i^1, v_k^2) \times \delta(v_j^1, v_p^2) \times d(v_i^1, v_j^1) \times d(v_k^2, v_p^2) \right] \tag{1}$$

where $\delta(v_x^1, v_y^2) = \frac{BlastScore(v_x^1, v_y^2) + BlastScore(v_y^2, v_x^1)}{2}$. $d(v_i^1, v_j^1)$ and $d(v_k^2, v_p^2)$ are the lengths of the shortest paths between v_i^1,v_j^1 and v_k^2,v_p^2 computed by the Floyd-Warshall algorithm. The runtime of the Floyd-Warshall Algorithm is $O(n^3)$. The

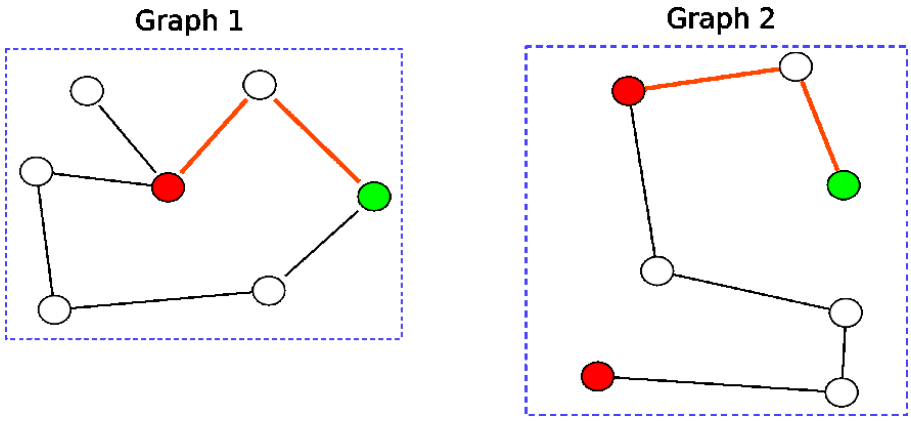

Fig. 3. An example of the graph matching conducted by the shortest path graph kernel. Similarly colored nodes are sequence homologous according to a BLAST search. As can be seen from the figure, the graph kernel compares the lengths of the shortest paths around homologous vertices across the two graphs. The red edges show the matching shortest path in both graphs as computed by the graph kernel. The shortest path distance graph kernel takes into account the sequence homology score for the matching vertices across the two graphs as well as the distances between the two matched vertices within the graphs.

shortest path graph kernel has a runtime of $O(n^4)$ (where n is the maximum number of nodes in larger of the two graphs being compared). Please see figure 3 for a general outline of the comparison technique used by the shortest-path graph kernel.

Modified Random Walk Graph Kernel. The random walk graph kernel [45] has been previously utilized by Borgwardt et al. [7] to compare protein-protein interaction networks. The random walk graph kernel for subgraphs Z_{G_1} and Z_{G_2} (e.g., k-hop subgraphs, bicomponent clusters extracted from G_1 and G_2 respectively) is given by:

$$K(Z_{G_1}, Z_{G_2}) = p \times (\mathbf{I} - \lambda K_x)^{-1} \times q \tag{2}$$

where $\mathbf{I}$ is the identity matrix, λ is a user-specified variable controlling the length of the random walks (a value of 0.01 was used for the experiments in this paper), K_x is an $nm \times nm$ matrix (where n is the number of vertices in Z_{G_1} and m is the number of vertices in Z_{G_2} resulting from the Kronecker product $K_x = Z_{G_1} \otimes Z_{G_2}$, specifically,

$$K_{\alpha\beta} = \delta(Z_{G_{1_{ij}}}, Z_{G_{2_{kl}}}), \alpha \equiv m(i-1) + k, \beta \equiv m(j-1) + l \tag{3}$$

Where $\delta(Z_{G_{1_{ij}}}, Z_{G_{2_{kl}}}) = \frac{BlastScore(Z_{G_{1_{ij}}}, Z_{G_{2_{kl}}}) + BlastScore(Z_{G_{2_{kl}}}, Z_{G_{1_{ij}}})}{2}$; p and q are $1 \times nm$ and $nm \times 1$ vectors used to obtain the sum of all the entries of the inverse expression $((\mathbf{I} - \lambda K_x)^{-1})$.

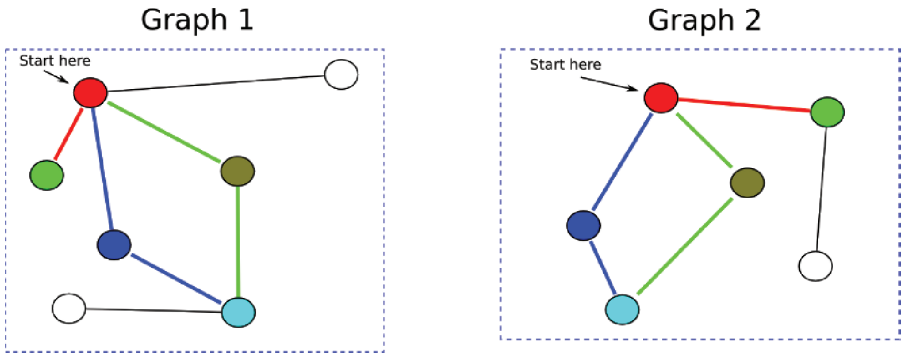

Fig. 4. An example of the graph matching conducted by the random walk graph kernel. Similarly colored vertices are sequence homologous according to a BLAST search. As can be seen from the figure, the graph kernel compares the neighborhood around the starting vertices in each graph using random walks. Colored edges indicate matching random walks across the two graphs of up to length 2. The random walk graph kernel takes into account the sequence homology of the vertices visited in the random walks across the two graphs as well as the general topology of the neighborhood around the starting vertex.

We adapted the random walk graph kernel to align protein-protein interaction networks by taking advantage of the reciprocal BLAST hits (RBH) among the proteins in the networks from different species [21]. Naive implementation of our modified random-walk graph kernel, like the original random-walk graph kernel [45], has a runtime complexity of $O(r^6)$ (where $r = max(n, m)$). This is due to the fact that the product graph's adjacency matrix is $nm \times nm$, and the matrix inverse operation takes $O(h^3)$ time, where h is the number of rows in the matrix being inverted (thus, the total runtime is $O((rm)^3)$ or $O(r^6)$ where $r = max(n, m)$). However, runtime complexity of the random walk graph kernel (and hence our modified random walk graph kernel) can be improved to $O(r^3)$ by making use of the Sylvester equations as proposed by Borgwardt et al. [7]. Figure 4 illustrates the computation of the random walk graph kernel.

2.4 Implementation

The the k-hop network neighborhood and bicomponent clustering based protein-protein interaction network alignment algorithms are implemented in BiNA (`http://www.cs.iastate.edu/~ftowfic`), an open source Biomolecular Network Alignment toolkit. The current implementation includes variants of the shortest path and random walk graph kernels for computing similarity between pairs of subnetworks. The modular design of BiNA allows the incorporation of alternative strategies for decomposing networks into subnetworks and alternative similarity measures (e.g., kernel functions) for computing the similarity between subnetworks.

3 Experiments and Results

We conducted experiments using k-hop subgraph and bi-component cluster based strategies for decomposing graphs into collections of subgraphs. In each case, both modified shortest path and modified random walk graph kernels were used to compute similarity between pairs of subgraphs. We compare the performance of the resulting algorithms with variants of NetworkBLAST [24] and HopeMap [44], which are among the state-of-the-art algorithms for pair-wise alignment of protein-protein interaction networks, using metrics proposed by Kalaev et al [24]. The NetworkBLAST algorithm uses BLASTp [2] to match the nodes across the different networks being aligned, whereas HopeMap uses InParanoid [34] orthology groups to match the nodes across different networks. The HopeMap and NetworkBLAST algorithm were adapted by Tian and Samatova to utilize KEGG Ortholog (KO) groups to match nodes across different species (NetworkBLAST-ko and HopeMap-ko) [44]. The results in table 1 for NetworkBLAST, HopeMap, NetworkBLAST-ko and HopeMap-ko are taken from Tian and Samatova's HopeMap paper [44].

In addition, we used our network alignment algorithms to generate phylogenetic trees from protein-protein interaction networks. We now proceed to describe the experimental setup and the results of this study.

3.1 Datasets

The yeast, fly, mouse and human protein-protein interaction networks were obtained from the Database of Interacting Proteins (DIP) release 1/26/2009 [38]. The sequences for each dataset were obtained from uniprot release 14 [4]. The DIP sequence ids were matched against their uniprot counterparts using a mapping table provided on the DIP website. All proteins from DIP that had obsolete uniprot IDs or were otherwise not available in release 14 of the uniprot database were removed from the dataset. The fly, yeast, mouse and human protein-protein interaction networks consisted of $6,645$, $4,953$, 424 and $1,321$ nodes and $20,010$, $17,590$, 384 and $1,716$ edges, respectively. The protein sequences for each dataset were downloaded from uniprot [4]. BLASTp [2] with a cutoff of 1×10^{-10} was used to match protein sequences across species.

3.2 Comparison with NetworkBLAST and HopeMap

To evaluate the alignments, Kalaev et al.'s approach was implemented as described in the NetworkBLAST [24] and the HopeMap [44] papers. Recall from section 2.1 that the output of the alignment algorithm is a set of subgraphs S_1 and S_2 (corresponding to the query and target networks, respectively). The set of subgraphs $S_2 = \{Z_1, Z_2, Z_3, ..., Z_w\}$ in the target network is evaluated by searching for overrepresented Gene Ontology (GO) categories from the biological process annotation [3]. The GOTermFinder [8] tool was utilized to compute enrichment p-values (p-value significance cutoff = 0.05) that have been corrected for multiple testing using the false discovery rate. Briefly, GOTermFinder computes p-values for a set of GO annotations for the set of proteins in subgraphs

$Z_{1..w}$ based on the number of proteins in the subgraph Z_x (where $1 \leq x \leq w$, and the number of vertices in Z_x is r) and the number of proteins in the genome of the target network (n) and their respective GO annotation. The p-value is computed based on the hypergeometric distribution as the probability of k or more out of r proteins being assigned a given annotation (where k is the number of proteins in the subgraph Z_x possessing the GO category of interest), given that y of n proteins possess such an annotation in the genome in general. The number of subgraphs, f, that had one or more GO categories overrepresented is computed (where $f \leq w$) and the fraction of subgraphs from the target network that had a significant number of GO categories overrepresented is then computed ($\frac{f}{w} \times 100$, % coherent subnetworks). The specificity of the alignment method is measured by the percent of coherent subnetworks discovered for each species. The sensitivity of the methods is indicated by the number of distinct GO categories covered by the functionally coherent subnetworks. The purpose of this evaluation approach is to determine whether or not the matching subgraphs found in the target network represent a functional module/pathway (functionally coherent subgraphs) based on the GO annotation of the proteins in the subgraph.

***k*-hop Neighborhood based Alignment.** The results in table 1 show a comparison of the performance of the k-hop neighborhood based alignment with variants of NetworkBLAST [24] and HopeMap [44].

As can be seen from the results in table 1, the performance measured in terms of % GO enrichment observed when the fly protein-protein interaction network is aligned with the yeast protein-protein interaction network, and vice-versa, using k-hop neighborhood based alignment (with k, the number of hops set equal to 1 and 2) is comparable to that of of variants of NetworkBLAST and Hopemap algorithms (as reported in [44]). The modified random walk graph kernel (RWKernel) yields higher % GO enrichment than the modified shortest path graph kernel (SPKernel) in the case of the fly dataset. The effectiveness of

Table 1. Comparison of the k-hop neighborhood based protein-protein interaction network alignment algorithm (using the SPKernel and RWKernels) with variants of NetworkBLAST and HopeMap (as reported in [44]) using the functional coherence measure: As can be seen from the table, the k-hop neighborhood based algorithm with k=2 is competitive with variants of NetworkBLAST and Hopemap that use sequence identity or KEGG ortholog groups (NetworkBLAST-ko and HopeMap-ko, respectively) for node-matching

Method	% GO enrichment in Yeast	# GO enriched in Yeast	% GO enrichment in Fly	# GO enriched in Fly
NetworkBLAST	94.87	67	84.62	62
HopeMap	98.73	65	78.48	46
NetworkBLAST-ko	100	9	100	8
HopeMap-ko	100	24	92.31	24
SPKernel-1Hop	100 (Score cutoff = 800)	51	78 (Score cutoff = 900)	22
SPKernel-2Hop	100 (Score cutoff = 1800)	46	76 (Score cutoff= 2400)	9
RWKernel-1Hop	100 (Score cutoff = 500)	71	85 (Score cutoff = 900)	19
RWKernel-2Hop	100 (Score cutoff = 800)	107	100 (Score cutoff = 3100)	1

Table 2. Sample comparison of the k-hop alignment algorithm with the SPKernel and RWKernel on the mouse and human DIP datasets

Method	% GO enrichment in Mouse	# GO enriched in Mouse	% GO enrichment in Human	# GO enriched in Human
SPKernel-1Hop	53 (Score cutoff = 40)	19	85 (Score cutoff = 70)	70
RWKernel-1Hop	100 (Score cutoff = 80)	1	100 (Score cutoff = 50)	8
SPKernel-2Hop	94 (Score cutoff = 450)	4	100 (Score cutoff = 200)	13
RWKernel-2Hop	94 (Score cutoff = 110)	4	100 (Score cutoff = 80)	17

k-hop neighborhood based network alignment algorithm is further confirmed by results of aligning the human and mouse protein-protein interaction networks shown in table 2.

It is worth noting that the k-hop based network algorithms which use only BLASTp hits to match nodes across networks are competitive with variants of NetworkBLAST and HopeMap (including those that use other evidence for orthology: InParanoid orthology groups in the case of HopeMap, phylogeny in the case of NetworkBLAST, and KEGG orthologs in the case of NetworkBLAST-ko, and HopeMap-ko) or utilize GO annotations as part of their scoring functions (in case of HopeMap).

Bicomponent Cluster based Alignment. The results in table 3 show the performance of the bicomponent cluster-based alignment algorithm using "bicomponent clusterer", as implemented in JUNG [35] (Java Universal Network/Graph Framework). The clustering algorithm produced 1,236, 2,110, 579 and 1,893 clusters, with 5, 4, 2 and 2.5 proteins per cluster, respectively, on the yeast, fly, mouse and human datasets extracted from DIP. As can be seen from table 3, the performance of the bicomponent clustering based alignment is comparable to that of k-hop neighborhood based alignment algorithm (see 1) when the modified random walk graph kernel is used for comparing subgraphs. However, the performance of the bicomponent clustering based alignment using the modified shortest path graph kernel is substantially worse than that obtained using the modified random walk kernel. This is consistent with the observation that random walk graph kernel is more sensitive to differences between the graphs being compared than the shortest path kernel.

Table 3. Performance for the cluster-based alignment algorithm with the Shortest Path graph kernel (SPKernel) and the Random Walk graph kernel (RWKernel) using the bicomponent clustering algorithm on the fly and yeast DIP datasets

Method	% GO enrichment in Yeast	# GO enriched in Yeast	% GO enrichment in Fly	# GO enriched in Fly
SPKernel with Bicomponent Clusterer	67 (Score cutoff = 5)	2	50	1
RWKernel with Bicomponent Clusterer	100 (Score cutoff = 4)	2	100 (Score cutoff = 4)	1

Table 4. Performance for the cluster-based alignment algorithm with the Shortest Path graph kernel (SPKernel) and the Random Walk graph kernel (RWKernel) using the bicomponent clustering algorithm on the mouse and human DIP datasets

Method	% GO enrichment in Mouse	# GO enriched in Mouse	% GO enrichment in Human	# GO enriched in Human
SPKernel with Bicomponent Clusterer	70 (Score cutoff = 16)	4	33 (Score cutoff = 20)	1
RWKernel with Bicomponent Clusterer	96 (Score cutoff = 4)	6	83 (Score cutoff 15)	4

3.3 Reconstructing Phylogenetic Relationships from Network Alignments

The accuracy with which known phylogenetic relationships between species can be recovered by a protein-protein interaction network alignment algorithm serves as an additional measure of the quality of network alignments produced by the algorithm. The pairwise similarity scores associated with a pairwise network alignment can be used to construct an *inter-species similarity graph* where the nodes denote the species and the weight on the links connecting pairs of nodes denote the pairwise alignment scores output by a network alignment algorithm. The resulting inter-species similarity graph can be partitioned (or alternatively, the nodes of the graph can be clustered) hierarchically to produce a phylogenetic tree.

We constructed the inter-species similarity graph using all possible pair-wise alignments of protein-protein interaction networks from yeast, fly, human and mouse obtained with k-hop neighborhood based alignment using the modified random walk kernel. The resulting inter-species similarity network is shown in figure 5 (left). We used spectral clustering algorithm [33], to recursively partition

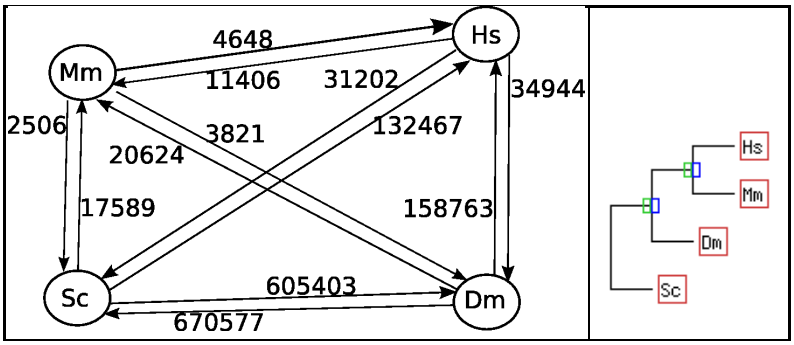

Fig. 5. (Left) Graph representation of the relationships between the mouse (Mm), human (Hs), yeast (Sc) and fly (Dm) networks compared using the 1-hop RWKernel algorithm. Higher scores indicate greater similarity between the two connected networks. **(Right)** The tree constructed using a hierarchical spectral clustering algorithm on the graph shown to the left. The tree figure was created using proWeb Tree viewer [43].

the inter-species similarity graph to obtain a hierarchical clustering of species which is shown in figure 5 (right). We observe that the tree shown in figure 5 is consistent with the generally accepted phylogenetic relationships among yeast, fly, human and mouse [20].

4 Summary and Discussion

Aligning biomolecular networks from different species, tissues and conditions allows offers a powerful approach to discover shared components that can help explain the observed phenotypes. Specifically, applications of network alignment allow the discovery of conserved pathways among different species [25,42], finding protein groups that are relevant to disease [22,31], discovery of the chemical mechanism of metabolic reactions [37,26] and more [48,27,39,5,1]. We have explored a novel class of graph kernel based polynomial time algorithms for aligning biomolecular networks. The proposed algorithms align large biomolecular networks by decomposing them into easy to compare substructures. The resulting subnetworks are compared using graph kernels as scoring functions. The modularity of kernels [11] offers the possibility of constructing composite kernel functions using existing kernel functions that capture different but complementary notions of similarity between graphs [7].

The runtime complexity of the k-hop neighborhood based alignment algorithm is $O(bmg)$ where m is the number of nodes in the query network G_1, b is the maximum number of matches in the target network G_2 for any node in the query network, and g is the running time of the similarity measure or scoring function used to compare a pair of k-hop subnetworks. The running time complexity of this algorithm is $O(lwg)$ where l is the number of clusters extracted from the query network G_1 , w is the number of clusters extracted from the target network G_2 , and g is the running time of the scoring function used to compare a pair of clusters (subgraphs). In comparison, the run-time complexity of NetworkBLAST-M ($O((np)^d s3^s)$), where n is the number of nodes in each of the networks, s the number of networks, p an upper bound on the node degree and d the number of *seed spines* used to generate the alignment. In the special case of pairwise network alignment (s=2), the run-time complexity of NetworkBLAST reduces to $O((np)^d)$. The runtime complexity of HopeMap is linear in terms of the total number of nodes and edges in the alignment graph [44], which is $O(2n+2n^2)$ in terms of the input graphs (where each input graph has at most n nodes).

The k-hop network neighborhood based and bicomponent clustering based protein-protein interaction network alignment algorithms are implemented in BiNA (`http://www.cs.iastate.edu/~ftowfic`), an open source Biomolecular Network Alignment toolkit. The current implementation includes variants of the shortest path and random walk graph kernels for computing similarity between pairs of subnetworks. The modular design of BiNA allows the incorporation of alternative strategies for decomposing networks into subnetworks and alternative

similarity measures (e.g., kernel functions) for computing the similarity between subnetworks.

Our experiments with the fly, yeast, mouse and human protein-protein interaction networks extracted from DIP (Database of Interacting Proteins) [38] demonstrate that the performance of the proposed algorithms (as measured by % GO term enrichment of subnetworks identified by the alignment) is competitive with variants of the NetworkBLAST and HopeMap, which are among the state-of-the-art algorithms for pair-wise alignment of large protein-protein interaction networks [24,44].

Our results show that the inter-species similarity scores computed on the basis of pair-wise protein-protein interaction network alignments can be used to cluster the species into a species tree that is consistent with the known phylogenetic relationships among the species. Taken together with the results reported by Frost et al. [14] and Kuchaiev et al. [30] on reconstruction of phylogenetic relationships by comparing metabolic networks, we conjecture that (a) the accuracy with which a network alignment algorithm can be used to recover known phylogenetic relationships among species can be used as useful metric for evaluating the algorithm and (b) protein-protein interaction networks can be used as a useful source of information in reconstructing phylogenies. As this evaluation approach works at a global level (it only considers the total alignment score between two species, not the specific alignment scores for the subnetworks), new evaluation approaches would need to be considered to determine the feasibility of the specific alignments/matches generated by network alignment algorithms.

Some interesting directions for further work on the biomolecular network alignment algorithms include:

- Design of alternative measures of performance for assessing the quality of the generated network alignments.
- Algorithms for aligning networks that contain directed links, such as transcriptional regulatory networks, multiple types of nodes (proteins, DNA, RNA) and multiple types of links.
- Extensions that allow the alignment of multiple networks.
- The use of more sophisticated graph-clustering algorithms (such as MCL [12]).
- Automated tuning of parameters (e.g λ for the random walk kernel) using parameter learning techniques [13].
- Optimizations that reduce the runtime memory requirements of the algorithm.

Acknowledgments. This research was supported in part by an Integrative Graduate Education and Research Training (IGERT) fellowship to Fadi Towfic, funded by the National Science Foundation grant (DGE 0504304) to Iowa State University and a National Science Foundation Research Grant (IIS 0711356) to Vasant Honavar. The authors are grateful to the WABI-09 anonymous referees for their helpful comments on the manuscript.

References

1. Aittokallio, T., Schwikowski, B.: Graph-based methods for analysing networks in cell biology. Briefings in Bioinformatics 7(3), 243 (2006)
2. Altschul, S.F., Madden, T.L., Schäffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Research 25(17), 3390 (1997)
3. Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., et al.: Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. Nature genetics 25(1), 25 (2000)
4. Bairoch, A., Apweiler, R., Wu, C.H., Barker, W.C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., et al.: The Universal Protein Resource (UniProt). Nucleic Acids Research 33, D154 (2005)
5. Barabasi, A.L., Oltvai, Z.N.: Network biology: understanding the cell's functional organization. Nature Reviews Genetics 5(2), 101–113 (2004)
6. Borgwardt, K.M., Kriegel, H.P.: Shortest-Path Kernels on Graphs. In: Proceedings of the Fifth IEEE International Conference on Data Mining, pp. 74–81 (2005)
7. Borgwardt, K.M., Kriegel, H.P., Vishwanathan, S.V.N., Schraudolph, N.N.: Graph Kernels For Disease Outcome Prediction From Protein-Protein Interaction Networks. In: Proceedings of the Pacific Symposium of Biocomputing (2007)
8. Boyle, E.I., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J.M., Sherlock, G.O.: GO: TermFinder–open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. Bioinformatics (Oxford, England) 20(18), 3710 (2004)
9. Bruggeman, F.J., Westerhoff, H.V.: The nature of systems biology. Trends Microbiol. 15(1), 45–50 (2007)
10. Burrus, L.W., McMahon, A.P.: Biochemical analysis of murine Wnt proteins reveals both shared and distinct properties. Experimental cell research 220(2), 363–373 (1995)
11. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines. Cambridge University Press, Cambridge (2000)
12. Enright, A.J., Van Dongen, S., Ouzounis, C.A.: An efficient algorithm for large-scale detection of protein families. Nucleic Acids Research 30(7), 1575 (2002)
13. Flannick, J., Novak, A., Do, C.B., Srinivasan, B.S., Batzoglou, S.: Automatic parameter learning for multiple network alignment. In: Vingron, M., Wong, L. (eds.) RECOMB 2008. LNCS (LNBI), vol. 4955, pp. 214–231. Springer, Heidelberg (2008)
14. Forst, C.V., Flamm, C., Hofacker, I.L., Stadler, P.F.: Algebraic comparison of metabolic networks, phylogenetic inference, and metabolic innovation. BMC Bioinformatics 7(1), 67 (2006)
15. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. WH Freeman & Co., New York (1979)
16. Ge, H., Walhout, A.J.M., Vidal, M.: Integrating 'omic' information: a bridge between genomics and systems biology. Trends in Genetics 19(10), 551–560 (2003)
17. Han, J.D., Bertin, N., Hao, T., Goldberg, D.S., Berriz, G.F., Zhang, L.V., Dupuy, D., Walhout, A.J., Cusick, M.E., Roth, F.P., Vidal, M.: Evidence for dynamically organized modularity in the yeast protein-protein interaction network. Nature 430(6995), 88–93 (2004)
18. Harary, F.: Graph theory (1969)

19. Hartwell, L.H., Hopfield, J.J., Leibler, S., Murray, A.W.: From molecular to modular cell biology. Nature 402(6761 suppl.), C47–C52 (1999)
20. Hedges, S.B.: The origin and evolution of model organisms. Nature Reviews Genetics 3(11), 838–849 (2002)
21. Hirsh, A.E., Fraser, H.B.: Protein dispensability and rate of evolution. Nature 411(6841), 1046–1049 (2001)
22. Ideker, T., Sharan, R.: Protein networks in disease. Genome Research 18(4), 644 (2008)
23. Kalaev, M., Bafna, V., Sharan, R.: Fast and accurate alignment of multiple protein networks. In: Vingron, M., Wong, L. (eds.) RECOMB 2008. LNCS (LNBI), vol. 4955, pp. 246–256. Springer, Heidelberg (2008)
24. Kalaev, M., Smoot, M., Ideker, T., Sharan, R.: NetworkBLAST: comparative analysis of protein networks. Bioinformatics 24(4), 594 (2008)
25. Kelley, B.P., Sharan, R., Karp, R., Sittler, E.T., Root, D.E., Stockwell, B.R., Ideker, T.: Conserved pathways within bacteria and yeast as revealed by global protein network alignment. Proc. Natl. Acad. Sci. 100, 11394–11399 (2003)
26. Kharchenko, P., Church, G.M., Vitkup, D.: Expression dynamics of a cellular metabolic network. Molecular Systems Biology 1 (2005)
27. Kirac, M., Ozsoyoglu, G.: Protein Function Prediction Based on Patterns in Biological Networks. In: Vingron, M., Wong, L. (eds.) RECOMB 2008. LNCS (LNBI), vol. 4955, pp. 197–213. Springer, Heidelberg (2008)
28. Koonin, E.: Orthologs, paralogs and evolutionary genomics. Annu. Rev. Genet. 39, 309–338 (2005)
29. Koyuturk, M., Kim, Y., Topkara, U., Subramaniam, S., Szpankowski, W., Grama, A.: Pairwise alignment of protein interaction networks. Journal of Computational Biology 13(2), 182–199 (2006)
30. Kuchaiev, O., Milenkovic, T., Memisevic, V., Hayes, W., Przulj, N.: Topological network alignment uncovers biological function and phylogeny. Arxiv, 0810.3280v2 (2009)
31. Lim, J., Hao, T., Shaw, C., Patel, A.J., Szabó, G., Rual, J.F., Fisk, C.J., Li, N., Smolyar, A., Hill, D.E., et al.: A Protein–Protein Interaction Network for Human Inherited Ataxias and Disorders of Purkinje Cell Degeneration. Cell 125(4), 801–814 (2006)
32. Manber, U.: Introduction to algorithms: a creative approach. Addison-Wesley Longman Publishing Co., Inc., Boston (1989)
33. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems 14: Proceedings of the 2002 [sic] Conference, p. 849. MIT Press, Cambridge (2002)
34. O'Brien, K.P., Remm, M., Sonnhammer, E.L.L.: Inparanoid: a comprehensive database of eukaryotic orthologs. Nucleic Acids Research 33(Database issue), D476 (2005)
35. O'Madadhain, J., Fisher, D., White, S., Boey, Y.: The JUNG (Java Universal Network/Graph) Framework. University of California, California (2003)
36. Ravasz, E., Somera, A.L., Mongru, D.A., Oltvai, Z.N., Barabasi, A.L.: Hierarchical organization of modularity in metabolic networks. Science 297(5586), 1551–1555 (2002)
37. Ross, J., Schreiber, I., Vlad, M.O.: Determination of Complex Reaction Mechanisms: Analysis of Chemical, Biological, and Genetic Networks. Oxford University Press, USA (2006)

38. Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U., Eisenberg, D.: The database of interacting proteins: 2004 update. Nucleic Acids Research 32(Database issue), D449 (2004)
39. Scott, J., Ideker, T., Karp, R.M., Sharan, R.: Efficient Algorithms for Detecting Signaling Pathways in Protein Interaction Networks. Journal of Computational Biology 13(2), 133–144 (2006)
40. Sharan, R., Ideker, T.: Modeling cellular machinery through biological network comparison. Nature Biotechnology 24, 427–433 (2006)
41. Steinbeck, C., Hoppe, C., Kuhn, S., Floris, M., Guha, R., Willighagen, E.L.: Recent Developments of the Chemistry Development Kit (CDK)-An Open-Source Java Library for Chemo-and Bioinformatics. Current Pharmaceutical Design 12(17), 2111–2120 (2006)
42. Stuart, J.M., Segal, E., Koller, D., Kim, S.K.: A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules. Science 302(5643), 249–255 (2003)
43. Taylor, N.: proWeb Tree Viewer, `http://www.proweb.org/treeviewer/`
44. Tian, W., Samatova, N.F.: Pairwise alignment of interaction networks by fast identification of maximal conserved patterns. In: Proc. of the Pacific Symposium on Biocomputing (2009)
45. Vishwanathan, S.V.N., Borgwardt, K.M., Schraudolph, N.N.: Fast Computation of Graph Kernels. Technical report, NICTA (2006)
46. White, S., Smyth, P.: Algorithms for estimating relative importance in networks. In: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 266–275. ACM, New York (2003)
47. Wong, S.L., Zhang, L.V., Tong, A.H.Y., Li, Z., Goldberg, D.S., King, O.D., Lesage, G., Vidal, M., Andrews, B., Bussey, H., et al.: Combining biological networks to predict genetic interactions. Proceedings of the National Academy of Sciences 101(44), 15682–15687 (2004)
48. Zhou, X., Kao, M.C.J., Wong, W.H.: Transitive functional annotation by shortest-path analysis of gene expression data. Proceedings of the National Academy of Sciences 99(20), 12783–12788 (2002)