JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 16, Number 8, 2009
© Mary Ann Liebert, Inc.
Pp. 989–999
DOI: 10.1089/cmb.2009.0136

Research Article

Fast and Accurate Alignment of Multiple Protein Networks

MAXIM KALAEV,[1] VINEET BAFNA,[2] and RODED SHARAN[1]

ABSTRACT

Comparative analysis of protein networks has proven to be a powerful approach for elucidating network structure and predicting protein function and interaction. A fundamental challenge for the successful application of this approach is to devise an efficient multiple network alignment algorithm. Here we present a novel framework for the problem. At the heart of the framework is a novel representation of multiple networks that is only linear in their size as opposed to current exponential representations. Our alignment algorithm is very efficient, being capable of aligning 10 networks with tens of thousands of proteins each in minutes. We show that our algorithm outperforms previous approaches for the problem, and produces results that are more in line with current biological knowledge.

Key words: combinatorial proteomics, computational molecular biology, gene expression, gene networks, genetic variation, sequence analysis.

1. INTRODUCTION

RECENT TECHNOLOGICAL ADVANCES enable the systematic characterization of protein-protein interaction (PPI) networks across multiple species. Procedures such as yeast two-hybrid (Ito et al., 2001a) and protein co-immunoprecipitation (Aebersold and Mann, 2003) are routinely employed nowadays to generate large-scale protein interaction networks for human and most model species (Uetz et al., 2000; Ito et al., 2001b; Ho et al., 2002; Gavin et al., 2002; Stelzl et al., 2005). Key to interpreting these data is the inference of cellular machineries. As in other biological domains, a comparative approach provides a powerful basis for addressing this challenge, calling for algorithms for protein network alignment.

In the network alignment problem, one has to identify network regions that are conserved in their sequence and interaction pattern across two or more species. While the general problem is hard, generalizing subgraph isomorphism, heuristic methods have been devised to tackle it. One heuristic approach for the problem creates a merged representation of the networks being compared, called a *network alignment graph*, facilitating the search for conserved subnetworks. In a network alignment graph, the nodes represent sets of proteins, one from each species, and the edges represent conserved PPIs across the investigated species.

The network alignment paradigm has been applied successfully by a number of authors to search for conserved pathways (Kelley et al., 2003) and complexes (Sharan et al., 2005a,b; Koyuturk et al., 2006).

[1]School of Computer Science, Tel Aviv University, Tel Aviv, Israel.
[2]Computer Science and Engineering, University of California San Diego, San Diego, California.

However, its extension to more than a few (three) networks proved difficult due to the exponential growth of the alignment graph with the number of species. Recently, an algorithm was suggested to overcome this difficulty, which is based on imitating progressive sequence alignment techniques (Flannick et al., 2006). The latter algorithm was successfully applied to align up to 10 microbial networks. Another framework for multiple network alignment was suggested by Dutkowski and Tiuryn (2007). Their method is based on clustering the proteins into orthology groups, reconstructing an ancestral network over representatives of these groups, and identifying conserved modules in this network. To date, this approach was applied to align three networks only.

Here we propose a new algorithm for multiple network alignment that is based on a novel representation of the network data. The algorithm avoids the explicit representation of every set of potentially orthologous proteins (which form a node in the network alignment graph), thereby achieving dramatic reduction in time and memory requirements. We compare our algorithm to those of Flannick et al. (2006) and Dutkowski and Tiuryn (2007), showing that it is extremely fast and accurate, providing results that are more in line with current biological knowledge.

2. METHODS

2.1. Data representation

Given k protein-protein interaction networks, we represent them using a k-layer graph, which we call a *layered alignment graph*. Each layer corresponds to a species and contains the corresponding PPI network. Additional edges connect proteins from different layers if they are sequence similar. Formally, layer i has a set V_i of vertices and a set E_i of edges. Additionally, we have a set of *inter-layer* edges denoted by E_H. Let $G_H = (\cup_i V_i, E_H)$ denote the graph restricted to the inter-layer edges. Let n be the maximum number of proteins in a species. Let m be the maximum number of sequence similarity edges in G_H between any pair of species.

The relation between an alignment graph and a layered alignment graph should be clear: while in the former every set of potentially orthologous proteins is represented by a vertex; in the latter such a set is represented by a subgraph of size k, which includes a vertex from each of the layers. We call such a subgraph a *k-spine*. Key to the algorithmic approach presented below is the assumption that a k-spine corresponding to a set of truly orthologous proteins must be connected and, hence, admits a spanning tree. Thus, we can identify all potential vertex sets inducing k-spines by looking for trees instead.

A collection of (connected) k-spines induces a candidate conserved subnetwork. We score it using a likelihood ratio score as described in Sharan et al. (2005b). The score evaluates the fit of the protein-protein interactions within this subnetwork to a conserved subnetwork model versus the chance that they arise at random. The conserved subnetwork model assumes that each pair of proteins from the same species in the subnetwork should interact, independently of all other pairs, with high probability β. Note that we do not score the conservation of interactions, but rather score the conservation of subnetwork density across species. More elaborate models that aim to score interaction conservation directly are described in Hirsh and Sharan (2006). The random model assumes that each species' network was chosen uniformly at random from the collection of all graphs with the same vertex degrees as the ones observed. This random model induces a probability of occurrence p_{uv} for each edge (u, v) of the graph. To accommodate for information on the reliability of interactions, the interaction status of every vertex pair is treated as a noisy observation, and its reliability is combined into the likelihood score. Overall, for a subnetwork with vertex set U, the likelihood ratio score factors over the vertex pairs in it: $\mathcal{L}(U) = \sum_{(u,v)\in U\times U} w(u,v)$ where $w(v,v) = 0$ and for $u \neq v$,

$$w(u,v) = \log \frac{\beta Pr(O_{uv}|T_{uv}) + (1-\beta)Pr(O_{uv}|F_{uv})}{p_{uv}Pr(O_{uv}|T_{uv}) + (1-p_{uv})Pr(O_{uv}|F_{uv})},$$

Here O_{uv} denotes the set of experimental observations on the interaction status of u and v, T_{uv} denotes the event that u and v truly interact, and F_{uv} denotes the event the u and v do not interact. The computation of $Pr(O_{uv}|T_{uv})$ and $Pr(O_{uv}|F_{uv})$ is based on the reliability assigned to the interaction between u and v (Sharan et al., 2005b).

This notion of a conserved subnetwork is extended easily to a layered alignment graph. If we consider every k-spine to correspond to a node in an alignment graph, then a d-node subgraph is a subgraph of

d k-spines (possibly sharing vertices among them) that are densely interconnected with PPI edges. Formally, define a *d-subnet* as a collection U of k multi-sets $U_i = \{u_i^1, \ldots, u_i^d\}$, corresponding to the species $1, \ldots, k$, with the following properties:

- For all $1 \le i \le k$ and $1 \le j \le d$, $u_i^j \in V_i$.
- For all $1 \le j \le d$, the set $U^j = \{u_1^j, u_2^j, \ldots, u_k^j\}$ forms a k-spine.

The score $\mathcal{G}(U)$ of the d-subnet is given by $\mathcal{G}(U) = \sum_{i=1}^{k} \mathcal{L}(U_i)$, where the definition of $\mathcal{L}$ is naturally extended to multi-sets by considering the corresponding sets.

In the following, it will be convenient for us to denote adjacency relations between multi-sets in a d-subnet. For two multi-sets U_1, U_2, denote $(U_1, U_2) \in E_H$ if and only if $(u_1^j, u_2^j) \in E_H$ for all $1 \le j \le d$. We also define a mapping $s(\cdot)$ from multi-sets of vertices from the same species to the index of the species they represent.

2.2. The search algorithm

The main algorithmic task is to look for high scoring d-subnets. This problem is computationally hard even when there is only a single network, and edge-weights are restricted to $+1$ for all edges, and -1 for all non-edges (Shamir et al., 2004). Thus, we resort to a greedy heuristic which starts from high weight seeds and expands them using local search. Such greedy heuristics have been successfully applied to search for conserved subnetworks in a network alignment graph (Sharan et al., 2005b; Flannick et al., 2006; Koyuturk et al., 2006).

There are two sub-tasks we need to tackle: (i) computing high weight seeds and (ii) extending a seed. We provide algorithmic solutions for both tasks below.

Computing seeds. We start by computing d-subnets as *seeds*, where d is a small constant. Notably, even when $d = 2$, we do not know of any algorithm better than the naive approach, which involves looking at all pairs of k-spines. This $O(n^{dk})$ time algorithm is intractable for typical networks, so we consider two alternative assumptions on the inter-layer edges that reduce the computational complexity while retaining the sensitivity of the algorithm.

The first assumption asserts that the k-spines of a seed support the same topology of inter-connections. This is motivated by the observation that proteins within the same pathway or complex are typically present or absent in the genome as a group (Pellegrini et al., 1999). Thus, we consider the following problem:

Problem 1. ***d-identical-spine-subnet***. *Compute a set of d k-spines with identical topologies and maximum score.*

Theorem 1. The *d-identical-spine-subnet* problem admits an $O(k^3 m^d 3^k)$ solution.

Proof. First, consider the case where each of the d k-spines is restricted to be a path. This implies that the d-subnet itself can be considered as a path, i.e., there exists some permutation $\pi : \{1, \ldots, k\} \to \{1, \ldots, k\}$ such that $U_{\pi(1)}, \ldots, U_{\pi(k)}$ is a path (see Figure 1 for an example in which π is the identity permutation). For a subset of species S, let $\mathcal{G}(U, S)$ denote the score of the best d-subnet that uses only species in S, and consists of a path that ends with U. To compute $\mathcal{G}(U, S)$, note that we only need to recurse using the predecessor of U in the path. Formally:

$$\mathcal{G}(U,S) = \begin{cases} \max\limits_{\substack{(U,W)\in E_H \\ s(W)\in S\setminus\{s(U)\}}} \mathcal{G}(W, S\setminus\{s(U)\}) + \mathcal{L}(U) & \text{if } |S| > 1 \\ \mathcal{L}(U) & \text{if } |S| = 1 \end{cases}$$

The recursion is computed for every pair $(U, W) \in E_H$ ($O(k^2 m^d)$ in total), and each subset S (2^k possibilities). For a proposed multi-set W, it takes $O(k)$ time to check whether $s(W) \in S$. Thus, the overall time is $O(k^3 m^d 2^k)$.

A similar recursion can be applied when searching for k-spines that are trees with identical topology. For a subset of species S, let $\mathcal{G}(U, S)$ denote the score of the best d-subnet that uses only the species in S, and consists of a tree rooted at U. Then for $|S| > 1$:

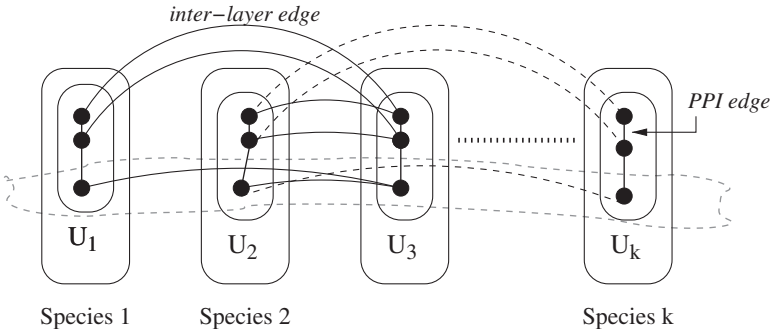

FIG. 1. A seed defined by a d-identical-spine subnet, where the k-spines are restricted to be paths with identical topology. Vertical lines denote PPIs; solid horizontal arcs denote edges in E_H; dashed arcs hint to paths connecting via sequence-similarity edges the nodes of the multi-sets $U_2, U_4, \ldots, U_k$; and the light gray dashed region encloses one of the three k-spines.

$$\mathscr{G}(U,S) = \max_{\substack{(U,W)\in E_H,\, S_1\subset S \\ s(U)\in S_1,\, s(W)\in S\setminus S_1}} \mathscr{G}(U,S_1)+\mathscr{G}(W,S\setminus S_1)$$

To analyze the complexity, notice that there are $O(3^k)$ variations of (S, S_1,$S\setminus S_1$). Thus, the overall time is $O(k^3 m^d 3^k)$. ■

A second, slightly different assumption is based on the phylogeny (described as a rooted, binary tree T) of the investigated species. Consider a set of vertices *a, b, c* whose underlying species form the phylogenetic triple ($s(\{a\})$, ($s(\{b\})$, $s(\{c\})$)). We assume that if *a, b, c* are connected via inter-layer edges, then there is an edge connecting b and c. This implies that we can restrict our attention to k-spines that are *guided* by the phylogeny T in the following sense: any restriction of the k-spine to species that form a clade in T is a subtree of the k-spine. Note that two guided spines can have very different topologies (Fig. 2).

Problem 2. The *d-guided-spine-subnet* problem. *Compute a set of d k-spines, guided by the underlying phylogeny, with maximum score.*
Unfortunately, we do not know of any efficient algorithm better than the naive $O(n^{kd})$ for this problem. However, we show a better solution when the k-spines are restricted to be paths guided by the phylogeny.

Theorem 2. The d-guided-spine-subnet problem can be solved in $O(k(kn)^{2d}|E_H|^d)$ time when restricted to paths.

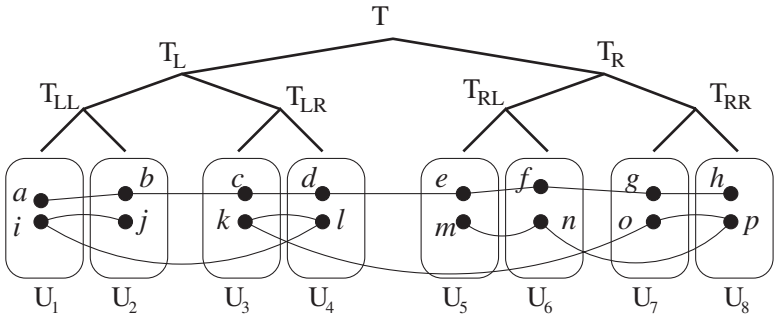

FIG. 2. Sketch of a 2-*guided-spine-subnet*. Note that while the paths of the two k-spines have different topologies, they are both guided by the underlying tree. Following the notation in the proof of Theorem 2, let $U=\{a,j\}$, $W=\{h,m\}$, and consider two possible distant sets $X=\{d,k\}$ and $Y=\{e,o\}$. By definition, $T_{LL}(U\cup X)=\{a,j\}$, $T_{LR}(U\cup X)=\{d,k\}$, $T_{RL}(Y\cup W)=\{e,m\}$, $T_{RR}(Y\cup W)=\{h,o\}$. Hence, $\mathscr{G}(U,W,T)\geq \mathscr{G}(\{a,j\},\{d,k\},T_L)+\mathscr{G}(\{e,m\},\{h,o\},T_R)\geq \mathscr{G}(\{a,i\},\{b,j\},T_{LL})+\mathscr{G}(\{c,k\},\{d,l\},T_{LR})+\mathscr{G}(\{e,m\},\{f,n\},T_{RL})+\mathscr{G}(\{g,p\},\{h,o\},T_{RR})\geq \mathscr{L}(a,i)+\mathscr{L}(b,j)+\mathscr{L}(c,k)+\mathscr{L}(d,l)+\mathscr{L}(e,m)+\mathscr{L}(f,n)+\mathscr{L}(g,o)+\mathscr{L}(h,p)$.

Proof. Consider a subtree T of the phylogeny with subtrees T_L, T_R, respectively. Clearly, each of the d paths will have one end-point in T_L, and the other in T_R. However, the order of the species along these paths is not identical. Therefore, we work with size d subsets U which are not restricted to be within a single species, but instead can span any species in T.

Let $\mathcal{G}(U, W, T)$ denote the best score of a set of d paths guided by a subtree T of the phylogeny, such that $s(U) \subseteq T_L, s(W) \subseteq T_R$ are the end nodes. At the base of the recursion T consists of a single node and $\mathcal{G}(U, U, T) = \mathcal{L}(U)$.

Denote the root of T by *root*(T). For node u, define its *distant set*:

$$\mathcal{D}_T(u) = \{x | \mathrm{LCA}_T(s(\{u\}), s(\{x\})) = \mathrm{root}(T)\}$$

where LCA_T (a, b) is the least common ancestor of a and b in T . Extend this to d elements by defining $\mathcal{D}_T(U) = \{X |\ \mathrm{LCA}_T(s(\{u^j\}), s(\{x^j\})) = \mathrm{root}(T)\ \forall j\}$. By definition, if $X \in \mathcal{D}_T(U)$ then for all $j : s(\{x^j\}) \in T_L, s(\{u^j\}) \in T_R$ or $s(\{x^j\}) \in T_R, s(\{u^j\}) \in T_L$. Define $T_L(X)$ ($T_R(X)$) as the set of all vertices in X with species in T_L (T_R) respectively. Finally, let T_{LL}, T_{LR} be the two subtree of T_L and let T_{RL}, T_{RR} be the two subtrees of T_R. Then, as exemplified in Figure 2:

$$\mathcal{G}(U, W, T) = \max_{\substack{X \in D_{T_L}(U) \\ Y \in D_{T_R}(W) \\ (X,Y) \in E_H}} (\mathcal{G}(T_{LL}(U \cup X), T_{LR}(U \cup X), T_L) + \mathcal{G}(T_{RL}(Y \cup W), T_{RR}(Y \cup W), T_R))$$

Intuitively, the recursion works by dividing the problem of finding a d-guided-subnet (U, W) with respect to a phylogenetic tree T, into two sub-problems on the two sub-trees, T_L and T_R. For the running time, note that there are $O((kn)^{2d}|E_H|^d)$ possible choices for (U, X, Y, W). Since a choice of U and W implies T, the overall time is $O(k(kn)^{2d}|E_H|^d)$. ■

Extending a seed. The next phase of the algorithm is performing an iterative expansion of the seed by adding, in each iteration, the k-spine that contributes the most to the score. The expansion is repeated while a k-spine contributing to the score can be found or the extended seed size reaches a predefined limit. Let us denote by M the current seed, by $\mathcal{G}_M(v, S)$ the score of the best partial extension of M by a subtree that is rooted at vertex v and visits the species in S, and by $\mathcal{L}_M(v)$ the contribution of vertex v to the the extension score. Then $\mathcal{G}_M(v, S)$ can be computed using the following recursive relation:

$$\mathcal{G}_M(v, S) = \begin{cases} \max_{\substack{(v,w) \in E_H S_1 \subset S \\ s(\{v\}) \in S_1, s(\{w\}) \in S \setminus S_1}} \mathcal{G}_M(v, S_1) + \mathcal{G}_M(w, S \setminus S_1) & \text{if } |S| > 1 \\ \mathcal{L}_M(v) & \text{if } |S| = 1 \end{cases}$$

The overall complexity is $O(k|E_H|3^k)$.

There are two speedups one can introduce to this basic extension scheme. The first is to set in advance the order of the species along the phylogenetic tree, eliminating the 3^k factor. We term this variant *restricted order* as opposed to the previous *relaxed order* variant. The second is to constrain k-spines to paths (rather than trees), obtaining an $O(k|E_H|2^k)$ time algorithm.

2.3. Implementation notes and quality assessment

We have designed a software package, *NetworkBLAST-M*, implementing the multiple network alignment approach outlined above. The implementation allows looking for 2-identical-spine seeds with relaxed and restricted orders. For efficiency reasons, we restricted the seed vertices in each network to be of distance at most 2 from one another.

The final collection of conserved subnetworks was filtered to remove redundant solutions. This was done using an iterative greedy procedure that selects each time the highest scoring subgraph and removes all subgraphs intersecting it by more than 50%. For two conserved subnetworks A and B, containing $|A|$ and $|B|$ proteins, respectively, the intersection level is computed as the number of common proteins over min $\{|A|, |B|\}$.

We evaluate the subnetworks output by the algorithm by computing the functional coherency of their member proteins with respect to the biological process annotation of the gene ontology (GO) (Ashburner

et al., 2000), for each species separately. To this end, we used the GO TermFinder tool (Boyle et al., 2004) to compute empirical enrichment *p*-values, and corrected for multiple testing using the false discovery rate (FDR) procedure (Benjamini and Hochberg, 1995), retaining results that passed an FDR threshold of 0.05.

For each species we report the percent of functionally coherent subnetworks discovered, and the number of distinct GO categories they cover. The first measure quantifies the *specificity* of the method, and the second provides an indication on the *sensitivity* of the method. We also provide analogous results with respect to the molecular function and cellular component branches of the gene ontology (see Tables A1 and A2 in Appendix).

To verify that using 2-identical-spines is adequate for our problem, we analyzed alignment nodes within conserved network regions output by NetworkBLAST (Sharan et al., 2005b) for different network sets. When aligning the networks of yeast, worm and fly (NetworkBLAST data set), in 85% of the cases the pertaining alignment nodes respected the yeast-worm-fly phylogeny-based ordering. In two additional microbial network sets (*C. jejuni*, *E. coli*, *H. pylori* and *C. crescentus*, *V. cholerae* and *H. pylori*; Graemlin data set), more than 95% of the alignment nodes respected the respective phylogeny-based orientation. Moreover, 72% of the alignment nodes actually formed cliques in G_H.

We have also experimented with the two seed extension variants presented here. Our results in this regard indicate that the relaxed-order variant yields higher sensitivity on certain data sets while showing similar specificity (Table 2). In the restricted order variant, the best sensitivity is achieved with spines whose orientation respects the phylogenetic tree. Interestingly, the difference in results between different orientations of spines in restricted order is not as significant as could be expected, probably because a large fraction of spines in functionally enriched alignments form cliques in G_H and such spines are robust to ordering restrictions.

3. RESULTS

We applied our algorithm to eukaryotic and microbial PPI networks (Table 1). The three eukaryotic networks were taken from Sharan et al. (2005b) (NetworkBLAST data set) or Dutkowski and Tiuryn (2007) (CAPPI data set), and the microbial networks were taken from (Flannick et al. (2006) (Graemlin data set). As in Sharan et al. (2005b), we used a BLAST E-value threshold of 10^{-7} for sequence similarity, ensuring a corrected significance value of 0.01.

To establish the validity of our method, we first compared it to NetworkBLAST (Sharan et al., 2005b). NetworkBLAST is an exhaustive approach that relies on explicitly constructing a network alignment graph

Table 1. Summary of the PPI Networks Analyzed in This Study

Species (taxon ID)	*No. of proteins*	*No. of PPIs*
Graemlin data set		
S. coelicolor (100226)	6678	230409
E. coli E12 (83333)	4087	216326
M. tuberculosis (83332)	3457	128932
S. typhimurium (99287)	4239	94609
C. crescentus (190650)	3341	40524
V. cholerae (243277)	2948	36038
S. pneumoniae (170187)	1843	25726
C. jejuni (192222)	1442	22116
H. pylori (85962)	1070	12943
Synechocystis sp. (1148)	2371	69439
NetworkBLAST data set		
S. cerevisiae (4932)	4738	15147
C. elegans (6239)	2853	4472
D. melanogaster (7227)	7165	23484
CAPPI data set		
S. cerevisiae (4932)	4726	15103
C. elegans (6239)	2619	3950
D. melanogaster (7227)	7032	20782

Table 2. NetworkBLAST-M Performance Evaluation of Relaxed versus Restricted Order Configurations for All Possible Orientations of Spines in a Yeast-Worm-Fly Data Set

Species	*Specificity (%)*	*No. of GO categories enriched*
Relaxed order		
S. cerevisiae	94.6	45
C. elegans	67.0	29
D. melanogaster	90.1	41
(Y,(W,(F))) order		
S. cerevisiae	100.0	32
C. elegans	65.6	29
D. melanogaster	98.4	37
(Y,(F,(W))) order		
S. cerevisiae	98.5	35
C. elegans	61.9	23
D. melanogaster	92.4	33
(W,(Y,(F))) order		
S. cerevisiae	98.0	27
C. elegans	62.0	17
D. melanogaster	88.0	29

and, hence, is limited in application to the alignment of up to 3 networks. Both methods use the same scoring function, and scoring parameters were set equal for both methods for fair comparison (which is the default configuration for both applications). The results in Table 3 show that the performance of NetworkBLAST-M is comparable to that of NetworkBLAST. The latter has higher specificity, but fewer GO categories enriched. The sensitivity of NetworkBLAST-M further improves when using the relaxed-order variant. Notably, the application of NetworkBLAST-M took less than 30 seconds in both configurations, while NetworkBLAST's run took more than six hours. The difference in the sensitivities of the two algorithms is due to a more relaxed definition of a *k*-spine (or an alignment graph node) that is used in the current work as compared to that in Sharan et al. (2005b).

To compare to the CAPPI method (Dutkowski and Tiuryn, 2007), we applied both methods to the data set which was used in the original publication, executing them with default parameters. In order to apply NetworkBLAST-M to these data we assigned identical scores (0.8) to all interactions, as no reliability information was available for them. As shown in Table 4, the NetworkBLAST-M yields generally higher specificity and consistently higher sensitivity than CAPPI.

Table 3. Comparison of NetworkBLAST-M and NetworkBLAST on Three Eukaryotic Networks from the NetworkBLAST Data Set

Species	*Specificity (%)*	*No. of GO categories enriched*
NetworkBLAST		
S. cerevisiae	100.0	14
C. elegans	88.0	13
D. melanogaster	94.9	16
NetworkBLAST-M restricted order		
S. cerevisiae	100.0	29
C. elegans	65.6	29
D. melanogaster	98.4	37
NetworkBLAST-M relaxed order		
S. cerevisiae	94.6	45
C. elegans	67.0	29
D. melanogaster	90.1	41

For these networks, NetworkBLAST produced 59 conserved regions, while NetworkBLAST-M identified 64 regions in the restricted-order variant and 92 in the relaxed-order variant, with sizes ranging from 8 to 15 *k*-spines.

Table 4. Comparison of NetworkBLAST-M and CAPPI on Three Eukaryotic Networks from the CAPPI Data Set

Species	*Specificity (%)*	*No. of GO categories enriched*
CAPPI		
S. cerevisiae	91.4	29
C. elegans	70.0	21
D. melanogaster	72.7	20
NetworkBLAST-M restricted order		
S. cerevisiae	97.3	40
C. elegans	59.2	25
D. melanogaster	85.1	39
NetworkBLAST-M relaxed order		
S. cerevisiae	99.0	46
C. elegans	61.1	21
D. melanogaster	86.7	48

For these networks, CAPPI reported 38 conserved regions and NetworkBLAST-M identified 74 conserved subnetworks in the restricted-order variant and 98 in the relaxed-order variant, with sizes ranging from 5 to 15 *k*-spines.

Next, we compared the performance of NetworkBLAST-M to that of Graemlin (Flannick et al., 2006) on a set of 10 microbial networks. Graemlin's results were taken from the original publication, considering only alignments which contain all 10 species (a total of 21 conserved regions). NetworkBLAST-M was applied only in the restricted-order variant due to the high computation burden. The algorithm detected a total of 33 conserved network regions. As summarized in Table 5, NetworkBLAST-M outperforms Graemlin, providing uniformly higher specificity and sensitivity.

Table 5. Comparison of NetworkBLAST-M and Graemlin on 10 Microbial Networks

Species	*Specificity (%)*	*No. of GO categories enriched*
NetworkBLAST-M restricted order		
S. coelicolor	100	17
E. coli E12	90	16
M. tuberculosis	87.9	17
S. typhimurium	93.1	14
C. crescentus	84.8	15
V. cholerae	90.6	16
S. pneumoniae	97.0	14
C. jejuni	96.2	12
H. pylori	92.3	13
Synechocystis	N/A	N/A
Graemlin		
S. coelicolor	71.4	12
E. coli E12	76.5	10
M. tuberculosis	76.9	8
S. typhimurium	81.3	10
C. crescentus	86.7	11
V. cholerae	80.0	9
S. pneumoniae	71.4	8
C. jejuni	76.9	9
H. pylori	56.3	8
Synechocystis	N/A	N/A

Results are provided for nine of the ten species for which we had gene ontology information (for Synechocystis, we did not have functional information readily available). The conserved subnetworks detected by NetworkBLAST-M ranged in size from 5 to 15 *k*-spines.

Table 6. NetworkBLAST-M Run-Time as a Function of the Number of Species and the Size of the Layered Alignment Graph

No. of species	*No. of nodes*	*No. of PPI edges*	*No. of sequence similarity edges*	*Restricted order run time (sec)*	*Relaxed order run time (sec)*
3	8132	102288	26834	40	44
5	11945	193843	57142	72	1587
7	17236	301365	103887	83	46686
10	31458	877032	327219	140	N/A

All the tests were performed on Intel Xeon 3.06-GHz 3-GB memory machine.

Statistics on the running times of NetworkBLAST-M on different sets of microbial networks with 3–10 species are given in Table 6. Evidently, the restricted-order variant is considerably faster than the relaxed-order variant and can process up to 10 networks in minutes.

4. CONCLUSION

We have provided a fast and accurate framework for multiple network alignment. Our framework is based on a novel representation of multiple protein-protein interaction networks and the orthology relations among their proteins. The framework performs comparably to an exhaustive approach while allowing dramatic reduction in running time and memory requirements. It is shown to outperform previous approaches for the problem.

Future research includes a more extensive comparison of the different seed computation variants presented here, as well as experimenting with other scoring functions. For example, it could be interesting to incorporate into the NetworkBLAST-M framework a scoring function which is based on modeling protein complex evolution, extending the method of Hirsh and Sharan (2006) to multiple networks.

The development of efficient network alignment techniques, such as the one described here, is crucial to the study of protein network evolution and is expected to become increasingly important as protein-protein interaction databases continue to grow in size and species coverage.

5. APPENDIX

Table A1. Comparison of NetworkBLAST-M and CAPPI with Respect to Cellular Component and Molecular Function Ontologies

	Cellular component		*Molecular function*	
Species	*Specificity (%)*	*No. of GO categories enriched*	*Specificity (%)*	*No. of GO categories enriched*
CAPPI				
S. cerevisiae	85.7	29	85.7	29
C. elegans	42.3	11	48.3	13
D. melanogaster	74.1	20	63.6	21
NetworkBLAST-M (restricted order)				
S. cerevisiae	74.3	29	98.6	24
C. elegans	40.6	21	70.3	19
D. melanogaster	50.0	29	81.1	30
NetworkBLAST-M (relaxed order)				
S. cerevisiae	78.6	33	96.9	26
C. elegans	46.3	20	69.4	15
D. melanogaster	43.3	29	84.7	34

Table A2. A Comparison of NetworkBLAST-M and Graemlin with Respect to Cellular Component and Molecular Function Ontologies

	Cellular component		*Molecular function*	
Species	*Specificity (%)*	*No. of GO categories enriched*	*Specificity (%)*	*No. of GO categories enriched*
NetworkBLAST-M (restricted order)				
S. coelicolor	95.5	7	100	18
E. coli E12	63.0	7	93.9	19
M. tuberculosis	66.7	5	93.9	20
S. typhimurium	61.5	7	100	18
C. crescentus	67.9	8	93.9	18
V. cholerae	88.5	8	97.0	20
S. pneumoniae	64.0	6	100	18
C. jejuni	57.1	5	96.7	15
H. pylori	63.6	6	90.9	15
Synechocystis	N/A	N/A	N/A	N/A
Graemlin				
S. coelicolor	66.7	6	82.4	13
E. coli E12	46.2	6	82.4	13
M. tuberculosis	63.6	4	76.9	9
S. typhimurium	76.9	7	87.5	13
C. crescentus	66.7	5	93.8	13
V. cholerae	76.9	7	75.0	12
S. pneumoniae	75.0	7	92.9	12
C. jejuni	62.5	3	78.6	11
H. pylori	47.7	4	68.8	11
Synechocystis	N/A	N/A	N/A	N/A

ACKNOWLEDGMENTS

V.B. was supported in part by a research gift from Glaxo SmithKline. This research was supported by the Israel Science Foundation (grant 385/06).

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

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

Ashburner, M., et al. 2000. The gene ontology consortium. gene ontology: tool for the unification of biology. *Nat. Genet.* 25, 25–29.

Benjamini, Y., and Hochberg, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc.* 57, 289–300.

Boyle, E., Weng, S., Gollub, J., et al. 2004. GO::TermFinder—open source software for accessing Gene Ontology information and finding significantly enriched Gene Ontology terms associated with a list of genes. *Bioinformatics* 20, 3710–3715.

Dutkowski, J., and Tiuryn, J. 2007. Identification of functional modules from conserved ancestral protein-protein interactions. *Bioinformatics* 23, 149–158.

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

Gavin, A., et al. 2002. Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature* 415, 141–147.

Hirsh, E., and Sharan, R. 2006. Identification of conserved protein complexes based on a model of protein network evolution. *Bioinformatics*.

Ho, Y., et al. 2002. Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. *Nature* 415, 180–183.

Ito, T., Chiba, T., and Yoshida, M. 2001a. Exploring the yeast protein interactome using comprehensive two-hybrid projects. *Trends Biotechnol.* 19, 23–27.

Ito, T., et al. 2001b. A comprehensive two-hybrid analysis to explore the yeast protein interactome. *Proc. Natl. Acad. Sci. USA* 98, 4569–4574.

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

Koyuturk, M., et al. 2006. Pairwise local alignment of protein interaction networks guided by models of evolution. *J. Comput. Biol.* 13, 182–199.

Pellegrini, M., Marcotte, E., Thompson, M., et al. 1999. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc. Natl. Acad. Sci. USA* 96, 4285–4288.

Shamir, R., Sharan, R., and Tsur, D. 2004. Cluster graph modification problems. *Discrete Appl. Math.* 144, 173–182.

Sharan, R., Ideker, T., Kelley, B., et al. 2005a. Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. *J. Comput. Biol.* 12, 835–846.

Sharan, R., et al. 2005b. Conserved patterns of protein interaction in multiple species. *Proc. Natl. Acad. Sci. USA* 102, 1974–1979.

Stelzl, U., et al. 2005. A human protein-protein interaction network: a resource for annotating the proteome. *Cell* 122, 830–832.

Uetz, P., et al. 2000. A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae*. *Nature* 403, 623–627.

Address correspondence to:
Dr. Roded Sharan
School of Computer Science
Tel Aviv University
Tel Aviv, 69978 Israel

E-mail: roded@post.tau.ac.il