Fast and Accurate Alignment of Multiple Protein Networks

Maxim Kalaev[1], Vineet Bafna[2], and Roded Sharan[1]

[1] School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel.
{kalaevma,roded}@post.tau.ac.il
[2] CSE, University of California San Diego, USA. vbafna@cs.ucsd.edu

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 a previous strategy for the problem that is based on progressive alignment, and produces results that are more in line with current biological knowledge.

1 Introduction

Recent technological advances enable the systematic characterization of protein-protein interaction (PPI) networks across multiple species. Procedures such as yeast two-hybrid ([1]) and protein co-immunoprecipitation ([2]) are routinely employed nowadays to generate large-scale protein interaction networks for human and most model species ([3–7]). 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 [8] and complexes [9–11]. However, its extension to more than a few (3) 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, proposing the idea of imitating progressive sequence alignment techniques [12]. The latter algorithm was successfully applied to align up to 10 microbial networks. Very recently, Dutkowsky and Tiuryn [13] proposed another framework for efficient alignment of multiple networks, but this approach was applied to date to 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 allows avoiding 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 previous approaches using various data sets, showing that it is extremely fast and accurate, outperforming the progressive alignment approach. For lack of space, some proofs are shortened or omitted.

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 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. For exposition purposes, assume that $|V_i| = n$ for all i. Additionally, we have a set of *inter-layer* denoted by E_H. Let $G_H = (\cup_i V_i, E_H)$ denote the graph restricted to the inter-layer edges. Let δ be the largest degree in G_H. 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 [11]. 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. 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 (see [11] for further details).

This notion of a conserved subnetwork is extended easily to a layered alignment graph. If we considered every k-spine to be a (super-)node in a graph, then an m-node subgraph is a subgraph of m k-spines, with a dense interconnection of PPI edges. Formally, define an *m-subnet* as a collection U of k multi-sets $U_i = \{u_i[1], \ldots, u_i[m]\}$ with the following properties:

- For all $1 \leq i \leq k$ and $1 \leq j \leq m$, $u_i[j] \in V_i$.
- For all $1 \leq j \leq m$, the set $U[j] = \{u_1[j], u_2[j], \ldots, u_k[j]\}$ is a k-spine.

The score $\mathscr{S}(U)$ of the m-subnet is given by $\mathscr{S}(U) = \sum_{i=1}^{k} \mathcal{L}(U_i)$.

2.2 The search algorithm

The main algorithmic task is to look for high scoring m-subnets, for a fixed m. 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 [14]. 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 [11].

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 $d << m$. 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 sized networks, so we consider two assumptions on the inter-layer edges that reduce the computational complexity while retaining sensitivity.

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 [15]. 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* $O((n\delta)^d k 3^k)$ *solution.*

Proof. Recall that a d-subnet can be described as a collection U of size d multisets $U_1, U_2, \ldots U_k$. Let $(U_{i_1}, U_{i_2}) \in E_H$ iff $(u_{i_1}[j], u_{i_2}[j]) \in E_H$ for all $1 \leq j \leq d$.

First, consider the case where each of the d k-spines is restricted to be a path (Figure 1). This implies that the d-subnet itself can be considered as a path $U_{i_1}, U_{i_2}, \ldots, U_{i_k}$. For a subset of species S, let $\mathscr{S}(U, S)$ denote the score of the best d-subnet that uses only species in S, and consists of a path that ends with U. Let $s(U)$ be the species corresponding to U. To compute $\mathscr{S}(U, S)$, note that we only need to recurse using the predecessor of U in the path. Formally:

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

Thus, for paths, the overall complexity is $O((n\delta)^d k 2^k)$.

A similar recursion can be applied when searching for k-spines that are trees with identical topology. For a subset of species S, let $\mathscr{S}(U, S)$ 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 $|S| > 1$:

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

The overall complexity is $O((n\delta)^d k 3^k)$. ♣

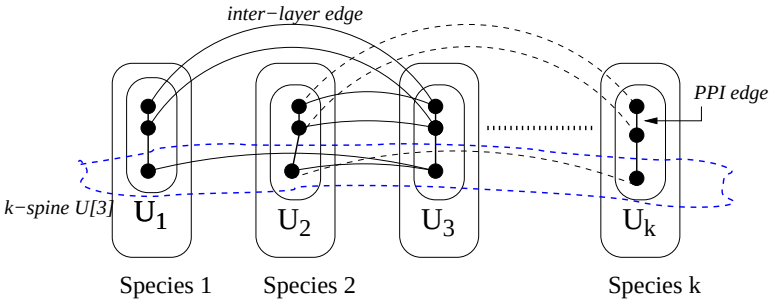

Fig. 1. A seed defined by a d-identical-spine subnet, where the k-spines are restricted to be paths with identical topology. The dashed line encloses one of the three k-spines.

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 nodes a, b, c whose underlying species follow the phylogenetic triple $(s(a), (s(b), s(c)))$.

We make the following *phylogenetic* assumption: if a, b, c are connected via inter-layer edges, then b and c must be connected. 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 (see Figure 2).

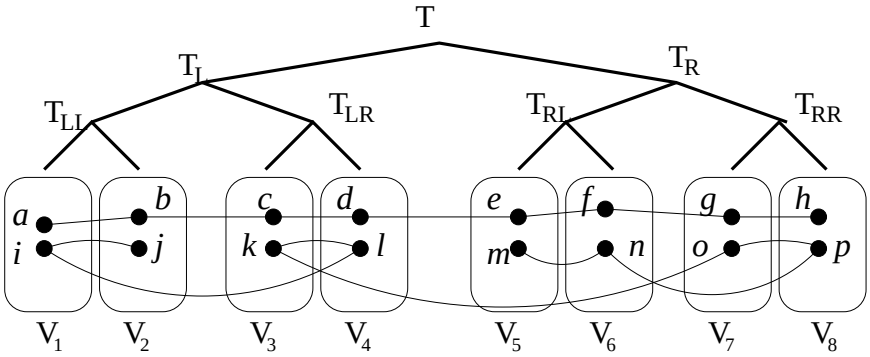

Fig. 2. Sketch of a 2-*guided-spine-subnet* . Note that while the paths of the two k-spines have different topologies, they are both guided by the underlying tree. Following the notation in the proof of Theorem 2, let $U = \{a, j\}, W = \{h, m\}$, and consider two possible distant sets $X = \{d, k\}$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

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 for *d-guided-path*s, where the k-spines are restricted to be paths guided by the phylogeny.

Theorem 2. *The d-guided-path-subnet problem can be solved in* $O(k^3(n^3\delta)^d)$.

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 species topology of 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 $\mathscr{S}(U, W, T)$ denote the best score of a d-guided-path-subnet restricted to 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 $\mathscr{S}(U, U, T) = \mathcal{L}(U)$.

Otherwise, let $U = \langle u[1], u[2] \dots u[d] \rangle \in T_L$, and $W = \langle w[1], w[2] \dots w[d] \rangle \in T_R$. Denote the root of T by $root(T)$.

For a node u, s.t. $s(u) \in T$, define its *distant set* $\mathcal{D}_T(u) = \{x | \text{LCA}_T(s(u), s(x)) = \text{root}(T)\}$, where $LCA_T(a, b)$ is the least common ancestor of a and b in T. Extend this to d elements by defining $\mathcal{D}_T(U) = \{X | \text{LCA}_T(s(u[j]), s(x[j])) = \text{root}(T) \forall j\}$ The key idea to note is that 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 $T_L(U \cup X)$ ($T_R(U \cup X)$) as the set of all vertices in $U \cup X$ with species in T_L (T_R). Then,

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

For an example see Figure 2. For the running time, note that there are $k^2 n^{2d}$ cells in the table $\mathscr{S}$. For each cell, there are kn^d choices for the set X and for each there are δ^d choices for a set Y s.t. $(X, Y) \in E_H$. The total time is therefore $O(k^3(n^3\delta)^d)$. ♣

In fact, we can improve the running time to $O((k^2 n^2 \delta)^d)$ (the proof will appear in the full version of the paper), but this is still not practical for reasonable values of n.

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. Let us denote by $H = (V', E')$ the current seed, and by $\mathscr{S}(v, S)$ the score of the best partial extension of H by a subtree that is rooted at vertex v and visits the species in S. Further denote by $s(v)$ the species corresponding to vertex v, and let $W(v) = \sum_{u \in V'} w(u, v)$. Then $\mathscr{S}(v, S)$ can be computed using the following recursive relation:

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

The overall complexity is $O(n\delta k 3^k)$.

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

2.3 Implementation notes

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 spines constrained to trees with relaxed and restricted topologies. For efficiency reasons, we restricted the seed vertices in each network to be of distance at most 2 from one another.

To verify that using 2-identical-spines is adequate for our problem, we analyzed alignment nodes within conserved network regions output by NetworkBlast [11] for different networks sets. When aligning yeast, worm and fly networks, in 85% of the cases, the pertaining alignment nodes respected the yeast-worm-fly phylogeny-based orientation. In two additional microbial network sets (C. jejuni, E. coli, H. pylori and C. crescentus, V. cholerae and H. pylori) more than 95% of the alignment nodes respected the same phylogeny-based orientation. Moreover, 72% of the alignment nodes actually formed cliques in G_H.

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 rate is computed as the number of common proteins over $\min\{|A|, |B|\}$.

3 Results

We applied our algorithm to eukaryotic and microbial PPI networks, summarized in Table 1. The three eukaryotic networks were taken from [11] and the microbial networks were taken from [12]. As in [11], we used a BLAST E-value threshold of 10^{-7} for sequence similarity, ensuring a corrected significance value of 0.01.

Table 1. A summary of the PPI networks analyzed in this study.

Species (tax id)	#Proteins	#PPIs
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
S. cerevisiae (4932)	4738	15147
C. elegans (6239	2853	4472
D. melanogaster (7227)	7165	23484

We evaluated the identified conserved subnetworks by computing the functional coherency of their member proteins with respect to the biological process annotation of the gene ontology (GO) [16], for each species separately. To

this end, we used the GO TermFinder tool [17] to compute empirical enrichment p-values, and corrected for multiple testing using the false discovery rate procedure [18]. For each species we report the percent of process 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.

To establish the validity of our method, we first compared it to NetworkBLAST [11]. NetworkBLAST is an exhaustive approach that relies on explicitly constructing a network alignment graph and, hence, is limited in application to the alignment of up to 3 networks. Both methods use same scoring function and scoring parameters were set equal for both methods for fair comparison. The results in Table 2 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.

Table 2. A comparison of NetworkBLAST-M and NetworkBLAST on three eukaryotic networks. 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.

Species	Specificity (%)	# 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	68.8	32
D. melanogaster	98.4	37
NetworkBLAST-M relaxed order		
S. cerevisiae	94.6	45
C. elegans	67.0	29
D. melanogaster	90.1	41

Next, we compared the performance of NetworkBLAST-M to that of Graemlin [12] 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 3,

NetworkBLAST-M outperforms Graemlin, providing uniformly higher specificity and sensitivity.

Table 3. A comparison of NetworkBLAST-M and Graemlin on 10 microbial networks. 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).

Species	Specificity (%)	# 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

Statistics on the running times of NetworkBLAST-M on different sets of microbial networks with 3-10 species are given in Table 4. As evident, the restricted-order variant is considerably faster and can process up to 10 networks in minutes.

4 Conclusions

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 a previous approach based on progressive alignment ideas.

Table 4. NetworkBLAST-M run-time as a function of the number of species and the size of the layered alignment graph. All the tests were performed on Intel Xeon 3.06GHz 3GB memory machine.

#Species	#Nodes	#PPI edges	#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

Future research includes a more extensive comparison of the different seed computation variants presented here. Our initial experiments in this regard indicate that the relaxed-order yields higher sensitivity on eukaryotic data sets, while the two perform similarly on microbial networks (data not shown). This may reflect the fact that sequence similarity among the pertaining microbial proteins tends to be transitive and, hence, any order of the species will form a tree in G_H. 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.

Acknowledgments

VB was supported in part by a research gift from Glaxo SmithKline. This research was supported by the Israel Science Foundation (grant no. 385/06).

References

1. Ito, T., Chiba, T., Yoshida, M.: Exploring the yeast protein interactome using comprehensive two-hybrid projects. Trends Biotechnology **19** (2001) 23–27
2. Aebersold, R., Mann, M.: Mass spectrometry-based proteomics. Nature **422** (2003) 198–207
3. Uetz, P., et al.: A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. Nature **403** (2000) 623–627
4. Ito, T., et al.: A comprehensive two-hybrid analysis to explore the yeast protein interactome. Proc. Natl. Acad. Sci. USA **98** (2001) 4569–4574
5. Ho, Y., et al.: Systematic identification of protein complexes in Saccharomyces cerevisiae by mass spectrometry. Nature **415** (2002) 180–183
6. Gavin, A., et al.: Functional organization of the yeast proteome by systematic analysis of protein complexes. Nature **415** (2002) 141–147
7. Stelzl, U., et al.: A human protein-protein interaction network: a resource for annotating the proteome. Cell **122** (2005) 830–2
8. Kelley, B., et al.: Conserved pathways within bacteria and yeast as revealed by global protein network alignment. Proc. Natl. Acad. Sci. **100** (2003) 11394–9

9. Sharan, R., Ideker, T., Kelley, B., Shamir, R., Karp, R.: Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. Journal of Computational Biology **12** (2005) 835–846
10. Koyuturk, M., et al.: Pairwise local alignment of protein interaction networks guided by models of evolution. Journal of Computational Biology **13** (2006) 182–99
11. Sharan, R., et al.: Conserved patterns of protein interaction in multiple species. Proc. Natl. Acad. Sci. **102** (2005) 1974–9
12. Flannick, J., Novak, A., Srinivasan, B., McAdams, H., Batzoglou, S.: Graemlin:general and robust alignment of multiple large interaction networks. Genome Research **16** (2006) 1169–1181
13. Dutkowsky, J., Tiuryn, J.: Identification of functional modules from conserved ancestral protein-protein interactions. Bioinformatics **23** (2007) 149–58
14. Shamir, R., Sharan, R., Tsur, D.: Cluster graph modification problems. Discrete Applied Mathematics **144** (2004) 173–182
15. Pellegrini, M., Marcotte, E.M., Thompson, M.J., Eisenberg, D., Yeates, T.O.: Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. PNAS **96** (1999) 4285–4288
16. Ashburner, M., et al.: The gene ontology consortium. gene ontology: Tool for the unification of biology. Nature Genetics **25** (2000) 25–29
17. Boyle, E., Weng, S., Gollub, J., Jin, H., Botstein, D., Cherry, J., Sherlock, G.: 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** (2004) 3710–15
18. Benjamini, Y., Hochberg, Y.: Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society **57 (1)** (1995) 289–300