Divide, Align and Full-Search for Discovering Conserved Protein Complexes

Pavol Jancura [1,*], Jaap Heringa [2], Elena Marchiori [1]

[1]ICIS, Radboud University Nijmegen, The Netherlands
[2] IBIVU, Vrije Universiteit Amsterdam, The Netherlands
{jancura,heringa,elena}@few.vu.nl

Abstract. Advances in modern technologies for measuring protein-protein interaction (PPI) has boosted research in PPI networks analysis and comparison. One of the challenging problems in comparative analysis of PPI networks is the comparison of networks across species for discovering conserved modules. Approaches for this task generally merge the considered networks into one new weighted graph, called alignment graph, which describes how interaction between each pair of proteins is preserved in different networks. The problem of finding conserved protein complexes across species is then transformed into the problem of searching the alignment graph for subnetworks whose weights satisfy a given constraint. Because the latter problem is computationally intractable, generally greedy techniques are used. In this paper we propose an alternative approach for this task. First, we use a technique we recently introduced for dividing PPI networks into small subnets which are likely to contain conserved modules. Next, we perform network alignment on pairs of resulting subnets from different species, and apply an exact search algorithm iteratively on each alignment graph, each time changing the constraint based on the weight of the solution found in the previous iteration. Results of experiments show that this method discovers multiple accurate conserved modules, and can be used for refining state-of-the-art algorithms for comparative network analysis.

Keywords: biological networks alignment, optimization

1 Introduction

With the recent advances in modern technologies for measuring protein-protein interaction, an exponential increase of data on protein-protein interactions has been generated. Data on thousands of interactions in human and most model species have become available (e.g. [1, 2]). Graph-representation of PPI interaction of proteins provides a powerful tool for analyzing and understanding modular organization of cells, for predicting biological functions and for providing insight into a variety of biochemical processes. Recent studies consider a comparative approach for the analysis of PPI networks from different species in order

* Corresponding author

to discover common protein groups which are likely to share relevant functions [3–5]. In particular, this problem is called *pairwise network alignment* when two PPI networks are considered. Algorithms for this problem generally construct a merged weighted graph representation of the two networks, called alignment (or orthology) graph, which describes how interaction between each pair of proteins is preserved in different networks. The problem of finding conserved protein complexes across species is then transformed into the problem of searching the alignment graph for subnetworks whose weights satisfy a given constraint. Due to the computational intractability of such problem, greedy algorithms are commonly used [6, 7]. Conserved modules, discovered by computational techniques such as [6], have in general small size compared to the size of the PPI network they belong to. Moreover, as indicated by recent studies, hubs whose removal disconnects the PPI network (articulation hubs) are likely to appear in conserved interaction patterns [8, 9]. Based on these motivation, in [10] we introduced an algorithm, called `DivA` for dividing a pair of PPI networks into small subnets which are expected to cover conserved modules, with the goal of performing modular network alignment. We used this algorithm for performing network alignment in a modular way, by merging pairs of resulting subnets from different species, and then applying an exact optimization algorithm for finding the heaviest subgraph of a weighted graph. Application of this algorithm generates one solution for each alignment subnet. In this paper we propose an extension of this search algorithm which allows to detect an higher number of conserved modules of biological interest. Specifically, the idea is to modify the exact search algorithm for finding the heaviest subgraph of an alignment network, by introducing an upper bound on the maximum weight of the subgraph to be found. Iterated runs of this constrained algorithm are performed, with different values of the upper bound generated at each iteration using the weight of the solution found in the previous iteration. We call this search approach *full-search*. In this way multiple subnets of the alignment network are discovered. The resulting method, called `DivAfull`, divides each PPI network into subnets using `DivA`, aligns pairs of subnets from different species, and performs full-search on each aligning pair. We use the state-of-the-art evolution-based alignment graph model introduced in [6] to construct an alignment graph. Results of experiments show effectiveness of the proposed approach, which is capable of detecting an high number of accurate conserved complexes. This number is considerably greater than the number of results identified only by using `DivA` whereas `DivAfull`'s results contain all `DivA`'s results. Furthermore, we show that improved performance is achieved by merging solutions discovered by `DivAfull` with those identified by Koyuturk et al.'s algorithm [6].

Recent overviews of approaches and issues in comparative biological networks analysis are presented in [4, 5]. Based on the general formulation of network alignment proposed in [3], a number of techniques for (local and global) network alignment have been introduced ([6, 7, 11, 12]). Techniques for local network alignment commonly construct an orthology graph, which provides a merged representation of the given PPI networks, and search for conserved subnets using

greedy techniques ([6],[7],[11]). In particular, in [11], d-clusters are defined for searching efficiently between a pair of networks, where a d-cluster consists of d proteins that are close together in the network, and d is a user-given parameter. Another parameter is used for identifying pairs of d-clusters, one from each network, called seeds, which provide starting regions of the alignment graph to be expanded. The algorithm searches for modules conserved across species by expanding these seeds using a greedy technique similar that used in [6],[7]. While the above algorithms focus on network alignment, we focus on 'modular' network alignment. Many papers have investigated the importance of hubs in PPI networks and functional groups [9, 13–17]. In particular, it has been shown that hubs with a central role in the network architecture are three times more likely to be essential than proteins with only a small number of links to other proteins [15]. Moreover, if we take functional groups in PPI networks, then, amongst all functional groups, cellular organization proteins have the largest presence in hubs whose removal disconnects the network [9]. Computational techniques for identifying functional modules in PPI networks generally search for clusters of proteins forming dense components [18, 19]. The scale-free topology of PPI networks makes difficult to isolate modules hidden inside the central core [20]. In [21] several multi-level graph partitioning algorithms are described addressing the difficulty of partitioning scale-free graphs. The approach we propose differs from the above mentioned works because it does not address (directly) the problem of identifying functional modules in a PPI network, but combines graph-theory, biology and heuristic search for discovering conserved protein complexes in a modular fashion.

2 Divide Align and Full-Search

Given a graph $G = (V, E)$, nodes joined by an edge are called *adjacent*. A *neighbor* of a node u is a node adjacent to u. The degree of u is the number of elements in E containing the vertex u.

Let $G(V, E)$ be a connected undirected graph. A vertex $v \in V$ is called *articulation* if the graph resulting by removing this vertex from G and all its edges, is not connected.

The *Divide algorithm* divides orthologous proteins of the PPI network into subsets. It consists of the following steps:

1. Detect orthologous articulations of the PPI network.
2. Reduce their number by constructing centers using preferential attachment property .
3. In parallel, incrementally expand from each center only alongside orthologous neighbors.
4. Stop when expanding sets are starting to overlap and if they do not have any orthologous neighbor which is not yet added to one of the actual sets.
5. If some orthologous nodes are not in any of the generated set, then join together neighboring ones.

The preferential attachment in the step 2 is a general property of scale-free networks. It means that if a new node is introduced into the network, it will more likely attach to a node of the network with very high degree than to a node with very low degree. Hence, based on this motivation, we construct centers by joining one orthologous articulation hub with its orthologous articulation neighbors, which will more likely to have low degree. The whole algorithm with all technical issues is described in [10].

After dividing, each set of orthologs proteins generates a subnetwork of the PPI network. Pairs of such subnetworks from distinct species can be merged into orthology graphs, which are mined for discovering alignments corresponding to protein complexes conserved across species.

To this aim we use a common approach, based on the construction of a weighted metagraph between two PPI networks of different species. In this metagraph each node corresponds to an homologous pair of proteins, one from each of the two PPI networks. The metagraph is called *alignment* or *orthology graph*. Weights are assigned either to edges, like in [6], or to nodes, like in [7], of the alignment graph using a scoring function. The function transforms conservation and eventually also evolution information to one real value for each edge or node. Induced subgraphs with total weight greater than a given threshold are considered to be relevant *alignments*. In this way one gets two subsets of proteins from each discovered subgraph from the two species, and each such subset provides a conserved complex of proteins.

The problem of finding induced subgraphs with weight greater than a given threshold is reduced in these methods to the problem of finding a maximal induced subgraph. Then an approximation greedy algorithm based on local search is used because the maximum induced subgraph problem is NP-complete (cf. [6]).

In our approach, we align only pairs of subnets from different species having more than one orthologous pair, yielding orthology graphs with more than one node. Because of the small size of the resulting subnets, we use exact optimization [22] for searching in each of such graphs, instead of greedy techniques employed in common approaches.

Specifically, the exact optimization algorithm [22] for finding the maximum weighted induced subgraph is first applied. Then the process is iterated by adding at each iteration the constraint which bounds the weight of the induced subgraph by the weight of the solution found in the previous iteration.

Formally, let f be a function which computes the weight of a subgraph in an input graph and C be a set of constraints which defines an induced subgraph of the input graph. Then we want to maximize the function f on the set defined by constrains C, that is, to solve the following optimization problem:

$$opt = \max_{C} f \qquad (OptP)$$

Algorithm 1 illustrates the resulting full-search procedure which uses the above constrained optimization problem at each iteration with different bound on the maximum allowed weight.

Algorithm 1 Full Search Algorithm

Input: G: alignment subnetwork, $\varepsilon \geq 0$
Output: List of heavy induced subgraphs of G with weight $> \varepsilon$

```
Formulate the problem of MaxInducedSubGraph for G as (OptP)
maxweight = ∞
C = C + {opt < maxweight}
while maxweight > ε do
    solve (OptP) by an exact method
    if opt > ε then
        record discovered solution
    end if
    maxweight = opt
end while
```

We call the resulting algorithm `DivAfull`. Finally, redundant alignments are filtered out as done in, e.g., [6]. A subgraph G_1 is said to be *redundant* if there exists another subgraph G_2 which contains $r\%$ of its nodes, where r is a threshold value that determines the extent of allowed overlap between discovered protein complexes. In such a case we say that G_1 *is redundant for* G_2.

3 Evaluation Criteria

In order to assess the performance of our approach, we use the state-of-the-art framework for comparative network analysis proposed in [23], where we change the proposed aligning procedure and searching algorithm to `MaWish` ([8]).

In order to filter out solutions that may also be found when a randomized protein-protein interaction relation between nodes is considered, we apply the following statistical procedure.

1. A collection of 10000 radomized networks are generated by shuffling the edges of the PPI networks while preserving vertex degrees, as well as by shuffling the pairs of homologous proteins while preserving the number of homologous partners per protein.
2. `MaWish` is used for finding solutions on each of the randomized networks.
3. The results are clustered into groups of solutions with equal size (that is, number of subnetwork's nodes). For each size and for each run, the best result (the one with highest score) is recorded. If there is no solution for a given size, we build an artificial cluster consisting of one zero weight solution.
4. For each size, the score at the 95%-percentile, of the corresponding cluster of random solutions, is chosen as treshold for removing 'insignificant' solutions.

We use known *yeast* complexes catalogued in the MIPS database. Category 550, which was obtained from high throughput experiments, is excluded and we retained only manually annotated complexes up to depth 3 in the MIPS tree category structure as standard of truth for quality assessment.

In order to measure statistically significant matches between a solution and a true complex we use the hypergeometric (HG) overlap score. The significance level of a solution is described by means of a function maximizing $-\log(HG)$ through the whole set of true complexes which intersect with the yeast PPI network at least in one protein. Solutions having no annotated protein in the MIPS catalogue are discarded.

We generate again a set of several (10000) radomized networks using the procedure described in the previous section. In each of such networks we find the most significant solution (which maximizes $-\log(HG)$) for each of the considered sizes, by modifying the algorithm `MaWish` in such a way that it outputs a solution of a given size (number of nodes). Specifically, in the incremental procedure `MaWish` at each cycle more than one node can be added in order to generate a subgraph with high weight. In the modified version of `MaWish` we use, if the size of subgraph has reached the given size, we stop. If the size of subgraph has exceeded the given size, we iteratively remove nodes with smallest gain for the actual subgraph, until a subgraph of the given size is obtained.

We compare significance levels of true solutions with those obtained from random networks. In this way we obtain empirical *p-values* for each of the solutions. These p-values are further corrected for multiple testing using the false discovery rate (FDR) procedure introduced in [24].

The following notions of specificity, sensitivity and purity are used to assess the quality of the results.

- Let C be the set of solutions with at least one annotated protein in MIPS catalogue and let $C^* \subseteq C$ be the subset of solutions with a significant match ($p < 0.05$). The *specificity* of the solution is defined as $|C^*|/|C|$.
- Let M be the set of true complexes that intersect with the yeast PPI network and let $M^* \subseteq M$ be the subset of complexes with a significant match by a solution. The *sensitivity* of the solution is defined as $|M^*|/|M|$.
- A solution is called *pure* if there exists a true complex whose intersection with the solution covers at least 75% of MIPS annotated proteins in the solution. Let D be the set of all solutions with at least 3 MIPS annotated proteins and let $D^* \subseteq D$ be the subset of pure solutions. The *purity* of the solutions is defined as $|D^*|/|D|$.

4 Results

The two following PPI networks, already compared in [8], are considered: *Saccharomyces cerevisiae* and *Caenorhabditis elegans*, which were obtained from BIND [1] and DIP [2] molecular interaction databases. The corresponding networks consist of 5157 proteins and 18192 interactions, and 3345 proteins and 5988 interactions, respectively. All these data are available at the webpage of `MaWish`[1]. Moreover, the data already contain the list of potential orthologous

[1] *www.cs.purdue.edu/homes/koyuturk/mawish/.*

and paralogous pairs, which are derived using BLAST E-values (for more details see [8]). We get 2746 potential orthologous pairs created by 792 proteins in S. cerevisiae and 633 proteins in C. elegans are identified.

We obtain 266 true complexes from the MIPS catalogue whose intersection with the yeast (Saccharomyces cerevisiae) PPI network consist of 876 proteins.

For *Saccharomyces cerevisiae*, 697 articulations, of which 151 orthologs, and 83 centers are identified. After expansion of these centers we covered 639 orthologs. The algorithm assigns the remaining 153 orthologous proteins to 152 new sets.

For *Caenorhabditis elegans*, 586 articulations, of which 158 orthologs, are computed, and 112 centers are constructed from them. Expansion of these centers covers 339 orthologs. The algorithm assigns the remaining orthologous 294 proteins to 288 new sets.

We observe that the last remaining orthologs assigned to new sets without expanding from centers are 'isolated' nodes, in the sense that they are rather distant from each other and not reachable from ortholog paths stemming from centers.

The dividing procedure generates 235 subnets of *Saccharomyces cerevisiae* and 400 subnets of *Caenorhabditis elegans*.

We perform network alignment with `MaWish` using the same parameter values as those reported in [8]. By constructing alignment graphs between each two subnets from different species containing more than one ortholog pair, we obtain 884 alignment graphs, where the biggest one consists of only 31 nodes.

We apply Algorithm 1 to each of the resulting alignment graphs. Zero weight threshold ($\varepsilon = 0$) is used for considering an induced subgraph as a heavy subgraph or a legal alignment. Redundant graphs are filtered using $r = 80\%$ as the treshold for redundancy.

In this way `DivAfull` discovers 151 solutions (alignments). By filtering out insignificant results we get 41 solutions.

Using only `DivA` we get 72 nonredundant alignments against 151 discovered by `DivAfull`. Because `DivA` takes only the first best possible solution from each alignment graph, all these solutions are also discovered by `DivAfull`. This happens in the first iteration of the latter algorithm. In the following iterations, `DivAfull` discovers other solutions, which have less weight than those discovered in the first iteration. Therefore the best solution can never be filtered out as redundant one. Hence after filtering, `DivAfull`'s results always contain all `DivA`'s solutions and a large number of other, potentially interesting, results identified by applying full search (Algorithm 1).

`MaWish` yields 83 solutions, and after filtering out insignificant results we get 34 solutions.

For both algorithms, we measure specificity, sensitivity and purity of all solutions and only of significant ones, in order to see whether results consider 'insignificant' are true noise in the data.

Moreover, we compare pairs of redundant alignments as well as new different results. A *paired redundant alignment* is a pair (G_1, G_2) of alignments, with G_1

discovered by `DivAfull` and G_2 discovered by `MaWish`, such that either G_1 is redundant for G_2 or vice versa. For a paired redundant alignment (G_1, G_2) we say that G_1 *refines* G_2 if the weight of G_1 is bigger than the weight of G_2.

Results of our experiments are summarized as follows.

Of the 83 solutions of `MaWish` 56 (67.5%) have at least one MIPS annotated protein and 15 (18.1%) have at least 3 annotated proteins. From the 151 `DivAfull` results, 103 (68.2%) have at least one annotated protein and 35 (23.2%) have at least 3 annotated proteins.

There are 70 redundant alignments, whose pair of weights are plotted on the left part of Fig. 1. Among these, 48 (31.8% of `DivAfull` results) are equal (red crosses in the diagonal) and 22 (14.6%) different. 8 (5.3%) (green crosses below the diagonal) with better `DivAfull` alignment weight, and 13(8.6%) (blue crosses above the diagonal) with better `MaWish` alignment weight (for 1 (0.7%) pair it is undecidable because of rounding errors during computation).

`DivAfull` finds 81 (53.6%) new alignments, that is, not discovered by `MaWish`. The right plot of Fig. 1 shows the binned distribution of weights of these alignments, together with the new 17 ones discovered by `MaWish` but not by `DivAfull`. There is no significant difference between the overall weight average of the `DivAfull` (0.8) and the the `MaWish` (0.86) results.

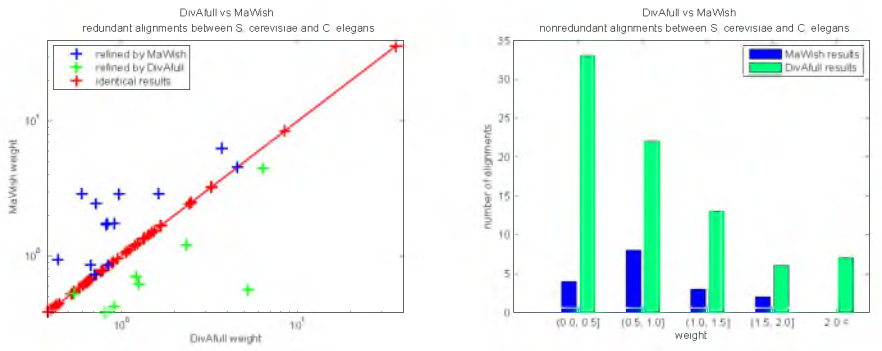

Fig. 1. Analysis all alignments discovered by `MaWish` and `DivAfull`. Left figure: Distribution of pairs of weights of paired redundant alignments, one obtained from `MaWish` and one from `DivAfull`. Weights of alignments found by `DivAfull` are on the x-axis, those found by `MaWish` on the y-axis. '+' is a paired redundant alignment. Right figure: Interval weight distributions of non-redundant alignments discovered by `MaWish` and `DivAfull` . The x-axis shows weight intervals, the y-axis the number of alignments in each interval.

By considering the union of all alignments discovered by `MaWish` and `DivAfull` and by filtering out the redundant ones, 164 alignments are obtained, from which 54.3% consist of refined or new `DivAfull` ones, and 29.3% consist of alignments discovered by both methods. Of all these alignments 111 have at least one annotated protein and 40 at least with 3 annotated proteins. This results indicate a

significant improvement (54.3%) of the performance of `MaWish` when augmented with `DivAfull`.

Statistical evaluation of all solution for `DivAfull` and `MaWish`, is reported in Table 1. One can observe that `DivAfull` outperforms `MaWish` and the number of `DivAfull` solutions is almost double of the number of `MaWish` ones. Combining results obtained by both algorithms generally increases sensitivity and purity, while specificity is increased only w.r.t `MaWish` solutions. The latter phenomenon can be justified by the effect of nonredundant `MaWish` results, since more of them do not have a significant match ($p < 0.05$) and therefore decrease overall specificity when combined with `DivAfull` solutions.

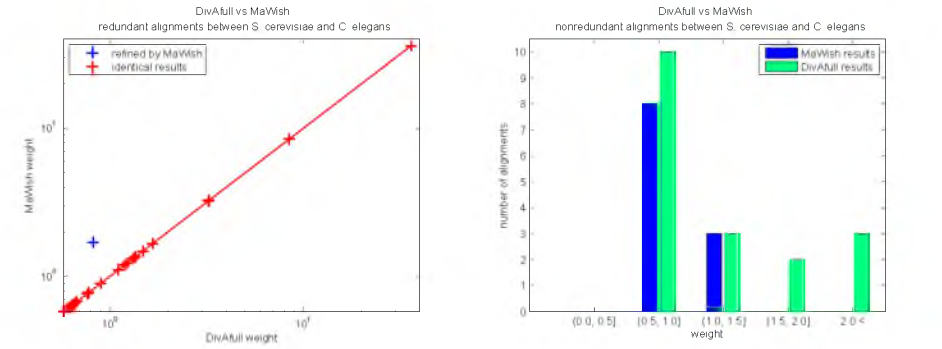

Fig. 2. Analysis significant alignments discovered by `MaWish` and `DivAfull`. Left figure: Distribution of pairs of weights of paired redundant alignments, one obtained from `MaWish` and one from `DivAfull`. Weights of alignments found by `DivAfull` are on the x-axis, those found by `MaWish` on the y-axis. '+' is a paired redundant alignment. Right figure: Interval weight distributions of non-redundant alignments discovered by `MaWish` and `DivAfull` . The x-axis shows weight intervals, the y-axis the number of alignments in each interval.

If the same analysis is performed only on the significant alignments then the following results are obtained.

From the significant 34 `MaWish` results, 25 (73.5%) have at least one annotated protein and 4 (11.8%) have at least 3 annotated proteins. From the significant 41 `DivAfull` results, 34 (83%) have at least one annotated protein and 10 (24.4%) have at least 3 annotated proteins.

`DivAfull` finds 18 new alignments not detected by `MaWish`. There are 23 redundant alignments. Among these, 22 (53.7% of `DivAfull` results) are equal and 1 (2.4%) different with better `MaWish` alignment weight.

The right plot of Fig. 2 shows the binned distribution of weights of the 18 (43.9%) found by `DivAfull` but not `MaWish`, and 11 found by `MaWish` and not by `DivAfull`. The overall weight average of the `DivAfull` ones (1.609) is greater than the overall average of the `MaWish` ones (0.8536).

By considering the union of all significant alignments of MaWish and DivAfull and by filtering out the redundant ones, we get together 52 alignments, from which 34.6% results are added as new ones by the DivAfull method and 42.3% are equal results discovered by both methods. This shows that performance of the MaWish model is improved by 34.6% when the algorithm is augmented with the DivAfull method. From all alignments, 41 have at least one annotated protein and 10 at least with 3 annotated proteins.

Table 2 report statistical evaluation of results of MaWish, DivAfull, and their union. DivAfull solutions have better specificity than MaWish solutions and similar sensitivity. Concerning purity, DivAfull has 7 pure solutions from 10 considered, while MaWish has 3 pure solutions from 4. Because of the small number of the considered alignments, the purity measure in this case does not provide sufficient information for comparing the two algorithms. Considering the union of MaWish and DivAfull generally increases sensitivity and specificity. Moreover, the new solutions added by DivAfull increase the number of pure alignments.

In summary, these results show that DivAfull can be successfully applied to discover conserved protein complexes and to 'refine' state-of-the-art algorithms for network alignment.

Table 1. Specificity, sensitivity and purity for all alignments discovered by DivAfull and MaWish. The first row of table shows results for combined solutions of both algorithms.

Algorithm	No. of alignments	Specificity (%)	Sensitivity (%)	Purity (%)
DivAfull & MaWish	164	44	6.8	92
DivAfull	151	46	6	91
MaWish	83	43	6	87

Table 2. Specificity, sensitivity and purity for significant alignments discovered by DivAfull and MaWish. The first row of table shows results for combined significant solutions of both algorithms.

Algorithm	No. of alignments	Specificity (%)	Sensitivity (%)	Purity (%)
DivAfull & MaWish	52	51	4.5	70
DivAfull	41	50	3.4	70
MaWish	34	48	3.8	75

5 Conclusion

This paper introduced a heuristic algorithm, DivAfull, for discovering conserved protein complexes, which is an extension of a previously proposed algorithm,

`DivA`. Results of the comparative experimental analysis indicated that `DivAfull` improves the search procedure of `DivA`. Moreover, comparison between `MaWish` and `DivAfull` indicated that `DivAfull` is able to discover new alignments which significantly increase the number of discovered complexes. `DivAfull` solutions showed also improved match with well-know yeast complexes measured by specificity, sensitivity and purity. Combination of solutions discovered by both `MaWish` and `DivAfull`, yielded new and refined alignments.

Although using an exact search in `DivAfull` requires higher computational time than the greedy searching of `MaWish` (in our experiment it took more than 4 hours on a desktop machine (AMD Athlon 64 Processor 3500+, 2 GB RAM)), the advantage of a modular approach relies also in possible parallelization of parts of the method. For instance, the full search algorithm can be run independently on each alignment graph. Moreover, ad-hoc internal parallelization can be applied to improve efficiency. We are actually working on such optimized implementation of `DivAfull`.

Results show that the filtering procedure used for removing 'insignificant' results seems to be rather strict, because it appears to discard a substantial number of solutions which seem to be biologically meaningful. A more thorough analysis of real biological functions of some of the new discovered results is still needed.

Finally, we intend to analyze instances of our approach based on other methods, such as [7].

References

1. Bader, G.D., Donaldson, I., Wolting, C., Ouellette, B.F.F., Pawson, T., Hogue, C.W.V.: Bind–the biomolecular interaction network database. Nucleic Acids Res **29**(1) (January 1 2001) 242–245
2. Xenarios, I., Salwiński, Ł., Duan, X.J., Higney, P., Kim, S.M., Eisenberg, D.: Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. Nucleic Acids Research **30**(1) (January 1 2002) 303–305
3. Kelley, B.P., Sharan, R., Karp, R.M., Sittler, T., Root, D.E., Stockwell, B.R., Ideker, T.: Conserved pathways within bacteria and yeast as revealed by global protein network alignment. Proceedings of the National Academy of Science **100** (September 2003) 11394–11399
4. Sharan, R., Ideker, T.: Modeling cellular machinery through biological network comparison. Nature Biotechnology **24**(4) (April 2006) 427–433
5. Srinivasan, B.S., Shah, N.H., Flannick, J., Abeliuk, E., Novak, A., Batzoglou, S.: Current Progress in Network Research: toward Reference Networks for kKey Model Organisms. Brief. in Bioinformatics (2007) Advance access.
6. Koyutürk, M., Grama, A., Szpankowski, W.: Pairwise local alignment of protein interaction networks guided by models of evolution. In: RECOMB. Volume 3500 of Lecture Notes in Bioinformatics., Springer Berlin / Heidelberg (May 2005) 48–65
7. Sharan, R., Ideker, T., Kelley, B.P., Shamir, R., Karp, R.M.: Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. Journal of Computional Biology **12**(6) (2005) 835–846

8. Koyutürk, M., Kim, Y., Topkara, U., Subramaniam, S., Grama, A., Szpankowski, W.: Pairwise alignment of protein interaction networks. Journal of Computional Biology **13**(2) (2006) 182–199
9. Pržulj, N.: Knowledge Discovery in Proteomics: Graph Theory Analysis of Protein-Protein Interactions. CRC Press (2005)
10. Jancura, P., Heringa, J., Marchiori, E.: Dividing protein interaction networks by growing orthologous articulations. IR-BIO-002, Available at http://www.cs.vu.nl/~elena/diva.pdf (2007)
11. Flannick, J., Novak, A., Srinivasan, B.S., McAdams, H.H., Batzoglou, S.: Graemlin: General and robust alignment of multiple large interaction networks. Genome Res. **16**(9) (2006) 1169–1181
12. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks. In: RECOMB. (2007)
13. Pržulj, N., Wigle, D., Jurisica, I.: Functional topology in a network of protein interactions. Bioinformatics **20**(3) (2004) 340–384
14. Rathod, A.J., Fukami, C.: Mathematical properties of networks of protein interactions. CS374 Fall 2005 Lecture 9, Computer Science Department, Stanford University (2005)
15. Jeong, H., Mason, S.P., Barabasi, A.L., Oltvai, Z.N.: Lethality and centrality in protein networks. NATURE v **411** (2001) 41
16. Ekman, D., Light, S., Björklund, A.K., Elofsson, A.: What properties characterize the hub proteins of the protein-protein interaction network of saccharomyces cerevisiae? Genome Biology **7**(6) (2006) R45
17. Ucar, D., Asur, S., Catalyurek, U., Parthasarathy, S.: Improving functional modularity in protein-protein interactions graphs using hub-induced subgraphs. In: 10th European Conference on Principle and Practice of Knowledge Discovery in Database (PKDD), Berlin, Germany (September 18-22 2006)
18. Bader, G.D., Lăssig, M., Wagner, A.: Structure and evolution of protein interaction networks: a statistical model for link dynamics and gene duplications. BMC Evolutionary Biology **4**(51) (2004)
19. Li, X.L., Tan, S.H., Foo, C.S., Ng, S.K.: Interaction graph mining for protein complexes using local clique merging. Genome Informatics **16**(2) (2005) 260–269
20. Yook, S.H., Oltvai, Z.N., Barabăsi, A.L.: Functional and topological characterization of protein interaction networks. PROTEOMICS **4** (2004) 928–942
21. Abou-Rjeili, A., Karypis, G.: Multilevel algorithms for partitioning power-law graphs. In: 20th International Parallel and Distributed Processing Symposium (IPDPS). (2006)
22. Wolsey, L.A.: Integer Programming. 1 edn. Wiley-Interscience (September 9 1998)
23. Hirsh, E., Sharan, R.: Identification of conserved protein complexes based on a model of protein network evolution. Bioinformatics **23**(2) (2007) e170–176
24. Benjamini, Yoav, Hochberg, Yosef: Controlling the false discovery rate: A practical and powerful approach to multiple testing. Journal of the Royal Statistical Society. Series B (Methodological) **57**(1) (1995) 289–300