Dividing Protein Interaction Networks by Growing Orthologous Articulations

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

[1]Intelligent Systems, ICIS, Radboud Universiteit Nijmegen, The Netherlands
{jancura,elenam}@cs.ru.nl
[2]IBIVU, Vrije Universiteit Amsterdam, The Netherlands
{heringa}@few.vu.nl

Abstract. The increasing growth of data on protein-protein interaction (PPI) networks has boosted research on their comparative analysis. In particular, recent studies proposed models and algorithms for performing network alignment, the comparison of networks across species for discovering conserved modules. Common approaches for this task construct a merged representation of the considered networks, called alignment graph, and search the alignment graph for conserved networks of interest using greedy techniques. In this paper we propose a modular approach to this task. First, each network to be compared is divided into small subnets which are likely to contain conserved modules. To this aim, we develop an algorithm for dividing PPI networks that combines a graph theoretical property(articulation) with a biological one (orthology). Next, network alignment is performed on pairs of resulting subnets from different species. We tackle this task by means of a state-of-the-art alignment graph model for constructing alignment graphs, and an exact algorithm for searching in the alignment graph. Results of experiments show the ability of this approach to discover accurate conserved modules, and substantiate the importance of the notions of orthology and articulation for performing comparative network analysis in a modular fashion.

Key words: Protein network dividing, modular network alignment.

1 Introduction

With the exponential increase of data on protein interactions obtained from advanced technologies, data on thousands of interactions in human and most model species have become available (e.g. [1, 2]). PPI networks offer a powerful representation for better 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 to discover common protein groups which are likely to be related to shared relevant functional modules [3–5].

This problem is also known as *pairwise network alignment*. Algorithms for this task typically construct a merged graph representation of the networks to be compared, called alignment (or orthology) graph, and model the problem as an optimization problem on such graph. Due to the computational intractability of such optimization problem, greedy algorithms are commonly used [6–10].

1.1 Problem Statement

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, PPI networks are known to have a scale-free topology where most proteins participate in a small number of interaction while a few proteins, called *hubs*, contains a high number of interaction. As indicated by recent studies, hubs

⋆ Corresponding author

whose removal disconnects the PPI network (articulation hubs) are likely to appear in conserved interaction patterns [11, 12]. These observations motivate the focus of this paper on the problem of performing modular network alignment. Specifically, we propose a two phases approach for this task: divide and align. The divide phase transforms each PPI network into a set of small subnets which are expected to cover conserved complexes. The align phase uses an existing evolution-based alignment graph model to merge suitable pairs of subnets from each species, and an exact search technique for extracting conserved modules from each alignment graph.

1.2 Contributions

We introduce an heuristic algorithm for dividing a PPI network into subnets, which combines biological (orthology) and graph theoretical (articulation) information. The algorithm starts by identifying groups of orthologous articulations, called centers, which are expanded into subsets consisting of orthologous nodes.

The algorithm automatically determines the number of subsets and has the property of being parameterless.

We use this algorithm for performing network alignment, by merging pairs of resulting subnets from different species, and applying exact optimization for searching conserved modules across species. We introduce a new notion, *modular alignment*, because we align only particular PPI subnets achieving conserved modules inside of them while current methods of global or local network alignment try to align whole PPI networks.

In order to test the performance of this approach, we consider an instance of the method that uses a state-of-the-art evolution-based alignment graph model [6]. Results of experiments show effectiveness of the proposed approach, which is capable of detecting accurate conserved complexes. Furthermore, we show that improved performance can be achieved by merging modules detected with our algorithm with those identified by Koyuturk et al. algorithm [6]. In general, these results substantiate the important role of the notions of orthology and articulation in modular comparative PPI network analysis.

1.3 Related Work

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 ([6–10, 13]).

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 ([6–10]).

While the above algorithms focus on alignment of whole global networks, we focus on 'modular' network alignment. Modular network alignment is an alignment of particular subnets of given networks to be compared. To the best of our knowledge, we propose the first algorithm which directly tackles the modularity issue in network alignment in the meaning that dividing step achieves conserved modules inside of particular subnets and therefore one can perform only modular alignment for local network alignment problem.

Many papers have investigated the importance of hubs in PPI networks and functional groups [12, 14–18]. 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 [16]. 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 [12]. Computational techniques for identifying functional modules in PPI networks generally search for clusters of proteins forming dense components [19, 20]. The scale-free topology of PPI networks makes difficult to isolate modules hidden inside the central core [21]. In [22] 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 uses homology information and articulations for dividing PPI networks into subnets in order to perform network alignment in a modular fashion.

2 Graph Theoretic Background

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

A *tree* is a connected graph not containing any circle. A tree is called *rooted tree* if one vertex of the tree has been designated as the root. Given a rooted tree $T(V, F)$, the depth of a vertex $v \in V$ is the number of edges from the root to v without repetition of edges. Leaves of the tree T are vertices which have only one neighbor. The depth of a tree is the highest depth of its leaves. A spanning tree $T(V, F)$ of a connected undirected graph $G(U, E)$ is a tree where $V = U$ and $F \subseteq E$.

Given an edge-weighted (or node-weighted) graph $G(U, E)$ with a scoring function $w : e \in E \rightarrow \Re$ (or $w : u \in U \rightarrow \Re$). *Total weight* $w(G)$ of G is the sum of weights of all edges (or nodes) in the graph:

$$w(G) = \sum_{\forall e \in E} w(e) \quad (or \;\; w(G) = \sum_{\forall u \in U} w(u) \;\;).$$

Suppose a connected undirected graph $G(U, E)$ and a vertex $u \in U$ are given. Let $N(u)$ a set of all neighbors of u and $N'(u) \subseteq N(u)$ be. A *center* of u is the set $C(u) \equiv N'(u) \cup \{u\}$.

Observe that a center can be expanded to a spanning tree of $G(U, E)$. Moreover, the center as an initial set of expansion can be consider as a root if we merge all vertices of center to one node. Such spanning tree created from a center, called *centered tree*, has zero depth all vertices of center and the vertices of i- depth are new nodes added in ith iteration of expansion to the spanning tree. Therefore a centered tree , can be generated as follows:

- the 0-depth of the centered tree is the center
- the i-th depth of the centred tree consists of all neighbors of $(i-1)$-th depth which are not yet in any lower depth of the centered tree yet.

Examples of a spanning and centered tree are on Figure 1.

A PPI network is represented by an undirected graph $G(U, E)$. U denotes the set of proteins and E denotes set of edges, where an edge $uu' \in E$ represents the interaction between $u \in U$ and $u' \in U$. Given PPI networks $G(U, E)$ and $H(V, F)$. A *vertex* $u \in U$ *is orthologous* if there exists at least one vertex $v \in V$ such that uv is an orthologous pair. *Orthologous articulation* is an orthologous vertex which is an articulation. An *orthology path* is a path containing only orthologous vertices.

3 From Orthologous Articulations through Centers to Trees

Given a PPI network $G(U, E)$ and the set of vertices $O \subseteq U$, which are orthologous w.r.t. the vertices of the other PPI network to be compared with G. Let $n = |O|$. We generate centers from orthologous articulations, and expand them into centered subtrees containing only orthologous proteins. The resulting algorithm, called `Divide`, is sketched in pseudo-code in Algorithm 1, and described in more detail below.

Computing Articulations (Line 1). Computation of articulations can be performed in linear time by using, e.g., Tarjan's algorithm described in [23] or [24].

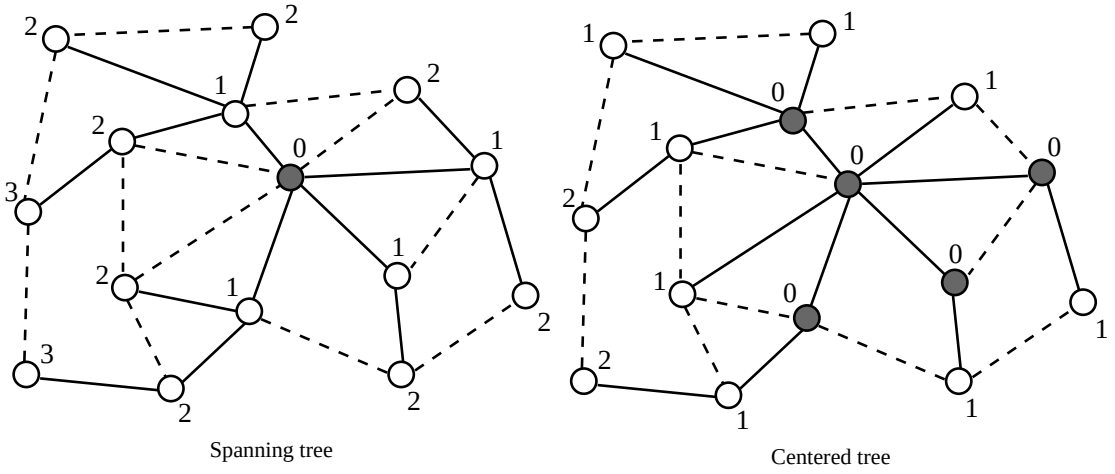

Fig. 1. Examples of spanning and centered tree in the same graph. The dark grey node in the left figure represents a root. Dark grey nodes in the right figure represent a center. Numbers indicate depths of nodes in trees. Solid edges are edges of a spanning tree. Dash edges are other edges of the graph.

Greedy Construction of Centers (Lines 3-10). The degree (in G) of all orthologous articulations is then used for selecting seeds for the construction of centers. Networks with scale-free topology appear to have edges between hubs systematically suppressed, while those between a hub and a low-connected protein seem favored [25]. Guided by this observation, we greedily construct centers by joining one orthologous articulation hub with its orthologous articulation neighbors, which will more likely have low degree.

Specifically, let A be the set of orthologous articulations of G. The first center consists of the element of A with highest degree and all its neighbors in A. The other centers are generated iteratively by considering, at each iteration, the element of A with highest degree among those which do not occur in any of the centers constructed so far, together with all its neighbors in A which do not already occur in any other center. The process terminates when all elements of A are in at least one center. Then an unambiguous label is assigned to each center.

Initial Expansion (Lines 11-16). By construction, centers cover all orthologous articulations. Articulation hubs are often present in conserved subnets detected by means of comparative methods such as [6]. Therefore, assuming that the majority of the remaining nodes belonging to conserved modules are neighbors of articulation hubs, we add to each center all its neighboring ortholog proteins, regardless whether they are or not articulations. We perform this step for all centers in parallel.

We mark these new added proteins with the label of the centers to which they have been added. These new added proteins form the first depth centered trees.

Observe that there may be a non-empty overlap between first depth centered trees (as illustrated in the right part of Figure 2).

Parallel Expanding of Trees (Lines 17-27) Successive depths of trees are generated by expanding all nodes with only one label which occur in the last depth of each (actual) centered tree. We add to the corresponding trees all orthologous neighbors of these nodes which are not yet labelled. Then we assign to the newly added nodes the labels of the centered trees they belong to. This process is repeated until it is impossible to add unlabeled orthologous proteins to at least one centered tree.

Observe that each iteration yields to possible overlap between newly created depths (see the left part of Figure 3).

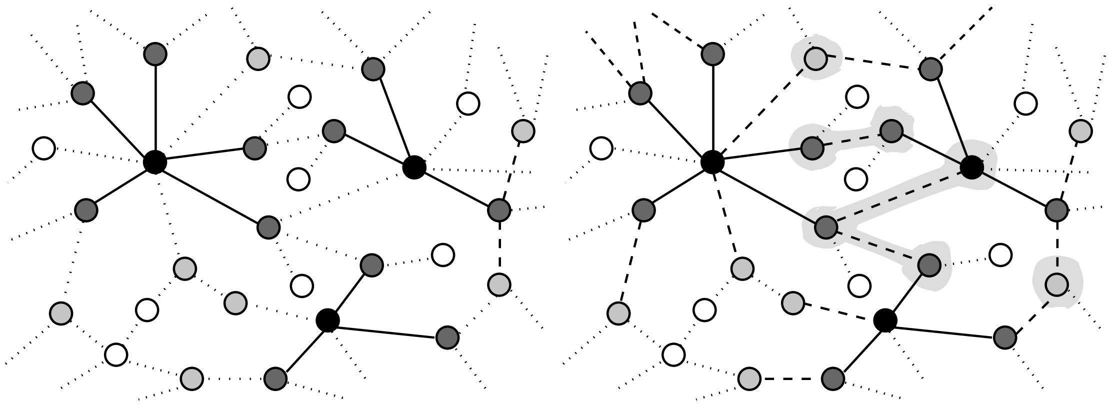

Fig. 2. Examples of centers of centered trees (left figure) and of their initial expansion (right figure). Seeds of centers are solid nodes. Dark grey nodes are the rest of centers connected to a seed by solid edges. Light grey nodes are orthologous proteins which are not articulations. Empty nodes are non-orthologous proteins. Dot edges are the rest of edges in the graph. In the second (right) graph dash edges indicate the expansion and connect nodes of centers (zero depth centered trees) with nodes of the first depth centered trees. Nodes on the grey background indicate the overlap among centered trees.

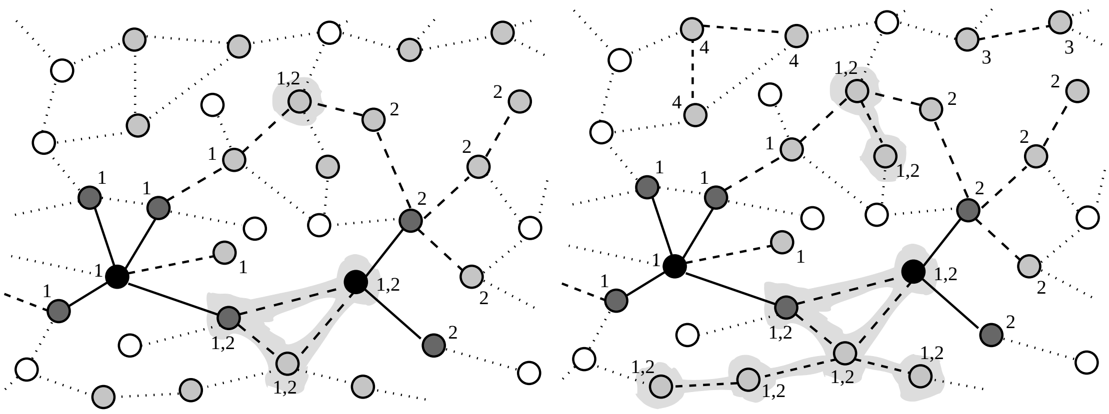

Fig. 3. Examples of parallel expansion of trees (left figure) and of the final assigning remaining nodes (right figure). Seeds of centers are solid nodes. Dark grey nodes are the rest of centers connected to a seed by solid edges. Light grey nodes are orthologous proteins which are not articulations. Empty nodes are non-orthologous proteins. Dash edges indicate the process of expansion. Dot edges are the rest of edges in the graph. Nodes on the grey background create the overlap. Numbers are labels of trees assigned to nodes during expansion.

Assigning Remaining Nodes to Trees (Lines 28-42). The remaining orthologous nodes, that is, those not yet labelled, are processed as follows. First, unlabeled nodes which are neighbors of multi-labelled nodes are added to the corresponding centered trees. Then the newly added nodes are marked with these labels. This process is iterated until there are no unlabeled neighbors of multi-labelled nodes.

Nodes which are not neighbors of any labelled protein are still unlabeled. We assume that they may possibly be part of conserved complexes which do not contain articulations. Hence we create new subtrees by joining together all unlabeled orthologous neighbor proteins.

An example of these final steps is shown on the right part of Figure 3.

Complexity. The algorithm divides only orthologs of a given PPI network where the number of all orthologs is $n = |O|$. It performs a parallel breadth-first search (BFS). It general, BFS has $O(|V| + |E|)$ complexity, where V and E denote the number of nodes and edges, respectively. However, `Divide` constructs trees considering only orthologous nodes, so the number of edges, which are traversed, is $|O'| - 1$, where $|O'|$ is the number of orthologs vertices of the constructed subtree. The possible overlap between trees can increase the number of traversed edges and visited vertices. In the worse case all orthologous vertices are visited by each center (all nodes are in the overlap). So, if the number of centers is k, the complexity of `Divide` is $O(kn)$.

4 Divide and Align Algorithm

The Divide algorithm divides orthologous proteins of the PPI network into overlapping subtrees. We separately apply this algorithm to each of the two PPI networks from the distinct species to be compared. Nodes of each constructed subtree induce a PPI subnetwork. Pairs of such induced subnetworks from different species are merged into small orthology graphs if at least two orthologous pairs exist between proteins of those subnetworks.

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.

In our experiments we use the evolution-based alignment graph model introduced in [6]. In that model, a weighted alignment graph is constructed from a pair of PPI networks and a similarity score S, which quantifies the likelihood that two proteins are orthologous. A node in the alignment graph is a pair of ortholog proteins. Each edge in the alignment graph is assigned a weight that is the sum of three scoring terms: for protein duplication, mismatches for possible divergence in function, and match of a conserved pair of orthologous interactions. We refer to [6] for a detailed description of these terms. Induced subgraphs of the resulting weighted alignment graph with total weight greater than a given threshold are considered as relevant *alignments*. This problem is reduced to the optimization problem of finding a maximal induced subgraph. In [6], an approximation greedy algorithm based on local search is used because the maximum induced subgraph problem is NP-complete. This greedy algorithm selects at first one seed which can likely contribute at most to the overall weight of a potential subgraph. Such seed is expanded by adding (removing) nodes to (from) the subgraph while the actual subgraph weight increases.

In this study, after the diving step and aligning possible pairs of PPI subnetworks a set of small alignment graphs is produced. We use exact optimization [26] for searching in those graphs. We call the resulting algorithm `DivA` (Divide and Align).

Finally, redundant alignments are filtered out as done, e.g., in [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.

Algorithm 1 Divide algorithm

Input: G: PPI network, O: orthologous nodes of G
Output: S: list of subsets of O

$A =$ { orthologous articulations of G}
$S =<>$
repeat {Construction of centers}
  $root =$ element of A with highest degree not already occurring in S
  $s = \{root\} \cup$ { neighbors of $root$ in A not already occurring in S}
  $S =< s, S >$
until all members of A occur in S
$d = 0$
Assign depth d to all elements of S
Assign label l_s to each s in S and to all its elements
for s in S **do**
  $s = s \cup$ { all neighbors of s in O}
  Assign label l_s to all neighbors of s in O
end for
$d = 1$
Assign depth d to all elements of S having yet no depth assigned
repeat {Expand one depth centered trees from nodes with one label}
  $N =$ { unlabeled neighbors in O of elements in s of depth d having only one label }
  for n in N **do**
    Assign to n all labels of its neighbors of depth d having only one label
    for $l_s \in n$ **do**
      $s = s \cup \{n\}$
    end for
  end for
  $d = d + 1$
  Assign depth d to all elements of S having yet no depth assigned
until S does not change
repeat {Expand centered trees from nodes multiple labels}
  $R =$ { unlabeled proteins in O with at least one multi-labelled protein as neighbor }
  for r in R **do**
    Assign to r all labels of its neighbors
    for $l_s \in r$ **do**
      $s = s \cup \{r\}$
    end for
  end for
until S does not change
repeat
  choose an unlabeled element u of O
  $t = \{u\} \cup$ {all elements of O which can be reached alongside an orthology path from u}
  Assign label l_t to t and to all its elements
  $S =< t, S >$
until O does not contain any unlabeled node

5 Experimental Results

In order to assess the performance of our approach, we use the state-of-the-art framework for comparative network analysis proposed in [6], called `MaWish`. The two following PPI networks, already compared in [6], 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 and paralogous pairs, which are derived using BLAST E-values (for more details see [11]). 2746 potential orthologous pairs created by 792 proteins in S. cerevisiae and 633 proteins in C. elegans are identified.

5.1 Divide phase

Results of application of the Divide algorithm to these networks are summarized as follows.

For *Saccharomyces cerevisiae*, 697 articulations, of which 151 orthologs, and 83 centers are identified. Expansion of these centers into centered trees results in 639 covered orthologs. The algorithm assigns the remaining 153 orthologous proteins to 152 new subtrees.

For *Caenorhabditis elegans*, 586 articulations, of which 158 orthologs, are computed, and 112 centers are constructed from them. Expansion of these centers into centered trees results in 339 covered orthologs. The algorithm assigns the remaining orthologous 294 proteins to 288 new subtrees.

We observe that the last remaining orthologs assigned to subtrees are 'isolated' nodes, in the sense that they are rather distant from each other and not reachable from ortholog paths stemming from centers.

The divide part of algorithm run only less than half of a second on a desktop machine (AMD Athlon 64 Processor 3500+, 2 GB RAM) in practical.

5.2 Alignment phase

We obtain 235 subtrees for *Saccharomyces cerevisiae* and 400 subtrees of *Caenorhabditis elegans*. Nodes of each such tree induce a PPI subnetwork. By constructing alignment graphs between each two PPI subnetworks containing more than one ortholog pair, we obtain 884 alignment graphs, where the biggest one consists of only 31 nodes. For each of such alignment graphs, the maximum weighted induced subgraph is computed by exact optimization. Zero weight threshold is used for considering an induced subgraph a legal alignment. Redundant graphs are filtered using $r = 80\%$ as the threshold for redundancy. In this way `DivA` discovers 72 alignments.

The computation of induced subgraphs by an exact search took a few minutes compared to around a second in `MaWish` on a desktop machine (AMD Athlon 64 Processor 3500+, 2 GB RAM).

5.3 Comparison between `DivA` and `MaWish`

We performed network alignment with `MaWish` using parameter values as reported in [11]. The algorithm discovered 83 conserved subnets.

A *paired redundant alignment* is a pair (G_1, G_2) of alignments, with G_1 discovered by `DivA` 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 total weight of G_1 is bigger than the total weight of G_2.

`DivA` finds 14 new alignments not detected by `MaWish`. Figure 5 shows the best new alignment found by `DivA` (left) and the alignment of `DivA` which best refines an alignment of `MaWish`.

There are 58 paired redundant alignments, whose total weights are plotted in the left part of Figure 4. Among these, 40 (55.6%) are equal (crosses in the diagonal), and 18 (25%) different. 5

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

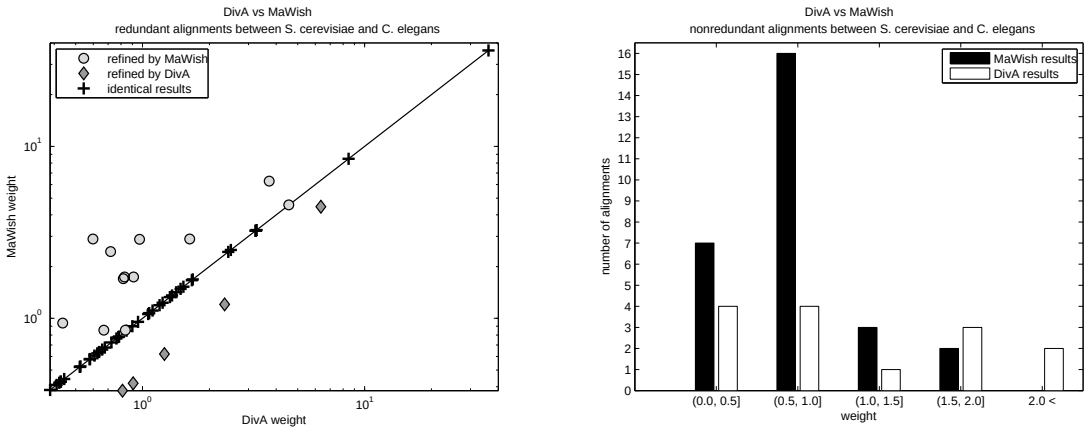

Fig. 4. Left figure: Distribution of pairs of weights of paired redundant alignments, one obtained from `MaWish` and one from `DivA`. Weights of alignments found by `DivA` are on the *x-axis*, those found by `MaWish` on the *y-axis*. Right figure: Interval weight distributions of non-redundant alignments discovered by `MaWish` (solid bars) and `DivA` (empty bars). The x-axis show weight intervals, the *y-axis* the number of alignments in each interval.

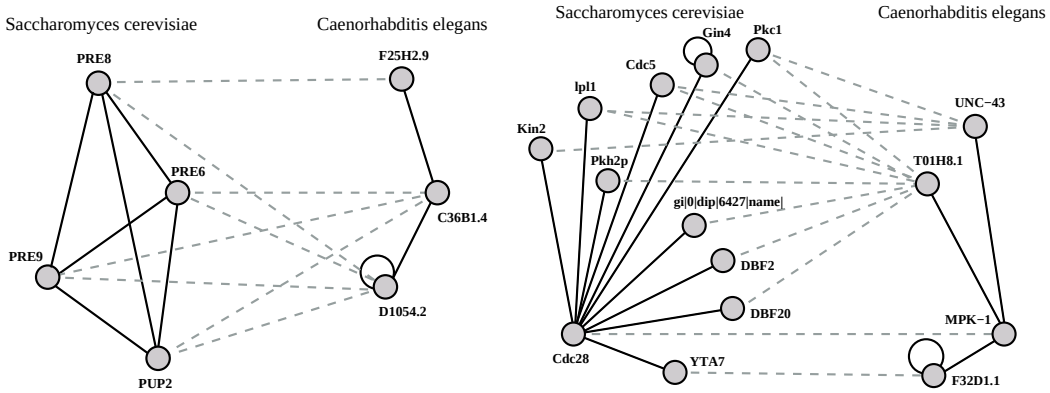

Fig. 5. Left: The best new alignment. Dash lines mark orthologous pairs. Solid line is protein-protein interaction. Right: The refined alignment with the greatest weight. Dash lines mark orthologous pairs. Solid line is protein-protein interaction. A loop on a protein means duplication.

(6.9%) (diamonds below the diagonal) with better `DivA` alignment weight, and 12 (16.7%) (circles above the diagonal) with better `MaWish` alignment weight (for 1 pair it is undecidable because of rounding errors during computation).

The right plot of Figure 4 shows the binned distribution of total weights of the 14 (19.4%) found by `DivA` but not `MaWish`, and 28 found by `MaWish` and not by `DivA`. The overall weight average of the `DivA` ones (1.197) is greater than the overall average of the `MaWish` ones (0.7501), indicating the ability of `DivA` to find high score subnets, possibly due to the exact search strategy used.

Of the 14 new alignments detected by `DivA`, 8 of them have a intersection with a true MIPS complex (cf. Table 1). Three of these alignments (6., 12. and 14.) have equal (sub)module in their true S. cerevisiae complex.

Table 1. HG= hypergeometric, Size = number of alignment nodes of an alignment, N = number of proteins of alignment nodes which are annotated in the best (according to hypergeometric score) true S. cerevisiae's MIPS complex of the alignment. M = number of proteins of alignment nodes in S. cerevisiae. Intersection = $|N|/|M|$

Align.	Score	Size	$\|M\|$	MIPS category	Intersection	$-log(HG)$
1.	4.28	8	4	20S proteasome	100(%)	7.25
4.	1.65	5	2	19/22S regulator	100(%)	3.45
6.	1.41	5	2	19/22S regulator	50(%)	1.71
7.	0.62	2	2	20S proteasome	100(%)	3.56
8.	0.61	2	2	Replication fork complexes	100(%)	3.22
9.	0.53	2	2	19/22S regulator	100(%)	3.45
12.	0.43	2	2	19/22S regulator	50(%)	1.71
14.	0.39	2	2	19/22S regulator	50(%)	1.71

From the refined alignments, three of them have intersection with a true MIPS complex.

Table 2. True complexes associated to `MaWish` refined alignments.

Align.	Score	Size	$\|M\|$	MIPS category	Intersection	$-log(HG)$
1.	4.46	10	10	Cdc28p complexes	10(%)	1.47
2.	0.62	2	2	Casein kinase II	100(%)	4.81
3.	0.38	2	2	SNF1 complex	50(%)	2.16

Table 3. True complexes associated to `DivA` refined alignments.

Align.	Score	Size	$\|M\|$	MIPS category	Intersection	$-log(HG)$
1.	6.35	15	11	Cdc28p complexes	9(%)	1.47
2.	1.26	4	4	Casein kinase II	100(%)	10.39
3.	0.81	3	2	SNF1 complex	50(%)	2.16

Note that alignments 1. and 3. in both Table 2 and 3 have equal hypergeometric score, showing that the coverage, that is, number of proteins of an alignment contained in its best true MIPS module, does not change. Alignment 2. in Table 2 covers 50% of the true complex, while its refinement in Table 3 covers the entire true complex (Casein kinase II, consisting of 4 proteins).

Three of these alignments have equal (sub)module in their true S. cerevisiae complex.

By considering the union of all alignments of `MaWish` and `DivA` and by filtering out the redundant ones, 97 alignments are obtained, from which 26% consist of new or refined `DivA` ones. In particular, conserved modules of three new true MIPS classes are detected: replication fork complexes, mRNA splicing, SCF-MET30 complex. Moreover, the alignment by `MaWish` which covers 50% of the true complex Casein kinase II (this complex consists of 4 proteins) is refined by `DivA` in such a way that the entire true complex is covered (all four proteins).

In this experiment we searched for the best solution in each orthology graph only. A full-search, where all possible solutions are found for each orthology graph, has been used in [27]. This yielded to a considerable increase of the number of results. Statistical evaluation of those results indicated their biological relevance. In general, the results show that `DivA` can be successfully applied to 'refine' state-of-the-art algorithms for network alignment.

6 Conclusion

The comparative experimental analysis with `MaWish` indicates that `DivA` is able to discover new alignments which seem to be on average more conserved because of higher weight than those discovered by `MaWish` but not by `DivA`. Improved performance is shown to be achieved by combining results of `MaWish` and `DivA`, yielding new and refined alignments.

The selection of centers is biased on the orthology information but it can be changed for another property. Hence the divide algorithm can be applied to perform modular network alignment of other type of networks.

Finally, we considered here an instance of our approach based on the evolution-based alignment graph model by Koyuturk et al. [11]. We intend to analyze instances of our approach based on other methods, such as [7].

Acknowledgments

We would like to thank Mehmet Koyuturk for discussion on the `MaWish` code.

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., Salwínski, 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. Hirsh, E., Sharan, R.: Identification of conserved protein complexes based on a model of protein network evolution. Bioinformatics **23**(2) (2007) e170–176
9. Sharan, R., Suthram, S., Kelley, R.M., Kuhn, T., McCuine, S., Uetz, P., Sittler, T., Karp, R.M., Ideker, T.: From the Cover: Conserved patterns of protein interaction in multiple species. Proceedings of the National Academy of Sciences **102**(6) (2005) 1974–1979

10. 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
11. 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
12. Pržulj, N.: Knowledge Discovery in Proteomics: Graph Theory Analysis of Protein-Protein Interactions. CRC Press (2005)
13. Singh, R., Xu, J., Berger, B.: Pairwise global alignment of protein interaction networks by matching neighborhood topology. (2007) 16–31
14. Pržulj, N., Wigle, D., Jurisica, I.: Functional topology in a network of protein interactions. Bioinformatics **20**(3) (2004) 340–384
15. Rathod, A.J., Fukami, C.: Mathematical properties of networks of protein interactions. CS374 Fall 2005 Lecture 9, Computer Science Department, Stanford University (2005)
16. Jeong, H., Mason, S.P., Barabasi, A.L., Oltvai, Z.N.: Lethality and centrality in protein networks. NATURE v **411** (2001) 41
17. 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
18. 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)
19. Bader, G.D., Lssig, 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)
20. 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
21. Yook, S.H., Oltvai, Z.N., Barabsi, A.L.: Functional and topological characterization of protein interaction networks. PROTEOMICS **4** (2004) 928–942
22. Abou-Rjeili, A., Karypis, G.: Multilevel algorithms for partitioning power-law graphs. In: 20th International Parallel and Distributed Processing Symposium (IPDPS). (2006)
23. Tarjan, R.: Depth-first search and linear graph algorithms. SIAM Journal on Computing **1**(2) (1972) 146–160
24. Hopcroft, J., Tarjan, R.: Algorithm 447: efficient algorithms for graph manipulation. Commun. ACM **16**(6) (1973) 372–378
25. Maslov, S., Sneppen, K.: Specificity and stability in topology of protein networks. Science **296** (2002) 910–913
26. Wolsey, L.A.: Integer Programming. 1 edn. Wiley-Interscience (September 9 1998)
27. Jancura, P., Heringa, J., Marchiori, E.: Divide, align and full-search for discovering conserved protein complexes. In Marchiori, E., Moore, J.H., eds.: EvoBIO. Volume 4973 of Lecture Notes in Computer Science., Springer (2008) 71–82