Dividing Protein Interaction Networks for Modular Network Comparative Analysis*

Pavol Jancura*, Elena Marchiori*

*Institute for Computing and Information Sciences, Radboud University Nijmegen, Postbus 9010, 6500 GL Nijmegen, The Netherlands

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, that is, the comparison of networks across species for discovering conserved functional complexes. In this paper, we present an algorithm for dividing PPI networks, prior to their alignment, into small sub-graphs that are likely to cover conserved complexes. This allows one to perform network alignment in a modular fashion, by acting on pairs of resulting small sub-graphs from different species. The proposed dividing algorithm combines a graph theoretical property (articulation) with a biological one (orthology). Extensive experiments on various PPI networks are conducted in order to assess how well the sub-graphs generated by this dividing algorithm cover protein functional complexes and whether the proposed pre-processing step can be used for enhancing the performance of network alignment algorithms. Source code of the dividing algorithm is available upon request for academic use.

Key words: protein interaction network division, 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. Bader et al., 2001; Xenarios et al., 2002). 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.

*This manuscript is an extended version of the conference paper Jancura et al. (2008b) presented at the Third IAPR International Conference on Pattern Recognition in Bioinformatics, Melbourne, Australia, 2008.

*Corresponding author. Tel.: +31 24 365 26 34; fax: +31 24 365 27 28.

Email addresses: jancura@cs.ru.nl (Pavol Jancura), elenam@cs.ru.nl (Elena Marchiori)
Recent studies consider a comparative approach for the analysis of PPI networks from different species in order to discover common protein groups, called conserved complexes, which are likely to be related and to share similar functionality in a cell (Sharan and Ideker, 2006; Srinivasan et al., 2007). This problem is known as protein network alignment. Algorithms for this task typically model this problem by means of a merged graph representation of the networks to be compared, called alignment (or orthology) graph, and then formalize the problem of searching (merged) conserved complexes in the alignment graph as an optimization problem. Due to the computational intractability of the resulting optimization problem, greedy algorithms are commonly used.

One can identify two main network alignment categories. Local network alignment, that identifies the best local mapping for each local region of similarity between input networks, and global network alignment, that searches for the best single mapping across all parts of the input networks, even if it is locally sub-optimal in some regions of the networks. If a method aligns networks of just two species, it is called pairwise network alignment, while if it can handle more than two networks, it is called multiple network alignment.

Many methods for network alignment have been proposed. We describe them briefly in the next section on related work.

The aim of this paper is not to propose yet another network alignment algorithm, but to show how PPI networks can be divided, prior to their alignment, into small sub-graphs that are likely to cover conserved complexes.

Conserved complexes discovered by computational techniques have in general small size (that is, number of proteins) 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 interactions while a few proteins, called hubs, contain a high number of interactions. As indicated by a recent study, hubs whose removal disconnects a PPI network (articulation hubs) are likely to appear in conserved interaction patterns (Pržulj, 2005).

These observations motivate the introduction of an algorithm for dividing PPI networks, called Divide, that combines biological (orthology) and graph theoretical (articulation) information: it detects small groups of ortholog articulations, called centers, which are then expanded into subsets of ortholog nodes. This algorithm has the desirable property of being parameterless.

The effectiveness and robustness of Divide is assessed experimentally in the following three ways.

First, we show that the sub-graphs generated by Divide indeed cover "true" conserved pro-
tein complexes. This is done by measuring the overlap of these sub-graphs with MIPS curated
functional complexes restricted to those proteins belonging to an orthologous pair.

Next, we show that the generated sub-graphs cover protein complexes computationally pre-
dicted. Specifically, we compare these sub-graphs with the conserved complexes predicted by
one state-of-the-art pairwise local alignment algorithm, called MaWish (Koyutürk et al., 2006b).
We investigate experimentally how Divide biases the search process of MaWish, and whether the
generated sub-graphs contain information to be used for discovering new conserved complexes.
Results of an extensive experimental analysis indicate that indeed Divide generates sub-graphs
containing conserved complexes that are not detected by MaWish.

Finally, we consider two case studies of modular network alignment. In the first case study,
Divide is used to generate sub-graphs, which are then pairwise merged using the networks merging
model of MaWish. We apply iterative exact search to the resulting alignment graphs. Results
of experiments show ability to detect a high number of accurate conserved complexes. In the
second case study, Divide is used for enhancing an existing method for discovering conserved
functional complexes, called MNAligner (Li et al., 2007). MNAligner consists of two main steps:
first, candidate functional complexes within one species are detected using a clustering algorithm
(MCODE); next, an exact optimization algorithm is applied for matching the resulting candidate
functional complexes with sub-graphs of the other species in order to extract conserved complexes.
Results of experiments show that by applying Divide to orthologs nodes prior to clustering
enhances the performance of this algorithm.

To the best of our knowledge, we propose the first algorithm which directly tackles the modular-
ity issue in network alignment by showing that Divide generates sub-graphs that cover conserved
complexes and can be used for performing modular pairwise network alignment.

In general, these results substantiate the important role of the notions of orthology and artic-
ulation in modular comparative PPI network analysis.

This paper contains and extends material from two previous conferences’ papers of Jancura
et al. (2008a,b). It is organized as follows. In the next section we discuss related works. Section
3 describes the graph-theoretic terminology used in the paper. The Divide algorithm is intro-
duced in Section 4. Section 5 summarizes the data and the type of assessment employed in the
experimental analysis. In Section 6 the robustness of Divide is assessed by analysing how the
generated sub-graphs cover "true" complexes. In Section 7 the sub-graphs generated by Divide
are compared with the complexes predicted by MaWish. In Section 8 modular network alignment
is performed on the two case studies above described. Finally, we conclude and briefly address future work in Section 9.

2. Related Work

Recent overviews of approaches and issues in comparative biological networks analysis have been presented by Sharan and Ideker (2006) and Srinivasan et al. (2007) since the first formulation of network alignment introduced by Kelley et al. (2003).

In general, network alignment methods have been proposed for discovering conserved metabolic pathways, conserved functional complexes, and for detecting functional orthologs. For instance, in Kelley et al. (2003) introduced an approach for detecting conserved metabolic pathways between two species. A local protein network alignment method based on this approach was proposed to discover conserved complexes (Sharan et al., 2004, 2005a). This method was further extended to the alignment of multiple species by Sharan et al. (2005b). Moreover, the approach of Kelley et al. (2003) motivated Bandyopadhyay et al. (2006) to develop a method for identifying functional orthologs.

Other alignment techniques for discovering conserved pathways based on the species conservation were proposed (e.g., Shlomi et al., 2006; Qian et al., 2009; Pinter et al., 2005; Cheng et al., 2008; Li et al., 2007; Koyutürk et al., 2006a), as well as methods handling network structures more general than single pathways (Wernicke and Rasche, 2007; Yang and Sze, 2007; Dost et al., 2008; Blin et al., 2009; Bruckner et al., 2009).

The main goal of local protein network alignment is to detect conserved protein complexes across species, by searching for local regions of input networks having both high topological similarity between the regions and high sequence similarity between proteins of these regions. Many pairwise local network alignment techniques have been introduced in recent years (see, e.g. Sharan et al., 2005a; Hirsh and Sharan, 2007; Liang et al., 2006; Koyutürk et al., 2005; Narayanan and Karp, 2007; Tian and Samatova, 2009). In particular, Berg and Lässig (2006) introduced an alignment framework based on Bayesian theory. Other approaches embed additional information into the local protein network alignment task (Guo and Hartemink, 2009; Ali and Deane, 2009).

A first attempt to perform multiple network alignment using three species was done by Sharan et al. (2005b). However, the method scales exponentially with the number of input species. Thus, new methods for aligning multiple species have been proposed (Flannick et al., 2006, 2009; Dutkowski and Tiuryn, 2007; Kalaev et al., 2009).
The main goal of global protein network alignment is functional orthologs detection, because, in contrast to local network alignment, each node in an input network is either matched to one node in the other network or has no match in the other network (Singh et al., 2007). Of course, global protein network alignment can be also used for identifying conserved complexes.

The first systematic identification of functional orthologs based on protein network comparison was done by Bandyopadhyay et al. (2006). Singh et al. (2008b) explicitly used global multiple network alignment for detecting functional orthologs.

The first formal global network alignment method was introduced by Singh et al. (2007). This method has been followed by more works on global pairwise network alignment (Evans et al., 2008; Zaslavskiy et al., 2009; Klau, 2009; Chindelevitch et al., 2010). Singh et al. (2008a); Liao et al. (2009); Flannick et al. (2009) tackled global alignment of multiple species.

While the above works focus on alignment of networks, we deal with protein networks preprocessing prior to their alignment, in order to perform modular network alignment (Jancura et al., 2008).

We turn now to the description of works related to the main graph topological ingredient used in our method: hub articulation. Many papers have investigated the importance of hubs in PPI networks and functional groups (Ekman et al., 2006; Jeong et al., 2001; Pržulj, 2005; Pržulj et al., 2004; Rathod and Fukami, 2005; Ucar et al., 2006). In particular, it has been shown by Jeong et al. (2001) 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. Moreover, if one takes functional groups in PPI networks, then, amongst all functional groups, cellular organization proteins have the largest presence in those hubs whose removal disconnects the network (Pržulj, 2005). These works justify the use of articulation hubs for dividing PPI networks prior to their alignment.

3. 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 \( u \).

A graph \( G = (U, E) \) is called undirected if \( uu' \) in \( E \) implies \( u'u \) also in \( E \); otherwise \( G \) is called directed. A directed acyclic graph is a directed graph that contains no cycles.

A sub-graph \( H(V, F) \) of an undirected graph \( G(U, E) \) is said to be induced by the set of nodes \( V \subset U \) if and only if the set of edges \( F \subset E \) consists of all the edges that appear in \( G \) over the same vertex set \( V \).
A graph is connected if there is a path from any node to any other node. 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 $w : e \in E \rightarrow \mathbb{R}$ (or $w : u \in U \rightarrow \mathbb{R}$). Total weight $w(G)$ of $G$ is the sum of weights of all edges (or nodes) in the graph:

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

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 $C(u) = N'(u) \cup \{u\}$.

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 $i$-th depth are new nodes added in $i$th 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 centered 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 shown in 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.
4. Divide Algorithm

Suppose given the PPI networks $G$ and $G_1$ of two species. Let $G(U, E)$ and $O \subseteq U$ be the set of vertices which are orthologous w.r.t. the vertices of $G_1$. Suppose $O$ contains $n$ elements. The divide algorithm is shown in pseudo-code in Algorithm 1. It generates centers from orthologous articulations and expands them into centered sub-trees containing only orthologous proteins. The main steps of divide are described in detail below.

**Computing Articulations** (Line 1). Computation of articulations can be performed in linear time by using, e.g., Tarjan’s algorithm described in Tarjan (1972) or Hopcroft and Tarjan (1973).

**Greedy Construction of Centers** (Lines 3-10). The degree (in $G$) of all orthologous articulations is 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 favoured (Maslov and Sneppen, 2002). 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 $A$.

Figure 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.
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.

![Figure 2: Examples of centers of centered trees (left figure) and of their initial expansion (right figure). Seeds of centers are solid nodes. Dark gray nodes are the rest of centers connected to a seed by solid edges. Light gray 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 gray background indicate the overlap among centered trees.]

**Initial Expansion** (Lines 11-16). By construction, centers cover all orthologous articulations. Articulation hubs are often present in conserved sub-graphs detected by means of comparative methods. Therefore, assuming that the majority of the remaining nodes belonging to conserved complexes are neighbors of articulation hubs, we add to each center all its neighbouring orthologous 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 labeled. 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 unlabelled 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).

Assigning Remaining Nodes to Trees (Lines 28-42). The remaining orthologous nodes, that is, those not yet labeled, are processed as follows. First, unlabelled nodes which are neighbors of multi-labeled 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 unlabelled neighbors of multi-labeled nodes.

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

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

In the end, the algorithm produces the list of subsets of orthologous nodes, where each subset of nodes corresponds to the nodes of one particular tree constructed by the algorithm. The subsets generate induced sub-graphs of the divided PPI network.

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). In general, BFS has \( O(|V| + |E|) \) complexity, where \( V \) and \( E \) denote the number of nodes and edges, respectively. However, the Divide algorithm constructs trees considering only orthologous nodes, so the number of edges, which are traversed, is \( |O'| - 1 \), where \( |O'| \) is the number of orthologous vertices of the
Algorithm 1 Divide algorithm

Input: $G, G_i$: PPI networks, $O$: orthologous nodes of $G$ with respect to $G_i$

Output: $S$: list of subsets of $O$

1: $A = \{\text{orthologous articulations of } G\}$
2: $S = \langle \rangle$
3: repeat {Construction of centers}
4: \text{root} = \text{element of } A \text{ with highest degree not already occurring in } S$
5: \text{s} = \{\text{root}\} \cup \{\text{neighbors of } \text{root} \text{ in } A \text{ not already occurring in } S\}$
6: $S = \langle s, S \rangle$
7: until all members of $A$ occur in $S$
8: $d = 0$
9: Assign depth $d$ to all elements of $S$
10: Assign label $l_s$ to each $s$ in $S$ and to all its elements
11: for $s$ in $S$ do
12: \text{s} = \text{s} \cup \{\text{all neighbors of } s \text{ in } O\}$
13: Assign label $l_s$ to all neighbors of $s$ in $O$
14: end for
15: $d = 1$
16: Assign depth $d$ to all elements of $S$ having yet no depth assigned
17: repeat {Expand one depth centered trees from nodes with one label}
18: $N = \{\text{unlabelled neighbors in } O \text{ of elements in } s \text{ of depth } d \text{ having only one label } \}$
19: for $n$ in $N$ do
20: Assign to $n$ all labels of its neighbors of depth $d$ having only one label
21: for $l_s \in n$ do
22: \text{s} = \text{s} \cup \{n\}$
23: end for
24: end for
25: $d = d + 1$
26: Assign depth $d$ to all elements of $S$ having yet no depth assigned
27: until $S$ does not change
28: repeat {Expand one depth centered trees from nodes with multiple labels}
29: $R = \{\text{unlabelled proteins in } O \text{ with at least one multi-labeled protein as neighbor } \}$
30: for $r$ in $R$ do
31: Assign to $r$ all labels of its neighbors
32: for $l_s \in r$ do
33: \text{s} = \text{s} \cup \{r\}$
34: end for
35: end for
36: until $S$ does not change
37: repeat {Assign labels to remaining elements}
38: choose an unlabelled element $u$ of $O$
39: $t = \{u\} \cup \{\text{all elements of } O \text{ which can be reached alongside an orthology path from } u\}$
40: Assign label $l_t$ to $t$ and to all its elements
41: $S = \langle t, S \rangle$
42: until $O$ does not contain any unlabelled node
constructed sub-tree. 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 \texttt{Divide} is \( O(kn) \).

5. Experimental Analysis

The effectiveness and robustness of the proposed pre-processing method is assessed experimentally in the following three ways.

First, we show that the sub-graphs generated by \texttt{Divide} indeed cover "true" conserved protein complexes. This is done by measuring the overlap of the generated sub-graphs with yeast MIPS curated functional complexes restricted to those proteins belonging to an orthologous pair.

Next, we show that the resulting sub-graphs cover protein complexes computationally predicted by one state-of-the-art alignment algorithm (Koyutürk et al., 2006b), \texttt{MaWish}, in order to investigate whether the sub-graphs contain information that could be used to discover new conserved complexes.

Finally, we consider two instances of modular network alignment. In the first instance, \texttt{Divide} is used to generate sub-graphs which are pairwise merged using the \texttt{MaWish} network alignment model. Then iterative exact search is applied to detect conserved complexes in the resulting alignment graphs. In the second instance of modular network alignment, \texttt{Divide} is used to generate sub-graphs, which are then used by the recent network alignment algorithm \texttt{MNAligner}, for discovering conserved functional complexes.

We conduct experiments on the following pairs of organisms:

- \textit{Saccharomyces cerevisiae} versus \textit{Caenorhabditis elegans} (yeast-nematode),
- \textit{Saccharomyces cerevisiae} versus \textit{Drosophila melanogaster} (yeast-fly) and
- \textit{Saccharomyces cerevisiae} versus \textit{Homo sapiens} (yeast-human).

Publicly available data were used, available at the web-page of \texttt{MaWish}\(^1\). These data consist of protein interactions obtained from the BIND (Bader et al., 2001) and DIP (Xenarios et al., 2002) molecular interaction databases, and the list of potential orthologous and paralogous pairs, which are derived using BLAST \( E \)-values (for more details see Koyutürk et al., 2006b). Table 1 and Table 2 report the number of interactions and proteins in the considered species, and the number of potential orthologous pairs between species considered in the alignment task, respectively.

\(^1\)http://vorlon.case.edu/~mxk331/software/
Table 1: Protein interaction network properties of yeast, nematode, fly and human.

<table>
<thead>
<tr>
<th>Pair of species</th>
<th>#orthologous pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>S. cerevisiae vs C. elegans</td>
<td>2746</td>
</tr>
<tr>
<td>S. cerevisiae vs D. melanogaster</td>
<td>15884</td>
</tr>
<tr>
<td>S. cerevisiae vs H. sapiens</td>
<td>6690</td>
</tr>
</tbody>
</table>

Table 2: Number of potential orthologous pair for considered species: yeast-nematode, yeast-fly and yeast-human.

5.1. Handling redundant alignments and complexes

A general issue in network alignment methods is that the solutions produced usually considerably overlap with each other; in other words they are highly redundant. Specifically, two clusters of nodes are said to be redundant if more than r% of the nodes in the smaller complex occur in the other complex, where r is a threshold value that determines the extent of allowed overlap between clusters.

Recall that most network alignment methods construct an alignment graph, which is a merged representation of the protein interaction networks being compared. Then, alignment solutions or alignments are network structures of interest found by searching the alignment graph. Each discovered alignment corresponds to a set of complexes, one for each given organism, which are conserved to each other. Thus, a set of alignment solutions gives separate collections of conserved complexes for the species being compared.

Obviously, one may observe the redundancy at two levels: alignment level and protein level. The first level is when alignments found in the alignment graph highly overlap. The second one is when conserved protein complexes in one collection highly overlap.

As mentioned above, in the experimental analysis of this study we use two alignment methods, MaWish and MNAligner. At the alignment level, MaWish filters out redundant solutions ($r = 80\%$) retaining only alignments with bigger score. MNAligner is a global network alignment method where the computed mapping between orthologs is a one-to-one mapping resulting in solutions that do not overlap.

At the node (protein) level MaWish does not handle possible occurrence of redundant complexes. Moreover, despite the fact that MNAligner performs global alignment, it may produce intersecting complexes when applied to sub-graphs generated by Divide, because such sub-graphs may overlap. Therefore, in both instances of modular network alignment here considered, we will have to handle redundant protein complexes.
In general if two complexes have a high intersection, one of them is discarded (see, e.g. Supporting Methods of Sharan et al. (2005b)). However, this approach for handling redundancy is not very satisfactory, since detected conserved complexes could possibly cover part of a ‘true’ functional module either due to constraints on the topology and homology similarity, or due to missing interactome data. Therefore, detected complexes having high overlap may still represent different parts of one bigger module.

In Liang et al. (2006) the following alternative method is proposed for merging redundant solutions. If two clusters are highly intersecting then they are merged into a single cluster by taking the union of the two clusters. Three or more clusters are merged by the rule of single linkage, that is, the merging relation is transitive. We refer to this method as chain-rule merging. A drawback of this procedure is that it may merge protein complexes whose intersection is not any more above the required threshold due to the transitive relation used. Therefore parts of different modules might be merged. Furthermore, application of the chain-rule merging can produce one or few very big modules containing several possible functional complexes.

These observations motivate the introduction of the following procedure for dealing with highly intersecting complexes. Specifically, we modify the chain-rule merging as follows. A set of complexes is merged if every possible pair of complexes contained in this set is redundant.

If we represent complexes by means of nodes and connect two nodes by an edge if they are redundant then the problem of finding a maximal set of complexes which can be merged according to the above rule can be reduced to the problem of finding a maximal clique in that graph. Consequently, finding all such maximal sets is equivalent to the problem of finding all maximal cliques, which is an intractable optimization problem. Nevertheless, in our setting the resulting graph is rather sparse and contains relatively few nodes, which allows us to apply an exact algorithm for finding all maximal cliques in graph (here we use the algorithm of Bron and Kerbosch (1973)).

We refer to the modified merging procedure as clique-rule merging. In our experimental analysis the redundancy threshold \( r = 80\% \) is used.

6. Divide generates sub-graphs covering "true" protein conserved complexes

Let \( \text{Divide sub-graphs} \) denote the sub-graphs generated by \( \text{Divide} \). We compared \( \text{Divide} \) sub-graphs with "true" protein conserved complexes. To this aim, we evaluated the quality of sub-graphs generated by \( \text{Divide} \) using known yeast complexes catalogued in the MIPS database\(^2\).

\(^2\)http://mips.helmholtz-muenchen.de/genre/proj/yeast/
(Guldener et al., 2005). 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. From each of these complexes we extracted the subset of proteins consisting of only orthologous proteins, where sets with less than three elements were filtered out. We call the resulting set of proteins yeast MIPS (conserved) complex.

Table 3 reports the number of yeast sub-graphs and yeast conserved complexes of the alignment tasks for the given pairs of species (yeast-nematode, yeast-fly and yeast-human) after and before the application of the filtering procedures above described.

<table>
<thead>
<tr>
<th>Alignment task</th>
<th>#sub-graphs</th>
<th>#yeast MIPS complexes</th>
<th>#yeast MaWish complexes</th>
</tr>
</thead>
<tbody>
<tr>
<td>S. cerevisiae vs C. elegans</td>
<td>53 (235)</td>
<td>56, 45 (185)</td>
<td>27 (83)</td>
</tr>
<tr>
<td>S. cerevisiae vs D. melanogaster</td>
<td>119 (408)</td>
<td>111, 99 (205)</td>
<td>99 (411)</td>
</tr>
<tr>
<td>S. cerevisiae vs H. sapiens</td>
<td>67 (253)</td>
<td>77, 63 (161)</td>
<td>57 (276)</td>
</tr>
</tbody>
</table>

Table 3: Number of yeast sub-graphs and yeast conserved complexes for a given alignment task: yeast-nematode, yeast-fly or yeast-human. In brackets the number of sub-graphs and complexes before removing sets with less than three elements is given. The second number in the yeast MIPS complexes column is the number of complexes after big-sized complexes have been removed.

The intersection rate between a sub-graph and a complex is used, computed as follow. Let $G = (U, E)$ be a sub-graph and let $C$ be a protein complex of one organism. The intersection rate of $G$ and $C$ is

$$|U \cap C|/|C|.$$ 

In case more Divide sub-graphs have equal intersection rate with a given complex, we chose the sub-graph of smallest size. This sub-graph provides a best coverage of the considered complex, because it needs the smallest number of proteins to achieve that intersection rate.

The relation between the intersection rate of yeast Divide sub-graphs and a "true" complex, and the size of a "true" complex are shown in the left column of Figure 4 for yeast-nematode, yeast-fly or yeast-human alignment task. Low intersection rates mostly correspond to complexes of big size (see left upper part of the plots).

Because conserved complexes have in general small size, we incorporated this prior information in our analysis and filtered out complexes of big size from the list of yeast MIPS complexes, since they were not considered to be conserved. To this end, we used the conserved complexes predicted by MaWish (see also the next section). For yeast-nematode and yeast-human the biggest yeast MaWish complex has size 12, for the yeast-fly alignment task the biggest MaWish complex consists of 21 proteins. Using these parameter values for the threshold to filter out yeast MIPS complexes considered too large, we got 45 yeast MIPS complexes w.r.t. nematode, 99 complexes w.r.t. fly...
Figure 4: Intersection rate vs. size of yeast complexes for the alignment task. Left column: yeast MIPS complexes. Right column: yeast MaWish complexes.
and 63 complexes w.r.t. human (see Table 3). As shown in Table 4 the average intersection rate increased for the small yeast MIPS complexes while the number of considered complexes does not decrease significantly.

Another issue concerns the selection of only one Divide sub-graph when computing the intersection rate with a complex. Divide sub-graphs having equal intersection rate with a complex may cover that complex in different ways. Therefore, one should consider the contribution to the coverage of that complex provided by all these sub-graphs. This may be formalized by defining a so-called union intersection rate as follows. Let $S$ be a set of Divide sub-graphs having the same intersection rate with a complex $C$. The union intersection rate is

$$I_{U} = \frac{| \bigcup_{G(U,E) \in S} U \cap C |}{|C|}.$$

The average union intersection rate between yeast MIPS complexes and sub-graphs is shown in Table 4 for three alignment tasks. The union intersection rate is higher than the intersection rate. Highest values are obtained for small-size complexes. For each alignment task, more than 70% coverage of yeast MIPS complexes is achieved. This means that some yeast MIPS conserved complexes are split among sub-graphs, hence different parts of conserved complexes can be discovered by searching in these sub-graphs.

<table>
<thead>
<tr>
<th>Considered yeast complexes</th>
<th>C. elegans (%)</th>
<th>D. melanogaster (%)</th>
<th>H. sapiens (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>small-sized (union)</td>
<td>71.4</td>
<td>75.7</td>
<td>81.9</td>
</tr>
<tr>
<td>all (union)</td>
<td>64.5</td>
<td>71.3</td>
<td>74.8</td>
</tr>
<tr>
<td>small-sized</td>
<td>64</td>
<td>68</td>
<td>69.3</td>
</tr>
<tr>
<td>all</td>
<td>56</td>
<td>64.2</td>
<td>63.4</td>
</tr>
</tbody>
</table>

Table 4: Average of (union) intersection rate of yeast MIPS complexes and sub-graphs given an alignment task: yeast-nematode, yeast-fly or yeast-human.

These results indicate that Divide is able to generate sub-graphs that highly cover "true" conserved complexes. Lower intersection rate for yeast MIPS complexes could be due to the fact that functional complexes in MIPS database are not biased on protein interaction conservation across species. Nevertheless, we achieved a satisfactory intersection rate for small-sized complexes, which are more likely to be (part of) conserved protein complexes.

7. Comparison of Divide Sub-graphs with Predicted Conserved Complexes

Here we investigate how Divide constrains the search process of MaWish, and whether the sub-graphs generated by Divide cover those produced by MaWish. To this end, we used the conserved complexes predicted by this alignment method and processed by the clique-rule merging procedure,
where complexes consisting of one or two proteins were filtered out. We call the resulting sets
MaWish complexes.

In the right column of Figure 4 one can observe that a number of yeast MaWish complexes are
fully covered and many of those, which are not fully covered, intersect with a sub-graph at a rate
higher than 0.5.

Next we computed the average intersection rate of MaWish complexes for each of the considered
alignment tasks of yeast-nematode, yeast-fly and yeast-human. For a given pair of organisms, we
computed the number of conserved complexes for the first and for the second organism, and the
intersection rate between the complexes and sub-graphs of the first organism and of the second
organism, respectively. In all cases, we got almost or more than 80% coverage of conserved
complexes (see Table 5).

<table>
<thead>
<tr>
<th>Alignment task</th>
<th>#conserved MaWish complexes</th>
<th>intersection rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>S. cerevisiae vs C. elegans</td>
<td>27, 24</td>
<td>87.0, 91.7</td>
</tr>
<tr>
<td>S. cerevisiae vs D. melanogaster</td>
<td>99, 80</td>
<td>79.9, 84.8</td>
</tr>
<tr>
<td>S. cerevisiae vs H. sapiens</td>
<td>57, 63</td>
<td>84.7, 89.8</td>
</tr>
</tbody>
</table>

Table 5: Average intersection rate of MaWish conserved complexes and sub-graphs for a given alignment task. In
each column, the first number contains the number of conserved MaWish complexes of yeast and the second one the
number of conserved complexes of the second organism in the considered alignment task.

Results of the experiments indicate that Divide can be used to perform modular network
alignment, since the sub-graphs it generates cover "true" as well as predicted conserved complexes.
In order to further substantiate this observation, we performed modular network alignment on a
case study for Saccharomyces cerevisiae and Caenorhabditis elegans. These organisms are generally
used to test the performance of alignment methods. Moreover, on these two organisms the worst
coverage for the yeast MIPS complexes and the best coverage for MaWish complexes were obtained.
Therefore they provide a hard benchmark instance problem for testing the performance of the
modular network alignment described below.

8. Applications of Modular Network Alignment

In this section we investigate the ability of Divide to enhance the performance of alignment
methods. Specifically, we apply Divide to two different alignment methods.

In the first case, we consider an instance of modular local network alignment, called DivAfull
(Jancura et al., 2008a). DivAfull employs Divide to generate sub-graphs, the MaWish alignment
model to align them, and iterative exact search to detect all possible solutions from the generated
alignments. Therefore, application of Divide allows one to improve the search process by replacing
the greedy search procedure of MaWish with an exact search algorithm.

In the second case we consider an instance of modular global network alignment. Specifically,
we show how MNAligner (Li et al., 2007) can be enhanced by prior application of Divide. In order
to detect conserved complexes using MNAligner a clustering algorithm is applied which detects
potential protein complexes in one PPI network. The resulting complexes are then aligned with
the second PPI network and the conserved protein (sub-)complexes are detected. Here, we apply
Divide before clustering in order to bias the search for complexes towards regions centered around
articulation hubs. Results of experiments indicate that this is an effective way of enhancing the
discovery of conserved complexes using MNAligner.

We discuss DivAfull and MNAligner more in detail in the next two sections. Then we
introduce validation measures in order to assess the quality of the discovered protein complexes
and the performance of the methods. Finally, we discuss results of experiments.

8.1. DivAfull

DivAfull uses the dividing procedure to generate sub-graphs for each of PPI networks given
by species to be compared. Next, pairs of the sub-graphs from different species are merged using
the MaWish network alignment model.

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, is computed.
A node in the alignment graph is a pair of orthologous 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 Koyutürk et al. (2006b) for a formal description of these terms.

Induced sub-graphs of the resulting weighted alignment graph with total weight greater than
a given threshold are considered as relevant alignments. Each relevant alignment corresponds to
two putative conserved complexes, one for each species.

After merging we search for these sub-graphs. This problem is reduced to the (optimization)
problem of finding a maximal induced subgraph. To tackle this problem, the search part of MaWish
consists of an approximation greedy algorithm based on local search, because the maximum in-
duced 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. The seed is expanded
by adding (removing) nodes to (from) the subgraph while the actual subgraph weight increases.
In contrast, DivAfull applies iteratively an exact optimization algorithm (Wolsey, 1998) for searching relevant alignments (maximum weighted induced sub-graphs) in the alignment graphs produced by merging possible pairs of the sub-graphs, since their size is small. This search algorithm is described in detail below.

8.1.1. Search Algorithm

First, an exact optimization algorithm for finding the maximum weighted induced subgraph is 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:

\[
\text{opt} = \max_C f \quad (\text{OptP})
\]

Algorithm 2 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 2 Full Search Algorithm**

**Input:** G: alignment sub-graph, \( \varepsilon \geq 0 \)

**Output:** List of heavy induced sub-graphs of G with weight > \( \varepsilon \)

1. Formulate the problem of MaxInducedSubGraph for G as (OptP)
2. \( \text{maxweight} = \infty \)
3. \( C = C + \{\text{opt} < \text{maxweight}\} \)
4. while \( \text{maxweight} > \varepsilon \) do
5. solve (OptP) by an exact method
6. if \( \text{opt} > \varepsilon \) then
7. record discovered solution
8. end if
9. \( \text{maxweight} = \text{opt} \)
10. end while

8.1.2. Handling Redundant Solutions

DivAfull employs the same filtering procedure as MaWish for dropping redundant alignments of worse weights. Still, a redundancy among protein complexes may occur. Hence, we apply clique-rule merging as described in Section 5.1.

8.2. Divide and MNAligner

MNAligner is a general tool for global alignment of molecular networks. It formalizes the problem of finding an optimal mapping between similar nodes of two different networks as an
integer quadratic programming optimization problem which is relaxed to quadratic optimization
problem (QP). The optimal integer solution is ensured if appropriate sufficient conditions on the
objective function are satisfied. However, QP may have an integer solution also if the conditions
are not satisfied. We refer to Li et al. (2007) for the detailed description of the objective function
and alignment algorithm.

Direct application of MNAligner is feasible only when small PPI networks are considered.
Furthermore, the alignment algorithm finds the global mapping but does not search for structures
of interest in this mapping, such as being dense sub-graphs. Therefore an additional search for such
structure is needed either before or after the alignment. For instance, in one of the applications
of MNAligner described in Li et al. (2007) two large PPI networks are aligned in order to detect
conserved complexes. To overcome the problem of large network size, and to bias the search
towards detection of protein complexes, the clustering algorithm MCODE (Bader and Hogue, 2003)
is applied to one of the networks prior the alignment. This algorithm generates a set of clusters
representing potential functional protein complexes. Each of these clusters is aligned with the
second PPI network, resulting in the detection of two sub-networks, one for each species being
compared.

Application of Divide yields sub-graphs representing regions of interest which potentially cover
a number of conserved complexes. We test whether the application of Divide prior to the use
of MCODE and MNAligner enhances the discovery of conserved complexes. Specifically, given two
PPI networks $G_1$ and $G_2$, we divide $G_1$ using Divide. Each of the resulting sub-graphs is further
processed by MCODE and aligned with $G_2$ using MNAligner. In this way two collections of conserved
complexes are generated, one for each species. We repeat this process by dividing $G_2$ instead of
$G_1$. This gives again two collections of complexes. For each species, the union of the collections
of detected complexes from that species is considered. Because sub-graphs of Divide overlap and
MCODE may also construct overlapping clusters we process each final collection by means of the
clique-rule merging procedure and retain only complexes of size greater or equal than 3. In such
way we get two complete collections of possible conserved complexes detectable by MNAligner.

These results are compared with complexes produced when MCODE and MNAligner are directly
applied to the PPI networks induced by orthologs. Specifically, we cluster the (orthologous sub-
network of) $G_1$ and align the resulting clusters with $G_2$. We repeat the process by interchanging
the role of $G_1$ and $G_2$. We again process the results using the clique-rule merging procedure and
removing complexes of size less than 3.
In this way, we allow a fair comparison of results when only MCODE and MNAligner are applied and when Divide is introduced prior these steps. We restrict the use of MCODE on sub-networks induced by orthologs, because Divide divides only the orthologs of a PPI network.

8.3. Evaluation Criteria for Conserved Complexes

We assess the performance of alignment methods by measuring the quality of detected complexes. A functional module may perform one or more functions in an organism and all proteins contained in that module are associated with these functions. Based on this assumption, computationally derived protein complexes may serve for predicting function of proteins. Then the quality of a complex can be assessed by the function prediction of the proteins it contains.

Therefore, we measure the enrichment of functional annotations of the protein set in a complex, as entailed by the gene ontology (GO) annotation (Ashburner et al., 2000), using one of the well-established tools, the Ontologizer\(^3\) (Robinson et al., 2004). Ontologizer measures statistical significance of an enrichment and assigns to the complex a p-value for each enriched function. The p-value is corrected for multiple testing by a classic Bonferroni correction procedure. Furthermore, Ontologizer also constructs a hierarchical directed acyclic graph (DAG) consisting of all significantly enriched annotations and all their ancestor annotations up to the root in the whole GO hierarchy. Given a DAG of enrichments, the level of an annotation is equal to the length of the shortest path from the root of GO hierarchy present in the DAG to that annotation.

A complex can be used as protein function predictor if the following criteria are satisfied:

1. a certain GO annotation is significantly enriched by the proteins in the complex (p-value < 0.05);
2. at least half of the proteins in the complex has this significant annotation;
3. the annotation is at least at GO level four from the root in GO hierarchy.

In such a case the significantly enriched GO annotation of the complex is used to predict protein function of each of the proteins in that complex. If a complex does not satisfy the above conditions, no prediction can be made. Similar criteria were used by, e.g. Liang et al. (2006). The condition on GO hierarchy guarantees that the prediction about biological functions is sufficiently specific and informative (Yon Rhee et al., 2008).

We validate the accuracy of the predictions, and consequently the quality of a protein complex, in a way similar to that proposed by Deng et al. (2003). Specifically, given a protein complex and

\(^3\)http://compbio.charite.de/index.php/ontologizer2.html
the corresponding DAG of enrichments, we restrict our validation only to the annotations which
are present in the DAG and are at GO level four or higher. A protein $p$ of the complex having such
annotations is assumed to be not annotated and its functions are predicted. The predictions are
then compared with the annotations of the protein $p$. The method is repeated for all annotated
proteins in the cluster. In the end, for each protein $p$ we have:

- $A_p$: the number of annotated functions for the protein $p$.
- $P_p$: the number of predicted functions for the protein $p$.
- $O_p$: the size of the overlap between the set of annotated functions and the set of predicted
  functions for the protein $p$.

Given this scheme, precision (PR) and recall (RC) are computed for each complex $C$ as follows:

$$PR(C) = \frac{\sum_{p \in C} O_p}{\sum_{p \in C} P_p},$$

$$RC(C) = \frac{\sum_{p \in C} O_p}{\sum_{p \in C} A_p}.$$

In the case of no prediction, precision and recall are set to zero. When both precision and
recall are close to one then function prediction of a protein complex is good. Therefore, we also
use the following well-established measure in information retrieval (Rijsbergen, 1979) as suggested
by Handl et al. (2005), the F-measure (FM), defined as

$$FM(C) = \frac{2 \cdot PR(C) \cdot RC(C)}{PR(C) + RC(C)},$$

where we assume that both precision and recall are equally important. We use the above
evaluation measure to validate the quality of a predicted complex with respect to its ability to
model the functions of the proteins it contains.

In order to assess whether Divide leads to the discovery of conserved complexes having a new
putative function, we introduce the following two additional measures, functional ratio ($FNR$) and
coverage ratio ($CVR$).

Let $A$ be the collection of all functions predicted by the complexes detected by the original
method and let $B$ be the collection of all function predicted by the complexes detected by the
combined method. Furthermore, denote by $C_X$ the set of all complexes which are predicted to
have a function from the function collection $X$. Then

$$FNR = \frac{|B \setminus A|}{|B|},$$

$$CVR = \frac{|C_X \setminus B|}{|C_X|}.$$
The first measure, FNR, computes the ratio of new functions discovered over the set of all functions discovered by the combined method. The latter one computes the ratio between the number of complexes which are predicted to have the new functions and the total number of complexes detected by the combined method.

Notice that all measures above defined treat each species separately rather than explicitly evaluating the conservation hypothesis implied by each pair of conserved complexes aligned. Such evaluation could, in principle, be performed by comparing the results to a reference set of conserved modules (Yosef et al., 2008). To date, however, most such references are not comprehensive enough and contain only a small number of cases to learn from (Yosef et al., 2008). One exception is the Biocarta database which contains many human-mouse conserved pathways.

Finally, it should be also noted that the functional annotations for the annotated proteins are incomplete. Thus, we may have a high confidence in the assignment of the function to a protein based on the GO annotation. However, that protein can have a particular true function which has not yet been annotated, that is, it has not been experimentally validated.

8.4. A Case Study: Saccharomyces cerevisiae vs Caenorhabditis Elegans

We present results on the considered case study as follows. We summarize the application of Divide algorithm on particular PPI networks. Then we show how the iterative exact search of DivAfull improves on MaWish results. Finally, we discuss results of MNAAligner combined with Divide.

8.4.1. Application of Divide

Results of application of the Divide algorithm to the PPI networks of Saccharomyces cerevisiae and Caenorhabditis elegans are following.

For Saccharomyces cerevisiae, 697 articulations, of which 151 orthologs, were computed, and 83 centers were constructed from them. Expansion of these centers into centered trees resulted in 639 covered orthologs. The algorithm assigned the remaining 153 orthologous proteins to 152 new sub-trees.

For Caenorhabditis elegans, 586 articulations, of which 158 orthologs, were computed, and 112 centers were constructed from them. Expansion of these centers into centered trees resulted in

\[ CVR = \frac{|C_B \backslash A|}{|C_B|}. \]
339 covered orthologs. The algorithm assigned the remaining orthologous 294 proteins to 288 new sub-trees.

We observed that the last remaining orthologs assigned to sub-trees were 'isolated' nodes, in the sense that they were rather distant from each other and not reachable from ortholog paths stemming from centers.

We obtained 235 sub-trees for *Saccharomyces cerevisiae* and 400 sub-trees of *Caenorhabditis elegans*. Nodes of each such tree induce a PPI sub-graph.

8.4.2. DivAfull and MaWish

DivAfull constructs alignment graphs between each two PPI sub-graphs containing more than one orthologous pair. In such way, we obtained 884 alignment graphs, where the biggest one consisted of only 31 nodes.

We applied Algorithm 2 to each of the resulting alignment graphs. Zero weight threshold ($\varepsilon = 0$) was used for considering an induced subgraph as a heavy subgraph or a legal alignment. Redundant graphs were filtered using $r = 80\%$ as the threshold for redundancy.

DivAfull discovered 151 solutions (alignments) while MaWish yielded 83 solutions. Between these two set of solutions we found 70 redundant alignments, whose pair of weights are plotted on the left part of Figure 5. Among these, 48 (31.8% of DivAfull results) were 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 was undecidable because of rounding errors during computation).

DivAfull found 81 (53.6%) new alignments, that is, not discovered by MaWish. The right plot of Figure 5 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 MaWish (0.86) results.

Further, we investigate conserved complexes derived from the alignments discovered. Recall, each set of discovered alignments gives two collections of conserved complexes, one for each species being compared, which are processed by clique-rule merging algorithm and only complexes of size greater than 2 are considered.

DivAfull discovered a higher number of protein complexes than MaWish and the same is observed when only those complexes which satisfy the criteria for being a functional predictor are considered. Concretely, for *Saccharomyces cerevisiae* DivAfull found 46 complexes of which 39
are functional predictors, and for *Caenorhabditis elegans* DivAfull found 28 complexes of which 18 are functional predictors. In contrast, MaWish found 27 complexes of which 24 are potential predictors for *Saccharomyces cerevisiae* and 24 complexes of which 13 are functional predictors for *Caenorhabditis elegans*.

We measured the GO enrichment of these complexes and computed the average of their precisions, the average of their recalls, and the average of their F-measures. The results are reported in Table 6 and Table 7 for *Saccharomyces cerevisiae* and *Caenorhabditis elegans*, respectively.

For *Saccharomyces cerevisiae*, when considering all modules, we observe lower average precision and average F-measure of DivAfull modules than of MaWish complexes (the upper part of Table 6). However, the difference in F-measures is subtle and average recalls are same. Thus, complexes of both methods are, in total, of comparable quality. When focused on functional predictors (the bottom part of Table 6), DivAfull clearly outperforms MaWish functional predictors.

<table>
<thead>
<tr>
<th>Method</th>
<th># Modules</th>
<th>Precision (±δ)</th>
<th>Recall (±δ)</th>
<th>F-measure (±δ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DivAfull</td>
<td>46</td>
<td>0.73 (±0.33)</td>
<td>0.52 (±0.30)</td>
<td>0.59 (±0.31)</td>
</tr>
<tr>
<td>MaWish</td>
<td>27</td>
<td>0.75 (±0.30)</td>
<td>0.52 (±0.26)</td>
<td>0.60 (±0.28)</td>
</tr>
<tr>
<td>DivAfull</td>
<td>39</td>
<td>0.86 (±0.13)</td>
<td>0.61 (±0.22)</td>
<td>0.70 (±0.20)</td>
</tr>
<tr>
<td>MaWish</td>
<td>24</td>
<td>0.84 (±0.13)</td>
<td>0.58 (±0.20)</td>
<td>0.67 (±0.18)</td>
</tr>
</tbody>
</table>

Table 6: The average of precisions, the average of recalls, and the average of F-measures of yeast protein modules. The upper part reports results for all complexes, the bottom part for all functional predictors.

For *Caenorhabditis elegans*, when considering all modules, a better average functional enrichment is achieved for DivAfull modules (the upper part of Table 7). Considering all functional predictors, MaWish complexes have a higher average precision but a better recall is obtained by
<table>
<thead>
<tr>
<th>Method</th>
<th>#Modules</th>
<th>Precision (±¿)</th>
<th>Recall (±5)</th>
<th>F-measure (±5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>DivAfull</td>
<td>28</td>
<td>0.56 (±0.44)</td>
<td>0.46 (±0.46)</td>
<td>0.49 (±0.40)</td>
</tr>
<tr>
<td>MaWish</td>
<td>24</td>
<td>0.50 (±0.48)</td>
<td>0.38 (±0.40)</td>
<td>0.41 (±0.42)</td>
</tr>
<tr>
<td>DivAfull</td>
<td>18</td>
<td>0.87 (±0.12)</td>
<td>0.71 (±0.25)</td>
<td>0.76 (±0.21)</td>
</tr>
<tr>
<td>MaWish</td>
<td>13</td>
<td>0.93 (±0.10)</td>
<td>0.70 (±0.27)</td>
<td>0.76 (±0.22)</td>
</tr>
</tbody>
</table>

Table 7: The average of precisions, the average of recalls, and the average of F-measures of nematode protein modules. The upper part reports results for all complexes, the bottom part for all functional predictors.

Furthermore, it is interesting to investigate whether DivAfull modules also provide new predictions. By computing functional and coverage ratio over all functions predicted by DivAfull functional predictions with respect to biological functions of MaWish predictions, Table 8 shows that there is a particular fraction of new discoveries for both species.

Table 8: The total number of biological functions predicted by DivAfull functional predictors and their functional ratio and the total number of DivAfull functional predictors and their coverage ratio computed with respect to MaWish results.

<table>
<thead>
<tr>
<th>Species</th>
<th>#Functions</th>
<th>FNR</th>
<th>#Predictors</th>
<th>CVR</th>
</tr>
</thead>
<tbody>
<tr>
<td>yeast</td>
<td>144</td>
<td>0.23</td>
<td>39</td>
<td>0.26</td>
</tr>
<tr>
<td>nematode</td>
<td>90</td>
<td>0.06</td>
<td>18</td>
<td>0.17</td>
</tr>
</tbody>
</table>

Furthermore, it is interesting to investigate whether DivAfull modules also provide new predictions. By computing functional and coverage ratio over all functions predicted by DivAfull functional predictions with respect to biological functions of MaWish predictions, Table 8 shows that there is a particular fraction of new discoveries for both species.

To sum up, we may conclude that DivAfull discovered a higher number of conserved complexes of the comparable or higher quality than MaWish. DivAfull also achieved new predictions.

8.4.3. Divide and MNAligner

MNAligner applies MCODE to each sub-graph produced by Divide before using the alignment procedure. Despite of the high number of generated sub-graphs generated by Divide, many of them have an empty set of complexes detected by MCODE. Indeed, the final number of conserved complexes is low, 12 complexes for Saccharomyces cerevisiae and 10 modules for Caenorhabditis elegans. However, almost the same number of complexes is discovered when MCODE is directly applied on orthologous sub-networks of the species being compared (see Tables 9 and 10, respectively). These results seem to indicate that the low number of discovered complexes is due to characteristics of MCODE’s clustering approach.

Tables 9 and 10 show the average of precisions, the average of recalls, and the average of F-measures of the detected complexes for Saccharomyces cerevisiae and Caenorhabditis elegans, respectively, after measuring their GO enrichment.

From Table 9 it can be seen that the complexes of Saccharomyces cerevisiae discovered when Divide was applied, and their subset of functional predictions, significantly outperformed the
Table 9: MNAligner: The average of precisions, the average of recalls, and the average of F-measures of yeast protein modules. The upper part reports results for all complexes, the bottom part for all functional predictions.

<table>
<thead>
<tr>
<th>Method</th>
<th>#Modules</th>
<th>Precision (±δ)</th>
<th>Recall (±δ)</th>
<th>F-measure (±δ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Divide + MNAligner</td>
<td>12</td>
<td>0.74 (±0.25)</td>
<td>0.53 (±0.28)</td>
<td>0.60 (±0.27)</td>
</tr>
<tr>
<td>MNAligner</td>
<td>13</td>
<td>0.69 (±0.24)</td>
<td>0.48 (±0.27)</td>
<td>0.53 (±0.28)</td>
</tr>
<tr>
<td>Divide + MNAligner</td>
<td>11</td>
<td>0.81 (±0.09)</td>
<td>0.58 (±0.24)</td>
<td>0.65 (±0.21)</td>
</tr>
<tr>
<td>MNAligner</td>
<td>12</td>
<td>0.75 (±0.12)</td>
<td>0.52 (±0.24)</td>
<td>0.58 (±0.24)</td>
</tr>
</tbody>
</table>

Table 10: MNAligner: The average of precisions, the average of recalls, and the average of F-measures of nematode protein modules. The upper part reports results for all complexes, the bottom part for all functional predictions.

<table>
<thead>
<tr>
<th>Method</th>
<th>#Modules</th>
<th>Precision (±δ)</th>
<th>Recall (±δ)</th>
<th>F-measure (±δ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Divide + MNAligner</td>
<td>10</td>
<td>0.72 (±0.39)</td>
<td>0.55 (±0.4)</td>
<td>0.59 (±0.37)</td>
</tr>
<tr>
<td>MNAligner</td>
<td>11</td>
<td>0.38 (±0.45)</td>
<td>0.34 (±0.43)</td>
<td>0.35 (±0.43)</td>
</tr>
<tr>
<td>Divide + MNAligner</td>
<td>8</td>
<td>0.90 (±0.11)</td>
<td>0.68 (±0.31)</td>
<td>0.74 (±0.23)</td>
</tr>
<tr>
<td>MNAligner</td>
<td>5</td>
<td>0.83 (±0.15)</td>
<td>0.75 (±0.29)</td>
<td>0.77 (±0.24)</td>
</tr>
</tbody>
</table>

For *Caenorhabditis elegans*, if we consider all modules, again better results are achieved when Divide is incorporated prior the clustering and alignment steps (the upper part of Table 10). When we focused on functional predictions, the application of Divide lead to results of higher precision but lower recall, which also affected the F-measure (the bottom part of Table 10). However, from 10 conserved complexes discovered when Divide is applied, 8 are potential functional predictions, but, in the case when orthologous sub-networks are not divided, more than the half of the results do not satisfy criteria for functional prediction.

In summary, the application of Divide resulted in new and in the majority of the cases better results despite the fact that the same clustering technique was applied on Divide sub-graphs as on the original whole orthologous sub-networks before the division. This shows that Divide can positively bias the search for improving detection of conserved complexes by means of modular global network alignment.
9. Conclusion

This paper introduced a heuristic algorithm, Divide, for dividing protein interaction networks in such a way that conserved functional complexes are covered by generated sub-graphs. To the best of our knowledge, this is the first algorithm for this task, which can be used to perform modular network alignment of protein interaction networks (Jancura et al., 2008a,b).

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.

We showed experimentally that the sub-graphs that were generated by Divide covered part of predicted conserved complexes. In some cases these sub-graphs covered different parts of one conserved complex. We tested experimentally the ability of Divide to be used for performing modular network alignment. Specifically, we performed two comparative experimental analysis.

In the first experiment we used the DivAfull algorithm, which uses Divide prior to the alignment phase, as done by Jancura et al. (2008a). Comparison between results of MaWish and DivAfull indicated that DivAfull is able to discover new alignments which significantly increase the number of discovered complexes. Moreover, complexes discovered by DivAfull showed comparable or improved GO enrichment, as measured by precision, recall, and F-measure, and provided new prediction of protein functions. This application shows that using Divide one can enhance the search strategy by replacing greedy with exact search in the alignment graph, resulting in the discovery of new conserved complexes.

In the second experiment an instance of global network alignment approach, called MNAAligner, was considered. This method employs a pre-processing step before computing the alignment of two PPI networks. The results showed that the application of Divide enhanced the quality of the results. This indicates the regions around articulation hubs constructed by Divide provide a beneficial search bias for detecting functional complexes and enhancing the performance of MNAAligner.

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

Another advantage of applying the Divide algorithm for performing modular protein network alignment is that it allows one to parallelise alignment methods. For instance, the full search algorithm DivAfull can be run independently on each alignment graph constructed on sub-graphs generated by Divide.
In future work we intend to employ the Divide algorithm for multiple network alignment problem.

Acknowledgments. We would like to thank Mehmet Koyutürk for providing the MaWish code.

References

URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/25/23/3166

URL http://dx.doi.org/10.1038/75556

URL http://www.biomedcentral.com/1471-2105/4/2


URL http://genome.cshlp.org/content/16/3/428.abstract

URL http://www.pnas.org/content/103/29/10967.abstract


URL http://dx.doi.org/10.1093/nar/gkp474


URL http://www.liebertonline.com/doi/abs/10.1089/106652703322756168


URL http://www.liebertonline.com/doi/abs/10.1089/cmb.2007.0172


URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/13/i149


URL http://www.biomedcentral.com/1471-2105/10/S1/S59


URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/13/1631

URL http://www.biomedcentral.com/1471-2105/7/457

32
URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/25/12/i253


URL http://www.liebertonline.com/doi/abs/10.1089/cmb.2007.0025

URL http://www.liebertonline.com/doi/abs/10.1089/152791601750294344

URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/16/3401

Pržulj, N., 2005. Knowledge Discovery in Proteomics: Graph Theory Analysis of Protein-Protein Interactions. CRC Press.


URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/20/6/979


URL http://www.pnas.org/cgi/content/abstract/102/6/1974


URL http://www.biomedcentral.com/1471-2105/7/199


URL http://dx.doi.org/10.1007/978-3-540-71681-5\_2


URL http://www.pnas.org/content/105/35/12763.abstract


URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/15/1978


URL http://dx.doi.org/10.1038/nrg2363


URL http://bioinformatics.oxfordjournals.org/cgi/content/abstract/25/12/i259