

Dividing Protein Interaction Networks for Modular Network Comparative Analysis[☆]

Pavol Jancura^{*,a}, Elena Marchiori^a

^a*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)

26 Recent studies consider a comparative approach for the analysis of PPI networks from different
27 species in order to discover common protein groups, called conserved complexes, which are likely
28 to be related and to share similar functionality in a cell (Sharan and Ideker, 2006; Srinivasan et al.,
29 2007). This problem is known as *protein network alignment*. Algorithms for this task typically
30 model this problem by means of a merged graph representation of the networks to be compared,
31 called *alignment (or orthology) graph*, and then formalize the problem of *searching* (merged) con-
32 served complexes in the alignment graph as an optimization problem. Due to the computational
33 intractability of the resulting optimization problem, greedy algorithms are commonly used.

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

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

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

45 Conserved complexes discovered by computational techniques have in general small size (that
46 is, number of proteins) compared to the size of the PPI network they belong to. Moreover, PPI
47 networks are known to have a scale-free topology where most proteins participate in a small
48 number of interactions while a few proteins, called *hubs*, contain a high number of interactions.
49 As indicated by a recent study, hubs whose removal disconnects a PPI network (articulation hubs)
50 are likely to appear in conserved interaction patterns (Pržulj, 2005).

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

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

57 First, we show that the sub-graphs generated by **Divide** indeed cover "true" conserved pro-

58 tein complexes. This is done by measuring the overlap of these sub-graphs with MIPS curated
59 functional complexes restricted to those proteins belonging to an orthologous pair.

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

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

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

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

83 This paper contains and extends material from two previous conferences' papers of Jancura
84 et al. (2008a,b). It is organized as follows. In the next section we discuss related works. Section
85 3 describes the graph-theoretic terminology used in the paper. The **Divide** algorithm is intro-
86 duced in Section 4. Section 5 summarizes the data and the type of assessment employed in the
87 experimental analysis. In Section 6 the robustness of **Divide** is assessed by analysing how the
88 generated sub-graphs cover "true" complexes. In Section 7 the sub-graphs generated by **Divide**
89 are compared with the complexes predicted by **MaWish**. In Section 8 modular network alignment

90 is performed on the two case studies above described. Finally, we conclude and briefly address
91 future work in Section 9.

92 **2. Related Work**

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

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

104 Other alignment techniques for discovering conserved pathways based on the species conserva-
105 tion were proposed (e.g., Shlomi et al., 2006; Qian et al., 2009; Pinter et al., 2005; Cheng et al.,
106 2008; Li et al., 2007; Koyutürk et al., 2006a), as well as methods handling network structures more
107 general than single pathways (Wernicke and Rasche, 2007; Yang and Sze, 2007; Dost et al., 2008;
108 Blin et al., 2009; Bruckner et al., 2009).

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

117 A first attempt to perform multiple network alignment using three species was done by Sharan
118 et al. (2005b). However, the method scales exponentially with the number of input species.
119 Thus, new methods for aligning multiple species have been proposed (Flannick et al., 2006, 2009;
120 Dutkowski and Tiuryn, 2007; Kalaev et al., 2009).

121 The main goal of *global protein network alignment* is functional orthologs detection, because,
122 in contrast to local network alignment, each node in an input network is either matched to one
123 node in the other network or has no match in the other network (Singh et al., 2007). Of course
124 global protein network alignment can be also used for identifying conserved complexes.

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

128 The first formal global network alignment method was introduced by Singh et al. (2007). This
129 method has been followed by more works on global pairwise network alignment (Evans et al., 2008;
130 Zaslavskiy et al., 2009; Klau, 2009; Chindelevitch et al., 2010). Singh et al. (2008a); Liao et al.
131 (2009); Flannick et al. (2009) tackled global alignment of multiple species.

132 While the above works focus on alignment of networks, we deal with protein networks pre-
133 processing prior to their alignment, in order to perform modular network alignment (Jancura
134 et al., 2008b).

135 We turn now to the description of works related to the main graph topological ingredient used
136 in our method: hub articulation. Many papers have investigated the importance of hubs in PPI
137 networks and functional groups (Ekman et al., 2006; Jeong et al., 2001; Pržulj, 2005; Pržulj et al.,
138 2004; Rathod and Fukami, 2005; Ucar et al., 2006). In particular, it has been shown by Jeong
139 et al. (2001) that hubs with a central role in the network architecture are three times more likely
140 to be essential than proteins with only a small number of links to other proteins. Moreover, if one
141 takes functional groups in PPI networks, then, amongst all functional groups, cellular organization
142 proteins have the largest presence in those hubs whose removal disconnects the network (Pržulj,
143 2005). These works justify the use of articulation hubs for dividing PPI networks prior to their
144 alignment.

145 3. Graph Theoretic Background

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

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

150 A sub-graph $H(V, F)$ of an undirected graph $G(U, E)$ is said to be *induced* by the set of nodes
151 $V \subset U$ if and only if the set of edges $F \subset E$ consists of all the edges that appear in G over the
152 same vertex set V .

153 A graph is connected if there is a path from any node to any other node. Let $G(U, E)$ be
 154 a connected undirected graph. A vertex $u \in U$ is called *articulation* if the graph resulting by
 155 removing this vertex from G and all its edges, is not connected.

156 A *tree* is a connected graph not containing any circle. A tree is called *rooted tree* if one vertex
 157 of the tree has been designated as the root. Given a rooted tree $T(V, F)$, the depth of a vertex
 158 $v \in V$ is the number of edges from the root to v without repetition of edges. Leaves of the tree T
 159 are vertices which have only one neighbor. The depth of a tree is the highest depth of its leaves.
 160 A spanning tree $T(V, F)$ of a connected undirected graph $G(U, E)$ is a tree where $V = U$ and
 161 $F \subseteq E$.

162 Given an edge-weighted (or node-weighted) graph $G(U, E)$ with a scoring function $w : e \in$
 163 $E \rightarrow \mathfrak{R}$ (or $w : u \in U \rightarrow \mathfrak{R}$). *Total weight* $w(G)$ of G is the sum of weights of all edges (or nodes)
 164 in the graph:

$$165 \quad w(G) = \sum_{\forall e \in E} w(e) \quad (\text{or } w(G) = \sum_{\forall u \in U} w(u)).$$

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

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

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

176 Examples of a spanning and centered tree are shown in Figure 1.

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

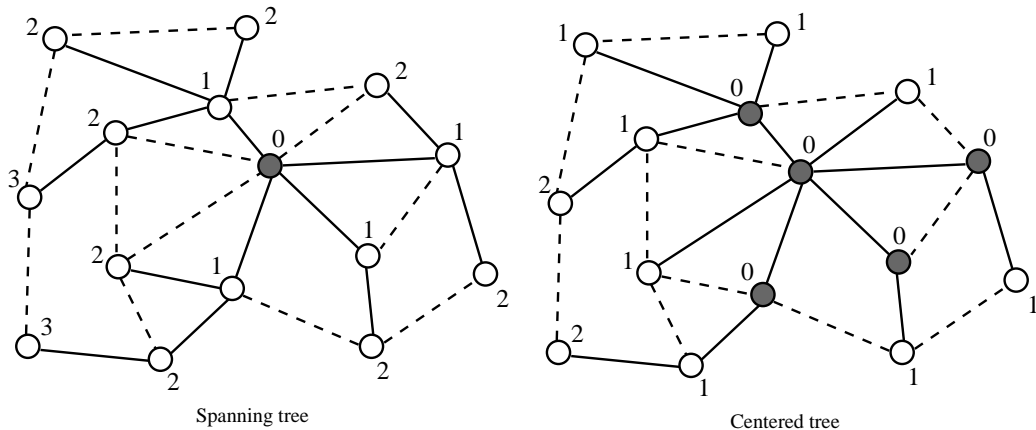


Figure 1: Examples of spanning and centered tree in the same graph. The dark grey node in the left figure represents a root. Dark grey nodes in the right figure represent a center. Numbers indicate depths of nodes in trees. Solid edges indicate edges of a spanning tree. Dash edges are other edges of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

183 **4. Divide Algorithm**

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

189
 190 **Computing Articulations** (Line 1). Computation of articulations can be performed in linear
 191 time by using, e.g., Tarjan’s algorithm described in Tarjan (1972) or Hopcroft and Tarjan (1973).

192
 193 **Greedy Construction of Centers** (Lines 3-10). The degree (in G) of all orthologous articu-
 194 lations is used for selecting seeds for the construction of centers. Networks with scale-free topology
 195 appear to have edges between hubs systematically suppressed, while those between a hub and a
 196 low-connected protein seem favoured (Maslov and Sneppen, 2002). Guided by this observation,
 197 we greedily construct centers by joining one orthologous articulation hub with its orthologous
 198 articulation neighbors, which will more likely have low degree.

199 Specifically, let A be the set of orthologous articulations of G . The first center consists of
 200 the element of A with highest degree and all its neighbors in A . The other centers are generated
 201 iteratively by considering, at each iteration, the element of A with highest degree among those
 202 which do not occur in any of the centers constructed so far, together with all its neighbors in A

203 which do not already occur in any other center. The process terminates when all elements of A
204 are in at least one center. Then an unambiguous label is assigned to each center.

205

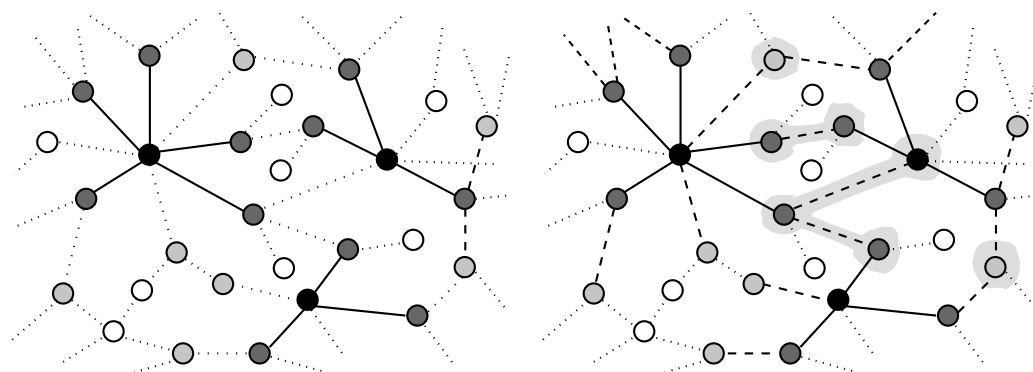


Figure 2: Examples of centers of centered trees (left figure) and of their initial expansion (right figure). Seeds of centers are solid nodes. Dark gray nodes are the rest of centers connected to a seed by solid edges. Light gray nodes are orthologous proteins which are not articulations. Empty nodes 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.

206 **Initial Expansion** (Lines 11-16). By construction, centers cover all orthologous articulations.
207 Articulation hubs are often present in conserved sub-graphs detected by means of comparative
208 methods. Therefore, assuming that the majority of the remaining nodes belonging to conserved
209 complexes are neighbors of articulation hubs, we add to each center all its neighbouring orthologous
210 proteins, regardless whether they are or not articulations. We perform this step for all centers in
211 parallel.

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

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

216

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

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

225

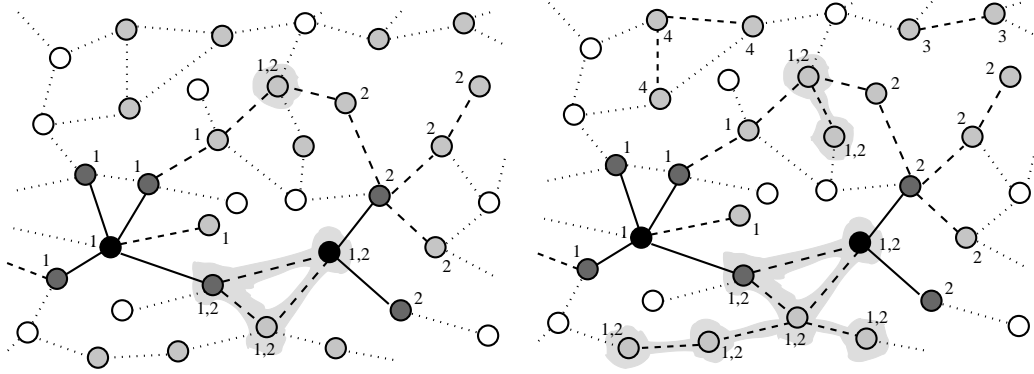


Figure 3: Examples of parallel expansion of trees (left figure) and of the final assigning remaining nodes (right figure). Seeds of centers are solid nodes. Dark gray nodes are the rest of centers connected to a seed by solid edges. Light gray nodes are orthologous proteins which are not articulations.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

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

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

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

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

238

239 **Complexity.** The algorithm divides only orthologs of a given PPI network where the number
 240 of all orthologs is $n = |O|$. It performs a parallel breadth-first search (BFS). In general, BFS
 241 has $O(|V| + |E|)$ complexity, where V and E denote the number of nodes and edges, respectively.
 242 However, the `Divide` algorithm constructs trees considering only orthologous nodes, so the number
 243 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_1 : PPI networks, O : orthologous nodes of G with respect to G_1

Output: S : list of subsets of O

```
1:  $A = \{\text{orthologous articulations of } G\}$ 
2:  $S = \langle \rangle$ 
3: repeat {Construction of centers}
4:    $root = \text{element of } A \text{ with highest degree not already occurring in } S$ 
5:    $s = \{root\} \cup \{\text{neighbors of } root \text{ in } A \text{ not already occurring in } S\}$ 
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:    $s = s \cup \{\text{all neighbors of } s \text{ in } O\}$ 
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:        $s = s \cup \{n\}$ 
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:        $s = s \cup \{r\}$ 
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
```

244 constructed sub-tree. The possible overlap between trees can increase the number of traversed
245 edges and visited vertices. In the worse case all orthologous vertices are visited by each center (all
246 nodes are in the overlap). So, if the number of centers is k , the complexity of `Divide` is $O(kn)$.

247 5. Experimental Analysis

248 The effectiveness and robustness of the proposed pre-processing method is assessed experimen-
249 tally in the following three ways.

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

253 Next, we show that the resulting sub-graphs cover protein complexes computationally pre-
254 dicted by one state-of-the-art alignment algorithm (Koyutürk et al., 2006b), `MaWish`, in order
255 to investigate whether the sub-graphs contain information that could be used to discover new
256 conserved complexes.

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

263 We conduct experiments on the following pairs of organisms:

- 264 • *Saccharomyces cerevisiae* versus *Caenorhabditis elegans* (yeast-nematode),
- 265 • *Saccharomyces cerevisiae* versus *Drosophila melanogaster* (yeast-fly) and
- 266 • *Saccharomyces cerevisiae* versus *Homo sapiens* (yeast-human).

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

¹<http://vorlon.case.edu/~mxk331/software/>

	S. cerevisiae	C. elegans	D. melanogaster	H. sapiens
#proteins	5157	3345	8577	4541
#interactions	18192	5988	28829	7393

Table 1: Protein interaction network properties of yeast, nematode, fly and human.

Pair of species	#orthologous pairs
S. cerevisiae vs C. elegans	2746
S. cerevisiae vs D. melanogaster	15884
S. cerevisiae vs H. sapiens	6690

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

273 5.1. Handling redundant alignments and complexes

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

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

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

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

293 At the node (protein) level **MaWish** does not handle possible occurrence of redundant complexes.
294 Moreover, despite the fact that **MNAligner** performs global alignment, it may produce intersecting
295 complexes when applied to sub-graphs generated by **Divide**, because such sub-graphs may overlap.
296 Therefore, in both instances of modular network alignment here considered, we will have to handle
297 redundant protein complexes.

298 In general if two complexes have a high intersection, one of them is discarded (see, e.g. Sup-
299 porting Methods of Sharan et al. (2005b)). However, this approach for handling redundancy is
300 not very satisfactory, since detected conserved complexes could possibly cover part of a ‘true’
301 functional module either due to constraints on the topology and homology similarity, or due to
302 missing interactome data. Therefore, detected complexes having high overlap may still represent
303 different parts of one bigger module.

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

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

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

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

324 **6. Divide generates sub-graphs covering ”true” protein conserved complexes**

325 Let *Divide sub-graphs* denote the sub-graphs generated by *Divide*. We compared *Divide*
326 sub-graphs with ”true” protein conserved complexes. To this aim, we evaluated the quality of
327 sub-graphs generated by *Divide* using known yeast complexes catalogued in the MIPS database²

²<http://mips.helmholtz-muenchen.de/genre/proj/yeast/>

328 (Güldener et al., 2005). Category 550, which was obtained from high throughput experiments,
 329 is excluded and we retained only manually annotated complexes up to depth 3 in the MIPS tree
 330 category structure as standard of truth for quality assessment. From each of these complexes we
 331 extracted the subset of proteins consisting of only orthologous proteins, where sets with less than
 332 three elements were filtered out. We call the resulting set of proteins *yeast MIPS (conserved)*
 333 *complex*.

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

Alignment task	#sub-graphs	#yeast MIPS complexes	#yeast MaWish complexes
S. cerevisiae vs C. elegans	53 (235)	56, 45 (135)	27 (83)
S. cerevisiae vs D. melanogaster	119 (408)	111, 99 (205)	99 (411)
S. cerevisiae vs H. sapiens	67 (253)	77, 63 (161)	57 (276)

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.

337 The *intersection rate* between a sub-graph and a complex is used, computed as follow. Let
 338 $G = (U, E)$ be a sub-graph and let C be a protein complex of one organism. The intersection rate
 339 of G and C is

$$340 \quad |U \cap C|/|C|.$$

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

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

348 Because *conserved complexes have in general small size*, we incorporated this prior information
 349 in our analysis and filtered out complexes of big size from the list of yeast MIPS complexes, since
 350 they were not considered to be conserved. To this end, we used the conserved complexes predicted
 351 by `MaWish` (see also the next section). For yeast-nematode and yeast-human the biggest yeast
 352 `MaWish` complex has size 12, for the yeast-fly alignment task the biggest `MaWish` complex consists
 353 of 21 proteins. Using these parameter values for the threshold to filter out yeast MIPS complexes
 354 considered too large, we got 45 yeast MIPS complexes w.r.t. nematode, 99 complexes w.r.t. fly

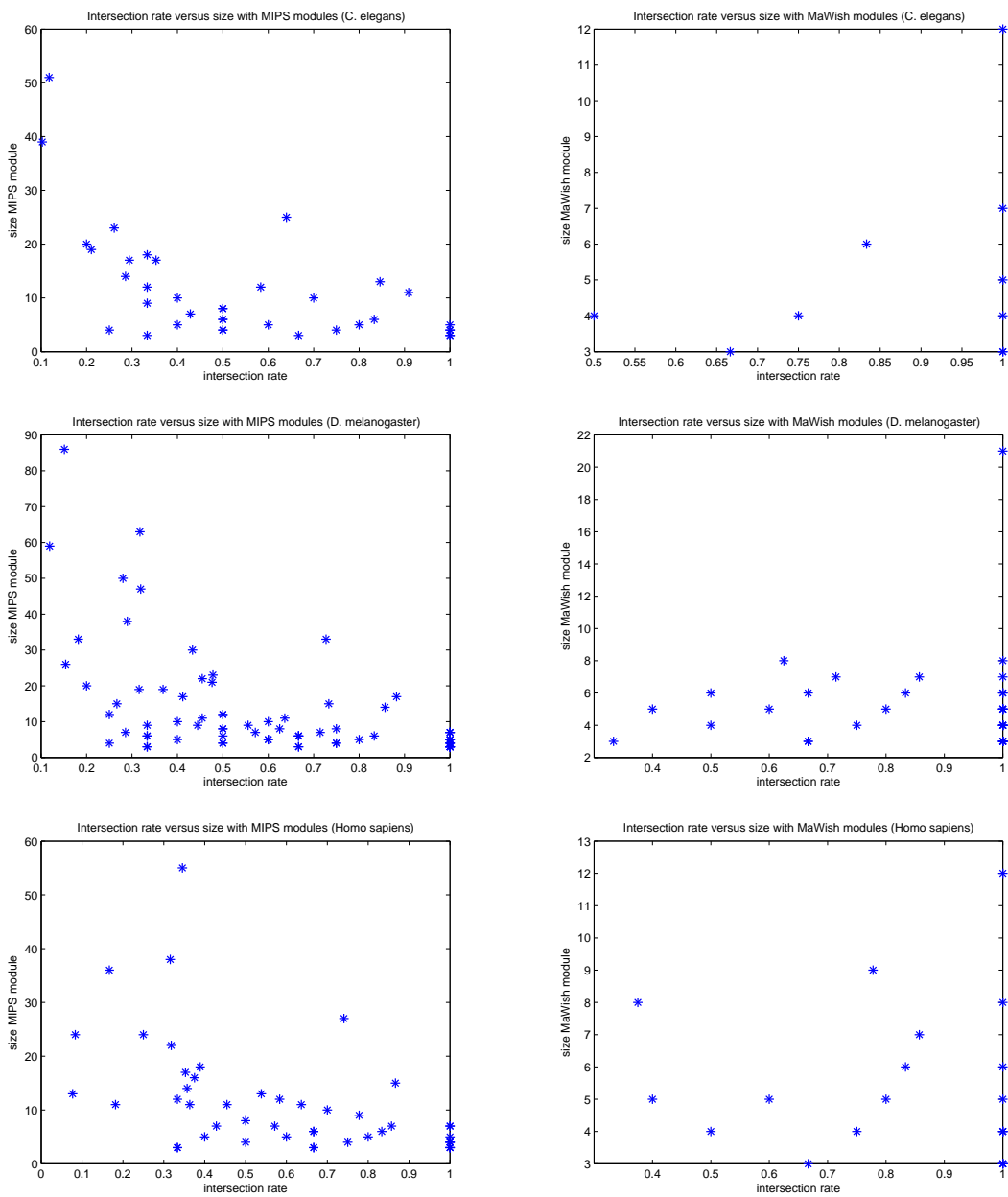


Figure 4: Intersection rate vs. size of yeast complexes for the alignment task. Left column: yeast MIPS complexes. Right column: yeast MaWish complexes.

355 and 63 complexes w.r.t. human (see Table 3). As shown in Table 4 the average intersection rate
 356 increased for the small yeast MIPS complexes while the number of considered complexes does not
 357 decrease significantly.

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

$$364 \quad \left| \bigcup_{G(U,E) \in S} U \cap C \right| / |C|.$$

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

Considered yeast complexes	C. elegans (%)	D. melanogaster (%)	H. sapiens (%)
small-sized (union)	71.4	75.7	81.9
all (union)	64.5	71.3	74.8
small-sized	64	68	69.3
all	56	64.2	63.4

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.

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

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

377 Here we investigate how `Divide` constrains the search process of `MaWish`, and whether the sub-
 378 graphs generated by `Divide` cover those produced by `MaWish`. To this end, we used the conserved
 379 complexes predicted by this alignment method and processed by the clique-rule merging procedure,

380 where complexes consisting of one or two proteins were filtered out. We call the resulting sets
381 *MaWish complexes*.

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

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

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

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.

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

399 8. Applications of Modular Network Alignment

400 In this section we investigate the ability of *Divide* to enhance the performance of alignment
401 methods.

402 Specifically, we apply *Divide* to two different alignment methods.

403 In the first case, we consider an instance of modular local network alignment, called *DivAfull*
404 (Jancura et al., 2008a). *DivAfull* employs *Divide* to generate sub-graphs, the *MaWish* alignment
405 model to align them, and iterative exact search to detect all possible solutions from the generated

406 alignments. Therefore, application of `Divide` allows one to improve the search process by replacing
407 the greedy search procedure of `MaWish` with an exact search algorithm.

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

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

419 8.1. `DivAfull`

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

423 In that model, a weighted alignment graph is constructed from a pair of PPI networks and a
424 similarity score S , which quantifies the likelihood that two proteins are orthologous, is computed.
425 A node in the alignment graph is a pair of orthologous proteins. Each edge in the alignment graph
426 is assigned a weight that is the sum of three scoring terms: for protein duplication, mismatches
427 for possible divergence in function, and match of a conserved pair of orthologous interactions. We
428 refer to Koyutürk et al. (2006b) for a formal description of these terms.

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

432 After merging we search for these sub-graphs. This problem is reduced to the (optimization)
433 problem of finding a maximal induced subgraph. To tackle this problem, the search part of `MaWish`
434 consists of an approximation greedy algorithm based on local search, because the maximum in-
435 duced subgraph problem is NP-complete. This greedy algorithm selects at first one seed which
436 can likely contribute at most to the overall weight of a potential subgraph. The seed is expanded
437 by adding (removing) nodes to (from) the subgraph while the actual subgraph weight increases.

438 In contrast, `DivAfull` applies iteratively an exact optimization algorithm (Wolsey, 1998) for
439 searching relevant alignments (maximum weighted induced sub-graphs) in the alignment graphs
440 produced by merging possible pairs of the sub-graphs, since their size is small. This search
441 algorithm is described in detail below.

442 8.1.1. Search Algorithm

443 First, an exact optimization algorithm for finding the maximum weighted induced subgraph is
444 applied. Then the process is iterated by adding at each iteration the constraint which bounds the
445 weight of the induced subgraph by the weight of the solution found in the previous iteration.

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

$$450 \quad \text{opt} = \max_C f \quad (\text{Opt}P)$$

451 Algorithm 2 illustrates the resulting full-search procedure which uses the above constrained
452 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 $(\text{Opt}P)$
 - 2: $\text{maxweight} = \infty$
 - 3: $C = C + \{\text{opt} < \text{maxweight}\}$
 - 4: **while** $\text{maxweight} > \varepsilon$ **do**
 - 5: solve $(\text{Opt}P)$ 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**
-

453 8.1.2. Handling Redundant Solutions

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

457 8.2. Divide and `MNAligner`

458 `MNAligner` is a general tool for global alignment of molecular networks. It formalizes the
459 problem of finding an optimal mapping between similar nodes of two different networks as an

460 integer quadratic programming optimization problem which is relaxed to quadratic optimization
461 problem (QP). The optimal integer solution is ensured if appropriate sufficient conditions on the
462 objective function are satisfied. However, QP may have an integer solution also if the conditions
463 are not satisfied. We refer to Li et al. (2007) for the detailed description of the objective function
464 and alignment algorithm.

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

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

487 These results are compared with complexes produced when `MCODE` and `MNAligner` are directly
488 applied to the PPI networks induced by orthologs. Specifically, we cluster the (orthologous sub-
489 network of) G_1 and align the resulting clusters with G_2 . We repeat the process by interchanging
490 the role of G_1 and G_2 . We again process the results using the clique-rule merging procedure and
491 removing complexes of size less than 3.

492 In this way, we allow a fair comparison of results when only `MCODE` and `MNAligner` are applied
493 and when `Divide` is introduced prior these steps. We restrict the use of `MCODE` on sub-networks
494 induced by orthologs, because `Divide` divides only the orthologs of a PPI network.

495 *8.3. Evaluation Criteria for Conserved Complexes*

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

501 Therefore, we measure the enrichment of functional annotations of the protein set in a com-
502 plex, as entailed by the gene ontology (GO) annotation (Ashburner et al., 2000), using one of the
503 well-established tools, the Ontologizer³ (Robinson et al., 2004). Ontologizer measures statistical
504 significance of an enrichment and assigns to the complex a p-value for each enriched function. The
505 p-value is corrected for multiple testing by a classic Bonferroni correction procedure. Furthermore
506 Ontologizer also constructs a hierarchical directed acyclic graph (DAG) consisting of all signifi-
507 cantly enriched annotations and all their ancestor annotations up to the root in the whole GO
508 hierarchy. Given a DAG of enrichments, the level of an annotation is equal to the length of the
509 shortest path from the root of GO hierarchy present in the DAG to that annotation.

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

- 511 1. a certain GO annotation is significantly enriched by the proteins in the complex (p-value
512 < 0.05);
- 513 2. at least half of the proteins in the complex has this significant annotation;
- 514 3. the annotation is at least at GO level four from the root in GO hierarchy.

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

520 We validate the accuracy of the predictions, and consequently the quality of a protein complex,
521 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>

522 the corresponding DAG of enrichments, we restrict our validation only to the annotations which
 523 are present in the DAG and are at GO level four or higher. A protein p of the complex having such
 524 annotations is assumed to be not annotated and its functions are predicted. The predictions are
 525 then compared with the annotations of the protein p . The method is repeated for all annotated
 526 proteins in the cluster. In the end, for each protein p we have:

- 527 • A_p : the number of annotated functions for the protein p .
- 528 • P_p : the number of predicted functions for the protein p .
- 529 • O_p : the size of the overlap between the set of annotated functions and the set of predicted
 530 functions for the protein p .

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

$$532 \quad PR(C) = \frac{\sum_{\forall p \in C} O_p}{\sum_{\forall p \in C} P_p},$$

$$533 \quad RC(C) = \frac{\sum_{\forall p \in C} O_p}{\sum_{\forall p \in C} A_p}.$$

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

$$539 \quad FM(C) = \frac{2 \cdot PR(C) \cdot RC(C)}{PR(C) + RC(C)},$$

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

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

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

$$550 \quad FNR = \frac{|B \setminus A|}{|B|},$$

551
$$CVR = \frac{|C_{B \setminus A}|}{|C_B|}.$$

552 The first measure, FNR, computes the ratio of new functions discovered over the set of all
553 functions discovered by the combined method. The latter one computes the ratio between the
554 number of complexes which are predicted to have the new functions and the total number of
555 complexes detected by the combined method.

556 Notice that all measures above defined treat each species separately rather than explicitly
557 evaluating the conservation hypothesis implied by each pair of conserved complexes aligned. Such
558 evaluation could, in principle, be performed by comparing the results to a reference set of conserved
559 modules (Yosef et al., 2008). To date, however, most such references are not comprehensive enough
560 and contain only a small number of cases to learn from (Yosef et al., 2008). One exception is the
561 Biocarta⁴ (Nishimura, 2001) database which contains many human-mouse conserved pathways.

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

566 8.4. A Case Study: *Saccharomyces cerevisiae* vs *Caenorhabditis Elegans*

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

571 8.4.1. Application of `Divide`

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

574 For *Saccharomyces cerevisiae*, 697 articulations, of which 151 orthologs, were computed, and
575 83 centers were constructed from them. Expansion of these centers into centered trees resulted in
576 639 covered orthologs. The algorithm assigned the remaining 153 orthologous proteins to 152 new
577 sub-trees.

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

⁴<http://www.biocarta.com/genes/allPathways.asp>

580 339 covered orthologs. The algorithm assigned the remaining orthologous 294 proteins to 288 new
581 sub-trees.

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

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

587 8.4.2. DivAfull and MaWish

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

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

594 DivAfull discovered 151 solutions (alignments) while MaWish yielded 83 solutions. Between
595 these two set of solutions we found 70 redundant alignments, whose pair of weights are plotted
596 on the left part of Figure 5. Among these, 48 (31.8% of DivAfull results) were equal (red
597 crosses in the diagonal) and 22 (14.6%) different. 8 (5.3%) (green crosses below the diagonal) with
598 better DivAfull alignment weight, and 13 (8.6%) (blue crosses above the diagonal) with better
599 MaWish alignment weight (for 1 (0.7%) pair it was undecidable because of rounding errors during
600 computation).

601 DivAfull found 81 (53.6%) new alignments, that is, not discovered by MaWish. The right plot
602 of Figure 5 shows the binned distribution of weights of these alignments, together with the new 17
603 ones discovered by MaWish but not by DivAfull. There is no significant difference between the
604 overall weight average of the DivAfull (0.8) and the MaWish (0.86) results.

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

609 DivAfull discovered a higher number of protein complexes than MaWish and the same is
610 observed when only those complexes which satisfy the criteria for being a functional predictor are
611 considered. Concretely, for *Saccharomyces cerevisiae* DivAfull found 46 complexes of which 39

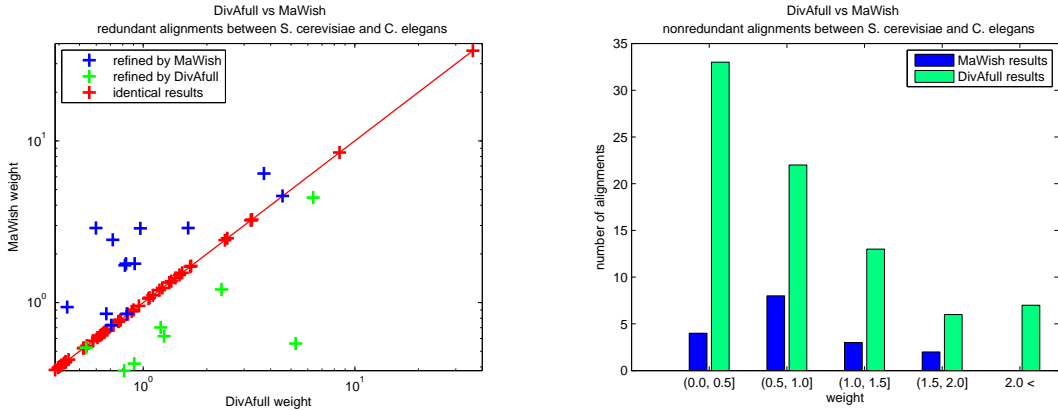


Figure 5: Analysis of all alignments discovered by **MaWish** and **DivAfull**. Left figure: Distribution of pairs of weights for paired redundant alignments, one obtained from **MaWish** and one from **DivAfull**. Weights of alignments found by **DivAfull** are on the x-axis, those found by **MaWish** on the y-axis. '+' is a paired redundant alignment. Right figure: Interval weight distributions of non-redundant alignments discovered by **MaWish** and **DivAfull**. The x-axis shows weight intervals, the y-axis the number of alignments in each interval.

612 are functional predictors, and for *Caenorhabditis elegans* **DivAfull** found 28 complexes of which
 613 18 are functional predictors. In contrast, **MaWish** found 27 complexes of which 24 are potential
 614 predictors for *Saccharomyces cerevisiae* and 24 complexes of which 13 are functional predictors for
 615 *Caenorhabditis elegans*.

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

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

Method	#Modules	Precision ($\pm\delta$)	Recall ($\pm\delta$)	F-measure ($\pm\delta$)
DivAfull	46	0.73 (± 0.33)	0.52 (± 0.30)	0.59 (± 0.31)
MaWish	27	0.75 (± 0.30)	0.52 (± 0.26)	0.60 (± 0.28)
DivAfull	39	0.86 (± 0.13)	0.61 (± 0.22)	0.70 (± 0.20)
MaWish	24	0.84 (± 0.13)	0.58 (± 0.20)	0.67 (± 0.18)

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.

624 For *Caenorhabditis elegans*, when considering all modules, a better average functional enrich-
 625 ment is achieved for **DivAfull** modules (the upper part of Table 7). Considering all functional
 626 predictors, **MaWish** complexes have a higher average precision but a better recall is obtained by

Method	#Modules	Precision ($\pm\delta$)	Recall ($\pm\delta$)	F-measure ($\pm\delta$)
DivAfull	28	0.56 (± 0.44)	0.46 (± 0.40)	0.49 (± 0.40)
MaWish	24	0.50 (± 0.48)	0.38 (± 0.40)	0.41 (± 0.42)
DivAfull	18	0.87 (± 0.12)	0.71 (± 0.25)	0.76 (± 0.21)
MaWish	13	0.93 (± 0.10)	0.70 (± 0.27)	0.76 (± 0.22)

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.

627 DivAfull modules. However, in total, they are of the same quality as shown by the average of
628 F-measures (the bottom part of Table 7).

Species	#Functions	FNR	#Predictors	CVR
yeast	144	0.23	39	0.26
nematode	90	0.06	18	0.17

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.

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

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

635 8.4.3. Divide and MNAAligner

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

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

647 From Table 9 it can be seen that the complexes of *Saccharomyces cerevisiae* discovered when
648 Divide was applied, and their subset of functional predictions, significantly outperformed the

Method	#Modules	Precision ($\pm\delta$)	Recall ($\pm\delta$)	F-measure ($\pm\delta$)
Divide + MNAaligner	12	0.74 (± 0.25)	0.53 (± 0.28)	0.60 (± 0.27)
MNAaligner	13	0.69 (± 0.24)	0.48 (± 0.27)	0.53 (± 0.28)
Divide + MNAaligner	11	0.81 (± 0.09)	0.58 (± 0.24)	0.65 (± 0.21)
MNAaligner	12	0.75 (± 0.12)	0.52 (± 0.24)	0.58 (± 0.24)

Table 9: MNAaligner: 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.

Method	#Modules	Precision ($\pm\delta$)	Recall ($\pm\delta$)	F-measure ($\pm\delta$)
Divide + MNAaligner	10	0.72 (± 0.39)	0.55 (± 0.4)	0.59 (± 0.37)
MNAaligner	11	0.38 (± 0.45)	0.34 (± 0.43)	0.35 (± 0.43)
Divide + MNAaligner	8	0.90 (± 0.11)	0.68 (± 0.31)	0.74 (± 0.23)
MNAaligner	5	0.83 (± 0.15)	0.75 (± 0.29)	0.77 (± 0.24)

Table 10: MNAaligner: 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.

649 complexes and predictions of the straightforward application of MNAaligner (with MCODE).

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

Species	#Functions	FNR	#Predictors	CVR
yeast	109	0.28	11	0.36
nematode	48	0.46	8	0.25

Table 11: The total number of predicted biological functions and their functional ratio and the total number of functional predictors and their coverage ratio as result when Divide is combined with (MCODE and) MNAaligner computed with respect to the results of straightforward application of (MCODE and) MNAaligner.

657 In the end, we computed functional and coverage ratio over all functions and their functional
658 predictions detected with the method which includes Divide with respect to the results of the
659 application of MCODE and MNAaligner. Table 11 indicates that in both species a quarter or even
660 more of the results are new discoveries.

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

666 9. Conclusion

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

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

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

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

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

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

694 Another advantage of applying the **Divide** algorithm for performing modular protein network
695 alignment is that it allows one to parallelise alignment methods. For instance, the full search
696 algorithm **DivAfull** can be run independently on each alignment graph constructed on sub-graphs
697 generated by **Divide**.

698 In future work we intend to employ the `Divide` algorithm for multiple network alignment
699 problem.

700

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

702 **References**

703 Ali, W., Deane, C. M., 2009. Functionally guided alignment of protein interaction networks for
704 module detection. *Bioinformatics* 25 (23), 3166–3173.

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

706 Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P.,
707 Dolinski, K., Dwight, S. S., Eppig, J. T., Harris, M. A., Hill, D. P., Issel-Tarver, L., Kasarskis,
708 A., Lewis, S., Matese, J. C., Richardson, J. E., Ringwald, M., Rubin, G. M., Sherlock, G., 05
709 2000. Gene ontology: tool for the unification of biology. *Nat Genet* 25 (1), 25–29.

710 URL <http://dx.doi.org/10.1038/75556>

711 Bader, G., Hogue, C., 2003. An automated method for finding molecular complexes in large protein
712 interaction networks. *BMC Bioinformatics* 4 (1), 2.

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

714 Bader, G. D., Donaldson, I., Wolting, C., Ouellette, B. F. F., Pawson, T., Hogue, C. W. V.,
715 January 1 2001. Bind—the biomolecular interaction network database. *Nucleic Acids Res* 29 (1),
716 242–245.

717 Bandyopadhyay, S., Sharan, R., Ideker, T., 2006. Systematic identification of functional orthologs
718 based on protein network comparison. *Genome Research* 16 (3), 428–435.

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

720 Berg, J., Lässig, M., 2006. Cross-species analysis of biological networks by Bayesian alignment.
721 *Proceedings of the National Academy of Sciences* 103 (29), 10967–10972.

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

723 Blin, G., Sikora, F., Vialette, S., 2009. Querying protein-protein interaction networks. In: IS-
724 BRA '09: Proceedings of the 5th International Symposium on Bioinformatics Research and
725 Applications. Springer-Verlag, Berlin, Heidelberg, pp. 52–62.

- 726 Bron, C., Kerbosch, J., 1973. Algorithm 457: finding all cliques of an undirected graph. Commun.
727 ACM 16 (9), 575–577.
- 728 Bruckner, S., Hüffner, F., Karp, R. M., Shamir, R., Sharan, R., July 2009. Torque: topology-free
729 querying of protein interaction networks. Nucleic acids research 37 (Web Server issue), W106–
730 108.
731 URL <http://dx.doi.org/10.1093/nar/gkp474>
- 732 Cheng, Q., Berman, P., Harrison, R., Zelikovsky, A., 2008. Fast alignments of metabolic networks.
733 In: BIBM '08: Proceedings of the 2008 IEEE International Conference on Bioinformatics and
734 Biomedicine. IEEE Computer Society, Washington, DC, USA, pp. 147–152.
- 735 Chindelevitch, L., Liao, C.-S., Berger, B., 2010. Local optimization for global alignment of protein
736 interaction networks. Pacific Symposium on Biocomputing 15, 123–132.
- 737 Deng, M., Zhang, K., Mehta, S., Chen, T., Sun, F., 2003. Prediction of protein function using
738 proteinprotein interaction data. Journal of Computational Biology 10 (6), 947–960.
739 URL <http://www.liebertonline.com/doi/abs/10.1089/106652703322756168>
- 740 Dost, B., Shlomi, T., Gupta, N., Ruppín, E., Bafna, V., Sharan, R., 2008. Qnet: A tool for
741 querying protein interaction networks. Journal of Computational Biology 15 (7), 913–925, PMID:
742 18707533.
743 URL <http://www.liebertonline.com/doi/abs/10.1089/cmb.2007.0172>
- 744 Dutkowski, J., Tiuryn, J., 2007. Identification of functional modules from conserved ancestral
745 protein protein interactions. Bioinformatics 23 (13), i149–158.
746 URL <http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/13/i149>
- 747 Ekman, D., Light, S., Björklund, A. K., Elofsson, A., 2006. What properties characterize the
748 hub proteins of the protein-protein interaction network of *saccharomyces cerevisiae*? Genome
749 Biology 7 (6), R45.
- 750 Evans, P., Sandler, T., Ungar, L., 2008. Protein-protein interaction network alignment by quan-
751 titative simulation. In: BIBM '08: Proceedings of the 2008 IEEE International Conference on
752 Bioinformatics and Biomedicine. IEEE Computer Society, Washington, DC, USA, pp. 325–328.
- 753 Flannick, J., Novak, A., Do, C. B., Srinivasan, B. S., Batzoglou, S., 2009. Automatic parameter
754 learning for multiple local network alignment. Journal of Computational Biology 16 (8), 1001–

755 1022, PMID: 19645599.
756 URL <http://www.liebertonline.com/doi/abs/10.1089/cmb.2009.0099>

757 Flannick, J., Novak, A., Srinivasan, B. S., McAdams, H. H., Batzoglou, S., 2006. Graemlin:
758 General and robust alignment of multiple large interaction networks. *Genome Res.* 16 (9), 1169–
759 1181.

760 Güldener, U., Münsterkötter, M., Kastenmüller, G., Strack, N., van Helden, J., Lemer, C.,
761 Richelles, J., Wodak, S. J., García-Martínez, J., Pérez-Ortín, J. E., Michael, H., Kaps, A.,
762 Talla, E., Dujon, B., André, B., Souciet, J. L., De Montigny, J., Bon, E., Gaillardin, C., Mewes,
763 H. W., 2005. CYGD: the Comprehensive Yeast Genome Database. *Nucl. Acids Res.* 33 (suppl_1),
764 D364–368.
765 URL http://nar.oxfordjournals.org/cgi/content/abstract/33/suppl_1/D364

766 Guo, X., Hartemink, A. J., 2009. Domain-oriented edge-based alignment of protein interaction
767 networks. *Bioinformatics* 25 (12), i240–1246.
768 URL <http://bioinformatics.oxfordjournals.org/cgi/content/abstract/25/12/i240>

769 Handl, J., Knowles, J., Kell, D. B., 2005. Computational cluster validation in post-genomic data
770 analysis. *Bioinformatics* 21 (15), 3201–3212.
771 URL <http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/15/3201>

772 Hirsh, E., Sharan, R., 2007. Identification of conserved protein complexes based on a model of
773 protein network evolution. *Bioinformatics* 23 (2), e170–176.
774 URL <http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/2/e170>

775 Hopcroft, J., Tarjan, R., 1973. Algorithm 447: efficient algorithms for graph manipulation. *Com-*
776 *mun. ACM* 16 (6), 372–378.

777 Jancura, P., Heringa, J., Marchiori, E., 2008a. Divide, align and full-search for discovering con-
778 served protein complexes. In: Marchiori, E., Moore, J. H. (Eds.), *EvoBIO*. Vol. 4973 of *Lecture*
779 *Notes in Computer Science*. Springer, pp. 71–82.
780 URL <http://dblp.uni-trier.de/db/conf/evow/evobio2008.html#JancuraHM08>

781 Jancura, P., Heringa, J., Marchiori, E., 2008b. Dividing protein interaction networks by growing or-
782 thologous articulations. In: *PRIB '08: Proceedings of the Third IAPR International Conference*
783 *on Pattern Recognition in Bioinformatics*. Springer-Verlag, Berlin, Heidelberg, pp. 187–200.

784 Jeong, H., Mason, S. P., Barabasi, A.-L., Oltvai, Z. N., 2001. Lethality and centrality in protein
785 networks. *NATURE* v 411, 41.
786 URL <http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0105306>

787 Kalaev, M., Bafna, V., Sharan, R., 2009. Fast and accurate alignment of multiple protein networks.
788 *Journal of Computational Biology* 16 (8), 989–999, PMID: 19624266.
789 URL <http://www.liebertonline.com/doi/abs/10.1089/cmb.2009.0136>

790 Kelley, B. P., Sharan, R., Karp, R. M., Sittler, T., Root, D. E., Stockwell, B. R., Ideker, T.,
791 Sep. 2003. Conserved pathways within bacteria and yeast as revealed by global protein network
792 alignment. *Proceedings of the National Academy of Science* 100, 11394–11399.

793 Klau, G., 2009. A new graph-based method for pairwise global network alignment. *BMC Bioin-*
794 *formatics* 10 (Suppl 1), S59.
795 URL <http://www.biomedcentral.com/1471-2105/10/S1/S59>

796 Koyutürk, M., Grama, A., Szpankowski, W., May 2005. Pairwise local alignment of protein inter-
797 action networks guided by models of evolution. In: *RECOMB*. Vol. 3500 of *Lecture Notes*
798 *in Bioinformatics*. Springer Berlin / Heidelberg, pp. 48–65.

799 Koyutürk, M., Kim, Y., Subramaniam, S., Szpankowski, W., Grama, A., 2006a. Detecting con-
800 served interaction patterns in biological networks. *Journal of Computational Biology* 13 (7),
801 1299–1322, PMID: 17037960.
802 URL <http://www.liebertonline.com/doi/abs/10.1089/cmb.2006.13.1299>

803 Koyutürk, M., Kim, Y., Topkara, U., Subramaniam, S., Grama, A., Szpankowski, W., 2006b.
804 Pairwise alignment of protein interaction networks. *Journal of Computational Biology* 13 (2),
805 182–199.

806 Li, Z., Zhang, S., Wang, Y., Zhang, X.-S., Chen, L., 2007. Alignment of molecular networks by
807 integer quadratic programming. *Bioinformatics* 23 (13), 1631–1639.
808 URL <http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/13/1631>

809 Liang, Z., Xu, M., Teng, M., Niu, L., 2006. Comparison of protein interaction networks reveals
810 species conservation and divergence. *BMC Bioinformatics* 7 (1), 457.
811 URL <http://www.biomedcentral.com/1471-2105/7/457>

812 Liao, C.-S., Lu, K., Baym, M., Singh, R., Berger, B., 2009. IsoRankN: spectral methods for global
813 alignment of multiple protein networks. *Bioinformatics* 25 (12), i253–258.
814 URL <http://bioinformatics.oxfordjournals.org/cgi/content/abstract/25/12/i253>

815 Maslov, S., Sneppen, K., 2002. Specificity and stability in topology of protein networks. *Science*
816 296, 910–913.
817 URL <http://www.citebase.org/abstract?id=oai:arXiv.org:cond-mat/0205380>

818 Narayanan, M., Karp, R. M., 2007. Comparing protein interaction networks via a graph match-
819 and-split algorithm. *Journal of Computational Biology* 14 (7), 892–907, PMID: 17803369.
820 URL <http://www.liebertonline.com/doi/abs/10.1089/cmb.2007.0025>

821 Nishimura, D., 2001. *Biocarta. Biotech Software & Internet Report* 2 (3), 117–120.
822 URL <http://www.liebertonline.com/doi/abs/10.1089/152791601750294344>

823 Pinter, R. Y., Rokhlenko, O., Yeger-Lotem, E., Ziv-Ukelson, M., 2005. Alignment of metabolic
824 pathways. *Bioinformatics* 21 (16), 3401–3408.
825 URL <http://bioinformatics.oxfordjournals.org/cgi/content/abstract/21/16/3401>

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

828 Pržulj, N., Wigle, D., Jurisica, I., 2004. Functional topology in a network of protein interactions.
829 *Bioinformatics* 20 (3), 340–384.

830 Qian, X., Sze, S.-H., Yoon, B.-J., FEB 2009. Querying Pathways in Protein Interaction Networks
831 Based on Hidden Markov Models. *Journal of Computational Biology* 16 (2), 145–157.

832 Rathod, A. J., Fukami, C., 2005. Mathematical properties of networks of protein interactions,
833 cS374 Fall 2005 Lecture 9, Computer Science Department, Stanford University.

834 Rijsbergen, C. J. V., 1979. *Information Retrieval*, 2nd Edition. Butterworth-Heinemann.

835 Robinson, P. N., Wollstein, A., Bohme, U., Beattie, B., 2004. Ontologizing gene-expression mi-
836 croarray data: characterizing clusters with Gene Ontology. *Bioinformatics* 20 (6), 979–981.
837 URL <http://bioinformatics.oxfordjournals.org/cgi/content/abstract/20/6/979>

838 Sharan, R., Ideker, T., April 2006. Modeling cellular machinery through biological network com-
839 parison. *Nature Biotechnology* 24 (4), 427–433.

840 Sharan, R., Ideker, T., Kelley, B. P., Shamir, R., Karp, R. M., 2004. Identification of protein
841 complexes by comparative analysis of yeast and bacterial protein interaction data. In: RECOMB
842 '04: Proceedings of the eighth annual international conference on Research in computational
843 molecular biology. ACM, New York, NY, USA, pp. 282–289.

844 Sharan, R., Ideker, T., Kelley, B. P., Shamir, R., Karp, R. M., 2005a. Identification of protein
845 complexes by comparative analysis of yeast and bacterial protein interaction data. *Journal of*
846 *Computational Biology* 12 (6), 835–846.

847 Sharan, R., Suthram, S., Kelley, R. M., Kuhn, T., McCuine, S., Uetz, P., Sittler, T., Karp, R. M.,
848 Ideker, T., 2005b. From the Cover: Conserved patterns of protein interaction in multiple species.
849 *Proceedings of the National Academy of Sciences* 102 (6), 1974–1979.
850 URL <http://www.pnas.org/cgi/content/abstract/102/6/1974>

851 Shlomi, T., Segal, D., Ruppin, E., Sharan, R., 2006. Qpath: a method for querying pathways in a
852 protein-protein interaction network. *BMC Bioinformatics* 7 (1), 199.
853 URL <http://www.biomedcentral.com/1471-2105/7/199>

854 Singh, R., Xu, J., Berger, B., 2007. Pairwise global alignment of protein interaction networks by
855 matching neighborhood topology. pp. 16–31.
856 URL http://dx.doi.org/10.1007/978-3-540-71681-5_2

857 Singh, R., Xu, J., Berger, B., 2008a. Global alignment of multiple protein interaction networks.
858 *Pacific Symposium on Biocomputing* 13, 303–314.

859 Singh, R., Xu, J., Berger, B., 2008b. Global alignment of multiple protein interaction networks
860 with application to functional orthology detection. *Proceedings of the National Academy of*
861 *Sciences* 105 (35), 12763–12768.
862 URL <http://www.pnas.org/content/105/35/12763.abstract>

863 Srinivasan, B. S., Shah, N. H., Flannick, J., Abeliuk, E., Novak, A., Batzoglou, S., 2007. Current
864 Progress in Network Research: toward Reference Networks for kKey Model Organisms. *Brief.*
865 *in BioinformaticsAdvance* access.

866 Tarjan, R., 1972. Depth-first search and linear graph algorithms. *SIAM Journal on Computing*
867 1 (2), 146–160.

868 Tian, W., Samatova, N. F., 2009. Pairwise alignment of interaction networks by fast identification
869 of maximal conserved patterns. *Pacific Symposium on Biocomputing* 14, 99–110.

- 870 Ucar, D., Asur, S., Catalyurek, U., Parthasarathy, S., September 18-22 2006. Improving func-
871 tional modularity in protein-protein interactions graphs using hub-induced subgraphs. In: 10th
872 European Conference on Principle and Practice of Knowledge Discovery in Database (PKDD).
873 Berlin, Germany.
- 874 Wernicke, S., Rasche, F., 2007. Simple and fast alignment of metabolic pathways by exploiting
875 local diversity. *Bioinformatics* 23 (15), 1978–1985.
876 URL <http://bioinformatics.oxfordjournals.org/cgi/content/abstract/23/15/1978>
- 877 Wolsey, L. A., September 9 1998. *Integer Programming*, 1st Edition. Wiley-Interscience.
- 878 Xenarios, I., Salwinski, L., Duan, X. J., Higney, P., Kim, S.-M., Eisenberg, D., January 1 2002.
879 Dip, the database of interacting proteins: a research tool for studying cellular networks of
880 protein interactions. *Nucleic Acids Research* 30 (1), 303–305.
- 881 Yang, Q., Sze, S.-H., 2007. Path matching and graph matching in biological networks. *Journal of*
882 *Computational Biology* 14 (1), 56–67, PMID: 17381346.
883 URL <http://www.liebertonline.com/doi/abs/10.1089/cmb.2006.0076>
- 884 Yon Rhee, S., Wood, V., Dolinski, K., Draghici, S., 07 2008. Use and misuse of the gene ontology
885 annotations. *Nat Rev Genet* 9 (7), 509–515.
886 URL <http://dx.doi.org/10.1038/nrg2363>
- 887 Yosef, N., Ruppin, E., Sharan, R., 2008. Protein-protein interactions and networks: identifica-
888 tion, computer analysis, and prediction. Springer, Ch. Cross-species analysis of protein-protein
889 interaction networks, pp. 163–186.
- 890 Zaslavskiy, M., Bach, F., Vert, J.-P., 2009. Global alignment of protein-protein interaction net-
891 works by graph matching methods. *Bioinformatics* 25 (12), i259–i267.
892 URL <http://bioinformatics.oxfordjournals.org/cgi/content/abstract/25/12/i259>