BIOINFORMATICS ORIGINAL PAPER
Vol. 30 no. 4 2014, pages 531–539
doi:10.1093/bioinformatics/btt713

Systems biology
Advance Access publication December 11, 2013

BEAMS: backbone extraction and merge strategy for the global many-to-many alignment of multiple PPI networks

Ferhat Alkan* and Cesim Erten
Department of Computer Engineering, Kadir Has University, Cibali, Istanbul 34083, Turkey

Associate Editor: Mario Albrecht

ABSTRACT

Motivation: Global many-to-many alignment of biological networks has been a central problem in comparative biological network studies. Given a set of biological interaction networks, the informal goal is to group together related nodes. For the case of protein–protein interaction networks, such groups are expected to form clusters of functionally orthologous proteins. Construction of such clusters for networks from different species may prove useful in determining evolutionary relationships, in predicting the functions of proteins with unknown functions and in verifying those with estimated functions.

Results: A central informal objective in constructing clusters of orthologous proteins is to guarantee that each cluster is composed of members with high homological similarity, usually determined via sequence similarities, and that the interactions of the proteins involved in the same cluster are conserved across the input networks. We provide a formal definition of the global many-to-many alignment of multiple protein–protein interaction networks that captures this informal objective. We show the computational intractability of the suggested definition. We provide a heuristic method based on backbone extraction and merge strategy (BEAMS) for the problem. We finally show, through experiments based on biological significance tests, that the proposed BEAMS algorithm performs better than the state-of-the-art approaches. Furthermore, the computational burden of the BEAMS algorithm in terms of execution speed and memory requirements is more reasonable than the competing algorithms.

Availability and implementation: Supplementary material including code implementations in LEDA C++, experimental data and the results are available at http://webprs.khas.edu.tr/~cesim/BEAMS.tar.gz.

Contacts: ferhat.alkan@stu.khas.edu.tr

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on July 11, 2013; revised on December 3, 2013; accepted on December 4, 2013

1 INTRODUCTION

Proteins and their interactions are at the core of almost every biological process. In protein–protein interaction (PPI) networks, nodes represent the proteins and the edges correspond to interactions between pairs of proteins. Several high-throughput techniques together with novel computational methods gave rise to extraction of large-scale PPI networks for many organisms in recent years (Aebersold and Mann, 2003; Finley and Brent, 1994; Goh and Cohen, 2002; Marcotte *et al.*, 1999; Skrabanek *et al.*, 2008). Parallel to this enormous growth in data, several problem formulations related to the analysis of such networks have been proposed and many computational methods have been developed for their comparative studies. In particular, biological network alignment problem has been of particular interest. The main motivation behind the problem is to detect functionally orthologous proteins across given networks from several organisms.

Two types of biological network alignments have been covered in literature: *local network alignments* and *global network alignments*. The former aims to extract local network motifs (subnetworks) from input networks; the motifs are expected to bear reasonable similarity both in terms of sequence and local network topologies (Flannick *et al.*, 2006; Kalaev *et al.*, 2009; Kelley *et al.*, 2004). Global network alignment on the other hand treats the problem globally and aims to find functionally orthologous mappings across all networks and proteins. Some of the proposed global alignment algorithms such as MI-GRAAL (Kuchaiev and Pržulj, 2011) and SPINAL (Aladağ and Erten, 2013) perform these alignments only for pairwise networks, whereas others such as IsoRank (Singh *et al.*, 2008) and IsoRankN (Liao *et al.*, 2009) perform alignments on multiple networks. Additionally, global alignment algorithms may also differ with respect to the types of mappings they provide. *One-to-one alignment* approaches aim to generate alignments where the output alignment either maps a protein in a network to exactly one protein from one of the networks or leaves the protein unmapped (Aladağ and Erten, 2013; Chindelevitch *et al.*, 2010; Singh *et al.*, 2008). *One-to-many alignments* have been proposed for the global alignment of other biological networks including metabolic pathways, where each metabolic reaction in a pathway is mapped to a subset of reactions from another pathway (Abaka *et al.*, 2013; Ay *et al.*, 2011). Finally, for *many-to-many alignments*, the goal is to extract clusters of proteins where each cluster may include any number of proteins from the input networks (Flannick *et al.*, 2009; Liao *et al.*, 2009; Sahraeian and Yoon, 2013). The proteins mapped to the same cluster as a result of the alignment are all expected to compose a functionally orthologous group. Among all three versions of the global network alignments, the many-to-many version is the most general. Furthermore, as far as constraints from evolutionary molecular biology are concerned, it provides a more intuitive definition; the evolutionary distance between organisms under study may have large variations, leading to different numbers of proteins functioning similarly when considered in different networks.

The focus of this article is on global many-to-many alignment of multiple PPI networks from different species. We first provide

*To whom correspondence should be addressed.

a formal combinatorial definition of the problem. We proceed with proving its computational intractability even in a restricted case. We next provide a general framework for the problem, where we decompose the original problem into two subproblems, that of *backbone extraction* and *backbone merging*. Informally, each backbone in this framework corresponds to a closely related central group of proteins, at most one from each network. Once all the backbones are determined, the latter subproblem involves merging together the backbones with higher chances of coexistence in a cluster of orthologous proteins. We provide heuristic methods for both subproblems that together form our proposed algorithm based on backbone extraction and merge strategy, *BEAMS*. We experimentally evaluate the algorithm with regards to several biological significance metrics proposed in literature and compare it against one of the most popular global many-to-many alignment methods, IsoRankN, and a recently proposed state-of-the-art alignment algorithm, SMETANA. The experimental results indicate that BEAMS alignments on real network data provide more consistent clusters than those of IsoRankN and SMETANA. Furthermore, considering the heavy computational load of the problem, the exceptional running time and memory requirements of BEAMS is a further improvement resulting from the provided framework and the algorithm.

2 METHODS AND ALGORITHMS

2.1 Problem definition

Although the one-to-one version of the problem has been formally defined in previous work, no formal combinatorial definition exists for the global many-to-many version of the interaction network alignment problem apart from parameter learning-based definitions of Graemlin 2.0 (Flannick *et al.*, 2009), which actually is defined as an intermediate subproblem for local alignments. We first provide a formally defined optimization goal for the problem that captures the essence of the informal definition provided in Section 1. The definition is based on an intuitive generalization of the global one-to-one network alignment problem definition provided in Singh *et al.* (2008) and Aladağ and Erten (2013).

Let $G_1(V_1, E_1)$, $G_2(V_2, E_2)$, $\ldots$, $G_k(V_k, E_k)$ be the input PPI networks, where G_i corresponds to the *i*th PPI network and V_i, E_i denote, respectively, the node set (proteins) and the edge set (interactions) of G_i. Let *S* indicate the edge-weighted complete *k*–partite *similarity graph* where the *i*th partition of *S* is V_i and each edge (*u*,*v*) in *S* is assigned a positive real weight *w*(*u*,*v*). This weight corresponds to the *sequence similarity score* *s*(*u*,*v*) between *u* and *v*, usually assumed to be the Basic Local Alignment Search Tool (BLAST) bit score of *u* and *v*, where $u \in G_i$, $v \in G_j$ and $i \neq j$. Let S_β be a subgraph of *S* with the same set of nodes. S_β represents a filtered version of the similarity graph *S*, so that only edges between pairs of proteins with relatively high sequence similarity are retained. For a fixed S_β, the *global many-to-many alignment* of all the input PPI networks is the problem of finding a *maximal* set of non-overlapping *clusters* $\mathcal{CL} = \{Cl_1, Cl_2, \ldots, Cl_m\}$ that maximizes the following score:

$$AS(\mathcal{CL}) = \alpha \times CIQ(\mathcal{CL}) + (1-\alpha) \times \frac{\sum_{\forall Cl_i \in \mathcal{CL}} ICQ(Cl_i)}{|\mathcal{CL}|} \tag{1}$$

Here α is a real number between 0 and 1. It is a balancing parameter that determines the contribution weight of network topology as compared with homological similarity in the construction of output alignments. Each cluster Cl_i is defined to be a complete *c*–partite subgraph of S_β, where $1 < c \leq k$. A set of clusters $\mathcal{CL}$ is maximal if no additional clusters can be added to $\mathcal{CL}$, i.e. no further complete *c*–partite subgraph remains in S_β. Maximizing the *AS* score does not automatically guarantee the maximality of the output set of clusters.

$CIQ(\mathcal{CL})$ in the equation denotes *cluster interaction quality* and is a measure of interaction conservation between all cluster pairs in $\mathcal{CL}$. We define a *conservation score*, denoted with *cs*(*m*,*n*), for each pair of clusters Cl_m, Cl_n. Let E_{Cl_m, Cl_n} denote the set of all PPI edges with endpoints in distinct clusters Cl_m, Cl_n. The score *cs*(*m*,*n*) is trivially 0, if $E_{Cl_m, Cl_n} = \emptyset$. Let $s_{m,n}$ denote the number of PPI networks shared by the nodes in both Cl_m, Cl_n, and let $s'_{m,n}$ be the number of distinct PPI networks containing the edges in E_{Cl_m, Cl_n}. We assign $cs(m,n) = 0$ if $s'_{m,n} = 1$ and $cs(m,n) = s'_{m,n}/s_{m,n}$ otherwise. The former assignment reflects the fact that there is no interaction conservation between the pair of clusters. The overall assignment is a generalization of edge conservation definition of pairwise network alignments. For pairwise alignments, edge conservation is assigned a binary value, i.e. a PPI edge in one network is either conserved in the other network or not. However, for multiple alignments the used definition may assign rational conservation values; see Figure 1. We formally define $CIQ(\mathcal{CL})$ as follows:

$$CIQ(\mathcal{CL}) = \frac{\sum_{\forall Cl_m, Cl_n} |E_{Cl_m, Cl_n}| \times cs(m,n)}{\sum_{\forall Cl_m, Cl_n} |E_{Cl_m, Cl_n}|} \tag{2}$$

In Equation (1), $ICQ(Cl_i)$ stands for the *internal cluster quality* of a given cluster Cl_i and is a measure of sequence similarities of involved proteins. Let $w_{max}(u)$ denote the maximum weight of any edge incident on *u* in S_β. Denote the S_β edges incident on nodes in Cl_i with $E(Cl_i)$. $ICQ(Cl_i)$ is defined as follows:

$$ICQ(Cl_i) = \frac{\sum_{\forall (u,v) \in E(Cl_i)} \sqrt{\frac{w(u,v)^2}{w_{max}(u) \times w_{max}(v)}}}{|E(Cl_i)|} \tag{3}$$

2.2 The BEAMS algorithm

We first show that for a fixed S_β, the global many-to-many network alignment problem is computationally intractable. Owing to space considerations we leave the proof to the Supplementary Document.

Proposition 2.1. *For all $\alpha \neq 0$, the global many-to-many alignment problem is NP-hard even for the restricted case where two PPI networks are aligned and all edge weights in S_β are equal.*

Considering this NP-hardness result, it is necessary to devise efficient heuristics for the problem. The general approach of the BEAMS algorithm can be described within the seed-and-extend framework. Several previous network alignment studies are also based on the same broad framework (Kuchaiev and Pržulj, 2011; Shih and Parthasarathy, 2012). However, how the seeds are

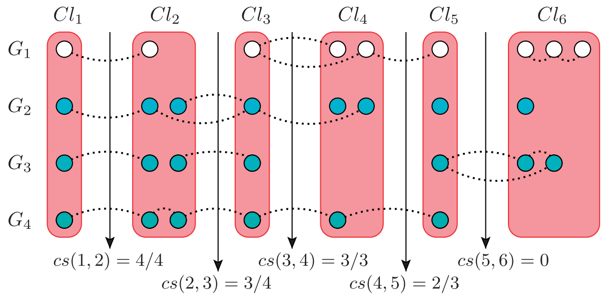

Fig. 1. Conservation scores on a sample alignment covering all notable cases. Rectangular groups represent the clusters of the alignment. The dotted edges represent the protein–protein interactions. Proteins of each PPI network are drawn at separate horizontal layers. The *CIQ* score for this alignment is $(4 \times 4/4 + 4 \times 3/4 + 4 \times 3/3 + 2 \times 2/3 + 0)/16 = 0.771$. Because no other PPI edges exist between any other pair of clusters, only the indicated *cs* scores contribute to *CIQ*

defined formally, how they are extracted and the formal definition of the extension that altogether constitute the main components of a seed-and-extend framework are the main novelties of our approach. Regarding the cluster definition of Equation (1), we make the following observation. Each cluster Cl_i, which is a complete c–partite graph, can be subdivided into a set of n_i disjoint cliques, where n_i denotes the size of the maximum partition of Cl_i. In fact, n_i is the minimum possible size for such a set and each clique in the set has size c' where $1 \leq c' \leq c$. Therefore, we view the original alignment problem of being composed of two subproblems: *backbone extraction* and *backbone merging*. A *backbone* is defined as a clique in S_β, and a set of appropriate backbones together form a cluster. Each backbone thus defined formally, can be considered to correspond to a seed within the general seed-and-extend framework. The first subproblem is that of extracting a *minimal* set of disjoint cliques from S_β, which covers S_β completely and that maximizes the alignment score AS when each non-trivial clique of size greater than one is considered a cluster in the definition of Equation (1). The set is minimal in the sense that no output pair of cliques can be merged together to form a larger clique. Informally, each backbone corresponds to an orthologous set of proteins with at most one protein from each of the input networks. Thus, the backbone extraction problem can actually be viewed as the global one-to-one alignment of multiple networks. A group of backbones is called *mergeable* if their union provides a valid cluster, i.e. a complete c–partite graph. We define the second subproblem as finding a minimal set of mergeable backbone groups such that no further mergeable group remains and that maximizes the resulting AS score when each mergeable backbone group is considered a cluster in the definition of Equation (1). A mergeable group represents a cluster of proteins that are highly homologous, as every pair of proteins from different networks are connected by large weight edges in the filtered similarity graph S_β. Thus, imposing the constraint that no further merging can be done on the set implies the intuition that no two pairwise homologous clusters should be part of the output alignment separately. We show that even these subproblems are computationally hard, and we provide efficient heuristics for each one. In what follows, we first present the details of S_β construction then proceed to provide descriptions of the two main steps of the BEAMS algorithm.

2.2.1 Construction of S_β Considering the sizes of the networks under consideration and the fact that multiple networks constitute the study subject, a suitable filtration on the complete sequence similarity graph S is necessary for mainly two reasons. First, even the suboptimal polynomial-time heuristic algorithms require large amounts of computational power as the size of S increases. Furthermore, taking into account the complete graph S may lead to incorrect alignments as far as biological significance measures are concerned; most protein pairs from different networks do not bear sufficient significance in terms of sequence similarity and using an alignment with the unfiltered similarity graph S may align proteins with almost no homology. As the evolutionary distance between pairs of input networks might be different, we use a relative filtration that takes into account the relative differences in sequence similarities of pairs of networks. For a user-defined threshold β, we construct the filtered similarity graph S_β, so that each edge (u,v) is removed from S if $w(u,v) < \beta \times max(u,v)$, where $max(u,v)$ denotes the maximum of $w(u,v')$ or $w(u',v)$ for any u', v' from the networks of u and v, respectively.

Algorithm 1: EXTRACT_BACKBONES

```
Input: $S_\beta, G_1, G_2, \ldots, G_k, \alpha$
Output: Set of backbones $B = \{B_1, B_2, \ldots, B_n\}$
$B = \emptyset$; $C = \emptyset$
//Initial candidate
$C_0 = MEWC(S_\beta)$;  $C = C \cup \{C_0\}$
repeat
    $B_{new} = Select\_Cand(C, B)$;  $B = B \cup \{B_{new}\}$
    Remove $B_{new}$ from $S_\beta$
    //Generate new candidate
    $C_{new} = Generate\_Cand(S_\beta, B_{new})$;  $C = C \cup \{C_{new}\}$
    //Update each candidate in C
    for all $C_i \in C$ do
        if $C_i \cap B_{new} \neq \emptyset$ then
            if $i == 0$ then
                $C_0 = MEWC(S_\beta)$
            else
                $C_i = Generate\_Cand(S_\beta, B_i)$
            end if
        end if
    end for
until $S_\beta$ contains only isolated nodes
//Each isolated node is a backbone itself
for all nodes $u \in S_\beta$ do
    $B_{new} = \{u\}$;  $B = B \cup \{B_{new}\}$
end for
```

2.2.2 Backbone extraction Regarding the first subproblem defined within the BEAMS framework, we show that the backbone extraction problem is NP-hard even for a restricted case. The full proof can be found in the Supplementary Document.

PROPOSITION 2.2. *For all values of $\alpha \neq 0$, the backbone extraction problem is NP-hard even for the case where there are two input networks and all edge weights in S_β are equal.*

Because the backbone extraction problem is NP-hard, we devise an iterative greedy heuristic that runs in polynomial time assuming the number of networks under consideration is constant. The pseudo-code is shown in Algorithm 1.

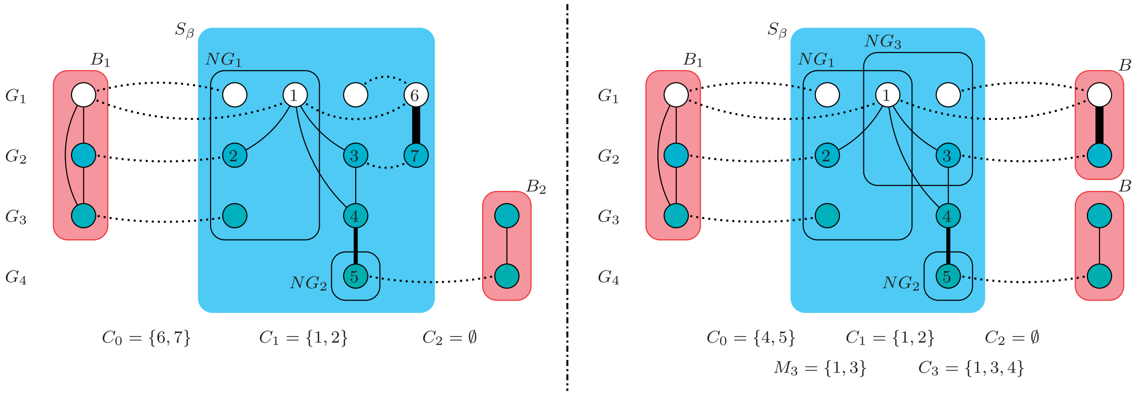

Fig. 2. The state of S_β, the backbones and the candidates on a sample input before (left) and after (right) the third iteration of the main *repeat* loop of **Algorithm 1**. The dotted edges represent protein interactions. Each network is drawn at a separate horizontal layer. Edges between different layers represent S_β edges. *Left:* assuming the *AS* score of C_0 when considered with existing backbones B_1,B_2 is greater than the corresponding score of C_1, the candidate C_0 becomes the newly generated backbone B_3. *Right:* B_3 is removed from S_β. To generate the new candidate C_3, first the neighborhood graph NG_3 of B_3 is constructed. The MEWC M_3 of NG_3 is computed, and the G-MEWC of M_3 in S_β becomes the new candidate C_3. Finally the candidate C_0, which is the only candidate sharing nodes with B_3 is generated anew. Assuming the MEWC of S_β is the edge (4,5), it becomes the updated candidate C_0

Our algorithm uses concepts related to *maximum edge weighted cliques* (MEWC), candidate generation based on neighborhood graph constructions and a greedy selection heuristic aiming to optimize the *AS* score. In the MEWC problem, the input graph is assumed to be edge-weighted with non-negative real values as weights, and the goal is to find a clique with maximum sum of edge weights.

We start with an empty backbone set and a candidate set that consists only of C_0, which is the MEWC of S_β. The *j*th iteration of the main loop of the algorithm consists of four main steps: selecting a new backbone B_j among already existing *j* candidates, removing the backbone from S_β, generating the new candidate C_j and finally updating all existing candidates. Figure 2 provides a depiction of each of these main steps on a sample instance for the third iteration. The first step simply involves selecting the new backbone as the candidate providing the maximum *AS* score when considered together with all existing backbones. In the first iteration, C_0 is selected trivially as the first backbone, B_1. Each candidate C_j is defined with respect to an already existing backbone B_j other than the special candidate C_0, which is updated throughout iterations as S_β is updated. To generate a new candidate C_j via the function call *Generate_Cand*(S_β, B_j), we first construct the *neighborhood graph* of B_j, which is the induced subgraph in S_β of the set of PPI neighbors of all the nodes in B_j. If the neighborhood graph does not contain any S_β edges, then the candidate C_j is empty. Otherwise, we find the MEWC, M_j, of this neighborhood graph, and we generate C_j by constructing the *G-MEWC* of M_j in S_β. Here, G-MEWC corresponds to *generalized* MEWC, which is defined as the maximum edge weighted clique in S_β that is required to include all the nodes of M_j. On top of the interaction conservation advantages brought by neighborhood graphs, constructing the MEWC of the neighborhood graph guarantees a highly similar backbone candidate as far as homological sequence similarities represented by S_β edges are concerned. The G-MEWC construction on the other hand is a precautionary measure to enable possible extensions of a candidate toward networks other than those of its respective backbone. As the last step within an iteration, we generate each candidate anew, again with respect to its corresponding backbone and the updated S_β, if it shares any nodes with the new backbone B_j. The iterations continue until S_β contains only isolated nodes, i.e. those of degree zero.

2.2.3 Computing generalized MEWC We use a depth-first branch-and-bound type algorithm to find the generalized maximum edge weighted clique of S_β that is required to contain a given set of nodes, M_j. The descriptions provided here assume basic familiarity with the general branch-and-bound framework; see Korf (2010) for further details on this framework. Assigning $M_j = \emptyset$, the problem reduces to that of finding the MEWC. As is the case with usual branch-and-bound type algorithms, we traverse the search tree $\mathcal{T}$ in a depth-first manner. Each node at level-*i* of $\mathcal{T}$ represents a clique of size $i + |M_j|$ in S_β that must include nodes in M_j. During the traversal, for each traversed node $\eta = \{u_1, \ldots, u_{i+|M_j|}\}$ of $\mathcal{T}$ representing clique containing nodes $u_1, \ldots, u_{i+|M_j|}$, we store the neighborhood set of η, denoted with N_η that contains nodes that are in the common S_β neighborhood of nodes $u_1, \ldots, u_{i+|M_j|}$. The total edge weight of η is denoted with $EW(\eta)$. Let $Rep(N_\eta)$ denote the set of partition numbers of S_β (the set of PPI networks) that has a node in the set N_η. Throughout the traversal, we store the best node of the search, denoted with $best_\eta$ and its weight with $EW(best_\eta)$. To complete the description of the algorithm, we need only to specify the rules for branching and the bound formulation of the search. An upper bound for the potential weight of a node η in $\mathcal{T}$ is assigned to $EW(\eta) + \sum_{\forall u_t \in \eta} \sum_{\forall r \in Rep(N_\eta)} w_{max}(u_t, r) + PW_{max}(N_\eta)$, where $w_{max}(u_t, r)$ denotes the weight of the maximum weighted edge between u_t and any node in the *r*-th partition of S_β, and $PW_{max}(N_\eta)$ represents the maximum potential weight of a possible clique in N_η. Formally, $PW_{max}(N_\eta)$ is defined as the sum of the edge weights of the $\frac{|Rep(N_\eta)| \times (|Rep(N_\eta)| - 1)}{2}$ heaviest edges of S_β. If the defined potential weight of a node η is greater than $EW(best_\eta)$, we branch at node η, which implies creating a new node η' at the next level $i + 1$, where $\eta' = \{u_1, \ldots, u_{i+|M_j|}, u_{i+|M_j|+1}\}$ such that $u_{i+|M_j|+1} \in N_\eta$.

2.2.4 Backbone merging With regards to the second main step of the BEAMS algorithm, we first state the following proposition about the computational complexity of the corresponding problem. The full proof can be found in the Supplementary Document.

PROPOSITION 2.3. *For all values of $\alpha \neq 0$, the backbone merging problem is NP-hard even for the case where there are two input networks and all edge weights in S_β are equal.*

We provide an iterative greedy heuristic for the backbone merging step. Let MB denote the set of mergeable backbone groups. Initially MB contains all backbones provided by the first backbone extraction step. It is updated at every iteration of the algorithm by a greedy selection strategy which, similar to the backbone extraction step, uses a candidate generation and selection idea. At each iteration, we construct all pairs of mergeable groups in MB that all together provide the set of all candidates of that iteration. For each candidate, we compute the AS score of MB considering the candidate pair as a single group. Some groups in MB may consist of a single node. Such groups are excluded from the AS score computations. We then select the candidate that provides the maximum score and update MB by merging the pair. The algorithm stops when no mergeable pair remains that provides a minimal set MB. We finally remove groups with a single node and provide the resulting set as the output set of clusters. A full discussion of several implementation details regarding this step and the algorithm as a whole are left to the Supplementary Document.

3 DISCUSSION OF RESULTS

We implemented the BEAMS algorithm in C++ using the LEDA library (Mehlhorn and Naher, 1999). The complete source code, evaluation tools, all the data and output results are available as part of the Supplementary Material. Two algorithms we compare BEAMS against are IsoRankN and SMETANA. IsoRankN is one of the most popular algorithms in the global many-to-many network alignment literature. It has been shown that compared with other popular alignment algorithms, such as Graemlin 2.0, NetworkBLAST-M and MI-GRAAL, it provides better performance under measures suitable for network alignment quality determination (Liao *et al.*, 2009; Sahraeian and Yoon, 2012). Furthermore, the informal optimization goals of both IsoRankN and the BEAMS algorithms are similar in the sense that they both aim at maximizing a suitable optimization scoring function that balances the contribution of homological similarities of clustered proteins and the edge conservation between pairs of clusters via a suitably assigned constant α. Therefore, IsoRankN is one of the algorithms that we extensively compare the BEAMS algorithm against. A second alignment algorithm that we use in our experimental evaluations is SMETANA (Sahraeian and Yoon, 2013), a recent approach proposed for probabilistic many-to-many alignment of multiple networks. We present the experimental results of BEAMS and IsoRankN for different values of α varying from 0.3 to 0.7 in the increments of 0.1. The BEAMS algorithm has an additional user-defined parameter β, the filtering ratio, which is set to 0.4. Regarding the settings of parameters used by SMETANA, we set n_{max}= 10, $\alpha^* = 0.9$ and $\beta^* = 0.8$. These are the settings used in the original article (Sahraeian and Yoon, 2013). Note that α^* and β^* do not correspond to α and β defined herein.

We experimented on both real and synthetic PPI networks. Regarding the former, we present a discussion of the global many-to-many alignment results for the PPI networks of five extensively studied species: *Caenorhabditis elegans* (worm), *Drosophila melanogaster* (fly), *Homo sapiens* (human), *Mus musculus* (mouse) and *Saccharomyces cerevisiae* (yeast). As input data, the BEAMS algorithm requires the PPI networks and the pairwise sequence similarity scores of aligned proteins. All these data are retrieved from the IsoBase (Park *et al.*, 2011) database, which is the same as that used by the IsoRank, IsoRankN, SPINAL and SMETANA algorithms. These PPI networks are formed by combining the network data from various databases including DIP (Salwinski *et al.*, 2004), BIOGRID (Breitkreutz *et al.*, 2008), HPRD (Keshava Prasad *et al.*, 2009), MINT (Ceol *et al.*, 2010) and IntAct (Aranda *et al.*, 2010). The *C.elegans* network has 19 756 proteins and 4884 interactions, the *D.melanogaster* network has 14 098 proteins and 25 054 interactions, the *H.sapiens* network has 22 369 proteins and 55 168 interactions, the *M.musculus* network has 24 855 proteins and 592 interactions, the *S.cerevisiae* network has 6659 proteins and 82 932 interactions and in total there are 87 737 proteins and 168 630 interactions. Pairwise sequence similarity scores correspond to the BLAST bit-values of the protein sequences retrieved from Ensembl (Hubbard *et al.*, 2009). With regards to the experimental results on synthetic data, we used synthetic PPI networks retrieved from the NAPAbench, which is a recently proposed synthetically constructed network alignment benchmark (Sahraeian and Yoon, 2012). Owing to space considerations, we present our experimental evaluations regarding these synthetic networks in the Supplementary Document.

Later in the text we provide a detailed evaluation of the alignment results produced by the three algorithms. In the next subsection, we analyze the output alignments in terms of quantitative properties. Following this discussion, we next provide an evaluation based on biological significance of the resulting alignments.

3.1 Analysis of output clusters

Table 1 provides a summary of a quantitative analysis of the alignments produced by the algorithms BEAMS, IsoRankN and SMETANA. For a more detailed analysis, in addition to the total coverage values provided by all the clusters, we also provide a separate analysis by subdividing the output set based on *c*, the number of networks represented in the clusters. The first four multirows provide these results for the instances of $c = 2,3,4,5$, respectively. The total coverage of BEAMS and SMETANA are close, although that of SMETANA is slightly larger. The clusters produced by the alignments of both algorithms have far better total coverage than those of the IsoRankN alignments; each algorithm aligns almost 50% more proteins than IsoRankN. Considering the clusters as claimed orthologies, this implies that BEAMS and SMETANA leave out much less unexplained data by proposing orthology relations for most of the proteins. The main reason behind this discrepancy is the lack of IsoRankN clusters containing only proteins

Table 1. Analysis of output clusters

	BEAMS					IsoRankN					SMETANA
	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	
$c = 2$	**7251** **20 540**	7238 20 359	7242 20 419	7249 20 399	7245 20 392	0 0	0 0	0 0	0 0	0 0	6104 14 956
$c = 3$	3259 12 089	3261 12 187	3277 12 259	3280 12 286	3277 12 204	**4717** **15 891**	4716 15 860	4708 15 827	4714 15 859	4699 15 807	2808 10 941
$c = 4$	3281 16 254	3287 16 353	3283 16 311	3286 16 322	**3291** 16 450	3058 14 651	3052 14 611	3036 14 540	3035 14 533	3040 14 550	3180 **18 189**
$c = 5$	2090 13 117	2092 13 094	2081 13 012	2081 12 978	2074 12 940	2099 12 834	2101 12 844	2104 12 868	2084 12 718	2083 12 697	**2412** **19 158**
Total coverage	15 881 62 000	15 878 61 993	15 883 62 001	**15 896** 61 985	15 887 61 986	9874 43 376	9869 43 315	9848 43 235	9833 43 110	9822 43 054	14 504 **63 244**
Interactions	7060 114 889 6.15%	7286 114 919 6.34%	7425 114 323 6.49%	7317 114 839 6.37%	7407 114 306 6.48%	5978 109 364 5.47%	5956 108 778 5.48%	6024 108 374 5.56%	5653 107 310 5.27%	5766 106 642 5.41%	**13 498** **122 450** **11.02%**
AS	**0.5261**	0.4560	0.3860	0.3153	0.2455	0.3970	0.3447	0.2932	0.2400	0.1882	0.4766

Note: For the first five multirows of the table, the top row corresponds to the number of generated clusters and the bottom row provides the total number of proteins in the output clusters. The first four multirows provide results for the instances of $c = 2,3,4,5$, respectively, where c denotes the number of networks in the clusters under consideration. In each row, highest value is shown in bold.

from two networks. Such a deficiency may lead to unreasonable conclusions, as it is natural to expect orthologous groups with proteins from only two species given that the pairwise evolutionary distances of the species under consideration have large variations.

The top row in the multirow indicated with *Interactions* provides the number of *conserved interactions (CI)* resulting from the output alignments, the middle row indicates the total number of interactions between clusters and the bottom row provides their ratios. A PPI is assumed to be conserved if its *cs* score is greater than zero, i.e. the interaction is between a pair of proteins from different clusters that further contain at least one more pair of interacting proteins from another PPI network. For all instances of α, the BEAMS algorithm provides more CI than IsoRankN. Furthermore, this superiority is not simply due to the large number of clusters produced by the BEAMS alignments; considering the ratio of the number of CI to the total number of interactions between clusters, it can be observed that the BEAMS alignments conserve a larger ratio of existing edges between all clusters. SMETANA performs better than BEAMS in terms of the number of CI. A reason that might account for this result is the sizes of produced clusters; the average cluster size for SMETANA alignments is 4.36, whereas that of BEAMS alignments is 3.90. An alignment with large cluster sizes has a better chance in providing larger number of CI. In the extreme case, simply subdividing the input networks into two clusters through the maximum cut of the networks provides a large interaction conservation even leading to 100% conservation ratio. On the other hand, larger cluster sizes may decrease the *ICQ* score, intended to measure the internal cluster quality, and thus the quality of the overall alignment. This becomes evident by inspecting the last row of the table that provides the *AS* scores of the alignments as defined in Equation (1). For each of the corresponding values of α used in the *AS* definition, the BEAMS alignments provide better results than both IsoRankN and SMETANA alignments. We note that for SMETANA the *AS* score provided in the table is computed under $\alpha = 0.3$ setting. The rest of the *AS* scores for SMETANA is 0.42, 0.36, 0.30, 0.24 and for α values is 0.4, 0.5, 0.6, 0.7, respectively. Furthermore, as noted in Sahraeian and Yoon (2013), simply comparing CI counts may be misleading, unless the interaction conservations are among orthologous groups. The next subsection provides a measure denoted with *COI (conserved orthologous interactions)*, which takes this fact into account.

3.2 Evaluations based on biological significance

Similar to previous PPI network alignment studies, our biological significance evaluations are based on the hierarchical Gene Ontology (GO) categorization, where proteins are annotated with appropriate GO categories organized as a directed acyclic graph (Ashburner *et al*., 2000). To standardize the GO annotations of proteins, similar to the evaluation methods of Aladağ and Erten (2013), Liao *et al*. (2009) and Singh *et al*. (2008), we restrict the protein annotations to level five of the GO directed acyclic graph by ignoring the higher-level annotations and replacing the deeper-level category annotations with their ancestors at the restricted level. The protein annotations are used to measure the consistency of generated clusters. A cluster is *annotated* if at least two of its proteins are annotated by some GO categories.

Table 2. Biological significance evaluations

	BEAMS					IsoRankN					SMETANA
	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	$\alpha = 0.3$	$\alpha = 0.4$	$\alpha = 0.5$	$\alpha = 0.6$	$\alpha = 0.7$	
$c = 2$	**2150**	2143	2147	2139	2132	0	0	0	0	0	1593
	1997	1992	**1997**	1992	1985	0	0	0	0	0	1489
	92.9%	93.0%	93.0%	93.1%	93.1%	–	–	–	–	–	**93.5%**
$c = 3$	1791	1787	1792	1786	1784	2523	2516	2524	**2528**	2524	1497
	1478	1469	1479	1468	1466	1926	1924	1938	**1944**	1943	1179
	82.5%	82.2%	**82.5%**	82.2%	82.2%	76.3%	76.5%	76.8%	76.9%	77.0%	78.8%
$c = 4$	2497	2503	2499	2503	**2517**	2275	2272	2253	2252	2255	2208
	1843	1852	1840	1842	**1853**	1616	1613	1608	1606	1601	1436
	73.8%	**74.0%**	73.6%	73.6%	73.6%	71.0%	71.0%	71.4%	71.3%	71.0%	65.0%
$c = 5$	1971	1974	1961	1962	1954	1958	1960	1963	1941	1943	**2233**
	1375	1382	**1384**	1382	1371	1309	1308	1305	1293	1298	1346
	69.8%	70.0%	**70.6%**	70.4%	70.2%	66.9%	66.7%	66.5%	66.6%	66.8%	60.3%
Total	**8409**	8407	8399	8390	8387	6756	6748	6740	6721	6722	7531
	6693	6695	**6700**	6684	6675	4851	4845	4851	4843	4842	5450
	79.59	79.64	**79.77**	79.67	79.59	71.8	71.8	71.97	72.06	72.03	72.37
Sensitivity	0.3780	0.3784	**0.3791**	0.3771	0.3783	0.3203	0.3203	0.3199	0.3189	0.3198	0.3606
Correct nodes	22 231	22 258	**22 304**	22 234	22 218	16 350	16 333	16 334	16 315	16 301	20 227
	71.1%	71.2%	**71.4%**	71.2%	71.1%	67.2%	67.1%	67.3%	67.3%	67.3%	64.1%
$RCNC_1$	11 397	**11 430**	11 425	11 370	11 406	3382	3330	3310	3377	3350	–
$RCNC_2$	6979	7036	**7056**	6966	6949	–	–	–	–	–	5325
MNE	**1.2881**	1.2908	1.2902	1.2909	1.2899	1.4685	1.4679	1.4672	1.4682	1.4672	1.3943
NGOC	0.3093	0.3075	0.3086	**0.3097**	0.3096	0.2413	0.2410	0.2424	0.2427	0.2422	0.2471
COI	3331	3541	**3590**	3469	3491	2374	2350	2359	2294	2335	2694

Note: For the first five multirows, the top row indicates the number of annotated clusters, the middle row provides the number of consistent clusters and the bottom row indicates the ratio of consistent clusters to annotated clusters. The *c* values are the same as those in Table 1. In each row, best performance is shown in bold.

An annotated cluster is considered *consistent* if all of its proteins share at least one common standard GO annotation. The consistency evaluations of the BEAMS, IsoRankN and SMETANA alignments are provided in the first five multirows of Table 2. The top row in each of these multirows indicates the number of annotated clusters, the middle row provides the number of consistent clusters and finally the bottom row indicates the ratio of consistent clusters to annotated clusters. This ratio is called *specificity* in some previous alignment studies (Sahraeian and Yoon, 2012). Considering the complete set of annotated clusters, it is clear that the BEAMS alignments outperform those of IsoRankN and SMETANA in terms of the number of consistent clusters. Furthermore, the aligned clusters of BEAMS are more specific than those produced by IsoRankN and SMETANA.

To measure how sensitive the provided alignment results are, we use the *sensitivity* definition as in Flannick *et al.* (2009). Analogous to that definition, for a given GO category, let its *closest cluster* denote the cluster that contains the maximum number of proteins annotated with this GO category. The sensitivity of an alignment is then defined as the average, over all GO categories, of the fraction of aligned nodes annotated with a GO category that are also in its closest cluster. *Correct nodes*, another measure that reflects sensitivity of an alignment (Sahraeian and Yoon, 2012), are defined as the total number of annotated proteins in all the consistent clusters. In the corresponding multirow, the top provides this number, whereas the bottom provides the ratio of correct nodes to the number of annotated nodes in the alignment. Additionally, we provide an alternative metric to measure the correct nodes of an alignment relative to an alternative alignment. An *RCNC1* value shown under a BEAMS column provides the number of annotated proteins in consistent clusters in a BEAMS alignment and in inconsistent clusters in an IsoRankN alignment under the same α settings. The *RCNC1* value under an IsoRankN column provides the exact opposite. Similarly, *RCNC2* measures analogous relative correct node counts between BEAMS and SMETANA alignments. We note that for SMETANA the *RCNC2* score provided in the table is relative to the BEAMS alignment with $\alpha = 0.3$ setting. The rest of the scores for SMETANA relative to the BEAMS alignments with $\alpha = 0.4$, 0.5, 0.6, 0.7 settings are 5330, 5332, 5400 and 5367, respectively. The BEAMS alignments provide much better sensitivity, correct node counts and relative correct node counts than those of IsoRankN and SMETANA.

Mean normalized entropy (MNE) is another consistency evaluation metric used in previous studies (Liao *et al.*, 2009; Sahraeian and Yoon, 2012). The normalized entropy of an annotated cluster

Cl_x is defined as $NE(Cl_x) = -\frac{1}{\log d} \times \sum_{i=1}^{d} p_i \times \log p_i$, where p_i is the fraction of proteins in Cl_x with the annotation GO_i, and d represents the number of different GO annotations in Cl_x. For *MNE* the sum of these values are averaged over the total number of annotated clusters. Lower *MNE* values indicate better consistency. Yet another consistency evaluation metric is *GO consistency (GOC)* defined in Aladağ and Erten (2013). Because GOC is defined for the one-to-one alignment of a pair of networks, we extend the definition to many-to-many alignments of multiple networks by normalizing the score. For an annotated cluster Cl_x, let $GO_{int}(Cl_x)$ and $GO_{uni}(Cl_x)$ indicate, respectively, the intersection set of GO annotations of proteins in Cl_x and the union set of GO annotations of all the proteins in Cl_x. The normalized *GOC* score, *nGOC*, is defined as the weighted mean of $|GO_{int}|/|GO_{uni}|$ over all annotated clusters, where the weight of each cluster is the number of annotated proteins it contains. In terms of better consistency larger *nGOC* values are desirable. With respect to both metrics, *MNE* and *nGOC*, the BEAMS algorithm clearly outperforms both IsoRankN and SMETANA.

Finally, as was noted at the end of the previous subsection, the *CI* score by itself may not be a proper measure. It is important to detect whether the provided interaction conservations are spurious or do actually correspond to real CI between orthologous nodes. Similar to Sahraeian and Yoon (2013), we use the *COI* measure for this purpose. For a given alignment, it represents the number of CI between consistent clusters. The *COI* scores of IsoRankN and SMETANA are somewhat similar, whereas BEAMS provides a noticeably large score. BEAMS provides almost 1000 more CI between orthologous clusters than IsoRankN and SMETANA. The *COI*/*CI* ratios may provide a good clue as to the success of SMETANA in achieving large *CI* score discussed in the previous subsection. The ratio is 48% for BEAMS, whereas it is as low as 20% for SMETANA. This indicates that SMETANA aggressively conserves interactions at the expense of possible spurious conservation between non-orthologous nodes.

In addition to these evaluation metrics, intended to measure biological significance of output alignments, we also provide a specific clustering instance resulting from the alignments of BEAMS, IsoRankN and SMETANA on the same dataset. Owing to space requirements details regarding a discussion of this alignment instance are provided in the Supplementary Document.

3.3 Running time requirements

Let V denote the set of nodes in all the PPI networks, Δ_{max} denote the maximum degree of any node in any of the input PPI networks and finally let Δ denote the maximum degree in S_β. With the reasonable assumptions that $\Delta_{max} = O(\Delta)$ and $|V| = O(\Delta^k)$, the running time of BEAMS is bounded by $O(V^2\Delta^{k+1})$. A formal running time analysis of the algorithm can be found in the Supplementary Document. An important advantage of BEAMS and SMETANA over IsoRankN is their superb execution speed. For the IsoBase data experiments of this section, IsoRankN required almost 40 h for execution completion on average. The time requirements of BEAMS and SMETANA were similar. Both required almost half an hour for completion under the same computational settings. Furthermore, the memory requirements of BEAMS is much better than those of SMETANA; the former requiring 2.5 Gb for the experiments on the IsoBase data, whereas the latter required almost 4.5 Gb on the same input. Details of all the required CPU times can be found in the Supplementary Document.

4 CONCLUSION

We provided a combinatorial optimization formulation for the global many-to-many alignment of multiple PPI networks. We showed that the problem is computationally intractable. Based on the general seed-and-extend framework, we then provided a novel heuristic, BEAMS for the problem. We compared the BEAMS algorithm against two popular state-of-the-art algorithms, IsoRankN and SMETANA. Using the network data of IsoBase, we showed that BEAMS outperforms both algorithms with regards to several biological significance metrics proposed in literature. We note that in addition to the many-to-many version of the network alignment problem, versions including one-to-one and one-to-many have also been studied previously. Owing to lack of standard criteria for evaluations of alignments produced by different versions, it was out of the scope of the current article to compare BEAMS against those algorithms proposed for one-to-one or one-to-many alignments. Further studies involving the design of evaluation criteria for various alignment problem versions would enhance our understanding of comparative biological network analysis.

ACKNOWLEDGEMENT

The authors would like to thank Ö. Yaşar Diner for fruitful discussions.

Funding: The Scientific and Technological Research Council of Turkey (TUBITAK) (112E137) (in part).

Conflict of Interest: none declared.

REFERENCES

Abaka,G. *et al.* (2013) Campways: constrained alignment framework for the comparative analysis of a pair of metabolic pathways. *Bioinformatics*, **29**, i145–i153.

Aebersold,R. and Mann,M. (2003) Mass spectrometry-based proteomics. *Nature*, **422**, 198–207.

Aladağ,A.E. and Erten,C. (2013) Spinal: scalable protein interaction network alignment. *Bioinformatics*, **29**, 917–924.

Aranda,B. *et al.* (2010) The intact molecular interaction database in 2010. *Nucleic Acids Res.*, **38**, 525–531.

Ashburner,M. *et al.* (2000) Gene Ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25–29.

Ay,F. *et al.* (2011) Submap: aligning metabolic pathways with subnetwork mappings. *J. Comput. Biol.*, **18**, 219–235.

Breitkreutz,B. *et al.* (2008) The biogrid interaction database: 2008 update. *Nucleic Acids Res.*, **36**, 637–640.

Ceol,A. *et al.* (2010) Mint, the molecular interaction database: 2009 update. *Nucleic Acids Res.*, **38**, 532–539.

Chindelevitch,L. *et al.* (2010) Local optimization for global alignment of protein interaction networks. *Pac. Symp. Biocomput.*, **2010**, 123–132.

Finley,R.L. and Brent,R. (1994) Interaction mating reveals binary and ternary connections between drosophila cell cycle regulators. *Proc. Natl Acad. Sci. USA*, **91**, 12980–12984.

Flannick,J. *et al.* (2006) Graemlin: general and robust alignment of multiple large interaction networks. *Genome Res.*, **16**, 1169–1181.

Flannick,J. *et al.* (2009) Automatic parameter learning for multiple local network alignment. *J. Comput. Biol.*, **16**, 1001–1022.

Goh,C.S. and Cohen,F.E. (2002) Co-evolutionary analysis reveals insights into protein-protein interactions. *J. Mol. Biol.*, **324**, 177–192.

Hubbard,T. *et al.* (2009) Ensembl 2009. *Nucleic Acids Res.*, **37**, 690–697.

Kalaev,M. *et al.* (2009) Fast and accurate alignment of multiple protein networks. *J. Comput. Biol.*, **16**, 989–999.

Kelley,B.P. *et al.* (2004) Pathblast: a tool for alignment of protein interaction networks. *Nucleic Acids Res.*, **32**, 83–88.

Keshava Prasad,T.S. *et al.* (2009) Human protein reference database-2009 update. *Nucleic Acids Res.*, **37**, 767–772.

Korf,R.E. (2010) Artificial intelligence search algorithms. In: Atallah,M.J. and Blanton,M. (eds) *Algorithms and Theory of Computation Handbook*. Chapman & Hall/CRC, Boca Raton, Florida, USA, pp. 22.1–22.23.

Kuchaiev,O. and Pržulj,N. (2011) Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics*, **27**, 1390–1396.

Liao,C.S. *et al.* (2009) Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, **25**, i253–i258.

Marcotte,E.M. *et al.* (1999) Detecting protein function and protein-protein interactions from genome sequences. *Science*, **285**, 751–753.

Mehlhorn,K. and Naher,S. (1999) *Leda: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, Cambridge.

Park,D. *et al.* (2011) Isobase: a database of functionally related proteins across PPI networks. *Nucleic Acids Res.*, **39**, 295–300.

Sahraeian,S.M. and Yoon,B.J. (2012) A network synthesis model for generating protein interaction network families. *PLoS One*, **7**, e41474.

Sahraeian,S.M.E. and Yoon,B.J. (2013) Smetana: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. *PLoS One*, **8**, e67995.

Salwinski,L. *et al.* (2004) The database of interacting proteins: 2004 update. *Nucleic Acids Res.*, **32**, 449–451.

Shih,Y.K. and Parthasarathy,S. (2012) Scalable global alignment for multiple biological networks. *BMC Bioinformatics*, **13** (**Suppl. 3**), S11.

Singh,R. *et al.* (2008) Global alignment of multiple protein interaction networks. *Pac. Symp. Biocomput.*, **2008**, 303–314.

Skrabanek,L. *et al.* (2008) Computational prediction of protein-protein interactions. *Mol. Biotechnol.*, **38**, 1–17.