Computer Science and Information Systems 18(4):1427–1444 https://doi.org/10.2298/CSIS200909030C

A Novel Network Aligner for the Analysis of Multiple Protein-protein Interaction Networks

Jing Chen[1,2,⋆] and Jia Huang[1]

[1] School of Artificial Intelligence and Computer Science,
Jiangnan University Wuxi, China
chenjing@jiangnan.edu.cn
[2] Jiangsu Provincial Engineering Laboratory of Pattern
Recognition and Computing Intelligence, Jiangnan University, Wuxi, China
6181914004@stu.jiangnan.edu.cn

Abstract. The analysis of protein-protein interaction networks can transfer the knowledge of well-studied biological functions to functions that are not yet adequately investigated by constructing networks and extracting similar network structures in different species. Multiple network alignment can be used to find similar regions among multiple networks. In this paper, we introduce Accurate Combined Clustering Multiple Network Alignment (ACCMNA), which is a new and accurate multiple network alignment algorithm. It uses both topology and sequence similarity information. First, the importance of all the nodes is calculated according to the network structures. Second, the seed-and-extend framework is used to conduct an iterative search. In each iteration, a clustering method is combined to generate the alignment. Extensive experimental results show that ACCMNA outperformed the state-of-the-art algorithms in producing functionally consistent and topological conservation alignments within an acceptable running time.

Keywords: graph data analysis, big data, protein-protein interaction network, network clustering, seed-and-extend strategy.

1. Introduction

Great progress has been made in constructing large amounts of biological networks of different species using high-throughput experimental techniques and computational predictions. In recent years, obtaining information on cell composition and function by analyzing network data has gradually become a popular research topic. In protein-protein interaction networks (PPINs), proteins are represented by nodes and interactions between two proteins by edges between two nodes. The alignment is usually generated according to the topological structure and sequence similarity information of the protein-protein interaction network. The topology of the network can extract much of the hidden information in the network [23], [26], which can be used for network research on different data. Functionally homogeneous proteins and protein complexes in different species can be discovered through network alignment, which is divided into pairwise network alignment (PNA) and multiple (*i.e.*, three or more) network alignment (MNA) according to the number of aligned networks. The purpose of PNA is the creation of node mappings

⋆ Corresponding author

between two networks. In addition to finding a mapping between multiple networks, the MNA can obtain correlation information of different species simultaneously. Therefore, a well-studied MNA can provide deeper network insight. With respect to the mapping types, network alignment algorithms are divided into one-to-one, one-to-many and many-to-many alignment algorithms. In one-to-one alignment, there is exactly one node from each aligned network, and not every node is required to be mapped; in one-to-many alignment, which is usually used in metabolic network alignments, one metabolic path can be mapped to another subset; in many-to-many alignment, there can be one or more nodes from the same network in each alignment cluster. Network alignment types are classified into local network alignment (LNA) and global network alignment (GNA). The purpose of LNA is to find highly conserved, unrelated sub-networks with a highly similar structure among the input network. However, LNAs only consider the similarity of local structures, which may lead to conflicts or ambiguities. On the other hand, GNA aims to construct node mappings between the overall nodes from the input networks with the cost of sub-optimal conservation in the local area and finally obtain a network with a larger coverage, which can produce a more consistent alignment compared to LNA.

Network alignment can be regarded as a subgraph isomorphism problem; however, subgraph isomorphism is an NP-hard complete problem [7],which makes it very difficult to find the network alignment solution. Heuristic algorithms are usually used for the solution of NP-hard problems. They have the ability of intelligence, generality and global search, which make them applicable in many fields, such as the cutting problem [44], image analysis [5], the graph matching problem and so on. Therefore, heuristics alignment algorithms are used to address the issue that the computational difficulty of network alignment increases exponentially with the increase of input network size.

The two most important aspects in evaluating network alignment results are network topology and biological consistency. Nevertheless, achieving high topological conservation while obtaining biological significance is often contradictory in present literature, even though they are both vital goals of network alignment [12]. The present study attempted to solve the problem of balancing network topological conservation and functional consistency. Moreover, the ACCMNA algorithm proposed in this paper introduces a network clustering method as a solution to the problem of network alignment. The MNA algorithm gathers many similar nodes in the same cluster and is, therefore, similar to a clustering algorithm.

The multiple network aligner ACCMNA is proposed in this paper, which could match as many consistent proteins together as possible and outperformed other state-of-the-art algorithms in real and synthetic network experiments. The ACCMNA algorithm is based on a seed-and-extend schema, inspired by the backbone extraction from the BEAMS algorithm [1]. To begin, the calculation of node weights is included in the initialization process and the topology and sequence information of the network are also considered. Second, a clustering method finds the maximum edge weighted cluster, and an expansion method is used so that similar proteins can be put into a cluster as much as possible.

In this section, we introduce network alignment, and the remainder of this paper is organized as follows: In Section 2, we introduce the current state of network alignment research. In Section 3, we describe the definition of the problem, the details of the algorithm, and the two important innovations of our algorithm. In Section 4, we present the

experimental results of the algorithm on different data sets and analyze the time complexity. Finally, the concluding remarks are discussed in Section 5.

2. Related Work

Network alignment algorithms have been widely studied in recent years. Research on pairwise network alignment was quite popular in the early years. Consequently, many excellent pairwise network alignment algorithms have been developed. The GRAAL family of algorithms consists of GRAAL [23], H-GRAAL [29], MI-GRAAL [24], C-GRAAL [28] and L-GRAAL [27],which use graphlet degree signatures and sequence similarity information to calculate the similarity between two networks. MAGNA [39] is optimized by a genetic algorithm to obtain the results, IsoRank [40] uses a method similar to Google's PageRank algorithm to calculate the similarity of nodes in a different network to find the alignment, NETAL [30] calculates the similarity score and obtains the alignment through the greedy search algorithm, and PINALOG [34] uses community detection to improve the alignment algorithm result. The above algorithms are compared and analyzed in detail in prior work [9]. With the increasing availability of PPI networks, the need for simultaneous alignment of multiple networks is growing, and the study of MNA algorithms has become increasingly popular. Multiple network alignment is different from the pairwise network alignment in that multiple networks can be aligned simultaneously,however the time complexity and computational difficulty of the algorithm become higher. Alignment algorithms that have been proposed include Graemlin [11], an early two-phase local MNA algorithm that learns the score function to optimize the vector of features while continuously iterating to produce the final alignment. However, Graemlin requires additional phylogenetic information as input. Both Graemlin1.0 and Graemlin2.0 [10] have been developed as local aligner and global aligner, respectively. IsoRankN [25] uses spectral graph theory to calculate similarity scores of nodes between any two networks. SMETANA [37] calculates the similarity score matrix based on a semi-Markov random walk model and uses probabilistic consistency transformations to enhance the similarity score matrix. The final alignment result is generated by a greedy searching method. BEAMS [1] establishes the alignment by generating the maximum edge weighted cliques. Then, the backbone extraction and merge strategy are used to produces alignment results of high biological consistency was proposed. CSRW [18] is an improved version of SMETANA in that it establishes a score matrix by using a context-sensitive random walk model. NetCoffee [16] is an extension of the T-Coffee algorithm [31], which uses a triplet approach that combines the third network information. Subsequently, the similarity score between any two networks is calculated and a simulated annealing method is used for continuous iteration until the final alignment is produced. Due to the limitation of the triplet method, NetCoffee can align only three or more networks. To address this issue, NetCofee2 [15] was proposed, which is based on graph feature vectors and is more accurate and efficient for aligning two or more networks. The MAPPIN [8] algorithm is an improved version of NetCoffee. MAPPIN can align two or more networks as well and combines the GO annotation information of proteins with topology and sequence similarity to calculate the similarity of nodes. Node Handprinting (NH) [35]is a global MNA, which solves the weighted bipartite graph matching problem by using the progressive alignment strategy to obtain the final optimal alignment. MultiMAGNA++ [42] is a global MNA designed to maxi-

mize the optimization objective function using genetic algorithms and is an extension of MAGNA and MAGNA++ [43], which are pairwise network alignment algorithms. FUSE [13] calculates similarity scores by using the non-negative matrix tri-factorization method and the k-partite matching algorithm to obtain the one-to-one alignment results. MPGM [20] generates seeds through sequence similarity and then obtains the final many-to-many alignment results through the percolation-based, graph-matching algorithm.

3. Method

3.1. Problem Definition

Let $G_1(V_1, E_1), G_2(V_2, E_2), \ldots, G_k(V_k, E_k)$ denote the k initial input PPI networks. Here, G_i represents the ith input network. V_i, E_i represent the nodes, that is, the proteins and edges (interactions), of the set of the ith input network, respectively. S represents the complete k-partite similarity graph of the weighted edge, where S has the same nodes as the input networks. The edges represent the interrelationship of proteins among different species. The value of the edge weight represents the sequence similarity score, where the value of weight is the bit score value between u and v, which are two nodes from different networks obtained through Basic Local Alignment Search Tool (BLAST), which was proposed in prior work [2]. S_β represents the filtered version of similarity graph S, which is a subgraph of S with some edges removed. If unfiltered sequence similarity data are used, then the computational complexity increases exponentially with the size of S and some similarity data may lead to incorrect alignment due to incompleteness in sequence similarity information. To avoid this, the S graph is filtered using beta, which is a user-defined threshold for each edge (x, y). If $w(x, y) < \beta \times max(x, y)$, then edge (x, y) in the similarity graph S is deleted. Here, $max(x, y)$ denotes the maximum value of the weight of an edge associated with x or y in S.

Assume that $A = \{Cl_1, Cl_2, \ldots, Cl_n\}$ is an alignment result of the input network, and alignment $A \in E$, where E is the edge set of all the networks mentioned above. In many-to-many network alignment A, for any cluster $Cl_i = \{V_{1,i}, V_{2,i}, \ldots, V_{k,i}\}$, $V_{c,i}$ is the node set of the ith cluster and nodes come from the cth network. $V_{c,i} \cap V_{c,j} = \emptyset, \forall i \neq j$, that is, a node belongs exclusively to one cluster. For a given network, the quality of alignment A is unknown and needs to be measured. Therefore, we quote the method in BEAMS [1] as the objective function of alignment A. Here, Formula 1 can be used as the objective function of the algorithm as follows:

$$AS(A) = \alpha \times CIQ(A) + (1 - \alpha) \times ICQ(A)\,, \tag{1}$$

where α is a real number from 0 to 1 that balances the contribution weight of topology and sequence scores.

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

In the equation above, $CIQ(A)$ stands for the cluster interaction quality and is a score function that measures the quality of the conservative edge between clusters; E_{Cl_m, Cl_n} represents the set of edges whose vertices are in the distinct cluster Cl_m, Cl_n; $cs(m, n)$

represents any two clusters Cl_m, Cl_n and the proportion of the conserved edge network calculated by the formula $cs(m,n) = c'_{m,n}/c_{m,n}$; $c_{m,n}$ represents the number of networks with nodes in both clusters Cl_m and Cl_n; and $c'_{m,n}$ is the number of networks in which the vertices of the edges in E_{Cl_m,Cl_n} are in different networks. We assign $cs(m,n) = 0$ if $c'_{m,n} = 1$, which indicates that there is no conservative edge between clusters Cl_m and Cl_n.

$$ICQ(A) = \frac{\sum_{Cl_i \in A} ICQ(Cl_i)}{|A|} \tag{3}$$

$$ICQ(Cl_i) = \frac{\sum_{\forall (u,v) \in E(Cl_i)} \sqrt{\frac{w(u,v)^2}{w_{max}(u) \times w_{max}(v)}}}{|E(Cl_i)|} \tag{4}$$

Here, $ICQ(A)$ stands for the internal cluster quality and is defined as a measure for the sequence similarity score between aligned nodes, expressed in Formula 3, where A denotes alignment result; $ICQ(Cl_i)$ represents the sequence similarity measure of Cl_i, expressed in Formula 4, where $w(u,v)$denotes the bit score of sequence similarity information between node u and node v; $w_{max}(u)$denotes the maximum value of the edge attached to the node u in S_β; and $E(Cl_i)$ is the number of edges from the S_β incident on nodes in cluster Cl_i.

3.2. Algorithm

Inspired by the backbone extraction of the BEAMS algorithm [1], the whole framework of the ACCMNA algorithm is a seed-and-extend strategy. The method is used in combination with clustering to generate the alignment cluster. The clusters are generated in each iteration as seeds and the seeds are expanded in the input network to generate new clusters. The pseudo-code for the ACCMNA algorithm is demonstrated in the following Algorithm description. The algorithm is initialized, while both the alignment set and the candidate set are empty. To begin, the weight of each node is calculated based on the topology and sequence information of the network, then the first candidate C_0 is generated by searching the cluster in the graph through the function $Generate\_Candidate(S_\beta)$. This function searches for the node with the largest weight and its neighbor nodes in the weighted graph S_β to generate a subgraph of S_β through these nodes, while a cluster is generated in this subgraph. The main part of the algorithm is the repeat loop. The first step involves selecting the candidate with the highest AS score in candidate set C as the new alignment A_{new} in this loop, adding A_{new} to the alignment set A, and deleting the nodes contained in A_{new} in S_β. The second step is generating the neighborhood node set in the PPIN according to the nodes in A_{new} and establishing the subgraph $N_{S_\beta}(A_{new})$ of this neighborhood node set. If $N_{S_\beta}(A_{new})$ contains only isolated nodes, then C_{new2} is empty; otherwise, a new prospective candidate $C_{new1}$is generated in the graph. In the third step, if C_{new1} contains nodes in each input network, then it is not extended; otherwise, candidate C_{new2} is generated by extending C_{new1} in the S_β. In the fourth step, if there is overlap between the newly generated cluster and the candidate set, then the candidate needs to be updated and the above four steps are repeated until the candidate set is empty.

```
Algorithm description
  Input: S_β, G_1, G_2, ..., G_k, α
```

```
Output: Set of cluster A
C = ∅;  A = ∅;
//Initial
Calculate node score NodeWeight;
C_0=Generate initial candidates in S_β;
C = C ∪ {C_0};
repeat
    Select a cluster A_new from candidate set C;
    A = A ∪ {A_new};
    remove A_new from S_β;
    generate new candidates C_new1 in A_new's neighbors;
    expand new Candidate of C_new1 in S_β;
    C = C ∪ {C_new2};
    for all C_i ∈ C do
        if C_i ∩ A_new ≠ ∅ then
            if i==0 then
                C_0=Generate initial candidates in S_β;
            else:
                generate new candidates C_i;
            end if
        end if
    end for
  until no more Candidate
end.
```

Calculation of the Node Weight. When the initial candidate is generated in S_β, the node with the highest weight needs to be located, and then the cluster is searched with this node as the center. Inspired by the HubAlign algorithm for computing the node similarity function, the HubAlign algorithm is used for pairwise network aligners. HubAlign uses a minimum-degree heuristic method to measure the role of nodes in the network and preferentially aligns the more important nodes [14]. The use of the HubAlign algorithm is extended to the calculation of the similarity of nodes among multiple networks. Using a similar approach to HubAlign, the topological importance score for all nodes in the network is calculated to begin with, and then the score of all nodes is calculated by combining the sequence similarity information between pairs of nodes from different networks. The degree is one of the properties that can reflect the importance of nodes, given that the importance of nodes in the network can be determined by the global topological property. The weights of nodes and edges are calculated by traversing from the node with the smallest degree to the node with a degree of 10. The weight of the node with a small degree is transferred to the node or edge with a larger degree at the adjacent node so that higher weight scores are assigned to nodes with a higher degree in the network and there is a greater weight of the edge connected with the node. The network nodes and edge weights are initialized as follows (see Fig. 1 for a simplified example):

$$we(u,v) = \begin{cases} 1, \text{(u, v)} \in E \\ 0, \text{otherwise} \end{cases}, wn(u) = 0, \forall u \in V\,, \tag{5}$$

where $wn(u)$ represents the weight of the nodes in V; $we(u, v)$ represents the weight of the edges between nodes u and v in PPINs; for a particular node u in a network, let $deg(u)$ be the degree of node u; $N(u)$ denotes the neighbor nodes set of node u; and $|N(u)|$ is the number of neighbors of node u and also the degree of node u. The node weight update starts from the node with the lowest degree and the topology information of the node is gradually transferred to the neighbors with the higher degree. Nodes with zero degrees are generally ignored. For a given node u, $\forall v \in N(u)$, $wn(v) = wn(v) + wn(u) + we(u, v)$, if $deg(u) = 1$. If $deg(u) > 1$ and $\forall v_1, v_2 \in N(u)$, then

$$we(v_1, v_2) = we(v_1, v_2) + \frac{wn(u) + \sum_{v \in N(u)} we(u, v)}{\frac{|N(u)||N(u)-1|}{2}}, \tag{6}$$

Following the weight calculation in Formula 6, the importance score of the node is calculated by combining the weight of the node with the weight of the edge, as follows:

$$importance(u) = wn(u) + \gamma \sum_{v \in V} we(u, v). \tag{7}$$

where $importance(u)$ is the importance score of node u, and γ is set $\gamma = 0.2$ and controls the contribution of the node related edge weights. The importance score obtained from the network topology information is combined with the sequence similarity information to obtain the final node weight. Nodes with zero degrees are generally ignored again. Formula 8 is proposed to calculate the sequence similarity score of node u in S_β. The formula is as follows:

$$B(u) = \frac{\sum_{v \in N_S(u)} B(u, v)}{|N_S(u)|}, \tag{8}$$

where $B(u, v)$ represents the sequence similarity information between nodes u and v, which in this paper was calculated by the BLAST bit score; and $B(u)$ represents the average value of sequence similarity values related to node u. Finally, the weight of each node in PPIN is obtained by combining the topology importance and sequence similarity score. The final node weight is calculated as follows:

$$Weight(u) = \alpha \times importance(u) + (1 - \alpha) \times B(u), \tag{9}$$

where α is a balancing parameter, see the definition of Equation 1.

Cluster Searching. For a graph with a given non-negative weight edge, the candidate is generated by searching similar nodes in the search graph according to the edge weight. Thus, nodes with high sequence similarity are clustered. The clustering method is combined with the network alignment and similar nodes are gathered to generate clusters through the clustering method in the search graph. Inspired by the clustering algorithm SPICi [19], a clustering method based on the seed-and-extend approach is adopted. The data of sequence similarity is incomplete, which may lead to similar nodes with no sequence similarity value between them and contribute to the incomplete alignment. The alignment is constructed by improving this method with the inclusion of similar nodes in the same cluster as much as possible.

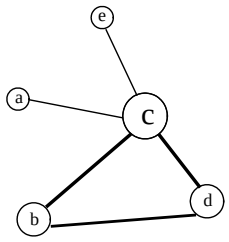

Fig. 1. An example to illustrate the calculation of node weights. This example network has five nodes. The thickness of an edge shows its weight, and the size of a node shows its weight. For example, for node a with degree one, N(a)={c}, wn(c)=wn(c)+wn(a)+we(a, c); however, for node b with degree greater than one, N(b)={c, d}, we(c, d)=we(c, d)+(wn(b)+we(b, c)+we(b, d))/(2(2-1)/2)=we(c, d)+wn(b)+we(b, c)+we(b, d)

The weights of all the nodes in the search graph are calculated according to the following formula:

$$deg_w(u) = \sum_{v \in N_S(u)} B(u, v)\,, \tag{10}$$

where $N_S(u)$ represents the set of neighbor nodes of node u in the filtered similarity graph S_β; and $B(u, v)$ represents the sequence similarity score between node u and node v in the S_β mentioned above. In each graph with weighted edges, the node with the highest weighted degree is selected as the first seed. The higher the weight of a node, the higher its importance in the graph, and it can be used as a meaningful seed node. A higher weight between nodes indicates a higher sequence correlation between two nodes, and, therefore, the weight of the neighbor of the first seed node is normalized and the nodes are divided into five bins according to the normalized weight between them. In this study, they were as follows: (0,0.2],(0.2,0.4],(0.4,0.6],(0.6,0.8],(0.8,1). Searching started from the bin with the highest node weight, that is, (0.8, 1], to the bin with the lowest node weight, that is, (0, 0.2]. If the bin being searched is not empty, then the node with the highest node degree weight in the current bin is the second seed; otherwise, searching continues in the next bin. The neighbor node that is most similar to the seed node is also an important node in the network.

After the initial seed node pair is obtained, the graph is extended through two seed nodes. First, S represents the nodes already included in the current cluster, and S contains only two seed nodes at the beginning of the seed extension. The search node set that could be added to S is composed of the neighbor nodes of the nodes in S. The node with the maximum value of $support(u, S)$ in the search set is selected in each iteration of the extension. The score of $support(u, S)$ is the sum of the weight of the edges in S related to u, indicating the correlation between node u and the node in S. Two constraint conditions decide whether to add node u to S. Node u is only added to S only if Formula 11 was satisfied; otherwise, the search loop is terminated.

$$\begin{cases} density\{S \cup \{u\}\} > T_d \\ \frac{|E_s(u)|}{|S| \times density\{S \cup \{u\}\}} \geq T_s \end{cases} \tag{11}$$

Here, $density\{S \cup \{u\}\}$ denotes the density of graph S after adding node u, and it reflects how close the current graph S is to clique; $|E_S(u)|$ denotes the number of edges related to u in S; and $|S|$ is the number of nodes in graph S. The Values for T_s, T_d were set to 0.5 in here.

After generating the prospective candidate in the neighborhood graph, the generated candidate cluster is extended only when the number of networks in the cluster is less than that of the input networks. The basic process of expansion is the same as that of the above search process. Here, S is the newly generated prospective candidate, and the search nodes are the neighbor nodes of the nodes in S. When the current node meets the two constraints above, the node is added. However, since there is no direct correlation between the extended search set and the nodes in the original alignment cluster, stricter constraints should be set. The values of T_s, T_d were set to be higher, namely 0.7 in the synthetic network and 0.9 in the real networks in here.

4. Results and Discussion

4.1. Datasets

The ACCMNA algorithm was compared with IsoRankN, SMETANA and BEAMS. IsorankN is the first global MNA. As one of the most popular two-phase alignment algorithms, many alignment algorithms have been compared to it. SMETANA is a multiple network aligner based on semi-Markov random walk and probabilistic consistency transformations. Several studies in the literature have proved that SMETANA can produce comparative results with relative topological significance. BEAMS is a heuristic algorithm that searches for the weighted maximum cluster, and the experimental results in many previous reports indicate that BEAMS can produce alignments with good functional consistency.

Table 1. The number of proteins and interactions of five eukaryotic species

	Node	Edge
S. cerevisiae	6659	82932
C. elegans	19756	4884
D. melanogaster	14098	25054
H. sapiens	22369	55168
M. musculus	24855	592

We used real and synthetic networks for the verification of our algorithm. Five eukaryotic network databases derived from the IsoBase [32]were used, S. cerevisiae, C. elegans, D. melanogaster, H. sapiens and M. musculus, which are consistent with the data used in SMETANA [37], IsoRankN [25] and BEAMS [1]. The PPINs data were constructed by combining data from BIOGRID [4], DIP [38], HPRD [21], IntAct [3] and MINT [6]. The node and edge data for each network are presented in Table 1. The sequence homology information of the network corresponded to the BLAST bit score retrieved from Ensembl [17].

The synthetic network used data provided by Network Alignment Performance Assessment Benchmark (NAPAbench) [36] and there were three different network growth models: crystal growth (CG) model [22], duplication-mutation-complementation (DMC) model [41] and duplication with random mutation (DMR) model [33]. Each model contained eight networks. Each network of the CG model contained 1000 nodes and 3985 edges. Each network of the DMC model consisted of 1000 nodes and the number of edges of each network was 1919, 1853, 1923, 1840, 1867, 1848, 1818 and 1867, respectively. The number of nodes in the DMR model network was also 1000 and the number of network edges was 2031, 2092, 1967, 1977, 1959, 1998, 2030 and 2056, respectively.

Table 2. Experimental results on a synthetic network CG model. best performance is shown in bold

	CIQ	SPE	Sen	MNE	nGOC
SMETANA	0.812	0.906	0.573	**0.071**	0.907
IsoRankN	0.692	0.620	0.679	0.276	0.575
BEAMS	0.702	0.879	0.588	0.112	0.910
ACCMNA	**0.892**	**0.920**	**0.713**	**0.071**	**0.947**

Table 3. Experimental results on a synthetic network DMC model. best performance is shown in bold

	CIQ	SPE	Sen	MNE	nGOC
SMETANA	0.754	**0.869**	0.631	**0.106**	**0.865**
IsoRankN	0.573	0.618	0.518	0.294	0.546
BEAMS	0.507	0.806	0.553	0.182	0.833
ACCMNA	**0.791**	0.858	**0.755**	0.119	0.850

Table 4. Experimental results on a synthetic network DMR model. best performance is shown in bold

	CIQ	SPE	Sen	MNE	nGOC
SMETANA	0.689	**0.872**	0.573	**0.106**	**0.873**
IsoRankN	0.545	0.607	0.566	0.304	0.544
BEAMS	0.640	0.815	0.558	0.181	0.841
ACCMNA	**0.748**	0.861	**0.714**	0.119	0.845

In the comparison experiment of the synthetic network, the parameters of our algorithm α and β were set as 0.5 and 0.2, respectively. The values of α and β in our algorithm were 0.5 and 0.3 on the real networks, respectively. The parameters of other compared algorithms were set as the recommended parameters from the literature. The parameters

Table 5. Performance of different algorithms on real networks. best performance is shown in bold

	CIQ	SPE	Sen	MNE	nGOC
SMETANA	**0.054**	0.724	0.360	1.394	0.247
IsoRankN	0.027	0.733	0.303	1.437	0.248
BEAMS	0.035	0.798	**0.379**	1.290	0.309
ACCMNA	0.041	**0.813**	0.345	**1.218**	**0.331**

of the BEAMS algorithm synthetic network were set as the same as our algorithm. The parameters α and β of the BEAMS algorithm real networks were set to 0.5 and 0.2, respectively. Parameter α of the IsoRankN algorithm was set to 0.6, and parameters $\alpha$and β of SMETANA were set to 0.9 and 0.8, respectively, and $n_{max} = 10$.

4.2. Analysis of the Alignment Result

The alignment results of the above algorithms were all many-to-many alignment, which indicated that, for each cluster, multiple nodes from the same network may exist. Protein coverage showed the total number of aligned nodes. The nodes were classified in each cluster according to their source network. The node k-coverage denotes the number of nodes that belong to clusters that contain nodes from k networks. To measure the biological significance of the alignment, GO annotation was used to evaluate the consistency of aligned proteins. If at least two proteins in a cluster were annotated by the GO category, then the whole cluster was considered to be annotated, and if all proteins in an annotated cluster shared the same GO category, then the whole cluster was considered to be consistent. The k-coverage of the consistent nodes denotes the number of consistent proteins present in clusters that contain proteins from k networks. As shown in Fig. 2, the total number of proteins aligned by ACCMNA, SMETANA and BEAMS was very close. IsoRankN aligned the least number of proteins. In general, each cluster is expected to contain proteins from as many species. The alignment generated by the ACCMNA algorithm had the largest number of 8-coverage of nodes. This indicates that the ACCMNA algorithm produced more high-quality clusters. From the consistent protein results in Fig. 2(b), the results of the ACCMNA algorithm indicate that its performance was the best among several aligners and that most of the consistent nodes belonged to clusters from k=8 species. Figure 2 shows that our algorithm produced the cluster that contained the highest number of proteins and consistent proteins from eight species. This can also demonstrate that our algorithm discovered more meaningful information and was more biologically consistent. The alignment results on real networks are displayed in Fig. 3, where A represents the protein coverage of alignment, and B represents the consistent protein coverage. There was little difference between the ACCMNA algorithm results and the BEAMS and SMETANA results, all of which were higher than IsoRankN. The coverage of proteins and consistent proteins on both real networks and synthetic networks revealed that the ACCMNA algorithm outperformed the other algorithms.

The alignment performance was measured using metrics established in the literature. The alignment measurement scores on the synthetic network and the real networks are displayed in Tables 2, 3, 4 and 5. CIQ has been proposed as a measurement for conserved edges between clusters and used in previous literature for result comparison [42],

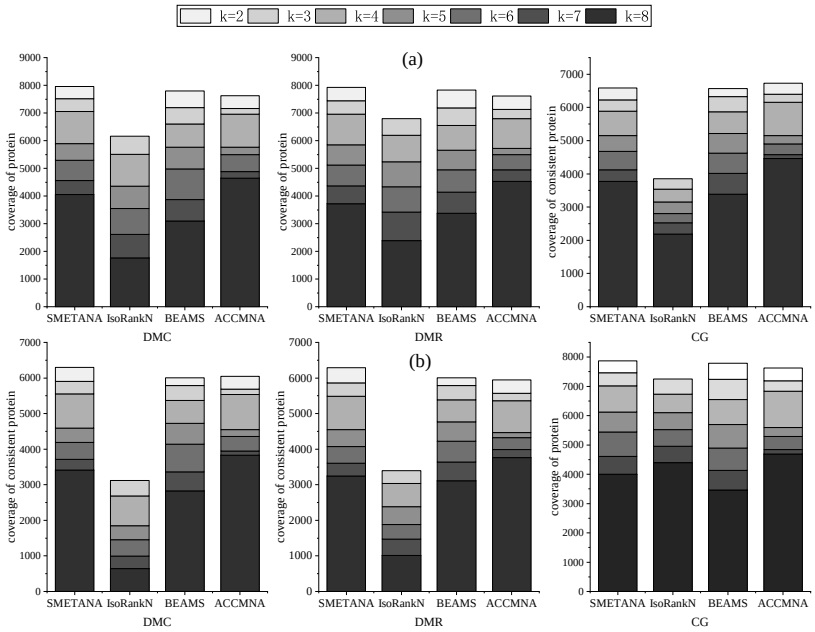

Fig. 2. Performance of various network alignment algorithms in the synthetic network. From left to right, respectively, are the results under the CG, DMC and DMR network model. (a) Node k-Coverage, where k denotes the number of input networks; (b) Consistent Node k-Coverage

[20]. Results on synthetic networks from Tables 2, 3 and 4 showed that our algorithm had the highest CIQ score on the three network sets CG, DMC and DMR, which suggests that our algorithm contained the highest proportion of conservation interaction. SPE stands for Specificity, which was proposed in prior work [37], and it is the proportion of the number of consistent clusters in the number of annotated clusters. Our algorithm had much higher SPE scores than IsoRankN and BEAMS in the three network sets and the highest score in the CG network. The other two network sets ranked second and were very close to the results of the first SMETANA. Sen represents Sensitivity, defined in previous literature [10], which indicates the sensitivity of the alignment. ACCMNA had the highest Sen score among the three network sets. MNE represents the Mean Normalized Entropy, which is an approach to measure the consistency of the alignment. The Mean Normalized Entropy was the average normalized entropy of all the clusters defined by prior work [25]. For a given cluster Cl_i, the normalized entropy is defined as $NE(Cl_i) = -\frac{1}{\log d} \times \sum_{i=1}^{d} p_i \times \log p_i$, where d denotes the number of different GO categories in cluster Cl_i, and p_i represents the proportion of proteins annotated by GO_i in cluster Cl_i. The biological consistency of the alignments increased with lower MNE values. Like the SPE results, the MNE value of our algorithm was the lowest in the CG model network, and our algorithm ranked second on the DMC and DMR datasets; however, the score was very close to that of the first SMETANA. nGOC has been also proposed for the measurement of the alignment consistency by prior researchers [1]. nGOC is an extension of GO Consistency(GOC), and the measurement used in one-to-one pairwise network alignment was extended to measure many-to-many alignment. nGOC is the average value of $nGOC(Cl_i)$ of all the clusters. For a given cluster Cl_i, nGOC is defined as $nGOC(Cl_i) = \frac{|GO_{int}|}{|GO_{uni}|} \times c$, where GO_{int} and GO_{uni} represent the intersection and union of the GO annotation items of proteins in cluster Cl_i, respectively, and c is the number of annotated proteins in cluster Cl_i. The consistency of alignment results increases with higher nGOC values. The ranking of nGOC for ACCMNA was the same as that of SPE and MNE. The main reason for this result may be that the number of nodes and edges of the eight networks in the CG model were the same, which indicated that our algorithm could get a better alignment in the case of a similar network size. However, the alignment generated by our algorithm was more consistent and specific. The result in Table 5 shows that the alignment generated by ACCMNA on real networks had the highest SPE, MNE and nGOC score, which also shows that our algorithm was more specific and consistent. ACCMNA scored second in the CIQ score, and it was only slightly lower than SMETANA. The SMETANA algorithm places high importance on the topology information of the network; therefore, the alignment on the real networks had a high topology score, but several biological scores were low. We believe that SMETANA performed well in the synthetic network, mainly because of its special network characteristics, namely, a relatively ideal network situation, which can explain the result on the synthetic network being slightly higher than ACCMNA. However, the alignment on the real networks was worse than ACCMNA.

To prove that the alignment generated by ACCMNA can perform well both in topological conservation and functional consistency, the product of CIQ and nGOC was plotted for all the algorithms and networks sets. This amplifies the advantages of the ACCMNA algorithm. CIQ is a measure to calculate the proportion of conservative edges between clusters, while nGOC measures the biological consistency of alignment. These are de-

picted in Fig. 4. Although some measures of the ACCMNA algorithm in Tables 2, 3, 4 and 5 were worse than SMETANA, the ACCMNA algorithm received the highest score among all the algorithms when the product of CIQ and nGOC was calculated. This proves that our algorithm can get a good result in both topology and biological consistency.

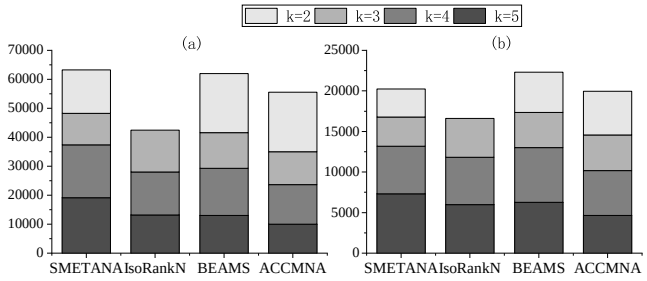

Fig. 3. Node and Consistent Node Coverage of different algorithms: (a)Node k-Coverage; (b) Consistent Node k-Coverage

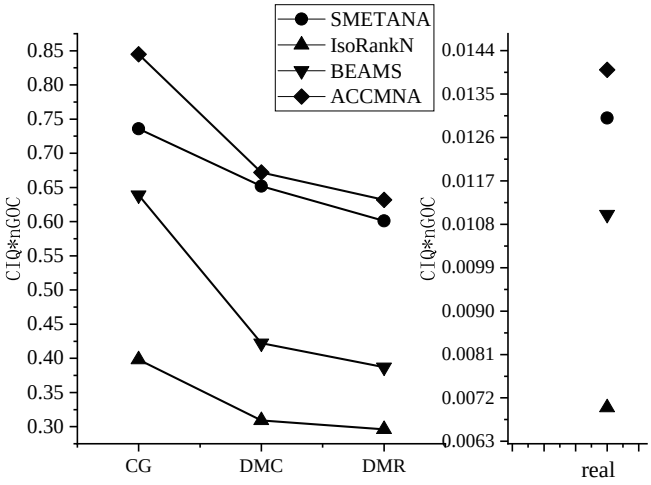

Fig. 4. The product CIQ and nGOC for all the algorithms. The figure on the left shows the scores on the three network models of the synthetic network, and the figure on the right shows the results on the real networks

4.3. Analysis of the Time Complexity

Let V be the set $V_1 \cup \ldots V_k$, and $n = max\{|V_1|, \ldots, |V_k|\}$. As we mentioned before, it takes $O(|V|)$ to calculate the NodeWeight of each node. Thus, the running time of

ACCMNA is mainly determined by the time spent in the main repeat loop. The number of iterations of the loop is $O(|V|)$, and, since the maximum number of output clusters can be $|V|$at most, each iteration finds a new cluster, and the iterations continue until no new clusters remain. The function Select_Candidate requires $O(|V|k^2\Delta_{max})$, where k is the number of PPINs and Δ_{max} the maximum degree in V. The function Generate_Candidate is made on the neighborhood graph of the new cluster. The total running time required by function Generate_Candidate is $O(\Delta(k\Delta_{max}))$, where Δ is the maximum degree in S_β. Function expand_Candidate requires $O(k\Delta)$. Note that the function Generate_Candidate is executed only once in the for-loop, but the functions Generate_Candidate and expand_Candidate in the for-loop are executed $O(|V|)$ times since the number of candidates at a specific iteration can be at most $|V|$. Thus, the overall time complexity of our algorithm is $O(|V|^2\Delta(k\Delta_{max}) + |V|^2k\Delta + |V|^2\Delta) = O(|V|^2k\Delta^2)$.

4.4. Discussion of the Alignment Result

In this section, we discuss the alignment results of the ACCMNA algorithm on the real networks and the synthetic networks along with the comparison experiments with other state-of-the-art algorithms. The above experimental results show that the algorithm proposed in this paper could obtain better alignment results than other state-of-the-art algorithms. The node coverage shows that the ACCMNA algorithm could produce more node coverage with a larger k, indicating its ability to produce higher quality alignment and more useful biological information. Moreover, the measurement results of the biological consistency, specificity and sensitivity show that the scores of our algorithm ranked high among several algorithms, which indicated that the alignment results produced by ACCMNA had good biological significance. When topological and biological consistency scores are combined, the alignment results of the ACCMNA algorithm can reach the balance between topological and biological consistency.

5. Conclusion

To solve the NP-hard problem of network alignment and the computational complexity of MNA gradually increasing with the increase of network size, a new and efficient ACCMNA aligner was proposed in this paper, which combines topology and sequence similarity information for alignment generation. ACCMNA is an aligner that utilizes the importance of nodes and combines clustering methods to produce better alignment results. The basic framework of ACCMNA is the seed-and-extend search method. The algorithm utilizes the degree and neighbors of nodes to calculate the node weight, which aims to reduce the complexity of alignment and make as many similar nodes as possible that can be successfully mapped by combining the clustering method to search the alignment. The ACCMNA algorithm was compared against excellent and representative MNA algorithms on both real and synthetic networks. Extensive evaluations showed that the ACCMNA algorithm performed well both in topological conservation and functional consistency. The superior experimental results also reflected that the ACCMNA algorithm is an efficient and accurate aligner that can be applied to PPINs of various sizes within an acceptable running time. In addition to proving the effectiveness of the method proposed in this paper, the alignment results generated by ACCMNA are of reference significance for the

study of real networks. Moreover, it has the potential to be extended to other types of complex networks in the future, rather than remain limited to PPINs.

References

1. Alkan, F., Erten, C.: Beams: Backbone extraction and merge strategy for the global many-to-many alignment of multiple ppi networks. Bioinformatics 30(4), 531–539 (2013)
2. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. Journal of Molecular Biology 215(3), 403–410 (1990)
3. Aranda, B., Achuthan, P., Alam-Faruque, Y., Armean, I., Bridge, A., Derow, C., Feuermann, M., Ghanbarian, A.T., Kerrien, S., Khadake, J.: The intact molecular interaction database in 2010. Nucleic Acids Research 38(suppl_1), 525–531 (2010)
4. Bobby-Joe, B., Chris, S., Teresa, R., Lorrie, B., Ashton, B., Michael, L., Rose, O., Lackner, D.H., Jürg, B., Valerie, W.: The biogrid interaction database: 2008 update. Nucleic Acids Research 36(suppl_1), 637–640 (2008)
5. Braovic, M., Stipanicev, D., Seric, L.: Retinal blood vessel segmentation based on heuristic image analysis. Computer Science and Information Systems 16(1), 227–245 (2019)
6. Ceol, A., Chatr Aryamontri, A., Licata, L., Peluso, D., Briganti, L., Perfetto, L., Castagnoli, L., Cesareni, G.: Mint, the molecular interaction database: 2009 update. Nucleic Acids Research 38(5), 32–39 (2010)
7. Cook, S.: The complexity of theorem-proving procedures. In: Proc Acm Symposium on the Theory of Computation. pp. 151–158 (1971)
8. Djeddi, W.E., Yahia, S.B., Nguifo, E.M.: A novel computational approach for global alignment for multiple biological networks. IEEE/ACM Transactions on Computational Biology Bioinformatics 15(6), 2060–2066 (2018)
9. Elmsallati, A., Clark, C., Kalita, J.: Global alignment of protein-protein interaction networks: A survey. IEEE/ACM Transactions on Computational Biology Bioinformatics 13(4), 689–705 (2016)
10. Flannick, J., Novak, A., Do, C.B., Srinivasan, B.S., Batzoglou, S.: Automatic parameter learning for multiple local network alignment. Journal of Computational Biology 16(8), 1001–1022 (2009)
11. Flannick, J., Novak, A., Srinivasan, B.S., McAdams, H.H., Batzoglou, S.: Graemlin: general and robust alignment of multiple large interaction networks. Genome Research 16(9), 1169–1181 (2006)
12. Gao, J., Song, B., Ke, W., Hu, X.: Balanceali: multiple ppi network alignment with balanced high coverage and consistency. IEEE Transactions on Nanobioscience 16(5), 333–340 (2017)
13. Gligorijević, V., Malod-Dognin, N., Pržulj, N.: Fuse: multiple network alignment via data fusion. Bioinformatics 32(8), 1195–1203 (2016)
14. Hashemifar, S., Xu, J.: Hubalign: an accurate and efficient method for global alignment of protein–protein interaction networks. Bioinformatics 30(17), i438–i444 (2014)
15. Hu, J., He, J., Gao, Y., Zheng, Y., Shang, X.: Netcoffee2: A novel global alignment algorithm for multiple ppi networks based on graph feature vectors. In: International Conference on Intelligent Computing. pp. 241–246 (2018)
16. Hu, J., Kehr, B., Reinert, K.: Netcoffee: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks. Bioinformatics 30(4), 540–548 (2014)
17. Hubbard, T.J., Aken, B.L., Ayling, S., Ballester, B., Beal, K., Bragin, E., Brent, S., Chen, Y., Clapham, P., Clarke, L., et al.: Ensembl 2009. Nucleic Acids Research 37(suppl_1), D690–D697 (2009)

18. Jeong, H., Yoon, B.J.: Accurate multiple network alignment through context-sensitive random walk. In: BMC Systems Biology. vol. 9, pp. 1–12. Springer (2015)
19. Jiang, P., Singh, M.: Spici: a fast clustering algorithm for large biological networks. Bioinformatics 26(8), 1105–1111 (2010)
20. Kazemi, E., Grossglauser, M.: Mpgm: Scalable and accurate multiple network alignment. IEEE/ACM Transactions on Computational Biology and Bioinformatics 17(6), 2040–2052 (2019)
21. Keshava Prasad, T.t., Goel, R., Kandasamy, K., Keerthikumar, S., Kumar, S., Mathivanan, S., Telikicherla, D., Raju, R., Shafreen, B., Venugopal, A., et al.: Human protein reference database—2009 update. Nucleic Acids Research 37(suppl_1), D767–D772 (2009)
22. Kim, W.K., Marcotte, E.M.: Age-dependent evolution of the yeast protein interaction network suggests a limited role of gene duplication and divergence. PLoS Comput Biol 4(11), e1000232 (2008)
23. Kuchaiev, O., Milenković, T., Memišević, V., Hayes, W., Pržulj, N.: Topological network alignment uncovers biological function and phylogeny. Journal of the Royal Society Interface 7(50), 1341–1354 (2010)
24. Kuchaiev, O., Pržulj, N.: Integrative network alignment reveals large regions of global network similarity in yeast and human. Bioinformatics 27(10), 1390–1396 (2011)
25. Liao, C.S., Lu, K., Baym, M., Singh, R., Berger, B.: Isorankn: spectral methods for global alignment of multiple protein networks. Bioinformatics 25(12), i253–i258 (2009)
26. Liu, X., Zhuang, C., Murata, T., Kim, K.S., Kertkeidkachorn, N.: How much topological structure is preserved by graph embeddings? Computer Science and Information Systems 16(2), 597–614 (2019)
27. Malod-Dognin, N., Pržulj, N.: L-graal: Lagrangian graphlet-based network aligner. Bioinformatics 31(13), 2182–2189 (2015)
28. Memišević, V., Pržulj, N.: C-graal: Common-neighbors-based global graph alignment of biological networks. Integrative Biology 4(7), 734–743 (2012)
29. Milenković, T., Ng, W.L., Hayes, W., Pržulj, N.: Optimal network alignment with graphlet degree vectors. Cancer Informatics 9, CIN–S4744 (2010)
30. Neyshabur, B., Khadem, A., Hashemifar, S., Arab, S.S.: Netal: a new graph-based method for global alignment of protein–protein interaction networks. Bioinformatics 29(13), 1654–1662 (2013)
31. Notredame, C., Higgins, D.G., Heringa, J.: T-coffee: A novel method for fast and accurate multiple sequence alignment. Journal of Molecular Biology 302(1), 205–217 (2000)
32. Park, D., Singh, R., Baym, M., Liao, C.S., Berger, B.: Isobase: a database of functionally related proteins across ppi networks. Nucleic Acids Research 39(suppl_1), D295–D300 (2010)
33. Pastor-Satorras, R., Smith, E., Solé, R.V.: Evolving protein interaction networks through gene duplication. Journal of Theoretical biology 222(2), 199–210 (2003)
34. Phan, H.T., Sternberg, M.J.: Pinalog: a novel approach to align protein interaction networks—implications for complex detection and function prediction. Bioinformatics 28(9), 1239–1245 (2012)
35. Radu, A., Charleston, M.: Node handprinting: a scalable and accurate algorithm for aligning multiple biological networks. Journal of Computational Biology 22(7), 687–697 (2015)
36. Sahraeian, S.M.E., Yoon, B.J.: A network synthesis model for generating protein interaction network families. PloS One 7(8), e41474 (2012)
37. Sahraeian, S.M.E., Yoon, B.J.: Smetana: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. PloS One 8(7), e67995 (2013)
38. Salwinski, L., Miller, C.S., Smith, A.J., Pettit, F.K., Bowie, J.U., Eisenberg, D.: The database of interacting proteins: 2004 update. Nucleic Acids Research 32(suppl_1), D449–D451 (2004)
39. Saraph, V., Milenković, T.: Magna: maximizing accuracy in global network alignment. Bioinformatics 30(20), 2931–2940 (2014)

40. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks with application to functional orthology detection. Proceedings of the National Academy of Sciences 105(35), 12763–12768 (2008)
41. Vázquez, A., Flammini, A., Maritan, A., Vespignani, A.: Modeling of protein interaction networks. Complexus 1(1), 38–44 (2003)
42. Vijayan, V., Milenković, T.: Multiple network alignment via multimagna++. IEEE/ACM Transactions on Computational Biology and Bioinformatics 15(5), 1669–1682 (2017)
43. Vijayan, V., Saraph, V., Milenković, T.: Magna++: Maximizing accuracy in global network alignment via both node and edge conservation. Bioinformatics 31(14), 2409–2411 (2015)
44. Yin, A., Chen, C., Hu, D., Huang, J., Yang, F.: An improved heuristic-dynamic programming algorithm for rectangular cutting problem. Computer Science and Information Systems 17(3), 717–735 (2020)

Jing Chen, born in 1977. Ph. D., associate professor, her research interests include complex networks, indoor positioning, etc.

Jia Huang, born in 1996. Master degree candidate, her research interests include complex networks.

Received: September 09, 2020; Accepted: April 06, 2021.