Research Article

Pairwise Biological Network Alignment Based on Discrete Bat Algorithm

Jing Chen [1,2] **Ying Zhang** [1] **and Jin-Fang Xia** [1]

[1]*School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, Jiangsu 214122, China*
[2]*Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computing Intelligence, Jiangnan University, Wuxi, Jiangsu 214122, China*

Correspondence should be addressed to Jing Chen; chenjing@jiangnan.edu.cn

The development of high-throughput technology has provided a reliable technical guarantee for an increased amount of available data on biological networks. Network alignment is used to analyze these data to identify conserved functional network modules and understand evolutionary relationships across species. Thus, an efficient computational network aligner is needed for network alignment. In this paper, the classic bat algorithm is discretized and applied to the network alignment. The bat algorithm initializes the population randomly and then searches for the optimal solution iteratively. Based on the bat algorithm, the global pairwise alignment algorithm BatAlign is proposed. In BatAlign, the individual velocity and the position are represented by a discrete code. BatAlign uses a search algorithm based on objective function that uses the number of conserved edges as the objective function. The similarity between the networks is used to initialize the population. The experimental results showed that the algorithm was able to match proteins with high functional consistency and reach a relatively high topological quality.

1. Introduction

With the development of high-throughput technology, such as the yeast two-hybrid system [1], an increasing amount of biological data are being modeled into biological networks. According to the different meanings of nodes and edges when the networks are built, the networks can be classified as protein-protein interaction (PPI) networks [2], gene regulatory networks [3], and metabolic networks [4]. Biological systems complete a series of biological processes through PPI, rendering the study of PPI networks of great significance [5]. Network alignment is a more efficient method for analyzing biological networks, in comparison to biological experiments [6], and can be used to discover functional modules among networks [7] and predict the unknown function of proteins [8]. Homologous protein pairs of less-studied biological networks can be discovered by comparison with biological networks that have been more extensively studied, to detect potential functions of unknown proteins [9, 10].

An aspect of the difference of biological network alignment from graph matching is that the goal is the production of biologically meaningful alignment results. Therefore, the sequence similarity, which is an internal property of proteins [11, 12], between network nodes needs to be used in biological alignment. Moreover, the topological structure information of the network, which is an external attribute of proteins, can also be incorporated. The majority of the PPI network alignment algorithms use a combination of sequence similarity and topological similarity [13–15].

Network alignment can be divided into local and global alignment, according to the alignment range [16, 17]. The purpose of local network alignment is to discover similar local subgraphs among networks [18]. MaWISH [19], Graemlin [20], PathBLAST [21], NetworkBLAST [22], and LePrimAlign [23] are examples of algorithms that can be used to solve the local network alignment problem. However, the disadvantage of local network alignment is that one module may be similar to several modules and a protein

may be mapped to dissimilar protein nodes [24]. Therefore, the concept of global network alignment has been proposed [25]. Global network alignment is aimed at discovering the overall similar mapping relationship between networks [26]. At present, a high number of global alignment algorithms have been proposed, such as IsoRank [25], GRAAL family algorithms [27–30], NETAL [31], MAGNA [32], MAGNA++ [33], SANA [34], ModuleAlign [35], AligNet [36], and IBNAL [37]. In IsoRank, the similarity of nodes between the networks calculated by the PageRank algorithm is used to guide the greedy algorithm to complete the alignment. The GRAAL family of algorithms includes GRAAL, MI-GRAAL, C-GRAAL, and L-GRAAL, all of which are based on the Graphlet degree similarity. NETAL first constructs an alignment score matrix, and then, a greedy strategy is adopted to update the scores until all nodes in the first network are aligned with nodes in the second network. MAGNA is an objective function-based alignment algorithm that uses a genetic algorithm for searching. MAGNA++ is the optimization of MAGNA that optimizes both structure and sequence similarities and provides a friendly graphical interface. SANA is also objective function-based and uses the simulated annealing search algorithm for alignment. Both the ModuleAlign and AligNet algorithms incorporate the idea of modularity into network alignment. IBNAL develops a clique-based index to measure the topology of the proteins.

Within the framework of objective function-based search algorithms, this paper discretizes the bat algorithm [38] and proposes the BatAlign algorithm. First, the similarity matrix is constructed by the combination of biological similarity and topological similarity information. The sequence similarity adopts BLAST bit-score [39] and evaluates the similarity of the network structure by considering the neighbors of the nodes, to further improve the similarity between networks. The greedy search is then used to generate the initial population, and the pair of nodes with the maximum score is chosen and aligned to each other. Finally, the alignment results are obtained by initial population optimization. By building a coarse similarity score matrix to guide the initialization, BatAlign can shorten the search time to convergence, in comparison to random initialization of the population. Our main contributions are summarized as follows.

(1) We propose BatAlign which uses a discrete bat algorithm for network alignment. The main idea of BatAlign is to iteratively update the bat position under the guidance of bat velocity

(2) The network topology information and node sequence information are combined to calculate the node similarity. The node similarity guides the construction of the initial population. With the initialization mechanism, BatAlign can obtain a good biological score and relatively high topological score

The related work on network alignment is introduced in the first section. The framework and theory of the BatAlign algorithm are explained in the second section. In the third section, BatAlign is compared with other state-of-

the-art algorithms based on synthetic and real networks. The work of this paper and future prospects are presented in Section 4.

2. Materials and Methods

2.1. Problem Definition.
Assume that the two networks to be aligned are $G_1(V_1, E_1)$ and $G_2(V_2, E_2)$, where V_1, V_2 are the node sets of networks G_1, G_2, respectively, and E_1, E_2 are the edge sets of G_1, G_2, respectively. Without loss of generalization, assuming that $|V_1| \leq |V_2|$, the small network G_1 is the source network and the large network G_2 is the target network. Global network alignment finds a mapping relationship $f : V_1 \longrightarrow V_2$, which aligns the nodes in the small network to the nodes in the large network one by one, to maximize the overall similarity between the networks.

The similarity of the node pairs between networks usually combines the similarity of topology and sequence. In this paper, the topology of the network is considered through the neighbors of a node, and the sequence similarity is combined to generate the similarity matrix between the networks:

$$S = \alpha B + (1 - \alpha) A_1 B A_2^T, \quad \alpha \in [0, 1], \tag{1}$$

where S represents the similarity matrix between networks, B represents the sequence similarity matrix of nodes between networks, and A_1 and A_2 that note the topological structure of the node represent the adjacency matrix of networks G_1 and G_2, respectively.

2.2. Bat Algorithm.
The bat algorithm [38] is a swarm intelligence optimization algorithm that simulates the echolocation behaviour of bats. The initial population is generated randomly, and then, the optimal solution is iteratively searched. The new solution is generated during searching by adjusting frequency f (Equations (3) and (4)); when the rate of pulse emission r is smaller than a random number, the local solution is generated around the selected best solution (Equation (5)):

$$f_i = f_{\min} + (f_{\max} - f_{\min})\beta, \quad \beta \in [0, 1], \tag{2}$$

$$v_i^t = v_i^{t-1} + (x_i^t - x_*)f_i, \tag{3}$$

$$x_i^t = x_i^{t-1} + v_i^t, \tag{4}$$

$$x_{\text{new}} = x_{\text{old}} + \varepsilon A^t, \quad \varepsilon \in [-1, 1]. \tag{5}$$

The bats can adjust the loudness (Equation (6)) and the rate of pulse emission (Equation (7)):

$$A_i^t = \theta A_i^{t-1}, \quad \theta \in (0, 1), \tag{6}$$

$$r_i^t = r_i^t[1 - \exp(-\gamma t)], \quad \gamma > 0. \tag{7}$$

2.3. Algorithm Overview

2.3.1. Individual and Population Initialization.
The BatAlign algorithm applies the discretized bat algorithm to the

(a)   (b)

	1 (b1)	2 (b2)	3 (b3)	4 (b4)	5 (b5)	6 (b6)	7 (b7)	8 (b8)
1 (a1)	0.49	0	0.26	0	0	0	0.19	0
2 (a2)	0	0	0.78	0	0	0	0	0
3 (a3)	0	0	0.65	0.53	0	0	0	0
4 (a4)	0	0	0	0	0.48	0	0	0.32

(c)

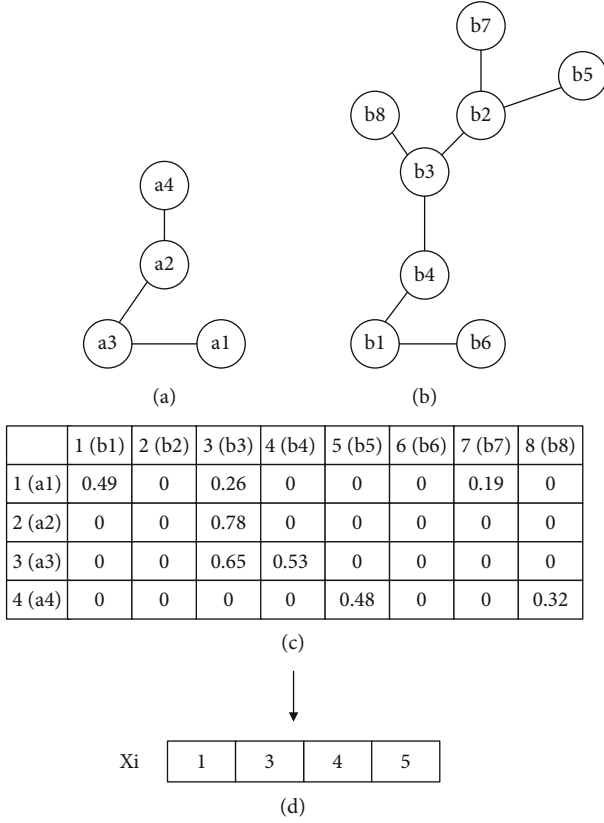| Xi	1	3	4	5

(d)

FIGURE 1: Initialization of individual position. In this figure, (a, b) are the source network and target network, respectively; (c) is the similarity matrix. Assuming the similarity matrix has been obtained, according to the matrix, the most similar node pair is chosen, and then, a2 is mapped to b3. Nodes are not aligned repeatedly; then, the most similar node pair is a3 and b4; a3 is mapped to b4; in a similar way, a1 is mapped to b1, and a4 is mapped to b5; thus, the individual position can be obtained as shown in (d).

network alignment, which is reflected in two aspects: position coding and velocity coding.

Position discretization: in both networks, the nodes are numbered from one and each node number is unique in its own network. Each individual position in the population represents an alignment of the entire network, the position is a vector X with n_1 components, and the entries are members of G_2.

Velocity discretization: an individual velocity is represented by a vector V with n_1 components, whose entries are 0 or 1. The values 0 or 1 were used to represent the flying velocity of one node in a network, where 0 means keeping the solution of this node, namely, not to fly, and 1 means that the solution of this node can randomly fly.

Individual initialization: the position and velocity of each individual need to be initialized. The initial position is an alignment generated by the greedy algorithm under the guidance of the similarity matrix. For example, assuming the similarity matrix of two networks has been obtained by Equation (1), the similarity matrix is shown in Figure 1(c). For each node in the source network, BatAlign identifies

(a)

xi	1	3	4	5

(b)

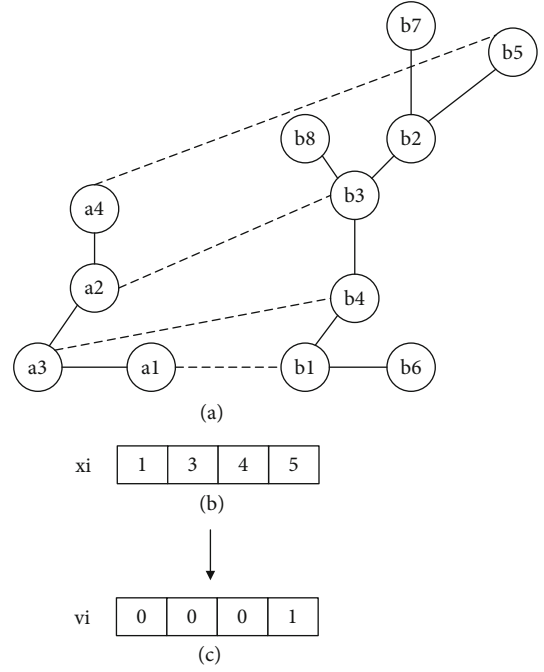| vi	0	0	0	1

(c)

FIGURE 2: Initialization of individual velocity. Assuming the individual position is as shown in (b), according to the position, the edges (a1, a3) and (a2, a3) are conserved; the nodes a1, a2, and a3 are conserved; thus, the velocity of these nodes is 0. Therefore, the individual velocity can be obtained as shown in (c).

the node with the highest similarity; therefore, position x_i is obtained, as shown in Figure 1(d). The method for initializing the velocity is given in Equation (8), and the velocity of the conserved node is 0. For example, assuming the individual position, shown in Figure 2(b), has been obtained, the velocity is obtained as shown in Figure 2(c):

$$
v_k = \begin{cases} 0, & \exists (u_k, u_l) \in E_1, \exists (\varnothing(u_k), \varnothing(u_l)) \in E_2, \\ 1, & \text{otherwise.} \end{cases} \tag{8}
$$

Due to the incompleteness of the similarity between network nodes, it is possible that simple greedy algorithms may not directly align all the nodes. Therefore, unaligned nodes were randomly mapped to generate all the individuals in the population (Figure 3). That is, only the nodes that have similarity are aligned first, while the nodes that do not have similarity are aligned randomly.

2.3.2. Individual and Population Iteration. The individual iteration process is composed of two parts: generating new individuals and updates.

Generating new individuals includes two parts: updating velocity and position. For an individual, the method for updating the velocity is given in

$$
v_{ij}^t = \begin{cases} v_{ij}^{t-1}, & \text{if } x_{ij} = x_{*j}, \\ {}^{\circ}v_{ij}^{t-1}, & \text{otherwise,} \end{cases} \tag{9}
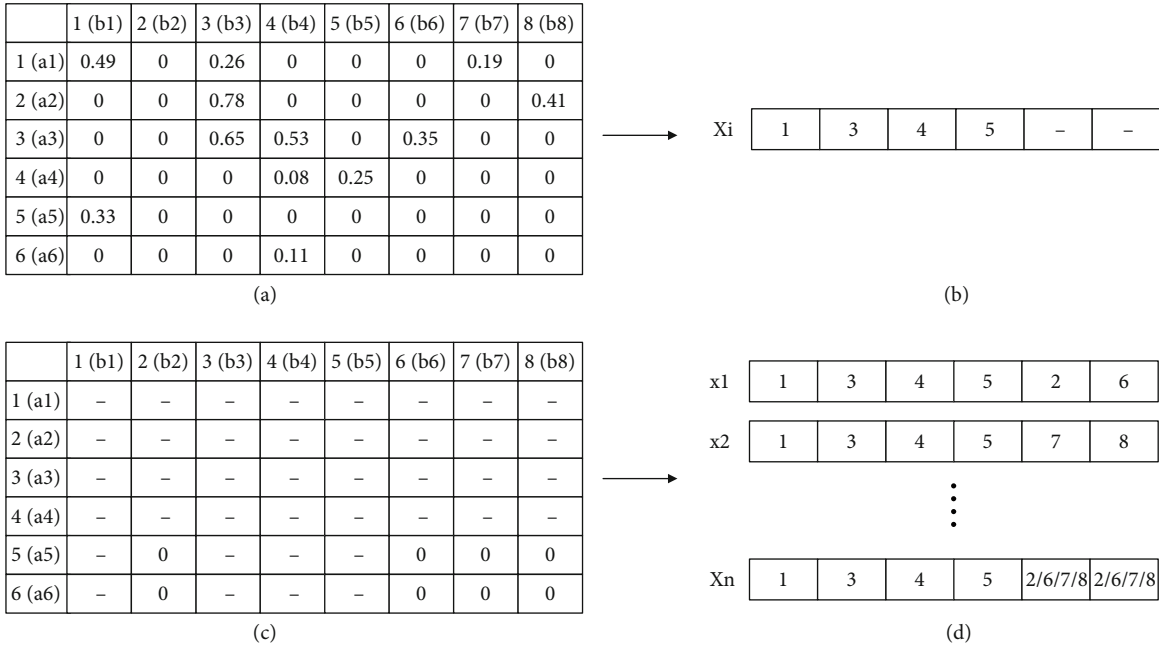$$

	1 (b1)	2 (b2)	3 (b3)	4 (b4)	5 (b5)	6 (b6)	7 (b7)	8 (b8)
1 (a1)	0.49	0	0.26	0	0	0	0.19	0
2 (a2)	0	0	0.78	0	0	0	0	0.41
3 (a3)	0	0	0.65	0.53	0	0.35	0	0
4 (a4)	0	0	0	0.08	0.25	0	0	0
5 (a5)	0.33	0	0	0	0	0	0	0
6 (a6)	0	0	0	0.11	0	0	0	0

(a)

X_i

1	3	4	5	–	–

(b)

	1 (b1)	2 (b2)	3 (b3)	4 (b4)	5 (b5)	6 (b6)	7 (b7)	8 (b8)
1 (a1)	–	–	–	–	–	–	–	–
2 (a2)	–	–	–	–	–	–	–	–
3 (a3)	–	–	–	–	–	–	–	–
4 (a4)	–	–	–	–	–	–	–	–
5 (a5)	–	0	–	–	–	0	0	0
6 (a6)	–	0	–	–	–	0	0	0

(c)

x_1

1	3	4	5	2	6

x_2

1	3	4	5	7	8

⋮

X_n

1	3	4	5	2/6/7/8	2/6/7/8

(d)

FIGURE 3: Initialization of individual position with population size n. Assuming the similarity matrix in (a) has been obtained, according to the matrix, the position is obtained as shown in (b); a2 is mapped to b3; a3 is mapped to b4; a1 is mapped to b1; and a4 is mapped to b5. As nodes are not aligned repeatedly, the matrix is obtained as shown in (c). In this case, the similarities of a5 with other nodes are 0; thus, a5 is randomly aligned with b2, b6, b7, or b8; and a6 is also aligned randomly. By mapping unaligned nodes randomly, the positions of individuals with population size n are obtained as shown in (d).
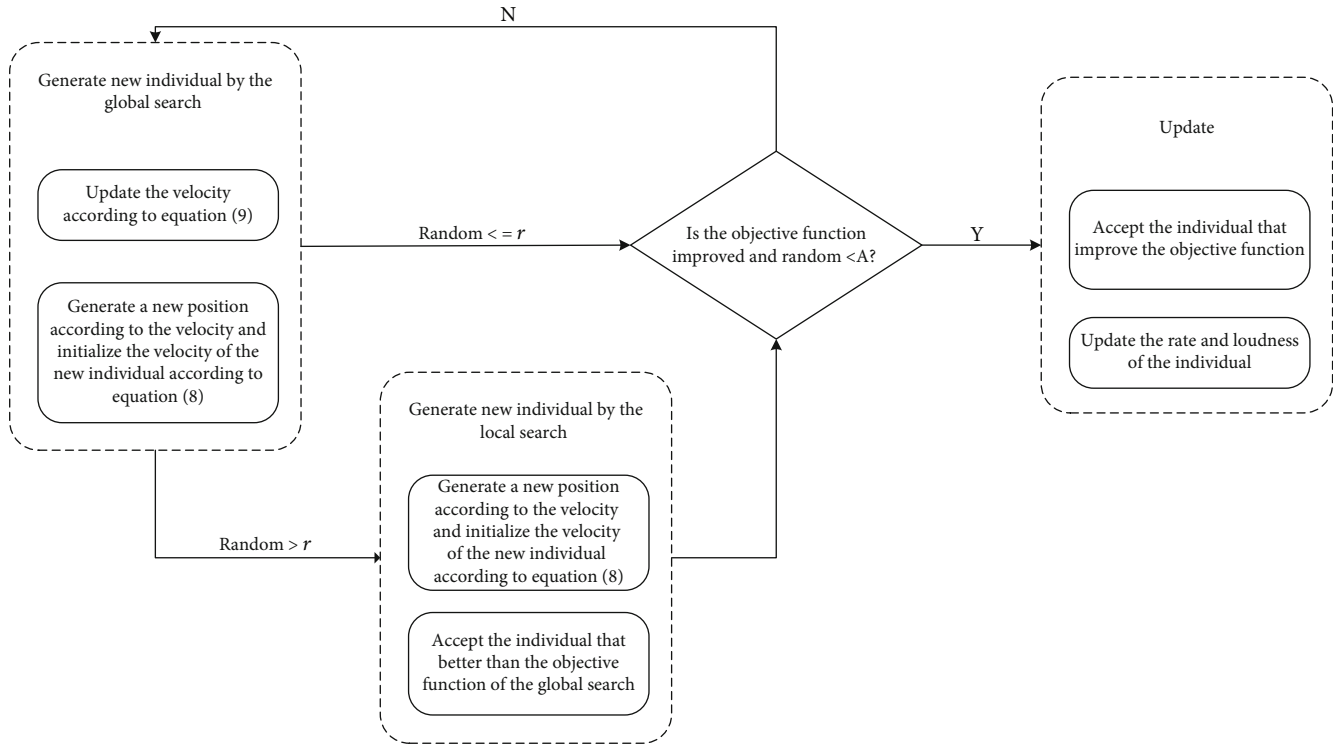

FIGURE 4: The individual iteration process.
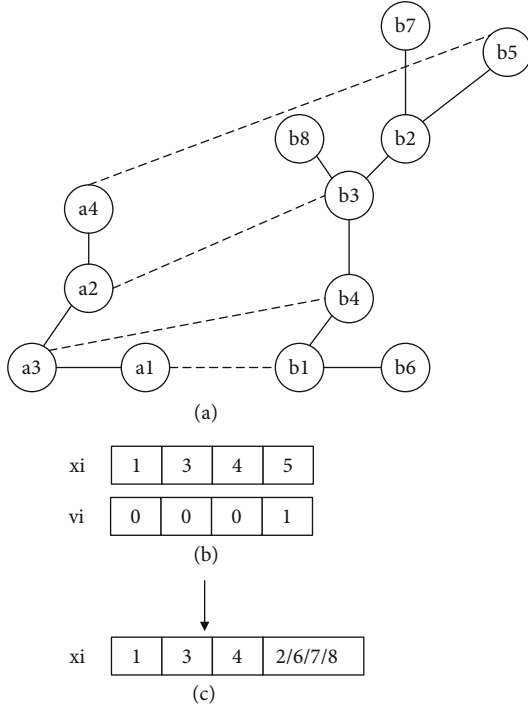
(a)

(b)

(c)

FIGURE 5: Example of global search. Assuming an individual position with the velocity shown in (b), according to the global search method, the node for which the velocity is 1 is changed through randomly choosing an unaligned node from $U = \{2, 6, 7, 8\}$; thus, the updated position is obtained as shown in (c).

where v_{ij}^t is the velocity of the ith individual on the jth dimension during the tth iteration, x_{ij} is the position of the ith individual on the jth dimension, and x_{*j} is the optimal solution on the jth dimension. The calculation of $°v_{ij}^{t-1}$ is as follows in

$$°v_{ij}^{t-1} = \begin{cases} °f_i, & \text{if } v_{ij}^{t-1} = 1, \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

where $°f_i$ is calculated as follows in

$$°f_i = \begin{cases} 1, & \text{if } f_i > 0.5 * (f_{max} - f_{min}), \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where f_{max} and f_{min} represent the maximum and minimum frequency, respectively. In BatAlign, f_{max} is set equal to 1 and f_{min} equal to 0. The calculation of f_i is Equation (2).

The method to update the position depends on the new velocity. An individual velocity entry of 1 means that this node needs to be remapped and the individual position of this node needs to be changed. Updating the position includes two processes: global search and local search. Each individual position is updated through global search. If the rate of the individual is less than a random number between 0 and 1, BatAlign performs a local search. On the other hand, if the objective function value of the position gener-
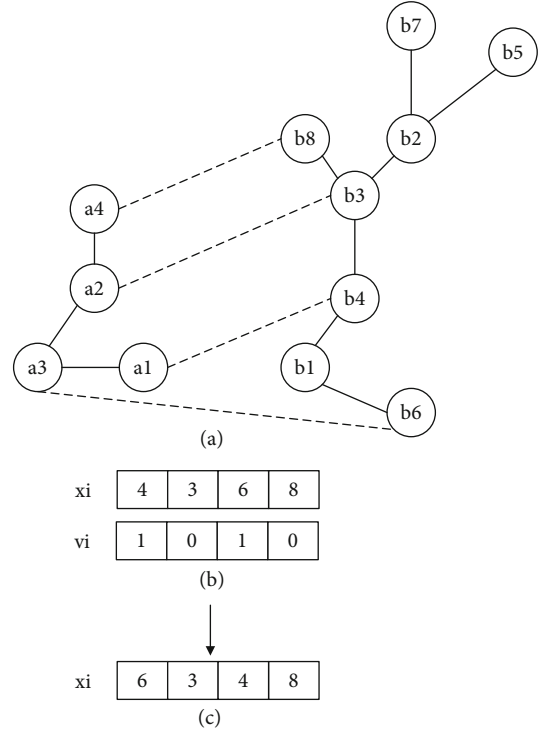

(a)

(b)

(c)

FIGURE 6: Example of local search. Assuming an individual position with the velocity shown in (b), according to the local search method, the node for which the velocity is 1 is changed through randomly choosing the node for which the velocity equals to 1 from $C = \{4, 6\}$; thus, the updated position is obtained as shown in (c).

ated by the local search is larger than the value of the position generated by the global search, then the local search position is accepted. The new position is obtained through global and local search, and if the objective function value of the new position is larger than the old position and the current loudness of the individual is greater than the random number between 0 and 1, BatAlign accepts this new position of the individual. Figure 4 shows the process of individual iteration.

The global search method is given in Equation (12). The node with a velocity of 0 is reserved, and the remaining nodes are those to be aligned and put into set U. The selection operation is represented by σ. Figure 5 shows the global search that generates a new position:

$$x_{ij}^t = \begin{cases} \sigma(U), & \text{if } v_{ij}^t = 1, \\ x_{ij}^{t-1}, & \text{otherwise}. \end{cases} \tag{12}$$

The local search method is given in Equation (13), where the set C is composed of nodes with a velocity of 1. An example of local search is shown in Figure 6:

$$x_{ij}^t = \begin{cases} \sigma(C), & \text{if } v_{ij}^t = 1, \\ x_{ij}^{t-1}, & \text{otherwise}. \end{cases} \tag{13}$$

The update operation is performed when the current loudness of the individual is greater than the random number between 0 and 1 and the value of the objective function of the new position is larger. The objects of the update operation include the velocity, position, rate, and loudness. The objective function used in this study is the number of conserved edges. The more conserved edges, the larger the objective function.

In each iteration, the individual with the highest score is chosen as the optimal solution in the population. When BatAlign runs T iterations or the optimal solution remains the same after N times, the optimal solution is output as the final alignment.

3. Results and Discussion

3.1. Experimental Dataset. Synthetic networks were used, retrieved from the NAPAbench2 [40], which was a synthetically constructed network alignment benchmark including three types of networks: Crystal Growth (CG), Duplication Mutation Complementation (DMC), and Duplication with Random Mutation (DMR). The number of nodes and edges of the three networks is shown in Table 1.

The dataset of the real networks was obtained from the BioGRID database [41]. The test species includes the Rattus norvegicus (RN), Schizosaccharomyces pombe (SP), Caenorhabditis elegans (CE), and Mus musculus (MM). The information of the real networks is provided in Table 2.

The similarity scores in the BioGRID datasets were BLAST bit scores computed by the BLAST package on NCBI (https://www.ncbi.nlm.nih.gov/). Gene Ontology terms [42] were used as standard functional annotations, and GO annotations were extracted from NCBI's Entrez Gene database [43].

3.2. Evaluation Metrics. The network alignment quality was evaluated in two aspects: topology and biology. The edge conservation under an alignment has been evaluated using three measures so far: Edge Correctness (EC) [27], Induced Conservative Structure (ICS) [44], and Symmetric Substructure Score (S^3) [32]. S^3 has been shown to be superior to EC and ICS, since EC only penalizes alignments from sparse graph regions to dense graph regions. ICS only penalizes alignments from dense graph regions to sparse graph regions; however, S^3 considers both aspects simultaneously. S^3 was used to evaluate the topological similarity of an alignment. The higher the S^3 value is, the more analogous structure the alignment has conserved:

S^3 was proposed in MAGNA, and it is formulated as

$$S^3 = \frac{|f(E_1)|}{|E_1| + |E_2(G_2(f(V_1)))| - |f(E_1)|}, \quad (14)$$

where $f : V_1 \longrightarrow V_2$ represents the alignment and $|f(E_1)|$ is the number of edges from the smaller network G_1 that is conserved by alignment. The formulation of $f(E_1)$ is as follows in Equation (15). $|E_2(G_2(f(V_1)))|$ is the number of edges from the induced subnet of G_2 with the aligned node set. The formulation of $f(V_1)$ is as follows in Equation (16):

TABLE 1: Information of the synthetic networks. $\bar{k}$ represents the average degree of the network.

Data	Network	Nodes	Edges	$\bar{k}$
CG	A	3000	11987	7.991
	B	4000	15987	7.994
DMC	A	3000	23700	15.8
	B	4000	34206	17.103
DMR	A	3000	30910	20.607
	B	4000	44386	22.193

TABLE 2: Information of the real networks. The species names are mentioned in the first col of the table. The number of nodes and edges are presented in the second and third col, respectively. The fourth col represents the average degree of the network.

Network	Nodes	Edges	$\bar{k}$
RN	2682	4604	3.433
SP	3269	10953	6.701
CE	6058	16463	5.435
MM	7282	21811	5.990

$$f(E_1) = \{(f(u), f(v)) \mid (u, v) \in E_1 : (f(u), f(v)) \in E_2\}, \quad (15)$$

$$f(V_1) = \{f(u) \mid u \in V_1, f(u) \in V_2\}. \quad (16)$$

The network alignment biological quality was evaluated by two measures, including Gene Ontology consistency (GOC) [45] and Average Functional Similarity (AFS) [46]. The high GOC and AFS values indicate the high functional consistency of the alignment.

GOC is based on the Gene Ontology (GO) consistency of the aligned pairs of proteins. GO terms describe some biological properties of a protein such as Cellular Component (CC), Molecular Function (MF), and Biological Process (BP). Proteins with similar GO terms are supposed to be functionally similar. GOC can be computed as follows in

$$\text{GOC}(u, f(u)) = \sum_{u \in V_1} \frac{|\text{GO}(u) \cap \text{GO}(f(u))|}{|\text{GO}(u) \cup \text{GO}(f(u))|}, \quad (17)$$

where $\text{GO}(u)$ denotes the set of GO terms annotating a protein u.

AFS is calculated based on the semantic similarity of the GO terms and depends on the distance between them in the ontology. Semantic similarity measures can be used to calculate the functional similarity in each category of BP, MF, and CC. The semantic similarity is calculated using a graph-based method, Wang. The detailed work of the Wang method is illustrated in [47]. AFS is defined as follows in

$$\text{AFS}_c = \frac{1}{|V_1|} \times \sum_{u \in V_1} s_c(u, f(u)), \quad (18)$$

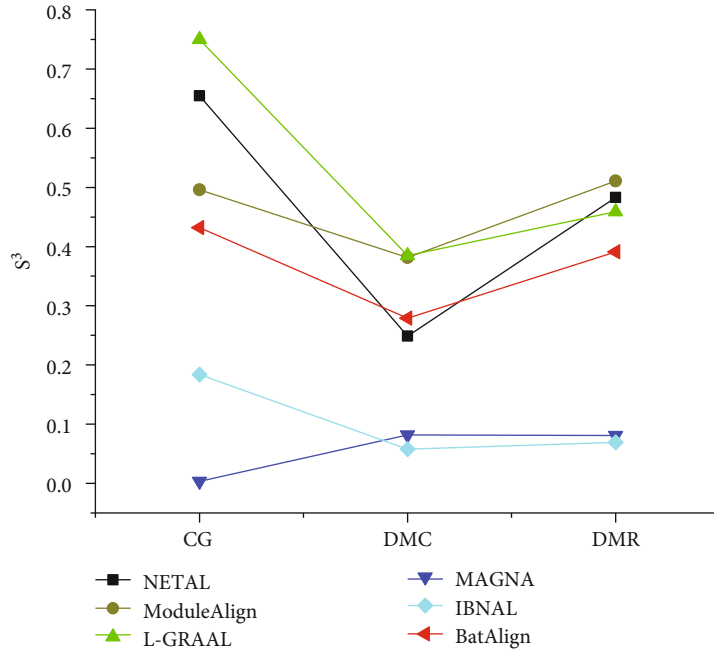where s_c is the semantic similarity of nodes u and $f(u)$, for

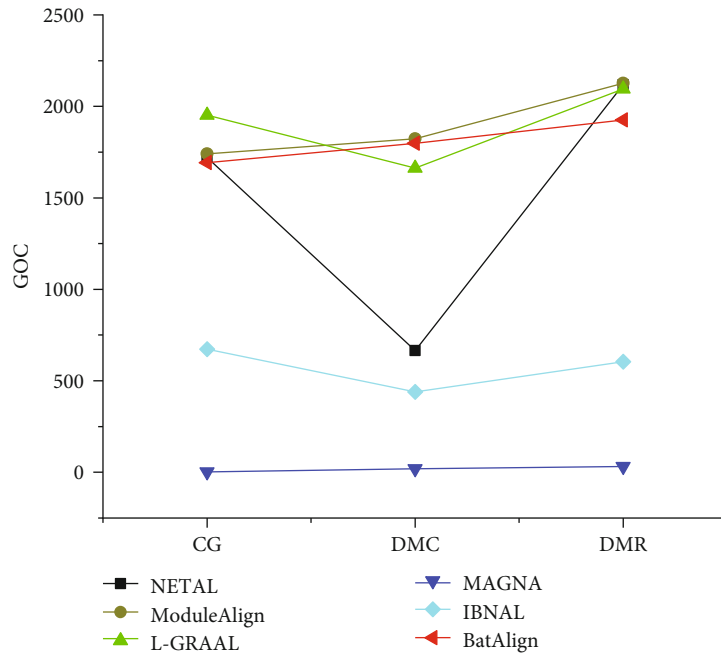FIGURE 7: S^3 of the different algorithms on synthetic networks.

FIGURE 8: GOC of the different algorithms on synthetic networks.

type $c(c \in \{BP, MF, CC\})$, calculated by Wang. GoSemSim [48] was used for semantic similarity calculation.

3.3. Experimental Results and Analysis. The number of iterations in BatAlign was set to 1000; the size of the population was set to 40; $N = 10$; that is, when the optimal solution is not updated after 10 times, the current optimal solution was output as the final alignment result. BatAlign makes use of parameter α in Equation (1), where α determines

the relative importance of sequence and topological similarity. Meanwhile, $\alpha = 1$ implies that only sequence information was used. In order to ensure the fairness of the comparison, parameter α was set to 0.4 in all the algorithms that use alpha to control the weight of topological similarity and sequence score, and this value was also recommended by ModuleAlign.

To verify the effectiveness of the BatAlign, the algorithm was tested on synthetic and real networks and compared to
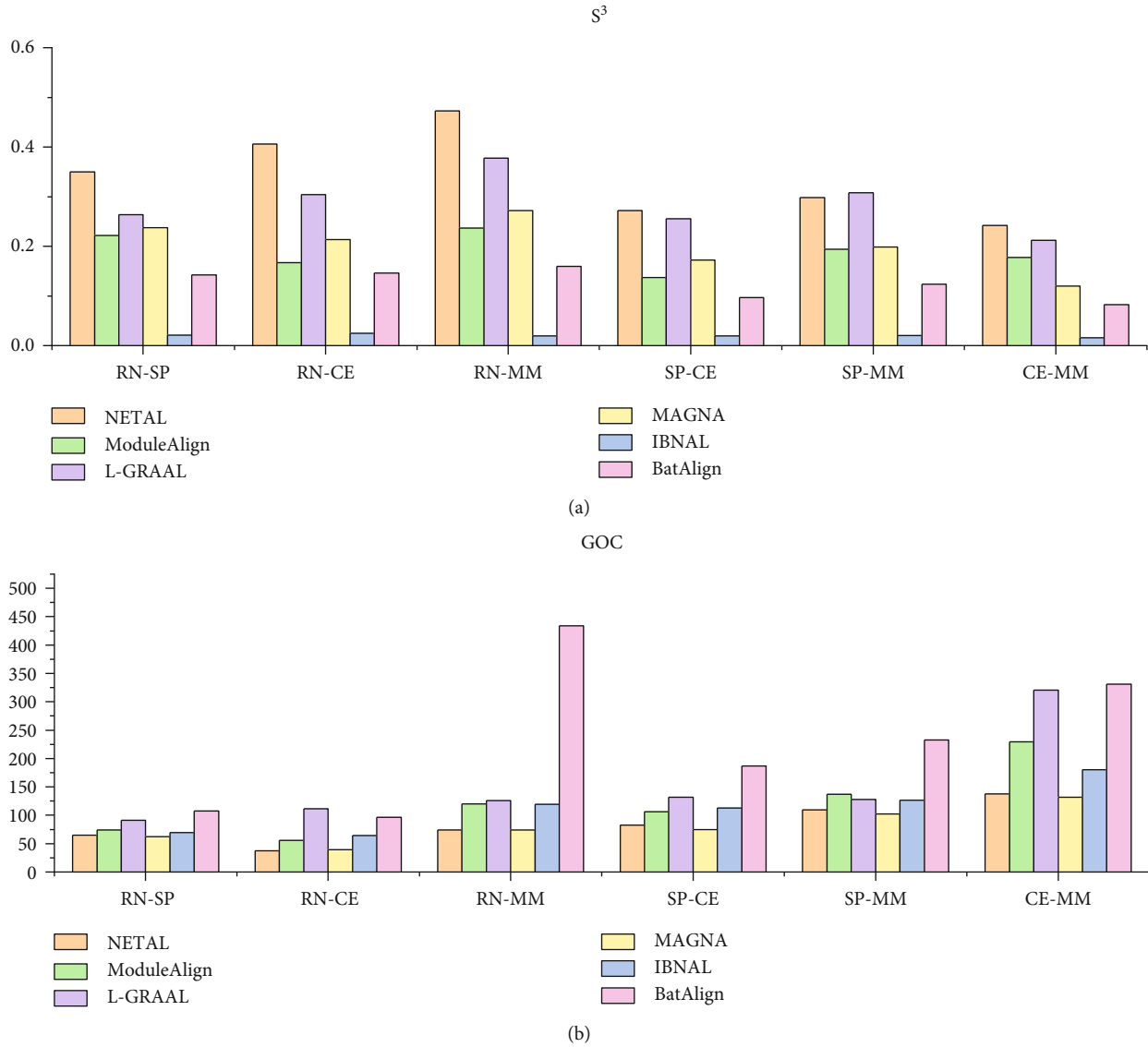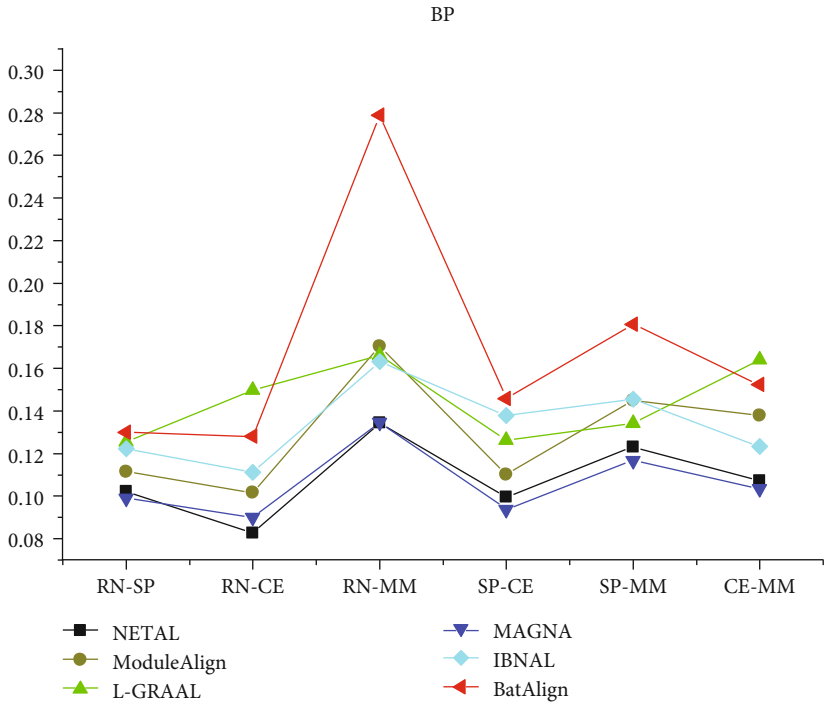
(a)

(b)

FIGURE 9: The performance of the different algorithms on real networks. (a) S^3 and (b) GOC of the different algorithms on real networks.

several state-of-the-art algorithms (i.e., NETAL [31], ModuleAlign [35], L-GRAAL [30], MAGNA [27], and IBNAL [37]). NETAL only adopts topological information to construct the alignment. ModuleAlign is an algorithm based on modularity. L-GRAAL is the representative of the GRAAL family algorithms and integrates Graphlet degree similarity and sequence similarity. MAGNA uses a genetic algorithm, only considers the topological similarity, and is based on the objective function. IBNAL makes use of a novel clique-based index.

The performance of the algorithms is evaluated in Figure 7, based on S^3 on the synthetic networks. In CG networks, the performance of BatAlign was inferior compared to L-GRAAL, NETAL, and ModuleAlign, and S^3 of BatAlign was 1.3-120 times higher than IBNAL and MAGNA. In the DMC networks, the score of BatAlign was 0.1-3.8 times higher than NETAL, MAGNA, and IBNAL. In DMR net-

works, the performance of BatAlign was inferior to ModuleAlign, L-GRAAL, and NETAL. S^3 of BatAlign was 3.8-4.6 times higher than MAGNA and IBNAL. The results show that the quality of the BatAlign is medium.

The algorithms based on the GOC score on the synthetic networks are compared in Figure 8. BatAlign presented good biological scores in DMC, while its score was slightly lower than ModuleAlign, outperforming the other algorithms. The score of BatAlign was lower than ModuleAlign and L-GRAAL in CG and DMC, while its performance was good compared to the other aligners. The topology of the real network is more complex than that of the synthetic network. Although the performance of BatAlign was not as good as ModuleAlign and L-GRAAL in synthetic networks, BatAlign performed well in real networks. BatAlign can identify functionally consistent proteins, which is helpful to biological research.

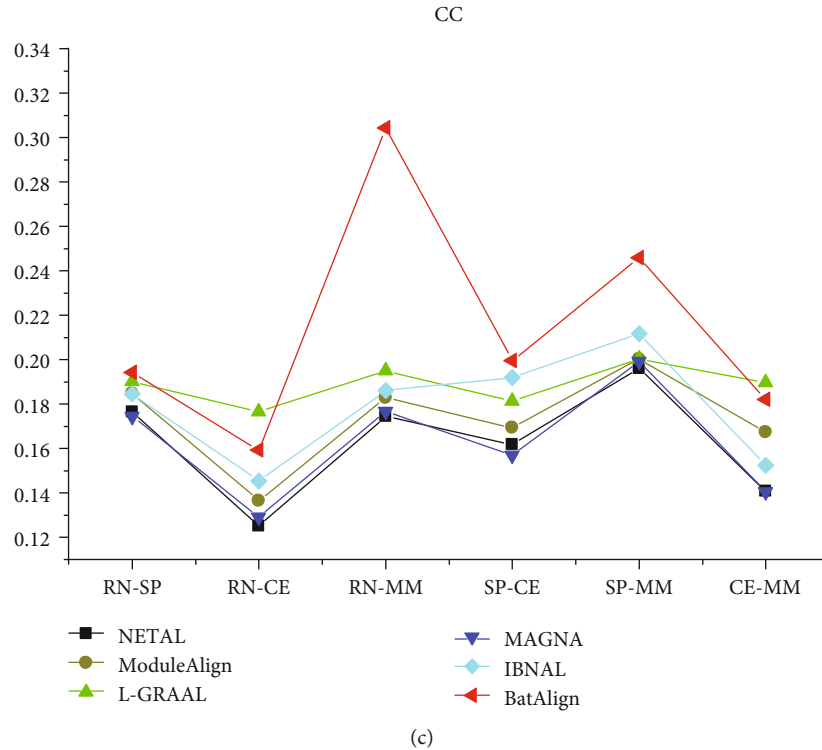BP

(a)

MF
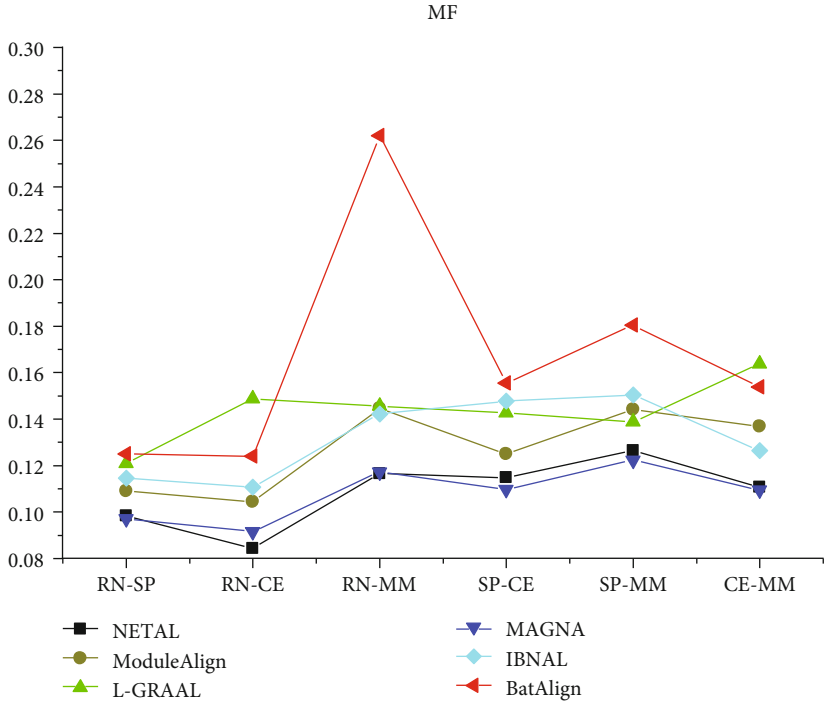
(b)

FIGURE 10: Continued.

FIGURE 10: The AFS of the different algorithms on real networks. (a) AFS with respect to BP; (b) MF and (c) CC of the different algorithms on real networks, respectively.

Figure 9 shows the results of the different algorithms on the real networks. The BatAlign performance with respect to S^3 was low, while BatAlign outperformed other aligners in terms of GOC; in particular, the alignment between RN and MM achieved an excellent biological score, which may be because these two species were closely related in genetic relationship. NETAL performed best with respect to the S^3 score, but it had a very low GOC score, may be because NETAL is a topological-only method; it can realize high topological quality at the expense of the biological quality. However, GOC carries more importance metrics than S^3 as a metric. In Figure 9, S^3 of ModuleAlign and MAGNA was higher than BatAlign, but they scored low GOC values, and their alignment result may miss node pairs with high functional similarity. The results showed that BatAlign performed much better than IBNAL in terms of S^3 and GOC scores. The GOC of BatAlign was slightly lower than L-GRAAL when aligning RN and CE. However, BatAlign was superior to L-GRAAL when aligning other networks.

AFS provides an alternative way to describe the biological quality of an alignment. Figure 10 represents the performance of each aligner in terms of AFS. The AFS of BatAlign was 20-50%, 19-54%, and 11-43% higher than NETAL, ModuleAlign, MAGNA, and IBNAL aligners, in terms of BP, MF, and CC, respectively. The performance of L-GRAAL was higher than BatAlign when mapping RN to CE and CE to MM. On the other hand, BatAlign outperformed L-GRAAL, when mapping other networks. Overall, BatAlign has good biological quality compared to other aligners.

On synthetic networks, BatAlign had high GOC scores among selected aligners and competitive S^3 scores. On real networks, BatAlign performed well in terms of the biological score with a relatively high topological score. Thus, BatAlign reached a relatively high topological quality and a superior biological quality. Experiments showed that BatAlign may be a useful tool for predicting the functions of unknown proteins in less studied species through network alignment with species that have been better studied.

4. Conclusions and Prospects

The BatAlign based on a discretized bat algorithm for the global alignment of two networks is proposed in this paper. BatAlign discretizes the bat algorithm and uses 0 or 1 to represent the flying velocity. The population of BatAlign is initialized according to the similarity score matrix. A new solution is generated according to a global and a local search, performed according to velocity. The number of conserved edges is used as the objective function. BatAlign overcomes the shortcoming of other search algorithms based on objective functions that initialize the population randomly and can only rely on a larger population and many iterations to find the optimal solution. The results of BatAlign are comparable to other state-of-the-art aligners. Experiments showed that BatAlign is a pairwise biological network global alignment algorithm that performs well in terms of biological quality. Future work will include parallelization of the BatAlign and expansion from two to multiple networks.

Data Availability

The data supporting the findings of the article is available at https://github.com/ComplexNetworkJN/BatAlign.

Conflicts of Interest

The authors declare that they have no competing interests.

Authors' Contributions

All authors contributed to the conception and design of the study. Jing Chen participated and coordinated the entire study. Jinfang Xia was involved in the conception of the study, participated in its design, and drafted the manuscript. Ying Zhang performed the statistical analysis and helped draft the manuscript. The final manuscript was approved by all authors.

Acknowledgments

References

[1] R. L. Finley and R. Brent, "Interaction mating reveals binary and ternary connections between Drosophila cell cycle regulators," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 91, no. 26, pp. 12980–12984, 1994.

[2] E. M. Phizicky and S. Fields, "Protein-protein interactions: methods for detection and analysis," *Microbiology Reviews*, vol. 59, no. 1, pp. 94–123, 1995.

[3] E. Davidson and M. Levin, "Gene regulatory networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 14, pp. 4935–4935, 2005.

[4] S. Schuster, D. A. Fell, and T. Dandekar, "A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks," *Nature Biotechnology*, vol. 18, no. 3, pp. 326–332, 2000.

[5] A. L. Barabási and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," *Nature Reviews Genetics*, vol. 5, no. 2, pp. 101–113, 2004.

[6] R. Sharan and T. Ideker, "Modeling cellular machinery through biological network comparison," *Nature Biotechnology*, vol. 24, no. 4, pp. 427–433, 2006.

[7] S. Bandyopadhyay, R. Sharan, and T. Ideker, "Systematic identification of functional orthologs based on protein network comparison," *Genome Research*, vol. 16, no. 3, pp. 428–435, 2006.

[8] J. Dutkowski and J. Tiuryn, "Identification of functional modules from conserved ancestral protein-protein interactions," *Bioinformatics*, vol. 23, no. 13, pp. i149–i158, 2007.

[9] M. Noël and P. Nataša, "GR-Align: fast and flexible alignment of protein 3D structures using graphlet degree similarity," *Bioinformatics*, vol. 30, no. 9, pp. 1259–1265, 2014.

[10] C. Clark and J. Kalita, "A comparison of algorithms for the pairwise alignment of biological networks," *Bioinformatics*, vol. 30, no. 16, pp. 2351–2359, 2014.

[11] V. Memišević, T. Milenković, and N. Pržulj, "Complementarity of network and sequence information in homologous proteins," *Journal of Integrative Bioinformatics*, vol. 7, no. 3, pp. 275–289, 2010.

[12] C. J. Ryan, P. Cimermančič, Z. A. Szpiech, A. Sali, R. D. Hernandez, and N. J. Krogan, "High-resolution network biology: connecting sequence with function," *Nature Reviews Genetics*, vol. 14, no. 12, pp. 865–879, 2013.

[13] H. T. Phan and M. J. Sternberg, "PINALOG: a novel approach to align protein interaction networks–implications for complex detection and function prediction," *Bioinformatics*, vol. 28, no. 9, pp. 1239–1245, 2012.

[14] A. Mir, M. Naghibzadeh, and N. Saadati, "INDEX: incremental depth extension approach for protein-protein interaction networks alignment," *Biosystems*, vol. 162, pp. 24–34, 2017.

[15] E. Kazemi, H. Hassani, M. Grossglauser, and H. Pezeshgi Modarres, "PROPER: global protein interaction network alignment through percolation matching," *BMC Bioinformatics*, vol. 17, no. 1, pp. 527–542, 2016.

[16] F. E. Faisal, L. Meng, J. Crawford, and T. Milenković, "The post-genomic era of biological network alignment," *EURASIP Journal on Bioinformatics and Systems Biology*, vol. 2015, no. 1, Article ID 3, 2015.

[17] G. Ciriello, M. Mina, P. H. Guzzi, M. Cannataro, and C. Guerra, "AlignNemo: a local network alignment method to integrate homology and topology," *PLoS One*, vol. 7, no. 6, pp. e38107–e38120, 2012.

[18] J. Berg and M. Lassig, "Local graph alignment and motif search in biological networks," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 101, no. 41, pp. 14689–14694, 2004.

[19] M. Koyutürk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama, "Pairwise alignment of protein interaction networks," *Journal of computational biology: a journal of computational molecular cell biology*, vol. 13, no. 2, pp. 182–199, 2006.

[20] J. Flannick, A. Novak, B. S. Srinivasan, H. McAdams, and S. Batzoglou, "Graemlin: general and robust alignment of multiple large interaction networks," *Genome Research*, vol. 16, no. 9, pp. 1169–1181, 2006.

[21] B. P. Kelley, B. Yuan, F. Lewitter, R. Sharan, B. R. Stockwell, and T. Ideker, "PathBLAST: a tool for alignment of protein interaction networks," *Nucleic Acids Research*, vol. 32, no. Web Server, pp. W83–W88, 2004.

[22] M. Kalaev, M. Smoot, T. Ideker, and R. Sharan, "Network-BLAST: comparative analysis of protein networks," *Bioinformatics*, vol. 24, no. 4, pp. 594–596, 2008.

[23] S. Maskey and Y.-R. Cho, "LePrimAlign: local entropy-based alignment of PPI networks to predict conserved modules," *BMC Genomics*, vol. 20, Suppl 9, pp. 964–975, 2019.

[24] R. Singh, J. Xu, and B. Berger, "Pairwise global alignment of protein interaction networks by matching neighborhood topology," in *Research in Computational Molecular Biology*, pp. 16–31, Springer, 2007.

[25] R. Singh, J. Xu, and B. Berger, "Global alignment of multiple protein interaction networks with application to functional orthology detection," *Proceedings of the National Academy of Sciences*, vol. 105, no. 35, pp. 12763–12768, 2008.

[26] G. W. Klau, "A new graph-based method for pairwise global network alignment," *BMC Bioinformatics*, vol. 10, Suppl 1, pp. S59–S67, 2009.

[27] O. Kuchaiev, T. Milenković, V. Memišević, W. Hayes, and N. Pržulj, "Topological network alignment uncovers biological function and phylogeny," *Journal of the Royal Society Interface*, vol. 7, no. 50, pp. 1341–1354, 2010.

[28] O. Kuchaiev and N. Pržulj, "Integrative network alignment reveals large regions of global network similarity in yeast and human," *Bioinformatics*, vol. 27, no. 10, pp. 1390–1396, 2011.

[29] V. Memišević and N. Pržulj, "C-GRAAL: common-neighbors-based global GRAph ALignment of biological networks," *Bioinformatics*, vol. 4, no. 7, pp. 734–743, 2012.

[30] N. Malod-Dognin and N. Pržulj, "L-GRAAL: Lagrangian graphlet-based network aligner," *Bioinformatics*, vol. 31, no. 13, pp. 2182–2189, 2015.

[31] B. Neyshabur, A. Khadem, S. Hashemifar, and S. S. Arab, "NETAL: a new graph-based method for global alignment of protein-protein interaction networks," *Bioinformatics*, vol. 29, no. 13, pp. 1654–1662, 2013.

[32] V. Saraph and T. Milenković, "MAGNA: maximizing accuracy in global network alignment," *Bioinformatics*, vol. 30, no. 20, pp. 2931–2940, 2014.

[33] V. Vijayan, V. Saraph, and T. Milenković, "MAGNA++: maximizing accuracy in global network alignment via both node and edge conservation," *Bioinformatics*, vol. 31, no. 14, pp. 2409–2411, 2015.

[34] N. Mamano and W. B. Hayes, "SANA: simulated annealing far outperforms many other search algorithms for biological network alignment," *Bioinformatics*, vol. 33, no. 14, pp. 2156–2164, 2017.

[35] S. Hashemifar, J. Ma, H. Naveed, S. Canzar, and J. Xu, "ModuleAlign: module-based global alignment of protein–protein interaction networks," *Bioinformatics*, vol. 32, no. 17, pp. i658–i664, 2016.

[36] A. Alcalá, R. Alberich, M. Llabrés, F. Rosselló, and G. Valiente, "AligNet: alignment of protein-protein interaction networks," *BMC Bioinformatics*, vol. 21, Suppl 6, pp. 265–286, 2020.

[37] A. Elmsallati, A. Msalati, and J. Kalita, "Index-based network aligner of protein-protein interaction networks," *IEEE/ACM Transactions On Computational Biology And Bioinformatics*, vol. 15, no. 1, pp. 330–336, 2018.

[38] X. S. Yang, "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization*, vol. 284, pp. 65–74, 2010.

[39] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of Molecular Biology*, vol. 215, no. 3, pp. 403–410, 1990.

[40] H. M. Woo, H. Jeong, and B. J. Yoon, "NAPAbench 2: a network synthesis algorithm for generating realistic protein-protein interaction (PPI) network families," *PLoS One*, vol. 15, article e0227598, pp. 1–20, 2020.

[41] R. Oughtred, C. Stark, B.-J. Breitkreutz et al., "The BioGRID interaction database: 2019 update," *Nucleic Acids Research*, vol. 47, no. D1, pp. D529–D541, 2019.

[42] M. Ashburner, C. A. Ball, J. A. Blake et al., "Gene Ontology: tool for the unification of biology," *Nature Genetics*, vol. 25, no. 1, pp. 25–29, 2000.

[43] D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova, "Entrez gene: gene-centered information at NCBI," *Nucleic Acids Research*, vol. 39, no. Database, pp. D52–D57, 2011.

[44] R. Patro and C. Kingsford, "Global network alignment using multiscale spectral signatures," *Bioinformatics*, vol. 28, no. 23, pp. 3105–3114, 2012.

[45] A. E. Aladag and C. Erten, "SPINAL: scalable protein interaction network alignment," *Bioinformatics*, vol. 29, no. 7, pp. 917–924, 2013.

[46] S. Hashemifar and J. Xu, "HubAlign: an accurate and efficient method for global alignment of protein-protein interaction networks," *Bioinformatics*, vol. 30, no. 17, pp. I438–I444, 2014.

[47] J. Z. Wang, Z. du, R. Payattakool, P. S. Yu, and C. F. Chen, "A new method to measure the semantic similarity of GO terms," *Bioinformatics*, vol. 23, no. 10, pp. 1274–1281, 2007.

[48] G. Yu, F. Li, Y. Qin, X. Bo, Y. Wu, and S. Wang, "GOSemSim: an R package for measuring semantic similarity among GO terms and gene products," *Bioinformatics*, vol. 26, no. 7, pp. 976–978, 2010.