Tsinghua Science and Technology

Volume 19 | Issue 6 Article 10

2014

PBNA: An Improved Probabilistic Biological Network Alignment Method

Muwei Zhao
School of Computer Science and Engineering, and also with MOE Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing 210096, China.

Wei Zhong
Division of Math and Computer Science, University of South Carolina Upstate, Spartanburg, SC 29303, USA,

Jieyue He
School of Computer Science and Engineering, and also with MOE Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing 210096, China.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 10/10 pp658-667
Volume 19, Number 6, December 2014

PBNA: An Improved Probabilistic Biological Network Alignment Method

Muwei Zhao, Wei Zhong, and Jieyue He*

Abstract: Biological network alignment is an important research topic in the field of bioinformatics. Nowadays almost every existing alignment method is designed to solve the deterministic biological network alignment problem. However, it is worth noting that interactions in biological networks, like many other processes in the biological realm, are probabilistic events. Therefore, more accurate and better results can be obtained if biological networks are characterized by probabilistic graphs. This probabilistic information, however, increases difficulties in analyzing networks and only few methods can handle the probabilistic information. Therefore, in this paper, an improved Probabilistic Biological Network Alignment (PBNA) is proposed. Based on IsoRank, PBNA is able to use the probabilistic information. Furthermore, PBNA takes advantages of Contributor and Probability Generating Function (PGF) to improve the accuracy of node similarity value and reduce the computational complexity of random variables in similarity matrix. Experimental results on dataset of the Protein-Protein Interaction (PPI) networks provided by Todor demonstrate that PBNA can produce some alignment results that ignored by the deterministic methods, and produce more biologically meaningful alignment results than IsoRank does in most of the cases based on the Gene Ontology Consistency (GOC) measure. Compared with Prob method, which is designed exactly to solve the probabilistic alignment problem, PBNA can obtain more biologically meaningful mappings in less time.

Key words: probabilistic biological network; network alignment; protein interaction network

1 Introduction

Comparative network analysis, namely biological network alignment, is an essential work in the field of biological network research. Through network aligning, one can discover the relationship between structures and features of organisms, and study the biological evolution. For example, you can predict functions and interactions of proteins using protein network alignment. You also can identify the conserved substructures and predict the protein complexes and functional modules using local network alignment[1,2].

Ogata et al.[3] started the biological network alignment research in 2000. They used the network alignment to discover the relationship between enzymes and positions of their corresponding gene encodings in the entire genome. Two biological networks were constructed to represent the gene distribution network and the metabolic network. By aligning these networks, the author found that these networks have similar local structures, which correspond to the adjacent functionally related enzyme clusters. This field of

• Muwei Zhao and Jieyue He are with School of Computer Science and Engineering, and also with MOE Key Laboratory of Computer Network and Information Integration, Southeast University, Nanjing 210096, China. E-mail: 220111461@seu.edu.cn; jieyuehe@seu.edu.cn.
• Wei Zhong is with Division of Math and Computer Science, University of South Carolina Upstate, Spartanburg, SC 29303, USA, E-mail: wzhong@uscupstate.edu.
∗ To whom correspondence should be addressed.
Manuscript received: 2014-06-16; accepted: 2014-06-23

research has gained a lot of interests since then. The work of Hirsh and Sharan[4] focused on the alignment of protein-protein interaction networks to detect and predict the conserved modules, the functions, and the interactions of proteins. Singh et al.[1,5] transformed the alignment problem into an eigenvalue problem first, then computed the eigenvalue and eigenvector. At last, the final alignment results are reverted. Narayanan and Karp[6] proposed an approach that divided networks into small sub networks, utilizing the structural characteristic of these networks. By aligning these sub networks, the final alignment results were obtained.

The existing methods of biological networks alignment can be divided into three categories[2]. The first category is heuristic search methods based on the graph models[7-12]. In these methods, an alignment graph is constructed based on the input networks. This alignment graph can be product graph or other kinds of graphs. Nodes of the alignment graph correspond to a set of compatible elements from different organisms, and similarity information is attached to the alignment graph as additional attributes. The original alignment problem is then solved by designing a heuristic search method over the align graph. The second category for biological networks alignment is constrained optimization methods based on objective functions[1,13,14]. These methods transform alignment problems into optimization problems that can be solved by some existing approaches. The original alignment problem is then solved using this approach. The third category for biological networks alignment is modular methods based on divide and conquer strategies[6,15,16]. Note that most biological networks are large and have modular structures[17-19]. These modular methods divide networks into sub networks, and then solve the original alignment problem by aligning these small sub networks.

It is worth noting that interactions in biological networks, like many other processes in the biological realm, are probabilistic events. For example, interactions in protein-protein interaction networks occur with certain probabilities. Many factors may have influences on the probabilities, such as the size, density, and redundancy of interacting molecules in the network, even errors in biological experiments. Because of these factors, we may lose confidence on the existence of some interactions[4]. Therefore, more accurate and better results can be potentially obtained if biological networks are characterized by probabilistic graphs, i.e., graphs having probabilistic values on edges. This probabilistic information, however, increases difficulties in analyzing networks.

In fact, only two methods including Weighted IsoRank[1] based on (non-weighted) IsoRank and an improved method proposed by Todor et al.[7] in 2007 (hereinafter referred as Prob) can be applied to the probabilistic network situation. However, the Weighted IsoRank[1] based on (non-weighted) IsoRank, when calculating the similarities between nodes, considered probabilities on edges as weights, rather than real probabilistic values. Therefore, Weighted IsoRank simplified probabilistic alignment problems into deterministic alignment problems, losing considerable amount of information. Prob modeled the probabilistic alignment problem by replacing deterministic values with random variables. Prob used Probabilistic Generating Function (PGF) to reduce computational complexity.

In this paper, an improved Probabilistic Biological Network Alignment (PBNA) is proposed. PBNA adopts the framework of IsoRank, and can utilize the probabilistic information in networks, allowing at least one network to be probabilistic. Also, Contributor and PGF[7] are adopted by PBNA to improve the accuracy of node similarity value and reduce the complexity of computing the expectation of random variables in similarity matrix. Experiment results demonstrate that PBNA can obtain alignment mappings that ignored by existing deterministic methods, and can obtain more biologically meaningful mappings in less time when compared with Prob[4]. To the best of our knowledge, Prob is the only method designed exactly to solve the probabilistic alignment problem.

2 Algorithm

2.1 Basic concepts

A biological network can be represented by Graphs: $G = (V, E)$, V is the node set of the graph, corresponding to nodes in the network, and E is the edge set, corresponding to interactions in the network. V and E may have attributes attached to, and E may be directed or undirected. Different types of biological networks are represented by different types of graphs. For example, an undirected graph with labels on nodes can be used to model a Protein-Protein Interaction (PPI) network. The labels of nodes are used to mark different proteins.

Probabilistic networks are networks with values (numbers) attached to edges. The values are probabilities in which edges exist (we assume that probabilities are independent of each other). As shown in Fig. 1, a probabilistic network is actually a summary of all possible deterministic networks that are determined by the subset of interactions that take place. This means that a probabilistic network represented as a graph with $|E|$ edges will actually describe the $2^{|E|}$ deterministic networks that could arise as instances of the probabilistic network, each with some probability. There are two probabilistic edges in the probabilistic network in Fig. 1, leading to four deterministic networks.

Figure 1 also shows that the traversal of all deterministic instances of a probabilistic network will consume exponential time, which is not practical. PBNA proposed in this paper will try to solve this problem later.

Given two biological networks, $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, the biological network alignment is defined as a mapping R from V_1 to V_2, and satisfies the following rules:

$$\mathrm{sim}(G_1, G_2) = \arg\max \sum_{<e_1, e_2> \in R} \mathrm{sim}(v_1, v_2) \quad (1)$$

$\mathrm{sim}(v_1, v_2)$ in Eq. (1) denotes the similarity score between a pair of nodes from G_1 and G_2 respectively. In other words, the objective of the alignment is to find proper mappings R (from V_1 to V_2), obtaining the greatest similarity $\mathrm{sim}(G_1, G_2)$.

As the definition describes, the computation of node similarities is the foundation of biological network alignment. The similarity represents the level of the matching between networks, and can be computed by the following rule:

$$\mathrm{sim}(G_1, G_2) = \alpha \sum f(v_1, v_2) + \beta \sum g(e_1, e_2) + \gamma h(G_1, G_2) \quad (2)$$

In Eq. (2), f is a function describing the node similarity; g is a function describing the edge similarity; h is an evolutionary similarity function between G_1 and G_2. For example, in the alignment of PPIs, f may measure the sequence similarity between proteins, while g may measure the topological structure similarity between networks. h is not included in most existing alignment methods. As the evolutionary similarity, the definition and calculation of h involves biological features and evolutions, which is a topic that still needs further research.

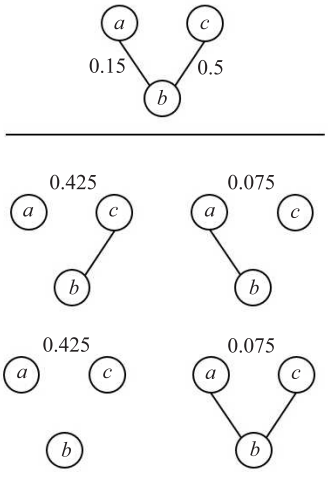

Fig. 1 A probabilistic network (top) and its four possible deterministic network instances (bottom).

2.2 PBNA algorithm

The PBNA method proposed by this paper adopts the framework of IsoRank[1], which is a classic deterministic alignment method. However, different from IsoRank, PBNA can utilize the probabilistic information in networks, allowing at least one network to be probabilistic. Also, Contributor and PGF[7] are employed by PBNA to improve the accuracy of node similarity value and reduce the complexity of computing the expectation of random variables in similarity matrix. The entire method can be roughly divided into three steps as Fig. 2: firstly, constructing the similarity matrix; secondly, computing the eigenvector of the matrix; and thirdly, extracting final alignment results from the eigenvector.

The procedures of PBNA include four parts: constructing similarity matrix without the probabilistic information; integrating the probabilistic information into similarity scores; computing the eigenvector; and extracting final alignment results.

2.2.1 Construction of the similarity matrix of deterministic graphs

Given two deterministic graphs $G_1 = (V_1, E_1)$ and

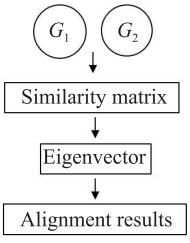

Fig. 2 The framework of PBNA.

$G_2 = (V_2, E_2)$ (G_2 will be extended to be probabilistic later in Section 2.2.2), let R_{ij} be the similarity score for the pair (i, j) where $i \in V_1$ and $j \in V_2$. The matrix composed of all the R_{ij}, namely $\boldsymbol{R}$, is the similarity matrix. R_{ij} should include both sequence similarity and topological structural similarity. PBNA adopts the approach proposed in Ref. [1] to handle the sequence similarity. Therefore, the topological structural similarity is mainly discussed here.

In order to calculate R_{ij}, PBNA sets up a system of constraints using the same recursive method as IsoRank: (i, j) is a good match if their respective neighbors are a good match with each other[1]. This constraint can be described as follows:

$$R_{ij} = \sum_{u \in N(i)} \sum_{v \in N(j)} \frac{1}{d_u d_v} R_{uv} \tag{3}$$

In Eq. (3), i is from V_1 and j is from V_2; $N(i)$ denotes the set of neighbors of node i; d_u denotes the degree of node u. This equation requires that the score R_{ij} for any pair (i, j) be equal to the total support provided to it by each of the $d_u d_v$ possible pairs between the neighbors of i and j[1]. Equation (3) can be rewritten in the following matrix form:

$$\boldsymbol{R} = \boldsymbol{A}\boldsymbol{R}, \text{where}$$

$$A[i, j][u, v] = \begin{cases} \dfrac{1}{d_u d_v}, & \text{if } (i, u) \in E_1, (j, v) \in E_2; \\ \dfrac{1}{mn}, & \text{if } d_u d_v = 0; \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

In Eq. (4), $m = |V_1|$, $n = |V_2|$; $\boldsymbol{A}$ is an $(mn) \times (mn)$ matrix whose rows and columns are both doubly indexed; $A[i, j][u, v]$ refers to the entry of $\boldsymbol{A}$ at the row$[i, j]$ and column $[u, v]$.

In equation $\boldsymbol{R} = \boldsymbol{A}\boldsymbol{R}$, $\boldsymbol{R}$ is rewritten from an $m \times n$ matrix into $(mn) \times 1$ vector, and is the eigenvector of $\boldsymbol{A}$. Therefore, through Eq. (4), the alignment problem is transformed into an eigenvector problem.

In Eq. (4), the similarity score of every pair in $N(i)$ and $N(j)$ contributes to the similarity score of (i, j). In fact, parts of the neighbors have more influences on R_{ij} than the rest ones[20]. We therefore introduce two concepts proposed in Ref. [20]: Neighborhood Bipartite Graph (NBG) and Contributor. They aim at increasing the similarity contributions of the pairs with higher chances of existence in the final alignment results.

Let S be a function mapping the pair of nodes $\prec u, v \succ$ to a real value, where $u \in V_1$ and $v \in V_2$. The set of neighbors of u is denoted with $N(u)$. The Neighborhood Bipartite Graph of the pair $\prec u, v \succ$, denoted with NBG $(\prec u, v \succ, S)$, is a complete bipartite graph with weights on edges defined on $N(u)$ and $N(v)$. The edge weights in NBG are $S(u, v)$. The Contributor, denoted with C, is the set of edges in the maximum weight matching of NBG $(\prec u, v \succ, S)$.

Equation (4) is converted into the following form with the help of C:

$$A[i, j][u, v] = \begin{cases} \dfrac{1}{d_u d_v}, & \text{if } (u, v) \in C; \\ \dfrac{1}{mn}, & \text{if } d_u d_v = 0 \text{ OR} \\ & \quad ((i, u) \in E_1, (j, v) \in E_2); \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

Equation (4) has no concept of Contributor. Therefore every pair in $N(i)$ and $N(j)$ all contributes to the similarity score. However, Eq. (5) only considers contributions of pairs with higher chances of existence in the final alignment results, namely those pairs as Contributor, and ignores contributions of pairs that have no chances of coexistence in the final results. The computation of similarity score in PBNA is therefore more accurate with the employment of Contributor[20].

Another effect resulting from Contributor is that the running time of PBNA decreases considerably, as the number of node pairs decreases from $|N(i)| \times |N(j)|$ to $\min(N(i), N(j))$. See detailed statistical results about running time in Section 3.

2.2.2 Integrating the probabilistic information into similarity scores

Now we generalize the alignment problem into the situation where G_1 is deterministic while G_2 is probabilistic. A main challenge of probabilistic network alignment is to compute $\boldsymbol{R}$. This is because that the nature of probabilistic network makes it almost impossible to choose one specific deterministic network and the corresponding $\boldsymbol{A}$ of it from all possible alternatives. There are $2^{|E_2|}$ alternative matrices. We solve this problem using an approach proposed in Prob[4]: All of these matrices are modeled with a matrix of random variables and $\boldsymbol{A}$ is replaced with its expected value, $E(\boldsymbol{A})$. Now the question is how to compute $E(\boldsymbol{A})$?

The degree of every node, namely d_v, is obviously probabilistic in probabilistic networks. d_v can be any value in the sequence $0, \cdots, d_v^{\max}$ ($d_v^{\max}$ is the largest possible degree of node v). Let a discrete random variable D_v model the degree of node v. Then

$P(D_v = k), k = 0, 1, \cdots, d_v^{\max}$ is the sequence of degree distribution of node v in probabilistic network G_2.

$\boldsymbol{A}$ is now a random matrix with entries that depend on D_v, and Eq. (5) is converted into Eq. (6) as the following form:

$$A[i,j][u,v] = \begin{cases} \dfrac{1}{d_u D_v}, & \text{if } (u,v) \in C; \\ \dfrac{1}{mn}, & \text{if } d_u D_v = 0 \text{ OR} \\ & ((i,u) \in E_1, (j,v) \in E_2); \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

In order to compute the expectation of $\boldsymbol{A}$ described in Eq. (6), we focus on its one specific entry $A[i,j][u,v]$. Note that the related information of node u is deterministic as it is from G_1. d_u is deterministic, and whether (u,v) is included in C of (i,j) is also deterministic (see Algorithm 1). According to Eq. (6), if node u is isolated, namely $d_u = 0$, $A[i,j][u,v]$ is a constant, $\dfrac{1}{mn}$; if u is not isolated and $(i,u) \notin E_1$, $A[i,j][u,v]$ is constant, 0; if $(u,v) \notin C$, $A[i,j][u,v]$ is also constant, $\dfrac{1}{mn}$. For the remaining cases, $A[i,j][u,v]$ is a random variable whose expectation needs to be computed. The definition of expectation of a discrete random variable is

$$E[A[i,j][u,v]] = \sum_k kP(A[i,j][u,v] = k) \tag{7}$$

The possible values of $A[i,j][u,v]$, denoted with k, can be 0, $\dfrac{1}{mn}$, or $\dfrac{1}{d_u D_v}$ according to Eq. (6). From the above discussion, we know that the prerequisite for $A[i,j][u,v]$ being a random variable is that $(u,v) \in C$. Therefore, node v has no possibility being an isolated node. According to Eq. (6), $A[i,j][u,v]$ can only be $\dfrac{1}{d_u D_v}$. Moreover, $(u,v) \in C$ leads to $(j,v) \in E_2$. And if the probabilistic edge (j,v) is guaranteed to appear in E_2, it implies that j and v are neighbors. The degree of v is at least 1. Thus, the probability distribution of the random variable D_v should take this prior information into consideration. In mathematical way, this can be expressed as a conditional probability $p(D_v = k|(j,v) \in E_2)$, for $k = 1, \cdots, d_v^{\max}$. Finally, the expectation of matrix $\boldsymbol{A}$ can be computed as following:

$$E[A[i,j][u,v]] = \begin{cases} \dfrac{1}{d_u} \times \displaystyle\sum_{k=1}^{d_v^{\max}} \frac{1}{k} P(D_v = k|(j,v) \in E_2), \\ \qquad \text{if } (u,v) \in C; \\ \dfrac{1}{mn}, \quad \text{if } d_u = 0 \text{ OR } (u,v) \notin C; \\ 0, \quad \text{otherwise} \end{cases} \tag{8}$$

The main technical difficulty in Eq. (8) is how to compute the conditional degree distribution $P(N_v = k|(j,v) \in E_2)$.

Definition 1[4] Let X be a discrete random variable taking integer values from 0 to N. The PGF of X is defined as the polynomial of z:

$$Q_X(z) = E[z^X] = \sum_{k=0}^{N} P(X = k) z^k \tag{9}$$

Example 1 Let X be a discrete random variable taking values from $\{0, 1, 2\}$. The probability distribution

Algorithm 1 Computing $E(A)$ according to Eq. (8).

Algorithm 1 Computation of $E(\boldsymbol{A})$.

Input: Deterministic graph $G_1 = (V_1, E_1)$,
Probabilistic graph $G_2 = (V_2, E_2)$
Output: $E(\boldsymbol{A})$
//Construct PGF of V_2
1. for all $v \in V_2$ do:
2. construct PGF of v
3. end for
//Construct the initial similarity matrix
//for finding Contributor later
4. for all $u \in V_1$, $v \in V_2$ do:
5. $M(u,v) = \alpha \times \text{DegDiff}(u,v) + (1-\alpha) \times \text{Seq}(u,v)$
6. end for
//Compute every entry in $\boldsymbol{A}$
7. for all $A[i,j][u,v] \in \boldsymbol{A}$ do:
//The first rule in Eq. (8)
8. if $(u,v) \in C$ then:
9. $Q_{D_v}^j = Q_{D_v}/1 - P(j,v) + P(j,v)z$
10. $S = 0$
11. for $k = 1 \to d_v^{\max}$ do:
12. $S+ = 1/k \times Q_{D_v}^j||_{k-1}$
13. end for
14. $E[A[i,j][u,v]] = S/d_u$
//The second rule in Eq. (8)
15. else if
16. $d_u = 0$ OR $(u,v) \notin C$ then:
17. $E[A[i,j][u,v]] = \dfrac{1}{mn}$
//The third rule in Eq. (8)
18. else
19. $E[A[i,j][u,v]] = 0$
20. end if
21. end for

is $\{0.09, 0.45, 0.7\}$. The PGF of X is the polynomial $Q_X(z) = 0.09 + 0.45z + 0.7z^2$.

Thus, we know that by simply listing the coefficients of PGF, the distribution of X will be obtained. And the following Theorem 1 can be used to compute PGF of D_v.

Theorem 1[4] Let $G = (V, E, P)$ be a probabilistic network and E_v be the set of edges incident on node v. The PGF of the degree distribution of v is

$$Q_{D_v}(z) = \prod_{e \in E_v} (1 - p_e + p_e z) \tag{10}$$

Example 2 In Fig. 1, the PGF of the distribution for node b is the multiplication of the polynomials corresponding to each incident edge: $(0.85 + 0.15z)(0.5 + 0.5z) = 0.425 + 0.5z + 0.075z^2$. The degree distribution of a is $P(N_b = 0) = 0.425$, $P(N_b = 1) = 0.5$, $P(N_b = 2) = 0.075$.

Thus, the degree distribution of node v can be computed by the PGF of D_v conveniently. The time complexity of the entire process then decreases from $O(2^{d_v^{\max}})$ (exhaustive method) to $O((d_v^{\max})^2)$[4].

Note that our goal in Eq. (8) is to compute the conditional degree distribution $P(D_v = k | (j, v) \in E_2)$. In fact, the condition that a particular edge e is present can be expressed by dividing the PGF of D_v by $(1 - p_e + p_e z)$, which is the PGF of D_v if edge e is the only edge incident on node v. The result of the division produces the PGF of D'_v, which is the degree of node v without considering the edge e at all. Finally, by shifting the distribution of D'_v by one position, we get the conditional degree distribution $P(D_v = k | e \in E_v) = P(D'_v = k-1)$. The intuitive explanation of this dividing-and-shifting procedure is that the probability of the degree of node v being k under the condition that edge e is already present equals to the probability of the degree of node v being $k - 1$ under the condition that node v doesn't have edge e at all.

The pseudo code for computing $E(\boldsymbol{A})$ according to Eq. (8) is shown in Algorithm 1.

Algorithm 1 demonstrates how to compute $E(\boldsymbol{A})$ in three major steps:

(1) Lines 1-3: construct the PGF for every node in probabilistic network G_2;

(2) Lines 4-6: for every possible pair (u, v) where $u \in V_1$ and $v \in V_2$, compute its initial similarity score for finding the set of Contributor according to Line 5;

(3) Line 7 to end: for every entry of $\boldsymbol{A}$, compute its similarity score according to three rules in Eq. (8): Lines 8-14 are for the first rule; Lines 15-17 are for the second rule; and Lines 18-20 are for the third.

2.2.3 Computing the eigenvector

Having computed $E(\boldsymbol{A})$, PBNA uses Power Iteration method to get $\boldsymbol{R}$, the eigenvector of $\boldsymbol{A}$. The Power Iteration, also known as Von Mises Iteration, is a simple iterative technique aiming to solve large sparse eigenvalue problems[1].

$\boldsymbol{R}$ begins with a random value before the first iteration, denoted with $\boldsymbol{R}_0$, and then is updated in every iteration by the following rule:

$$\boldsymbol{R}_{k+1} \leftarrow \frac{\boldsymbol{A}\boldsymbol{R}_k}{||\boldsymbol{A}\boldsymbol{R}_k||}.$$

$\boldsymbol{R}_k$ is the value of the vector $\boldsymbol{R}$ in the k-th iteration. The method will converge to the principal eigenvector.

2.2.4 Extracting final alignment results

At this step of PBNA, we have a score R_{ij} for every pair, where nodes come from G_1 and G_2. This score indicates how good a mapping $\prec i, j \succ$ is. In order to extract the final alignment results, PBNA adopts an almost breadth first searching approach by using seed-and-extend technique[20,21]. This approach aims at increasing the size of conserved interactions.

We now introduce the Alignment Graph. The Alignment Graph is one of the descriptions of the network alignment results, denoted with $A_{12} = (V_{12}, E_{12})$. Each node of A_{12} is corresponding to a pair of mapping $\prec v_1, v_2 \succ$, where $v_1 \in V_1, v_2 \in V_2$. For any two nodes $\prec v_1, v_2 \succ \in V_{12}, \prec v_3, v_4 \succ \in V_{12}$, it should not be the case $v_1 = v_3$ or $v_2 = v_4$. In other words, one node from G_1 can be mapped to at most one node from G_2 in the final results, and vice versa. The edge of A_{12}, namely $(\prec v_1, v_2 \succ, \prec v_3, v_4 \succ) \in E_{12}$ is present, if and only if both $(v_1, v_3) \in E_1$ and $(v_2, v_4) \in E_2$ are present in G_1 and G_2.

The process of extracting alignment results from $\boldsymbol{R}$ in PBNA is repeatedly by adding nodes to the alignment graph A_{12}. In every iteration, one connected component A_{sub} is added to A_{12}. All the A_{sub} assemble A_{12}.

A_{sub} starts with the best available seed. It is the pair with the greatest score in $\boldsymbol{R}$, and neither of the nodes in this pair is aligned. The NBG of the pair is then constructed. Finally, a maximum weight matching[21] provides us the Contributor to be added to A_{sub}. Repeat this operation until no Contributor being found. One A_{sub} is now being constructed. It is connected as in

every iteration the neighbors of existing nodes in A_{sub} are added.

We repeatedly construct A_{sub} and add it to A_{12} until every node in one network has been mapped to another node or a gap in the other network (a gap means no node in the second network is mapped to the node in the first network, because the numbers of nodes in two networks are not necessarily equal).

3 Experiments and Results

This section presents the experimental evaluation of PBNA. All experiments can be divided into two parts: Experiments 1 and 2 verify the necessity and effectiveness of PBNA; Experiments 3, 4, and 5 evaluate the biological significance of the alignment produced by PBNA and the efficiency of PBNA. The biological significance of the alignment results produced by PBNA and its efficiency are both compared with Prob[4], which is now the only one method, as far as we know, designed exactly for the probabilistic alignment problem.

We implement our PBNA, IsoRank, and Prob using C++, with the help of Qt Library (http://qt-project.org/) for accessing and sorting array and matrix data, and OGDF Library (http://ogdf.net/) for manipulating the networks. All the experiments run on a standard PC with a 3.3 GHz CPU and 4 GB of RAM.

We adopt the suggestion of the literature[12] when setting the parameter α, the relative contribution of topology and pairwise similarity in Eq. (4), to 0.6. Setting α to 0.6 may produce the best alignment results. Please refer to Ref. [12] for further information about choosing appropriate α.

The dataset of all experiments is the PPI networks provided by Todor et al.[7] They decomposed the set of proteins in the MINT[9] networks according to their underlying functions to get smaller, yet biologically coherent sub networks, with the help of the KEGG database[22]. Then they removed the sub networks that had fewer than 10 nodes. The final dataset contains 198 networks from 10 organisms, shown in Table 1.

Table 1 Dataset statistics.

Organism	Number of networks	Number of proteins		Number of interactions	
		Average	Max	Average	Max
Cel	7	14.00	22	9.57	21
Dme	7	17.14	28	12.42	26
Eco	6	16.83	27	21.16	26
Hpy	1	11.00	11	7.00	7
Has	83	36.50	96	46.55	168
Mmu	43	16.23	40	11.16	33
Rno	13	14.69	30	11.00	22
Sce	34	32.91	106	80.32	313
Spo	3	11.00	11	10.00	10
Tpa	1	20.00	20	21.00	21

3.1 Agreement with IsoRank

The fundamental question we need to answer first is: Does the probabilistic method PBNA produce any alignment that is ignored by the deterministic methods? In other words, is there any need for our PBNA?

In order to answer this, we introduce a measurement, Agreement[7], defined as the ratio of the number of alignment that produced by both methods to the size of the entire alignment (the size of the smaller aligned network):

$$\frac{|\text{Alignments in common}|}{|\text{All alignments}|}.$$

Agreement is a value between 0 and 1. The greater value indicates that two alignments differ with each other LESS significantly. The value of 1 means that two alignments are identical.

Each network from one organism is aligned with its corresponding network from another organism, which yields 244 experiments. In every experiment, PBNA and IsoRank are both executed. This means that 244 Agreement values are obtained. The distribution of these values, after being sorted, is demonstrated in Fig. 3, in which Agreement values are denoted as y-coordinate and experiment indexes are denoted as x-coordinate.

Figure 3 shows that all of the Agreement values distribute within the range from 0.15 to 0.9. This means that the alignment results produced by our PBNA are noticeably different than those produced by IsoRank. In other words, PBNA obtains many alignment results

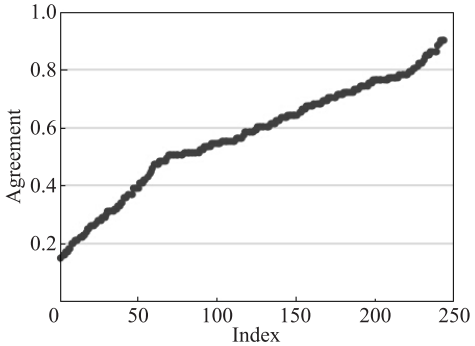

Fig. 3 Agreement statistics.

that IsoRank cannot obtain because two methods adopt different approaches when extracting alignment from similarity matrix (i.e., the eigenvector). IsoRank discards the probabilistic interaction information in the PPI networks, which however PBNA integrates into the computation of similarity matrix.

Note that the results of Experiment 1 can only indicate that PBNA produce novel alignment results. They do not however evaluate the biological significance of the alignment. Experiment 2 tries to do this.

3.2 Gene ontology consistencies of PBNA and IsoRank

In this experiment, we take one of the common measures to test the biological quality of the alignment, Gene Ontology (GO) Consistency (GOC)[20]:

$$\mathrm{GOC} = \frac{|\mathrm{GO}(u) \cap \mathrm{GO}(v)|}{|\mathrm{GO}(u) \cup \mathrm{GO}(v)|}.$$

$\mathrm{GO}(u)$ denotes the set of GO terms annotating a protein u. The GOC is computed for every pair of proteins in the alignment results. The greater GOC indicates the closer relationship of the two proteins in terms of biological functions (Please refer to Refs. [1, 20] for detailed discussion about GOC). The GO dataset is provided by Ref. [20] and GO Consortium[11].

Similar to Experiment 1, PBNA and IsoRank are both ran on 244 same datasets. On every dataset (a pair of networks), two alignments are obtained. After computing GOC for every mapping in the alignment and the sum of GOCs for the entire alignment, we get 244 pairs of sum values, in which one for PBNA and the other for IsoRank. The distribution of these values is demonstrated in Fig. 4, where values of PBNA as x-coordinate and those of IsoRank as y-coordinate.

For a majority of points (224 points, 92% of total) in Fig. 4, their x-coordinate values are 5%-15% greater than their y-coordinate values; only for the few remaining ones, their x-coordinate and y-coordinate values are almost identical. This indicates that based on the GOC measure, PBNA produces more biologically meaningful alignment results than IsoRank does in most of the cases. And there is no clear difference between PBNA and IsoRank in remaining cases.

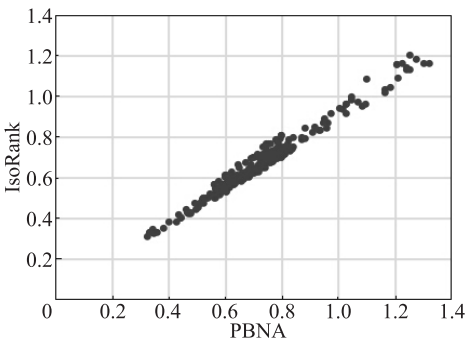

Fig. 4 GOC statistics of PBNA and IsoRank.

Here is an interesting observation: The alignment results of PBNA are mostly more biologically significant than those of IsoRank (Experiment 2), although they differ from each other greatly (Experiment 1). This leads to a conclusion that deviations may rise in those methods that ignore the probabilistic information. And by taking this information into consideration, PBNA produces novel and biologically meaningful alignment results.

We have now verified the necessity and effectiveness of PBNA.

3.3 GNAS of PBNA and Prob

In this experiment, we take Global Network Alignment Score (GNAS)[20] as the measurement. GNAS is one of the specific forms describing the general basic goal of biological network alignment methods (Eqs. (1) and (2)). Greater GNAS values indicate better alignments: more conserved interactions in alignment graph A_{12} (Section 2.2.4) and higher node sequence similarities. GNAS is defined as

$$\mathrm{GNAS} = \alpha \times |E| + (1-\alpha) \times \sum \mathrm{seq}(u, v).$$

$\mathrm{seq}(u, v)$ denotes the sequence similarity score between a pair of proteins. We employ the data from BLAST[20]. $|E|$ denotes the number of edges in A_{12}. We have both PBNA and Prob ran on the same datasets, and get 2 groups of GNAS values, each containing 244 values. The average values of them are shown in Table 2.

Table 2 shows us that $|E|$ and GNAS of PBNA are both better than Prob since PBNA adopts an approach aiming at increasing $|E|$ when extracting alignments from similarity matrix $\boldsymbol{R}$ (Section 2.2.4). This approach repeatedly adds mappings to the final alignment result, and these mappings are chosen from the neighbors of existing mappings in the result. The increase of $|E|$ then results in the increase of GNAS.

Table 2 $|E|$ and GNAS statistics.

	$\lvert E\rvert$	GNAS
PBNA	5.30	3.42
Prob	4.87	3.26

3.4 Gene ontology of PBNA and Prob

This experiment follows almost the same procedure as Experiment 2. The results are shown in Fig. 5.

Figure 5 shows a similar distribution pattern compared to Fig. 4. For most of the points (202 points, 83% of total), their x-coordinate values are 2%-10% greater than their y-coordinate values; for the remaining ones, their x-coordinate and y-coordinate values are almost identical. This indicates that based on the GOC measure, PBNA produces more biologically meaningful alignment results than Prob does in most cases. And there is no noticeable difference between PBNA and Prob in the remaining cases.

3.5 Time analysis

In this experiment, we evaluate the running time of PBNA and Prob.

The most time-consuming step for both PBNA and Prob is constructing similarity matrix, which takes about 90% of the entire running time. Therefore, it is reasonable that we measure only this step's running time in order to evaluate the entire algorithm time efficiency. The results are shown in Table 3.

Table 3 shows us that the construction of similarity matrices in PBNA consumes much less time than the counterpart in Prob since PBNA only considers the contributions of the Contributor set when computing the similarity score of a pair of nodes (u, v), while Prob takes the similarity score of every node pair in the neighbor set into consideration. The complexity thus decreases from $|N(i)| \times |N(j)|$ to $\min(N(i), N(j))$, which then results in the dramatic reduction of running time.

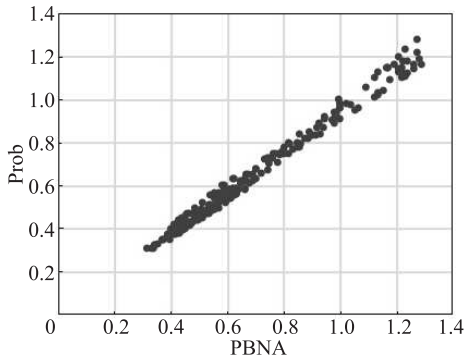

Fig. 5 GOC statistics of PBNA and Prob.

Table 3 PBNA and Prob algorithm time statistics.

Method	Average (s)	Max (s)
PBNA	1.1	5.7
Prob	125.4	545.5

We now can draw the conclusion based on the results of Experiments 3, 4, and 5, that the introduction of the concept of Contributor improves the alignment method. In the construction of similarity matrices, Contributor ignores the contributions of the node pairs that have small chances of coexistence in the final alignment; and when extracting alignment results, Contributor helps to increase the $|E|$ in A_{12}.

4 Conclusions

More and more biological network data produced by high-throughput techniques need to be analyzed in bioinformatics methods, such as biological network alignment. However, almost every existing alignment method can only solve the deterministic alignment problem. In this paper, we introduce an improved probabilistic biological network alignment, called PBNA. PBNA is based on IsoRank, but it can utilize the probabilistic information, allowing at least one network to be probabilistic. Also, Contributor and PGF are employed by PBNA to improve the accuracy of node similarity value and reduce the complexity of computing the expectation of random variables in similarity matrix. Moreover, experiments using GOC and GNAS as evaluation metrics validate the necessity and effectiveness of our PBNA, and demonstrate that PBNA can produce more biologically significant alignment results and has lower time complexity as compared to the similar approach.

The future work based on results of this paper may include: (1) the alignment of multiple biological networks, (2) the efficient preprocessing of biological network data, and (3) the practical evaluation metrics considering both biological significance and network topology information.

Acknowledgements

This work was supported by the Natural Science Foundation of Jiangsu Province under Grant No. BK2012742.

References

[1] R. Singh, J. Xu, and B. Berger, Global alignment of multiple protein interaction networks with application to functional orthology detection, *PNAS*, vol. 105, no. 35, pp. 12763-12768, 2008.

[2] X. L. Guo, L. Gao, and X. Chen, Models and algorithms for alignment of biological networks, (in Chinese), *Journal of Software*, vol. 21, no. 9, pp. 2089-2106, 2010.

[3] H. Ogata, W. Fujibuchi, and S. Goto, A heuristic graph comparison algorithm and its application to detect functionally related enzyme clusters, *Nucleic Acids Research*, vol. 28, no. 20, pp. 4021-4028, 2000.

[4] E. Hirsh and R. Sharan, Identification of conserved protein complexes based on a model of protein network evolution, *Bioinformatics*, vol. 23, no 2, pp. e170-e176, 2007.

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

[6] M. Narayanan and R. M. Karp, Comparing protein interaction networks via a graph match-and-split algorithm, *Journal of Computational Biology*, vol. 14, no. 7, pp. 892-907, 2007.

[7] A. Todor, A. Dobra, and T. Kahveci, Probabilistic biological network alignment, *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 10, no. 1, pp. 109-121, 2013.

[8] Z. Liang, M. Xu, M. Teng, and L. Niu, Comparison of protein interaction networks reveals species conservation and divergence, *BMC Bioinformatics*, vol. 7, no. 1, pp. 457-472, 2006.

[9] A. Chatr-Aryamontri, A. Ceol, L. M. Montecchi, G. Nardelli, M. Schenider, L. Castagnoli, and G. Cesareni, MINT: The molecular interaction database, *Nucleic Acids Research*, vol. 35, no. s1, pp. 572-574, 2007.

[10] M. Koyuturk, Y. Kim, U. Topkara, S. Subramaniam, W. Szpankowski, and A. Grama, Pairwise alignment of protein interaction networks, *Journal of Computational Biology*, vol. 13, no. 2, pp. 182-199, 2006.

[11] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, and H. Butler, Gene ontology: Tool for the unification of biology, *Nat. Genet*., vol. 25, no. 1, pp. 25-29, 2000.

[12] W. Tian and N. F. Samatova, Pairwise alignment of interaction networks by fast identification of maximal conserved patterns, *Pacific Symposium on Biocomputing*, vol. 14, pp. 99-110, 2009.

[13] E. Almaas, Biological impacts and context of network theory, *The Journal of Experimental Biology*, vol. 210, pp. 1548-1558, 2007.

[14] G. W. Klau, A new graph-based method for pairwise global network alignment, *BMC Bioinformatics*, vol 10, pp. 59-67, 2009.

[15] F. Towfic, M. H. W. Greenlee, and V. Honavar, Aligning biomolecular networks using modular graph kernels, in *Algorithms in Bioinformatics.* Springer Berlin Heidelberg, 2009, pp. 345-361.

[16] W. Guan, J. Wang, and F. He, The advance in research methods for large-scale protein-protein interactions (in Chinese), *Chinese Bulletin of Life Sciences*, vol. 18, no. 5, pp. 507-512, 2006.

[17] E. Silva and M. P. H. Stumpf, Complex networks and simple models in biology, *Journal of The Royal Society Interface*, vol. 2, no. 5, pp. 419-430, 2005.

[18] J. Sun, J. Xu, Y. Li, and T. Shi, Analysis and application of large-scale protein-protein interaction data, (in Chinese), *Chinese Science Bulletin*, vol. 50, no. 19, pp. 2055-2060, 2005.

[19] P. Jancura, J. Heringa, and E. Marchiori, Divide, align and full-search for discovering conserved protein complexes, in *Machine Learning and Data Mining in Bioinformatics.* Springer Berlin Heidelberg, 2008, pp. 71-82.

[20] A. E. Aladag and C. Erten, SPINAL: Scalable protein interaction network alignment, *Bioinformatics,* vol. 29, no. 7, pp. 917-924, 2013.

[21] H. W. Kuhn, The Hungarian method for the assignment problem, *Naval Research Logistics Quarterly,* vol. 2, nos. 1-2, pp. 83-97, 1955.

[22] M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori, The KEGG resource for deciphering the genome, *Nucleic Acids Research*, vol. 32, no. s1, pp. D277-D280, 2004.

Muwei Zhao is currently a student in School of Computer Science and Engineering of Southeast University for his master degree. His research interests include bioinformatics and data mining. He received his bachelor's degree in 2010 from Jilin University, China.

Wei Zhong is an associate professor in the Division of Math and Computer Science at University of South Carolina Upstate. He is an elected fellow of International Society of Intelligent Biological Medicine and IEEE Senior Member. He received his PhD degree in Computer Science from Georgia State University, USA, in 2006. His research interests include big data, data mining, computer security, and bioinformatics.

Jieyue He received her BS and MS degrees in Department of Computer Science and Technique from Nanjing University, China, and her PhD degree in the School of Computer Science and Engineering from Southeast University, China. She is currently a professor of the School of Computer Science and Engineering, Southeast University, China. Her current research interests include bioinformatics, data mining, machine learning, and big data.