Published in final edited form as:

IEEE/ACM Trans Comput Biol Bioinform. 2016 ; 13(3): 483–493. doi:10.1109/TCBB.2015.2465957.

An Adaptive Hybrid Algorithm for Global Network Alignment

Jiang Xie,
School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China

Chaojuan Xiang,
School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China

Jin Ma,
School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China

Jun Tan,
School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China

Tieqiao Wen,
School of Life Sciences, Shanghai University, Shanghai, 200444, China

Jinzhi Lei, and
Zhou PeiYuan Center for Applied Mathematics, MOE Key Laboratory of Bioinformatics, Tsinghua University, Beijing 100084, China

Qing Nie
Department of Mathematics, Center for Mathematical and Computational Biology, and Center for Complex Biological Systems, University of California Irvine, Irvine, CA 92697, USA

The School of Computer Engineering and Science, Shanghai University, Shanghai, 200444, China

Jiang Xie: jiangx@shu.edu.cn; Chaojuan Xiang: cjxiang@shu.edu.cn; Jin Ma: jma518@shu.edu.cn; Jun Tan: tanjun_2525@shu.edu.cn; Tieqiao Wen: tqwen@staff.shu.edu.cn; Jinzhi Lei: jzlei@tsinghua.edu.cn; Qing Nie: qnie@math.uci.edu

Abstract

It is challenging to obtain reliable and optimal mapping between networks for alignment algorithms when both nodal and topological structures are taken into consideration due to the underlying NP-hard problem. Here, we introduce an adaptive hybrid algorithm that combines the classical Hungarian algorithm and the Greedy algorithm (HGA) for the global alignment of biomolecular networks. With this hybrid algorithm, every pair of nodes with one in each network is first aligned based on node information (e.g., their sequence attributes) and then followed by an adaptive and convergent iteration procedure for aligning the topological connections in the networks. For four well-studied protein interaction networks, i.e., *C.elegans*, yeast, *D.melanogaster* and human, applications of HGA lead to improved alignments in acceptable running time. The mapping between yeast and human PINs obtained by the new algorithm has the largest value of common Gene Ontology (GO) terms compared to those obtained by other existing algorithms, while it still has lower Mean normalized entropy (MNE) and good performances on several other measures. Overall, the adaptive HGA is effective and capable of providing good mappings between aligned networks in which the biological properties of both the nodes and the connections are important.

Index Terms

global alignment; hybrid algorithm; protein interaction network

I. Introduction

NEtwork alignment algorithms are often used to compare different biomolecular networks to explore signaling pathways, conserved function modules, regulatory relationships, and the evolution of species [1]. For example, in protein interaction networks (PINs), the protein-protein interaction (PPI) data such those in yeast [2], and a variety of other organisms, including bacteria, worm, fly and human, may be used with the network alignment approach to determine whether the conserved sequences in different species have similar functions [3]. In this way, the network annotation of any species can be efficiently transferred to another species [4] as a means to study human disease using the model organism [5].

Typically, a biomolecular network consists of thousands of biomolecules with each node having biological attributes that influence its functions. In addition, the connections between nodes are also directly linked to their biological functions [6], [7]. For example, it was reported that the accuracy of the prediction of protein functions based on a network was approximately 58%–63% versus an accuracy of 37%–53% predicted on the basis of sequence similarity. In the same study, the success rate of PPI prediction using a network was 40%–52% versus a success rate of 16%–31% from sequence information [8]. However, when aligning biomolecular networks, optimal mapping among networks is difficult when both nodes and topological structures are taken into consideration. Such mapping becomes more intricate because of the addition of false-positive and false-negative results from experimental data.

Two major approaches have been developed to address this complexity: heuristic methods and parameterized algorithms.

A typical feature of heuristic methods is the use of the Greedy algorithm (GA). The PathBlast family tools [9]–[12] are representative of heuristic approaches. NetworkBLAST-M [13], an improved version of NetworkBLAST [11], attempts to align two or more networks by greedily searching conserved regions. Graemlin 2.0 [14] considers phylogenetic relationships to infer a network, then optimizes the learned objective function. IsoRank [15] formulates the alignment as an eigenvector problem and uses a greedy algorithm to obtain the final alignment. The algorithm is extended to IsoRankN [16], which uses spectral clustering on a graph to improve the global network alignment. Also based on IsoRank, a new algorithm that uses interaction probabilities is developed to explore more meaningful alignments [17]. The PISwap algorithm [18] computes a PINs alignment based on a local optimization heuristic. GRAAL [7] is a greedy “seed-and-extend” approach analogous to the popular BLAST [19] algorithm for sequence alignment. MI-GRAAL [20] is a further improvement of GRAAL that is designed to build a matrix of confidence scores based on different measures of similarity between nodes and that aligns networks using a greedy algorithm. GEDEVO [21] is an ingenious method based on the Graph Edit Distance (GED) model that aligns networks using a novel evolutionary algorithm that attempts to minimize

the GED. NETAL [22] aligns networks very quickly; it pre-processes topological scores and biological scores separately, then uses GA to find the global alignment and focuses on topological similarities in the current version. Both GEDEVO and NETAL work with large networks and attain high EC (edge correctness) values. Each of these algorithms has its advantages and contributes to the comparison of biomolecular networks. However, heuristic approaches are limited because they do not guarantee the quality of the solution. In some cases, these methods are unstable and depend largely on the experience of the researcher.

Compared to typical heuristic methods, parameterized algorithms can achieve an optimal alignment at the cost of more simulation time. An example is the color-coding method [23], which can be used to obtain a better alignment by finding simple paths and circles of a specified edge length along with other small subgraphs within a graph. However, the search target is usually limited to a small number of proteins due to the computational cost. QPath [24] incorporates the query with an inexact match to speed up the original color-coding method but has similar time complexity issues. Another example is the H-GRAAL [25] method, which is based on the Hungarian algorithm (HA) [26] and which finds an optimal assignment for a given cost matrix. Although H-GRAAL is more accurate than its corresponding heuristic counterpart GRAAL [7], it is substantially more expensive [25].

We present a hybrid algorithm that combines HA and GA (HGA) to align biomolecular networks in this paper. Considering both similarities between biomolecules and interactions, we use HA to obtain better node assignment and GA to reduce the computation time, which is critical for resolving the alignment of large-scale networks.

Typically, the alignment of networks can be carried out by two distinct types of methods: local and global algorithms [8]. Global algorithms compare entire node-and-edge structures among networks [7], [10], [14]–[16], [18], [20], whereas local algorithms identify local regions in networks that exhibit similar node and edge structures [9], [13]. HGA explores global algorithms to align a pair of networks.

The rest of this paper is organized as follows. In Section 2, network alignment is illustrated, and the hybrid algorithm HGA is presented. In Section 3, we explore the adaptive parameters for HGA and examine the effect of the key parameters. In Section 4, we use PINs as an example of an application of HGA and compare it with several other algorithms. Finally, Section 5 presents a discussion and our conclusions.

II. Methods

A. Problem Formulation

Biomolecular networks can be represented as graphs in which nodes (or vertices or points) and edges denote biomolecules and their connections, respectively. For example, in PINs, the nodes are the proteins and the edges are their interactions. The object of optimizing the alignment of two networks is to obtain a mapping that best describes the similarity of the nodes as well as that of their connections. Such an alignment problem is described as follows.

Two networks A and B are represented by $G_A = (V_A, E_A)$ and $G_B = (V_B, E_B)$, respectively, in which $V_A = \{u_1, u_2, \ldots, u_n\}$ and $V_B = \{v_1, v_2, \ldots, v_m\}$ are nodes and $n \leq m$ and $E_A = \{e_{ij}^A = (u_i, u_j) \,|u_i, u_j \in G_A\}$, $E_B = \{e_{ij}^B = (v_i, v_j) \,|v_i, v_j \in G_B\}$ are edges.

A network is represented by an adjacent matrix X for which each entry x_{ij} is 0 if no edge connects nodes u_i and u_j; otherwise, x_{ij} is 1. For a network A of n nodes and a network B of m nodes, the adjacent matrices are $A_{n\times n} = (a_{ij})$, $1 \leq i, j \leq n$, $a_{ij} \in \{0,1\}$, and $B_{m\times m} = (b_{ij})$, $1 \leq i, j \leq m$, $b_{ij} \in \{0,1\}$, respectively.

We define a set N to denote the neighbors of a node x_i in network A or B:

$$N_A(u_i) = \{u_l | a_{il} = 1\} \quad (1)$$

$$N_B(v_i) = \{v_l | b_{il} = 1\} \quad (2)$$

Here, $N_A(u_i)$ is the set including neighbors of node u_i in network A, and $N_B(v_i)$ is the set including neighbors of node v_i in network B.

The alignment of two networks is to find a mapping $\varphi = \{\varphi_1, \varphi_2\} : G_A \rightarrow G_B$ such that

1. $\forall u_i \in G_A$, there exists a unique $\varphi_1(u_i) \in G_B$;
2. $\forall e = (u_i, u_j)$, there exists a unique $\varphi_2(e) = (\varphi_1(u_i), \varphi_1(u_j))$.

This mapping is obtained by optimizing an objective score that may include the node alone or both the edge and the node. In the matrix representation of this mapping, there is only one nonzero (i.e., one "1") element in each row and all values of "1" must be in different columns. That is, for each node in network A, there is one node in network B that it matches.

To enable the mapping, first, the "similarity" between a pair of nodes (u_i, v_j), $u_i \in G_A$, $v_j \in G_B$, $1 \leq i \leq n$, $1 \leq j \leq m$ must be estimated. This quantity, the similarity coefficient s_{ij}, may or may not contain information for the edge connecting the two particular nodes. For the two networks, the similarity matrix is $S = (s_{ij})$, $i = 1 \ldots n$, $j = 1 \ldots m$.

Such an alignment problem is of high computational complexity because of the underlying subgraph isomorphism problem, which is known to be NP-hard [27] and only special cases of which can be solved in polynomial time [28].

B. Algorithm

Here, we present an algorithm that integrates HA and GA to find an alignment for two networks. During the iterative process, at each re-matching step we first use HA for the preliminary assignment and then use GA to complete the matching. In this way, we can take advantage of the strength of GA, which is simple and effective (but is less accurate), and the strength of HA, which optimally matches the nodes between the aligned networks (but is computationally expensive).

There are two major steps in the HGA algorithm. First, because we consider both the node similarity and topological similarity of the two networks, we iteratively compute the overall similarity matrix by gradually incorporating the topological and node information for the two networks; second, we construct the mapping matrix to establish the matching of the networks.

1) An iterative procedure to compute the similarity matrix

i. Initialization:

The initial similarity matrix, denoted by $S^{(t)} \equiv (s_{ij}^{(t)})$, where $t = 0$, can be based only on node information, for example, using homologous coefficients of proteins computed by BLAST for PINs.

ii. Updating the similarity matrix:

The similarities of neighbors for each pair of matching nodes (u_p, v_q) are then rewarded with a positive number ω, leading to an updated similarity matrix with the following entries:

$$s_{kl}^{*} = \begin{cases} s_{kl}^{(t)} + \omega, & \text{if } u_k \in N_A(u_p), v_l \in N_B(v_p) \\ s_{kl}^{(t)}, & \text{otherwise} \end{cases} \quad (3)$$

All matching nodes can be found by the mapping matrix $M^{(t)}$ described in Section II-B2.

This approach using ω for the neighbors is based on the Neighbor Biased Mapping (NBM) [29] method; thus, neighbors have more opportunities to match with each other when two nodes are matched. Moreover, HGA defines the reward parameter (ω) that is adaptively updated according to the degree of the neighbors. The choice of ω will be discussed in Section II-C1.

iii. Adding topology information:

Given any two nodes u_i, v_j in the networks A and B, respectively, their topological similarities are computed based on an approach previously used for the topological similarity of biomolecular networks [5], which we have called the topological similarity parameter (TSP). The TSP includes θ_{ij}^{1} and θ_{ij}^{2}, which are updated according to the rule that two nodes are similar if they link or do not link to similar nodes [30].

θ_{ij}^{1} represents the average similarity between the neighbors of u_i and v_j, and θ_{ij}^{2} represents the average similarity between the non-neighbors of u_i and v_j. For θ_{ij}^{1}, we first obtain the sum of similarities of each pair of nodes u_k and v_l that are the neighbors of u_i and v_j and then normalize the sum by the total number of neighbor pairs. If both u_i and v_j are isolated points, θ_{ij}^{1} is defined as the sum of similarities of every pair (u_k, v_l) normalized by the total number of such

pairs, $n \times m$. For θ_{ij}^2, we first obtain the sum of similarities of each pair of nodes u_k and v_l that are the non-neighbors of u_i and v_j and then normalize the sum by the total number of non-neighbor pairs, $(n - |N_A(u_i)|)(m - |N_B(v_j)|)$. If both u_i and v_j link to all of the nodes in their networks, then θ_{ij}^2 is defined as the sum of similarities of every pair (u_k, v_l) normalized by the total number of such pairs, $n \times m$.

$$\theta_{ij}^1 = \begin{cases} \frac{\sum_{(u_k \in N_A(u_i), v_l \in N_B(v_j))} s_{kl}^*}{|N_A(u_i)||N_B(v_j)|}, & \text{if } |N_A(u_i)| \neq 0 \text{ and } |N_B(v_j)| \neq 0 \\ \frac{\sum_{(u_k \in A, v_l \in B)} s_{kl}^*}{m \times n}, & \text{if} |N_A(u_i)| = |N_B(v_j)| = 0 \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

$$\theta_{ij}^2 = \begin{cases} \frac{\sum_{(u_k \notin N_A(u_i), v_l \notin N_B(v_j))} s_{kl}^*}{(n - |N_A(u_i)|)(m - |N_B(v_j)|)}, & \text{if } |N_A(u_i)| \neq n-1 \text{ and } |N_B(v_j)| \neq m-1 \\ \frac{\sum_{(u_k \in A, v_l \in B)} s_{kl}^*}{m \times n}, & \text{if} (n - |N_A(u_i)|) = (m - |N_B(v_j)|) = 1 \\ 0, & \text{otherwise} \end{cases}$$

(5)

When PINs evolve from a common ancestor of two closely related species, proteins within the two networks might have essentially identical amino acid sequences. During evolution, proteins may be duplicated, inserted or deleted in networks [31]; hence, the topology of the evolved networks changes. To find the most similar alignment between the networks, topological information must be incorporated along with sequence information [5].

$$s_{ij}^{(t+1)} = s_{ij}^{(0)} + \frac{\theta_{ij}^1 + \theta_{ij}^2}{2} \quad (6)$$

iv. Continue to step ii) until one of the following conditions is satisfied:

a. $\left| S^{(t)} - S^{(t-1)} \right|_{max} \leq \varepsilon,$

b. $\left| S^{(t)} - S^{(t-2)} \right|_{max} \leq \varepsilon,$

where ε is a prescribed tolerance. Typically, we choose $\varepsilon = 0.01$ to allow 1% error.

c. A sum score, which will be given later in formula (14), does not change in three continuous iterations. The sum score is calculated

in each iteration to record the score of the mapping matrix at that iteration, as described in Section II-B2.

The convergence of a similar iteration procedure for obtaining a similarity matrix has been previously proved based on the power method [30].

2) Construction of the mapping matrix—Now that a sequence of similarity matrices $S^{(t)}$ for $t = 0, \ldots, k$ has been obtained, the mapping matrix for each of the similarity matrices must be constructed.

i. Initial matching:

For $S^{(0)}$, which might only contain initial similarities of the nodes in network A and B, it is simple to calculate the corresponding mapping matrix $M^{(0)}$ using HA:

$$m_{ij}^{(0)} = \begin{cases} 1, & \text{if } u_i \in G_A \text{ is matching with } v_j \in G_B \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

in which $i = 1 \ldots n$, $j = 1 \ldots m$.

ii. Re-matching:

Finding the mapping between the nodes of two networks is similar to assigning tasks to workers. HA is good at such assignment problems but is with time consuming especially for a large number of tasks. $S^{(t)}(t > 0)$ contains the topological information for the network, the similarity matrix becomes more complex and it becomes more difficult to obtain the mapping matrix directly using HA alone. We use HGA to divide $S^{(t)}$ into two matrixes: the H-matrix, in which each row has at least h nonzero entries, and the G-matrix, which collects the remaining entries of $S^{(t)}$. The value h is an empirical value that might depend on the number of homologs of proteins in the other network, providing a balance of good performance and good mapping. A large h corresponds to a small H-matrix, resulting in good mapping among fewer nodes, whereas a small h indicates a large H-matrix that is time-consuming to solve. For example, the H-matrix of PINs addresses those proteins that correspond to a greater number of similar proteins in the other network in their initial state. We explored a wide range of values for h, and we set it as 5 for our experiments. For the alignment between the yeast and human PINs, yeast proteins that have at least 5 homologous proteins in the human network are collected into the H-matrix. HA is used to assign proteins of yeast in the H-matrix to match their best counterparts in human PIN. The G-matrix is sparser. We then apply GA to the yeast proteins in the G-matrix that have not been matched in the H-matrix.

Specifically, suppose $S^{(t)}(t > 0)$ is divided into the H-matrix and G-matrix as follows:

H-matrix:

$$\begin{pmatrix} s_{h_1 1} & s_{h_1 2} & \cdots & s_{h_1 p} & \cdots & s_{h_1 m} \\ s_{h_2 1} & s_{h_2 2} & \cdots & s_{h_2 p} & \cdots & s_{h_2 m} \\ \vdots & \vdots & & \vdots & & \vdots \\ s_{h_k 1} & s_{h_k 2} & \cdots & s_{h_k p} & \cdots & s_{h_k m} \\ \vdots & \vdots & & \vdots & & \vdots \\ s_{h_j 1} & s_{h_j 2} & \cdots & s_{h_j p} & \cdots & s_{h_j m} \end{pmatrix} \quad (8)$$

G-matrix:

$$\begin{pmatrix} s_{g_1 1} & s_{g_1 2} & \cdots & s^*_{g_1 p} & \cdots & s_{g_1 m} \\ s_{g_2 1} & s_{g_2 2} & \cdots & s^*_{g_2 p} & \cdots & s_{g_2 m} \\ \vdots & \vdots & & \vdots & & \vdots \\ s_{g_k 1} & s_{g_k 2} & \cdots & s^*_{g_k p} & \cdots & s_{g_k m} \\ \vdots & \vdots & & \vdots & & \vdots \\ s_{g_i 1} & s_{g_i 2} & \cdots & s^*_{g_i p} & \cdots & s_{g_i m} \end{pmatrix} \quad (9)$$

where $g_i + h_j = n$. First, we assign proteins in the H-matrix to match their best counterparts using HA and mark the corresponding columns in the G-matrix. For example, if $u_{h_k} \in G_A$ matches with $v_p \in G_B$, that is, $m_{h_k p}$ in $M^{(t)}$ is 1, then we mark column p in the G-matrix. Next, we use GA to assign those unmatched proteins in the G-matrix, and we no longer consider those in the marked columns (e.g. column p). Finally, we obtain $M^{(t)}$, which is computed by HA and GA. The final output of the mapping matrix M^F is the desired mapping that has the best sum score based on equation (14).

3) Scoring the mapping matrix to obtain the optimal mapping—For a typical global network alignment problem, each node in one network is only matched with one node in the other network. Edge Correctness (EC) is often used to measure the degree of topological similarity [7], [15], [20], [32] and can be estimated as the percentage of matched edges,

$$\mathrm{EC} = \frac{|\{(u_i, u_j) \in E_A \Lambda(\phi(u_i), \phi(u_j)) \in E_B\}|}{|E_A|} \times 100\%. \quad (10)$$

Another way to measure alignment is to use the Largest Common Connected Subgraph (LCCS) [20]. A greater number of nodes and edges in the LCCS implies a better alignment. Obviously, both the EC and the LCCS only indicate topological similarities other than the original attributes of the nodes, such as the amino acid sequences or functional similarities of the proteins in PINs.

If the correct mappings between the aligned networks G_A and G_B are known, Node Correctness (NC) and Interaction Correctness (IC) can also be used to indicate the alignment quality. NC is the percentage of nodes in G_A that are correctly mapped to nodes in G_B, and IC is the percentage of correctly mapped interactions [25].

For PINs, one can posit an objective function based on similarities of network topology, the sequences and their relative weights [33]. Another popular approach is Gene Ontology (GO) [34], in which an alignment has a higher biological relevance when more matched proteins in this alignment share more GOs [20], [25].

Mean normalized entropy (MNE) is also an useful indicator of the function coherence of the alignments. Lower MNE means that the matched proteins have greater consistency, which is a better indication of proteins sharing the same function [35].

Here, we include information on both nodes and edges in the score (the points and edges score, PE) that is defined similarly to the measures used by INM [5].

First, we define $sim(u_i, v_j)=s_{ij}^{(0)}$.

Next, we set $wt\left(e_{ij}^{A}\right)$ as the edge weight between nodes u_I and u_j in network A, which indicates the reliability of this edge (it is usually set to be 1 by default). We define the following:

$$\mathrm{PE}\left(G_A, G_B\right) = ES + PS = \frac{1}{2}\sum_{i,j=1}^{n}\mathrm{ES}\left(e_{ij}^{A}\right) + \sum_{i=1}^{n}\mathrm{PS}\left(u_i\right) \tag{11}$$

in which,

$$\mathrm{ES}\left(e_{ij}^{A}\right) = \begin{cases} wt\left(e_{ij}^{A}\right), \exists e_{\phi(u_i),\phi(u_j)} \in E_B, \\ \quad \text{and } sim\left(u_i, \phi\left(u_i\right)\right) > 0, \\ \quad \text{and } sim\left(u_j, \phi\left(u_j\right)\right) > 0 \\ 0, \quad \text{otherwise} \end{cases} \tag{12}$$

$$\mathrm{PS}\left(u_i\right) = \begin{cases} sim\left(u_i, \phi\left(u_i\right)\right), \sum_{j=1}^{n}\mathrm{ES}\left(e_{ij}^{A}\right) > 0 \\ 0, \quad \text{otherwise} \end{cases} \tag{13}$$

PE is clearly stricter than EC because it reflects the status of both the node and edge matches in the mapping. The score for an edge (the Edge Score, ES) equals zero if any of its nodes does not match with its similar nodes, and the score for a node (the Point Score, PS) equals zero if none of its edges has a score.

To illustrate the scoring, we show an example for the alignment of a three-node network to a four-node network (see Fig. 1).

Because EC is a popular quantity used by many algorithms and PE indicates the level of matching of both nodes and edges, we compute the final sum score of a mapping as follows:

$$\mathrm{SS}=100\times \mathrm{EC}+\mathrm{PE} \quad (14)$$

For each iteration, we calculate the SS of its mapping matrix, select the one with the highest SS as the final mapping when the computation is completed, and then obtain its corresponding PE, EC and so on.

4) Flow chart—Fig. 2 shows a summary flow chart for HGA.

The computation of topological similarities for each pair of nodes in the two networks, θ_{ij}^{1} and θ_{ij}^{2}, consumes the dominant portion of the CPU time, and it has a time complexity of $O(n\times m)^2$.

C. Parameters

1) An adaptive parameter—Two similar proteins may often interact with other similar proteins. In other words, if protein u in network A is similar to v in network B, protein a, a neighbor of u, is likely to be similar to protein b that is a neighbor of v. HGA adopts NBM [29] when it aligns networks A and B, and it rewards the similarity coefficient of a and b with ω so that they have a greater opportunity to match each other [5].

In particular, we define ω as follows:

$$\omega=sim\,(u,v)\,/\,|N_{A}\,(a)| \quad (15)$$

where $sim(u, v)$ is the homologous coefficient between u and v and $|N_A(a)|$ is the degree, which is the number of its neighbors, of a in the network A. This relation implies that the greater the similarity between u and v (corresponding to a higher $sim(u, v)$), the greater the likelihood that their neighbors are also similar. The value of ω reaches a maximum when all of the neighbors of a and b are similar to each other. We note that it is important to normalize by the total number of neighbors ($|N_A(a)|$) to account for the relative contribution of the neighbors in formula (15).

Because the similarity of the neighbors of a node affects the overall match between the nodes, it is natural that the award parameter ω should depend on the topological locations for each node rather than remaining as a fixed or an empirical value as seen in the previous study [5].

2) Relaxation factor—When the sequence similarity and the topological similarity are considered equally important in the network alignment process, the iteration of the similarity

matrix can be updated with equation (6). A more flexible approach is the introduction of a relaxation factor (α) to weight the sequence or the topology information when HGA is employed. With this weighting, equation (6) becomes

$$s_{ij}^{(t+1)} = \alpha \cdot s_{ij}^{(0)} + (1-\alpha)\frac{\theta_{ij}^1 + \theta_{ij}^2}{2}, \quad (16)$$

implying that when α is small, the alignment is mainly affected by the topological information, and when α is close to 1, the alignment is mostly controlled by the sequence homology information.

We will explore more details about these two parameters in Section III.

III. Explore the Key Parameters

A. Data Sets

1) Yeast and human—The yeast *S.cerevisiae* PIN, Ynet2390n, contains 2390 proteins with 16,127 interactions [36] [20]. To perform this comparison, we extracted a subnetwork (Ysubnet38n) from Ynet2390n. Ysubnet38n was chosen to have 131 edges and 38 nodes such that each node had homologous proteins in the human PIN and the average homologous proteins were greater than 10. The human PIN [20], Hnet9141n, contained 9141 proteins with 41,456 interactions.

In our experiments, we downloaded 6641 yeast sequences from UniProt [37] on May 18, 2012, and 9588 human sequences were taken from [38].

The homologous coefficient, $sim(u_i, v_j)$, was calculated based on the E-value of BLAST [19]. We used the following equation (17) to transform the E-value to a value between 0 and 1, which was then used as the entry for the initial similarity matrix $S^{(0)}$.

$$sim(u_i, v_j) = 1 \Big/ \left(1 - \frac{1}{log\mathrm{Evalue}(u_i, v_j)}\right) \quad (17)$$

2) C.elegans and D.melanogaster—The *C.elegans* (CE) and the *D.melanogaster* (DM) PINs have been collected by Bonnie Berger's group [39]. The former has 5851 interactions among 2745 proteins, in which there are 1469 multiple edges and 40 loops. The latter has 26,712 interactions among 6709 proteins, in which there are 6952 multiple edges and 110 loops. The H-matrix collects 687 of 2745 rows of this CE-DM similarity matrix.

Using the PISwap approach [18], we normalized the BLAST scores retrieved from the IsoRank website [39] and used them as the initial similarity coefficients for HGA.

B. The Adaptive Parameter

There are two ways to award a similarity coefficient through ω. The first approach is to award it by constructing a new similarity matrix during each iteration of HGA. The similarity coefficient can also be awarded after all iterations are completed, which is similar to the approach in INM [5]. In the first approach, ω is involved in the iteration itself for better integration of the topological information and the sequence information.

Applications of these two approaches to the real data indicate that for large data sets (the first pairs of the networks shown in Table I), updating ω during each iteration is necessary. INM [5], based on a fixed ω, requires a very long computation time such that the algorithm becomes impractical. This computational burden is due to the time complexity of the matching process in INM. For smaller networks, such as the second set, Ysubnet38n and Hnet9141n, INM can compute the matching. All evaluation indicators, including PE, ES, EC and LCCS, are less than the corresponding cases using HGA, in which ω is updated during each iteration. For the alignment of the third set, INM generates a much higher EC and LCCS but the lowest PE and HGp.

Table I also contains the results for the case in which the “best” alignment was found only based on sequence information ($\alpha = 1$) without any iteration. Although HA was effective at task assignment, it was difficult to obtain a better PE or EC because the topological information was not integrated with the sequence information. For the third pair of networks, even the PE (62.08) and HGp (1523) based on sequences were slightly higher than those for HGA (57.96, 1488), and the EC and LCCS were much lower than HGA and INM.

If the topological information is integrated with the node information, for example, by setting $\alpha = 0.4$, alignment of these networks by HA alone, but not by a hybrid of HA and GA, becomes impractical because of the unacceptable computation time required.

C. The Relaxation Factor

As mentioned in Section II-C2, the alignment is more controlled by the sequence homology information when α is closer to 1. As shown in Fig. 3, PE was low, and the maximum EC was highest when α=0.1. When α increased (i.e., more sequence information was included), more homologous proteins were matched, leading to an increase in PE and a decrease in the maximum EC. However, PE cannot increase continuously because the number of matched edges is likely to decrease when more homology information is incorporated with a higher weight, leading to a decrease in PE. $\alpha = 0$ is a special case, which means that the original node information is ignored during the iteration after integration of the node information with the topological structure. When $\alpha = 0$, topological information is paramount, and the EC is almost equal to its lowest value when $\alpha = 0.1$. Although the PE is not at a minimum, the SS defined by formula (14) cannot reach a maximum; thus, we do not consider it a good choice for the best mapping.

In general, α may be set to be a value close to 0.5 for equal weighting of the topological and the homology information.

Of course, depending on the specific alignment objective and potential differing reliability of the source data for topology and homology, α can be adjusted to obtain the best mapping. PE may be used as an overall indicator for alignment even when one might not aim at achieving the highest PE. Similarly, EC may be used as a good indicator of similar topology between two networks when the alignment algorithm does not include both homology and topological information.

As noted in Section II-B, the alignment is obtained by maximizing SS among all iterations. When SS is at a maximum, the PE may be at a maximum, but the EC may not achieve the maximum value among all iterations. Fig. 3(b) plots the maximum EC among all iterations as well as the value of EC when the SS is a maximum. We found that these two ECs agree very well except when $\alpha = 0.1$ and $\alpha = 0$, indicating that HGA finds the alignment with the highest PE and the nearly highest EC. This finding suggests that the alignment found by HGA approximates the best mapping considering both sequence and topological information.

IV. Comparison with other methods

In this section, we use PINs as an example to compare HGA with several other algorithms. In particular, we aligned the yeast and human PINs.

A. Case 1

Because it was difficult to compare the accuracy and performance of different algorithms for the alignment of two networks that had no “exact solution”, the same network as the aligned networks was used instead. Specifically, we used Ysubnet38n and aligned it with itself using HGA. The H-matrix collected 22 of 38 rows of this yeast-yeast similarity matrix. As expected, for all α (described in Section II-C2) from 0 to 0.9, all nodes matched with themselves, and the two networks completely coincided with each other in alignment with HGA, in which all of NC, IC and EC were 100%.

NETAL [22] and GEDEVO [21] could generate the same mapping, in which all nodes in the two networks were matched with their exact counterparts. This reflects the fact that the two networks were duplicated.

We also aligned this subnetwork with itself using MI-GRAAL by exploring different combinations of the four topological and sequence measures presented in [20]. We found that no alignment could make the two networks completely coincide with each other except for a particular choice of parameter of *SeqD* (p=16). The number of correctly mapped nodes obtained by other parameters was typically in the range of 2 to 28, with the corresponding NC ranging from 5.26% to 73.68%, and the number of correctly mapped edges ranged from 47 to 131, with the corresponding mapped rate ranging from 35.88% to 100%. Interestingly, we found that even when the rate was 100%, the NC was not necessarily 100% because the nodes of some edges were not matched to themselves.

None of IsoRank, PISwap and GRAAL was able to find the exactly matched mapping. Although the EC of IsoRank was 100%, the NC was 86.84%. The NC and EC of PISwap

were 57.89% and 47.33%, respectively. We carried out 10 alignments to calculate the NC and EC of GRAAL, in which the NC was from 63.16% to 86.84% and the EC was from 99.24% to 100% with the parameter α=0.8, which was the core alignment of GRAAL.

B. Case 2

We next aligned Ysubnet38n with Hnet9141n using HGA, IsoRank, GRAAL, MI-GRAAL, NETAL, GEDEVO and PISwap. The H-matrix contained 28 of 38 rows of this yeast-human similarity matrix.

For this alignment, HGA had a PE=76.41 with an ES=55 and a PS=21.41, whereas the other algorithms in Table II had PE, ES and PS=0 except for MI-GRAAL and PISwap. Moreover, the number of HomoloGene pairs among matched proteins (HGp) of HGA was 38, which was the same in PISwap, meaning that all of the proteins in Ysubnet38n were matched with their homologous proteins. In contrast to the mapping of HGA and PISwap, none of the proteins matched by IsoRank, GRAAL, and GEDEVO were homologous (HGp =0), and far fewer pairs of proteins were homologous for the alignment of NETAL or MI-GRAAL.

Table II also shows that the EC by HGA was higher than the corresponding score by PISwap, IsoRank, GRAAL, MI-GRAAL and GEDEVO and was very close to that from NETAL. Moreover, the LCCS of HGA was greater than that for PISwap, IsoRank, GRAAL, and MI-GRAAL.

Comparing the alignment of the networks in Table III and Table II, the main difference between the two groups of aligned networks is that all proteins in Ysubnet38n had homologous proteins in Hnet9141n (Table II), but some of the proteins (approximately 20%) in Ynet2390n had no homologous proteins in Hnet9141n (Table III). The homologous relationship was observably different between these two pairs of networks. Corresponding to the increased rate of homologous proteins between the two aligned networks (from 80% to 100%), the EC of HGA was increased by 28.58% (from 13.4% to 41.98%). The EC of NETAL ranged from 36.1% to 42.75%, an increase of 6.65%, and the EC of GEDEVO ranged from 22.45% to 35.11% with the parameter *maxsame* = 5000, an increase of 12.66%. IsoRank and GRAAL were similar to GEDEVO. Their ECs were increased by 13.36% and 15.12%, respectively. Using the parameter *SeqD*, the sequence matrix presented in [20], MI-GRAAL had an EC that ranged from 14.78% to 38.17%, an increase of 23.39%; the EC of PISwap ranged from 8.3% to 32.82%, an increase of 24.52%; these values are close to that of HGA.

In most cases, it should be expected that the more homologous proteins exist between the two networks, the more similar the two networks become, leading to a larger value of EC. HGA correctly captured this network alignment property.

C. Case 3

Next, we explored larger networks: the yeast PIN Ynet2390n and the human PIN Hnet9141n. The H-matrix collected 773 of 2390 rows of the yeast-human similarity matrix. HGA was applied to these two PINs along with six other algorithms.

As shown in Table III, all values of IsoRank [15] and GRAAL [7] were less than HGA.

Although MI-GRAAL [20], NETAL [22] and GEDEVO [21] showed higher EC values and more LCCS edges, the PE, ES and PS for HGA were dramatically higher than those values for these three algorithms. In particular, the PE of NETAL was 0. Although NETAL found 36.1% of edges conserved, the score from these edges (ES) was zero. One possible reason is that none of the nodes of the matched edges were homologous proteins; the other possibility is that the two nodes for each edge were not homologous with their counterparts in the aligned network. For example, there are three pairs of matched edges e_{12}^{A} and e_{12}^{B}, e_{13}^{A} and e_{13}^{B}, e_{23}^{A} and e_{23}^{B} in Fig. 1: thus, e_{12}^{A}, e_{13}^{A} and e_{23}^{A} may contribute to the calculation of EC. However, based on equation (12), only the edge e_{12}^{A} has an ES because its nodes u_1 and u_2 are similar to v_1 and v_2, respectively. Moreover, based on equation (13), both nodes u_1 and u_2 have a PS because they are similar to the nodes to which they are matched and belong to the edge e_{12}^{A}, which has an ES. The node u_3 will not receive a PS because none of its edges has an ES. If none of the pairs of the matched nodes belong to the same edge, the score of all nodes is zero because none of the edges they belong to has a score. As a result, both ES and PS are zero, leading to a zero PE. Similarly, the high ES computed by HGA in Table III indicates that for the majority of the matched interactions, the nodes in Ynet2390n were homologous with the nodes in Hnet9141n. This result occurred because HGA considered both node attribution and the topological structure to achieve the mapping, whereas NETAL considered only the topological information in the current version. Moreover, compared with PISwap [18], HGA had lower PE, ES and PS. However, its EC was higher, and the LCCS was larger.

Table III also shows the substantial difference in HGp between HGA and the other algorithms except PISwap, which means that HGA matched more homologous proteins with each other and the number of HGp was very close to that of PISwap.

This finding suggests that neither EC nor LCCS is a gold standard for the evaluation of network alignment. In other words, if one emphasizes EC alone while focusing on global alignment, some important local similarity information might be ignored. To avoid this situation, our PE score balances both nodes and topological attributes and provides a good measurement that accounts for both local and global information. Accordingly, a better way to evaluate an alignment is to compare EC, LCCS and PE together.

To measure the biological relevance of the alignment obtained by HGA, we counted the matched protein pairs that had at least 1–6 common Gene Ontology (GO) terms [34]. The GO terms used here were downloaded from UniProt [37] in July and August 2012. The results presented in Table IV indicate that the alignment using HGA was more highly enriched for GO terms, implying that HGA matched more proteins with similar functions.

Furthermore, we explored the ratio of matched proteins in clusters and MNE based on IsoBase, which is for functionally related proteins querying across species [35]. Proteins within an IsoBase cluster stand a good chance to share similar GO annotations, and lower MNE means that the proteins have greater consistency. We first found the matched proteins that are within IsoBase clusters and then computed the average mean entropy (ME) and

MNE of the protein pairs within different clusters. As shown in Table V, none of the pairs of proteins matched by NETAL and GEDEVO belonged to any cluster in IsoBase, and the ratio of GRAAL was only 0.11%, whereas HGA had a larger ratio of 24.81%, as high as PISwap and higher than IsoRank and MI-GRAAL. At the same time, the average ME and MNE of HGA were very close to those of IsoRank and PISwap, and lower than that of MI-GRAAL. Overall, the higher ratio of matched proteins in the same clusters of IsoBase along with a lower MNE implies that HGA matched more functionally related proteins.

We also investigate the statistical significance of our alignment results using HGA. By randomly changing edges but holding the original degree of each node in human PIN [40], we generated 50 random networks that were used to align with the yeast PIN to obtain 50 different alignments. We estimated that the p-value was approximately 1.85×10^{-102} (T-test).

On a personal computer running 64-bit CentOS release 6.3, 8 Intel Xeon CPUs and with 8 GB RAM, the typical CPU time for HGA or GEDEVO is more than 50 hours for a typical simulation. GEDEVO might take more time when its EC value is greater than 30%, although the program can run multithreads according to the number of CPU cores. For similar cases, IsoRank, GRAAL and MI-GRAAL took approximately 50 minutes, 90 minutes and 3 hours, respectively, whereas PISwap took approximately 10 minutes. NETAL is significantly faster than all other algorithms studied, taking only approximately 150 seconds, but it is based on topological information alone. To accelerate HGA, we have implemented a preliminary parallel algorithm using MPI for distributing the simulation. For the same case, HGA now only took approximately 6 hours on 4 computing nodes with 16 cores.

Taken together, the comparison among the different algorithms indicate that HGA generated alignments that had a lower MNE. The matched proteins using the HGA mapping had the most common GO terms. PE and HGp of HGA were close to PISwap and much higher than other algorithms, whereas HGA had larger LCCS and higher EC than PISwap. Althouth the computational efficiency of HGA needs further improvement, HGA is capable of finding more conserved interactions with strong biological relevance for PINs.

V. Conclusion

The global alignment of biomolecular networks is important for the development of network medicine and the study of species evolution. In this work, we present a hybrid algorithm (HGA) that combines HA and GA for more accurate alignment but with decreased computation time compared to HA. In particular, our experiments show that HGA aligns large networks well with an acceptable computation time, whereas HA fails. The programs, data sets and alignments are available at http://biocenter.shu.edu.cn/software/index.php/hga.

Including information from both the biomolecules themselves and the network topology, HGA places an emphasis on the influence of neighbors. We have introduced an award coefficient in the neighbor bias method, which adaptively updates similarities between nodes based on their locations and their neighbors' information. In our method, we also have introduced a new factor (PE) to evaluate the alignment quality, which infers the similarity

between aligned networks based not only on topological information but also on biomolecular attributes, in addition to using popular measurements such as EC, LCCS, common GO and HGp.

Because HGA is adaptive in considering the node neighbor's information, the common GO terms found using HGA are higher than those obtained by other existing algorithms. For example, the number of common GO terms in PINs of yeast and human that are greater than six was found to be 27.17% using HGA alignment, whereas the corresponding numbers for other alignment approaches were 9.21% to 26.77%. To the best of our knowledge, the mapping between yeast and human found by HGA has the highest rate of common GO terms.

Another advantage of HGA is on evaluating the factor HGp (the HomoloGene pairs among matched proteins). Typically, HGA mapping has significantly more HGp, although it might not achieve the highest EC value. Other algorithms that consider only topological structure usually find it difficult to increase HGp (and also common GO terms), whereas these algorithms have better computational efficiency or achieve higher EC values. In addition, the application of HGA to IsoBase provides lower ME and MNE and a higher ratio of matched proteins in clusters of IsoBase, whereas the three factors of NETAL and GEDEVO are zero.

For networks whose edges have different weights, the presented HGA requires modifications. The choice of parameters in HGA will most likely depend on various network features, and a systematic exploration on how the network structure affects those parameters needs to be conducted. HGA has not shown major improvements in time complexity, and further improvements in computational efficiency are also needed for HGA, especially for analyzing large-scale networks. It would also be interesting to perform biological experiments to test or validate those alignments obtained by HGA. Although HGA was developed for the alignment of biomolecular networks, it might also be useful for comparing other types of complex networks in which both nodes and edges are important and, in particular, when the edge information may be only partially complete or unreliable.

Acknowledgments

We thank Prof. Wu Zhang for his helpful discussions and suggestions. We thank Rashid Ibragimov from the Max-Planck-Institut für Informatik Saarbrücken 66123, Germany, Seyed Shahriar Arab from Tarbiat Modares University, Iran, and Chung-Shou Liao and Cheng-Yu Ma from the Department of Computer Science and Information Engineering, National Taiwan University, Taiwan, for their help with their programs. We thank Prof. Nataša Pržulj from the Department of Computing, Imperial College London, and Predrag Radivojac from Indiana University, Indiana, for their help with their data sets.

This research is partially supported by NIH grant P50GM76516. It is also partially supported by the Specialized Research Fund for the Doctoral Program of Higher Education [SRFDP 20113108120022] and the Major Research Plan of NSFC [No. 91330116].

Biographies

Jiang Xie obtained a Ph.D. in Computer Application Technology at Shanghai University in 2008. From Sept. 2011 – Dec. 2012, she was a Visiting Associate Researcher in the Department of Mathematics at the University of California, Irvine, USA. She is an associate professor of the School of Computer Engineering and Science at Shanghai University. She has been working in the research area of Computational Biology and Bioinformatics, high-performance computing and applications since 2004, supported by the Specialized Research Fund for the Doctoral Program of Higher Education, partly supported by NSFC and the Key Project of Science and Technology Commission of Shanghai Municipality. She is currently a member of the Chinese Computer Federation (CCF) and a member of the Technical Committee of Open System and Parallel Computing under the CCF.

Chaojuan Xiang is a postgraduate student at the School of Computer Engineering and Science, Shanghai University. Her research interests include high performance computing and its applications.

Jin Ma is a postgraduate student at the School of Computer Engineering and Science, Shanghai University. His research interests include bioinformatics and high performance computing.

Jun Tan is a postgraduate student at the School of Computer Engineering and Science, Shanghai University. His research interests include high performance computing and its applications.

Tieqiao Wen is a Professor of Neuroscience and Systems Biology at Shanghai University. He received his Ph.D. in Microbiology with the theoretical and practical combination of molecular biology and cell engineering at the Huazhong Agricultural University in 1996. After his molecular genetics and neurobiology study of neural signals in Fudan University in 1999 and the University of Georgia in 2002, he joined the School of Life Sciences at Shanghai University in 2002 as Professor, where he directed his research on the differentiation of neural stem cells. Dr. Wen focuses on problems in the molecular mechanisms of neural stem cell differentiation. Emphasis is placed on elucidating the cell signal pathways using the methods of systems biology.

Jinzhi Lei has been a member of the Zhou-Pei Yuan Center for Applied Mathematics (ZCAM), Tsinghua University, Beijing, China, since 2003, where he was a Research Assistant from 2003–2006 and an Associate Researcher since 2007. Before joining ZCAM, he was a Postdoctoral Researcher in the Department of Mathematical Science in Tsinghua University from 2001–2003. From Aug. 2004 – Jul. 2005, he was a Visiting Assistant Professor in the Department of Mathematics at the University of California, Irvine, USA. From Jul. – Dec. 2005, he was a fellow of the Sino-Canada Exchange Scholar Program in the Centre for Nonlinear Dynamics in the Department of Physiology at McGill University, Canada. He has published over 50 journal and conference papers in different areas of applied

mathematics. His previous research interests lie in the field of ordinary differential equations and dynamical systems. His current research areas are systems biology in different levels and mathematical biology, including the protein folding problem, the gene regulation networks related to apoptosis and autophagy, hematopoiesis disease, and their related mathematical problems.

Qing Nie is a Professor of Mathematics and Biomedical Engineering at the University of California, Irvine (UCI). He received his Ph.D. in Mathematics (1995) at The Ohio State University. Prior to joining UCI in 1999, he was at the University of Chicago working in the areas of computational fluid and solid mechanics. Since 2001, his research interests have included systems biology, stem cells, regulatory networks, and morphogenesis. He has been an AAAS fellow since 2013, and a Fellow of American Physical Society (APS) since 2014. Currently, he is the director of the Center for Mathematical and Computational Biology, the director of a Ph.D. training program on Mathematical and Computational Biology at UCI, and one of the principal investigators for the NIH National Center of Excellence for Systems Biology at UCI. He is also a member of several NIH and NSF panels and is on the editorial boards of several journals.

References

1. Barabasi A-L, Gulbahce N, Loscalzo J. Network medicine: a network-based approach to human disease. Nature Reviews Genetics. 2011; 12:56–68.
2. Uetz P, Giot L, Cagney G, Mansfield TA, Judson RS, Knight JR, Lockshon D, Narayan V, Srinivasan M, Pochart P, Qureshi-Emili A, Li Y, Godwin B, Conover D, Kalbfleisch T, Vijayadamodar G, Yang M, Jonston M, Fields S, Rothberg JM. A comprehensive analysis of protein-protein interactions in Saccharomyces cerevisiae. Nature. 2000; 403:623–627. [PubMed: 10688190]
3. Pellegrini M, Marcotte EM, Thompson MJ, Eisenberg D, Yeates TO. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. PNAS. 1999; 96(8):4285–4288. [PubMed: 10200254]
4. Sharan R, Ideker T. Modeling cellular machinery through biological network comparison. Nature Biotechnology. 2006; 24(4):427–433.
5. Xie, J.; Zhang, S.; Wen, T.; Ding, G.; Yu, S.; Gu, Z.; Zhang, W. Proceedings of First IEEE conference on Healthcare Informatics, Imaging and Systems Biology. San Jose, California, USA: IEEE Computer Society; Jul. 2011 A querying method with feedback mechanism for protein interaction network; p. 351-358.
6. Milo R, ltzkovitz S, Kashtan N, Levitt R, Shen-Orr S, Ayzenshtat I, Sheffer M, Alon U. Superfamilies of evolved and designed networks. Science. 2004; 303:1538–1542. [PubMed: 15001784]
7. Kuchaiev O, Milenkovic T, Memisevic V, Hayes W, Przulj N. Topological network alignment uncovers biological function and phylogeny. Journal of the Royal Society Interface. 2010; 7:1341–1354.

8. Atias N, Sharan R. Comparative analysis of protein networks: Hard problems, practical solutions. Communications of the ACM. 2012; 55(5):88–97.
9. Kelley BP, Yuan B, Lewitter F, Sharan R, Stockwell BR, Ideker T. PathBLAST: a tool for alignment of protein interaction networks. Nucleic Acids Research. 2004; 32:w83–w88. DOI: 10.1093/nar/gkh411 [PubMed: 15215356]
10. Kelley BP, Sharan R, Karp RM, Sittler T, Root DE, Stockwell BR, Ideker T. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. PNAS. 2003; 100(20): 11 394–11 399.
11. Sharan R, Suthram S, Kelley RM, Kuhn T, McCuine S, Uetz P, Sittler T, Karp RM, Ideker T. Conserved patterns of protein interaction in multiple species. PNAS. 2005; 102(6):1974–1979. http://www.pnas.org/cgi/doi/10.1073/pnas.0409522102. [PubMed: 15687504]
12. Suthram S, Sittler T, Ideker T. The Plasmodium protein network diverges from those of other eukaryotes. Nature. 2005; 438:108–112. [PubMed: 16267557]
13. Kalaev M, Bafna V, Sharan R. Fast and accurate alignment of multiple protein networks. Journal of Computational Biology. 2009; 16:989–999. [PubMed: 19624266]
14. Flannick J, Novak A, Do C, Srinivasan B, Batzoqlou S. Automatic parameter learning for multiple local network alignment. Journal of Computational Biology. 2009; 16(8):1001–1022. [PubMed: 19645599]
15. Singh, R.; Xu, J.; Berger, B. Research in Computational Molecular Biology. Springer; 2007. Pairwise global alignment of protein interaction networks by matching neighborhood topology; p. 16-31.
16. Liao C, Lu K, Baym M, Singh R, Berger B. IsoRankN: spectral methods for global alignment of multiple protein networks. Bioinformatics. 2009; 25:253–258.
17. Todor A, Dobra A, Kahveci T. Probabilistic biological network alignment. IEEE/ACM Transactions on Computational Biology and Bioinformatics. 2013; 10(1):109–121. [PubMed: 23702548]
18. Chindelevitch, L.; Liao, C-S.; Berger, B. Proceedings of Pacific Symposium on Biocomputing. Kamuela, Hawaii, USA: World Scientific Publishing; Jan. 2010 Local optimization for alignment of protein interaction networks; p. 123-132.
19. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. Journal of Molecular Biology. 1990; 215:403–410. [PubMed: 2231712]
20. Kuchaiev O, Przulj N. Integrative network alignment reveals large regions of global network similarity in yeast and human. Bioinformatics. 2011; 27:1390–1396. DOI: 10.1093/bioinformatics/btr127 [PubMed: 21414992]
21. Ibragimov R, Malek M, Guo J, Baumbach J. GEDEVO: An evolutionary graph edit distance algorithm for biological network alignment. 2013:68–79.
22. Neyshabur B, Khadem A, Hashemifar S, Arab SS. NETAL: a new graph-based method for global alignment of protein-protein interaction networks. Bioinformatics. 2013; 29(13):1654–1662. [PubMed: 23696650]
23. Alon N, Yuster R, Zwick U. Color-coding. Journal of ACM. 1995; 42:844–856.
24. Shlomi T, Segal D, Ruppin E, Sharan R. QPath: a method for querying pathways in a protein-protein interaction network. BMC Bioinformatics. 2006; 7:199.doi: 10.1186/1471-2105-7-199 [PubMed: 16606460]
25. Milenkovic T, Ng WL, Hayes W, Przulj N. Optimal network alignment with graphlet degree vectors. Cancer Informatics. 2010; 9:121–137. [PubMed: 20628593]
26. Kuhn HW. The Hungarian method for the assignment problem. Naval Research Logistics. 1955; 2:83–97.
27. Cook, SA. Proceedings of 3rd Annual ACM Symposium on Theory of Computing. New York: 1971. The complexity of theorem-proving procedures; p. 151-158.
28. Eppstein D. Subgraph isomorphism in planar graphs and related problems. Journal of Graph Algorithms and Applications. 1999; 3(3):1–27.
29. He, H.; Singh, AK. ICDE'06: Proceedings of the 22nd International Conference on Data Engineering. Washington, DC, USA: IEEE Computer Society; 2006. Closure-Tree: An index structure for graph queries; p. 38http://dx.doi.org/10.1109/ICDE.2006.37.

30. Heymans M, Singh AK. Deriving phylogenetic trees from the similarity analysis of metabolic pathways. Bioinformatics. 2003; 19:i138–i146. DOI: 10.1093/bioinformatics/btg1018 [PubMed: 12855450]
31. Koyuturk M, Kim Y, Topkara U, Subramaniam S, Szpankowski W, Grama A. Pairwise alignment of protein interaction networks. Journal of Computational Biology. 2006; 13(2):182–199. [PubMed: 16597234]
32. Zaslavskiy M, Bach F, Vert J-P. Global alignment of protein-protein interaction networks by graph matching methods. Bioinformatics. 2009; 25:i259–i267. [PubMed: 19477997]
33. Singh R, Xu J, Berger B. Global alignment of multiple protein interaction networks with application to functional orthology detection. PNAS. 2008; 105(35):12 763–12 768.
34. The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. Nature Genetics. 2000; 25:25–29. [PubMed: 10802651]
35. Park D, Singh R, Baym M, Liao C-S, Berge B. Isobase: a database of functionally related proteins across ppi networks. Nucleic Acids Research. 2011; 39:D295–D300. [PubMed: 21177658]
36. Collins SR, Kemmeren P, Zhao X, Greenblatt JF, Spencer F, Holstege FCP, Weissman JS, Krogan NJ. Toward a comprehensive atlas of the physical interactome of Saccharomyces cerevisiae. Molecular and Cellular Proteomics. 2007; 6:439–450. [PubMed: 17200106]
37. The Uniprot Consortium. Activities at the universal protein resource (uniprot). Nucleic Acids Research. 2014; 42:D191–D198. [PubMed: 24253303]
38. Radivojac P, Peng K, Clark WT, Peters BJ, Mohan A, Boyle SM, Mooney SD. An intergrated approach to inferring gene-disease associations in humans. Proteins. 2008; 72(3):1030–1037. [PubMed: 18300252]
39. IsoRank and IsoRankN. 2012; 6 http://isorank.csail.mit.edu.
40. Maslov S, Sneppen K. Specificity and stability in topology of protein networks. Science. 2002; 296:910–913. [PubMed: 11988575]

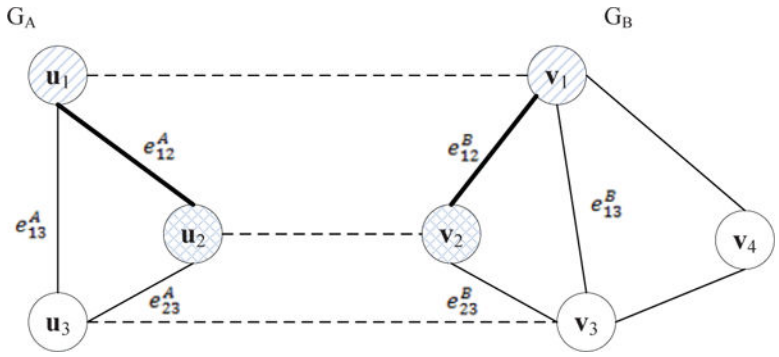

Fig. 1.

Alignment for two small networks. The dotted lines indicate the mapping between G_A and G_B. In this alignment, u_1, u_2 and u_3 are matched with v_1, v_2 and v_3, respectively. e_{12}^A, e_{13}^A and e_{23}^A are conserved edges that are matched with e_{12}^B, e_{13}^B and e_{23}^B, respectively. Therefore, EC is 100%. Supposing that u_1 is similar to v_1 and u_2 is similar to v_2, then e_{12}^A is the only edge that will receive ES, and u_1 and u_2, but not u_3, are nodes that will receive PS.

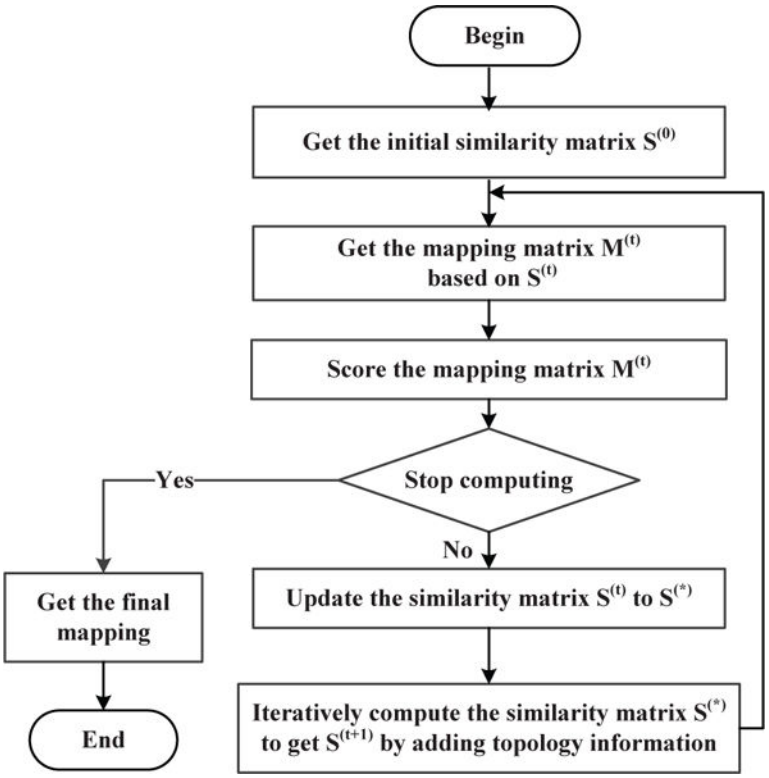

Fig. 2.
Flow chart of HGA

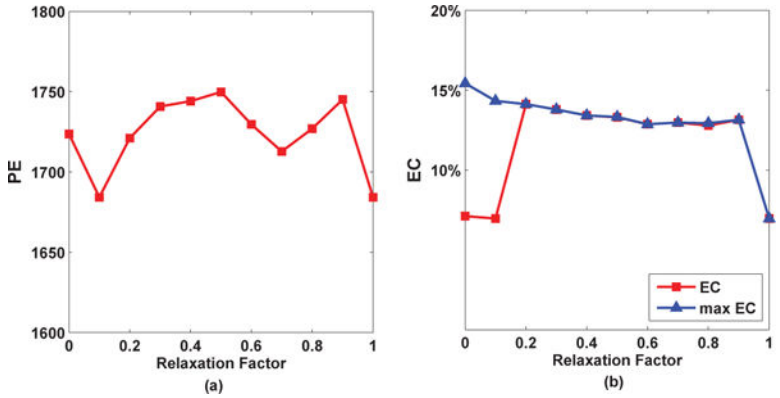

Fig. 3.

Alignment of Ynet2390n and Hnet9141n using different Relaxation Factors $\alpha = 0, 0.1, \ldots, 0.9, 1$. PE, EC in the final mapping with the maximum SS and the maximum EC as functions of the Relaxation Factors.

TABLE I

A study on choice of ω for three data sets. (1): ω is updated during each iteration; (2): ω is only updated once after all iterations; (3): Alignment is computed only based on sequences without iteration.

Networks	ω Iteration	PE	ES	PS	Matched Edges	EC (%)	LCCS (Edge, Node)	HGp	Remarks
Ynet2390n/ Hnet9141n	**1 (HGA)**	**1744.11**	**1161**	**583.11**	**2164**	**13.4**	**(584, 193)**	**1735**	**α=0.4**
	2 (INM)	–	–	–	–	–	–	–	α=0.4
	3 (No_Iteration)	1684.23	1123	561.23	1123	6.69	(255, 71)	1761	α=1
Ysubnet38n/ Hnet9141n	**1 (HGA)**	**76.41**	**55**	**21.41**	**55**	**41.98**	**(46, 12)**	**38**	**α=0.4**
	2 (INM)	62.74	38	24.74	38	29.0	(26, 11)	38	α=0.4
	3 (No_Iteration)	63.55	43	20.55	43	32.8	(34, 11)	38	α=1
CE PIN/ DM PIN	**1 (HGA)**	**57.96**	**42**	**15.96**	**789**	**18.17** *	**(36, 37)**	**1488**	**α=0.4**
	2 (INM)	42.50	29	13.50	1817	41.84	(1326, 1206)	778	α=0.4
	3 (No_Iteration)	62.08	45	17.08	45	1.03	(2, 3)	1523	α=1

* Equal to 789/(5851-1469-40) excluding multiple edges and loops.

TABLE II

Alignments between Ysubnet38n with Hnet9141n, EC and LCCS are given in Section II-B, and the HGp are homologous pairs among the matched proteins

Algorithm	PE (ES, PS)	EC (%)	LCCS (Edge)	HGp	Remark
IsoRank	0 (0,0)	20.61	20	0	default parameters
GRAAL	0 (0,0)	19.85	16	0	default parameters
MI-GRAAL	58.24 (38,20.24)	38.17	38	25	*SeqD* [20]
NETAL	0 (0,0)	42.75	56	1	a=0.0001, i=2 [22]
GEDEVO	0 (0,0)	35.11	90	0	*maxsame*= 5000 [21]
PISwap	63.55 (43, 20.55)	32.82	34	38	default parameters
HGA	**76.41 (55, 21.41)**	**41.98**	**46**	**38**	**α=0.4**

TABLE III

Comparison between HGA and other algorithms for alignment between Yeast PIN (Ynet2390n) and Human PIN (Hnet9141n). EC, LCCS, PE, ES and PS are given in Section II-B, and the HGp are homologous pairs among the matched proteins.

Algorithm	PE (ES, PS)	EC (%)	LCCS (Edge)	HGp	Remark
IsoRank	805.39 (572,233.39)	7.25	446	666	default parameters
GRAAL	0 (0,0)	4.73	415	4	default parameters
MI-GRAAL	662.68 (444,218.68)	14.78 *	2083 *	522	*SeqD*
NETAL	0 (0,0)	36.1 [22]	5370 [22]	4	a=0.0001, i=2
GEDEVO	2.73 (1,1.73)	38.14 **	4526	5	[21]
PISwap	1911.79 (1281, 630.79)	8.30	393	1738	default parameters
HGA	**1744.11 (1161, 583.11)**	**13.4**	**584**	**1735**	***α*=0.4**

* the highest EC is 23.26% and the LCCS is 3467 in [20]

** the EC is 22.45% with the parameter *maxsame* = 5000

TABLE IV

Matched protein pairs in the yeast-human alignment that have common GO terms (CGO)

CGO	≥1	≥2	≥3	≥4	≥5	≥6
IsoRank (%)	55.23	42.97	35.27	28.37	23.77	20.04
GRAAL (%)	34.23	22.38	17.49	13.31	11.00	9.21
MI-GRAAL * (%)	47.32	35.52	28.12	22.09	17.28	14.06
NETAL ** (%)	39.41	26.99	20.79	16.82	13.97	11.84
GEDEVO *** (%)	35.23	22.47	17.49	13.47	11.17	9.46
PISwap (%)	67.08	57.30	47.44	38.81	31.73	26.77
HGA (%)	**69.29**	**58.72**	**48.49**	**39.62**	**32.02**	**27.17**

* The alignment with EC=14.78

** The alignment with EC=36.1 in [22]

*** The alignment with EC=38.14 in [21]

TABLE V

Mean entropy, mean normalized entropy of the yeast-human alignment based on IsoBase

	ratio in IsoBase (%)	average ME	average MNE
IsoRank	7.62	0.1034	0.0805
GRAAL	0.11	0	0
MI-GRAAL *	4.57	0.1346	0.0945
NETAL **	0	0	0
GEDEVO ***	0	0	0
PISwap	25.37	0.096	0.0699
HGA	**24.81**	**0.1053**	**0.0757**

* The alignment with EC=14.78

** The alignment with EC=36.1 in [22]

*** The alignment with EC=38.14 in [21]