Novel Geometric Approach for Global Alignment of PPI Networks*

Yangwei Liu,[1] Hu Ding,[2] Danyang Chen,[1] Jinhui Xu[1]
[1]Department of Computer Science and Engineering, State University of New York at Buffalo
[2]Department of Computer Science and Engineering, Michigan State University

Abstract

In this paper we present a novel geometric method for the problem of global pairwise alignment of protein-protein interaction (PPI) networks. A PPI network can be viewed as a node-edge graph and its alignment often needs to solve some generalized version of the subgraph isomorphism problem which is notoriously challenging and NP-hard. All existing research has focused on designing algorithms with good practical performance. In this paper we propose a two-step algorithm for the global pairwise PPI network alignment which consists of a **Geometric Step** and an **MCMF Step**. Our algorithm first applies a graph embedding technique that preserves the topological structure of the original PPI networks and maps the problem from graph domain to geometric domain, and computes a rigid transformation for one of the embedded PPI networks so as to minimize its Earth Mover's Distance (EMD) to the other PPI network. It then solves a Min-Cost Max-Flow problem using the (scaled) inverse of sequence similarity scores as edge weight. By using the flow values from the two steps (*i.e.,* EMD and Min-Cost Max-Flow) as the matching scores, we are able to combine the two matching results to obtain the desired alignment. Unlike other popular alignment algorithms which are either greedy or incremental, our algorithm globally optimizes the problem to yield an alignment with better quality.

Introduction

A protein-protein interaction (PPI) network is a graph that describes the interaction of proteins, where a node represents a protein, and an edge means that two corresponding proteins interact with each other. The alignment of two PPI networks is thus the alignment of two undirected graphs. Since this is usually a generalized subgraph isomorphism problem which is NP-hard, researches on this problem are mostly heuristic algorithms aimed at achieving good practical efficiency. Current alignment algorithms can be broadly classified into two categories: local alignment and global alignment. Local alignment algorithms are designed to find isomorphic subgraphs of two (or more) PPI networks. Popular algorithms include Mawish (Koyutürk et al. 2006) and AlignNemo (Ciriello et al. 2012). While local alignment algorithms can find isomorphic substructures, global alignment algorithms can better capture the global picture of how conserved substructure motifs are organized.

A great deal of recent research has focused on global alignment algorithms. Some algorithms are designed to handle the alignment of multiple PPI networks, such as IsoRankN (Liao et al. 2009), NetCoffee (Hu, Kehr, and Reinert 2013), ConvexAlign (Hashemifar, Huang, and Xu 2016) and an algorithm framework in (Kalaev, Bafna, and Sharan 2008) Some alignment algorithms are designed to work with the alignment of two PPI networks, such as IsoRank (Singh, Xu, and Berger 2008), MI-GRAAL (Kuchaiev and Pržulj 2011), GHOST (Patro and Kingsford 2012), MAGNA (Saraph and Milenković 2014), Prob (Todor, Dobra, and Kahveci 2013), NETAL (Neyshabur et al. 2013), HubAlign (Hashemifar and Xu 2014) and an embedding algorithm based on graphlet frequency (Przulj 2007) . IsoRank is the first algorithm of such kind, and is also one of the most popular ones. It defines the similarity of two nodes recursively, meaning that two nodes are similar if their neighbors are similar. MI-GRAAL is an algorithm that uses both topological and biological information, and produces an alignment in a greedy way using a seed-and-extend approach. GHOST defines the distance between two nodes as the difference of spectral signatures of them, and then generates the alignment in a greedy way. NETAL first defines topological similarity between nodes in a similar way to IsoRank, then tries to optimize the number of conserved edges, and finally builds the alignment greedily.

In this paper we propose a novel global pairwise alignment algorithm, called **GeoAlign**, to align two PPI networks based on techniques including graph embedding, geometric algorithms and flow algorithms. GeoAlign is a two-step algorithm, with a *Geometric step* which gives us a topological matching score, and an *MCMF step* which gives us a biological matching score. The general ideas can be summarized as follows.

Geometric Step. Since directly aligning graphs is challenging and the mostly used information in PPI network alignment is the local topology of each node, a natural way

*The research of this work was supported in part by NSF through grants CCF-1422324, IIS-1422591, and CNS-1547167. Ding was supported by a start-up fund from Michigan State University. Part of the work was done when Ding was in Simons Institute, UC Berkeley. Ding also wants to thank Prof. Roded Sharan for his helpful discussion.

of thinking is to transform the problem from the graph domain to some other domain and hope that the target domain preserves these topological information and has better matching algorithms to solve the original alignment problem.

Since matching is a very thoroughly researched topic in computational geometry, we choose to transform the problem to the geometric domain. Given two PPI networks, we first use a graph embedding technique to embed both networks into a low dimensional Euclidean space. The graph embedding technique has the ability to conserve the original topological structures of the graph. In this way the original node-edge graph is transformed into a low dimensional point set, and local topological properties (such as connectivity between nodes, length of shortest path between nodes) are preserved in a geometric form. Then, with some preprocessing, we apply a geometric matching algorithm called Earth Mover's Distance under Rigid Transformation (EMDRT) (Ding and Xu 2016) to the two point sets. EMDRT establishes a matching between the two point sets. Based on this matching, we can then obtain a matching score for each pair of nodes. EMDRT ensures that if two nodes have similar local topology, they are more likely to have a higher matching score. This score is later used as the topological score.

MCMF step. In addition to the topological information used in the Geometric Step, we also make use of the available sequence similarity information. We build a Min-Cost Max-Flow (MCMF) model (Gabow and Tarjan 1989) using the inverse of the similarity score as edge weights. Solving the MCMF problem gives us another matching score. Since MCMF favors node pairs that have higher sequence similarity (thus smaller cost), we can consider this matching score as the sequence score (or biological score).

Combining the two types of scores, we obtain a final matching score between node pairs. A pair of nodes are matched if their combined matching score is higher than a certain threshold.

Method

Problem Definition

Definition 1. ***PPI network*** *A PPI network is an undirected graph $G = (V, E)$, where the node set V represents the proteins and the edge set E represents interactions between proteins.*

Based on Definition 1, in this paper, the notions "graph" and "PPI network" are often used interchangeably.

Definition 2. ***Pairwise Global Alignment of PPI Networks*** *The pairwise global alignment of two PPI networks $G_1 = (V_1, E_1), G_2 = (V_2, E_2)$ is a matching (one-to-one or many-to-many) $\mathcal{M} = V_1 \cup V_2 = \mathcal{M}_1 \cup \mathcal{M}_2 \cup \cdots \mathcal{M}_k$ of the node sets (V_1, V_2).*

For simplicity, in the rest of the paper, we refer to "global pairwise alignment" simply as "alignment". The goal of the algorithm is to find a matching so that certain evaluation metrics are optimized (detailed in later sections). There can be many such metrics, however most of them follow the same intuition: if a node v_1 from PPI network A has similar local topology to another node v_2 from PPI network B, they should have a high probability to be matched. Also, if v_1 and v_2 have high sequence similarity score, they also should have high probability to be matched. Based on these intuition, we introduce the proposed method in the following subsections.

Geometric Step

Graph Embedding Algorithm. As discussed in the introduction, we need to first transform the problem from graph domain to geometric domain. To achieve this, we need a graph embedding algorithm that is able to preserve the topological properties of the original graph. Intuitively, by preserving topology we mean that if the length of the shortest path between two nodes in a PPI network is short, the geometric distance between embedded points should also be short. More specifically, we want to be able to recover the connectivity of the original graph based only on the positions (coordinates) of the embedded points using some connectivity algorithm. There are multiple connectivity algorithms to choose from, in this paper we used the k-nearest-neighbor algorithm, where a point in a point set is connected to its k nearest neighbors. So when given an input graph, the desired graph embedding algorithm should be able to produce an embedding (point set) which, if fed as the input to the k-nearest-neighbor algorithm, will result in a graph that is close to the original graph. In this way, a good matching between the embedded point sets will also lead to a good matching between the original graphs.

In this paper we choose to use an embedding algorithm called the Structure Preserving Embedding (SPE) algorithm (Shaw and Jebara 2009). The algorithm satisfies the aforementioned requirements, and also have some other nice properties, such as a low dimensionality embedding result that will reduce the computation time.

Roughly speaking, for each connectivity algorithm $\mathcal{G}$, the SPE algorithm enumerates a certain number of linear constraints on the kernel matrix K of $\mathcal{G}$ (in our case the $k$nn algorithm). It is shown that as long as these constraints are satisfied, the topological properties of the original graph can be preserved. Then the SPE algorithm tries to optimize an objective function that ensures a low dimensional embedding. The pseudo code of the algorithm is as follows:

Algorithm 1 Structure Preserving Embedding

```
INPUT: Adjacency matrix $A$ of graph $G$; connectivity algorithm $\mathcal{G}$; parameter $C$
Solve SDP $\tilde{K} = \arg\max_{K \in \mathcal{K}} tr(KA) - C\xi$ s.t. $D_{ij} > (1 - A_{ij}) \max_m (A_{im} D_{im} - \xi)$
Apple SVD to $\tilde{K}$ and return the top eigenvectors as embedding coordinates.
```

In the above algorithm, $\mathcal{K} = \{K \succeq 0, tr(K) \leq 1, \sum_{ij} K_{ij} = 0, \xi \geq 0\}$ is a set of constraints to ensure that the embedding is centered at the origin. $D_{ij} = K_{ii} + K_{jj} - 2K_{ij}$ is a distance function based on the kernel matrix K, and $D_{ij} > (1 - A_{ij}) \max_m (A_{im} D_{im})$ is the linear constraints mentioned above. The parameter C is used to allow some constraints be violated to make sure that there will always be a solution to the SDP. For more detailed explanation on SPE we refer the reader to (Shaw and Jebara 2009).

Earth Mover's Distance under Rigid Transformation. After transforming the problem from graph domain to geometric domain, we need an algorithm to match the two point sets resulting from the graph embedding step. Since in the original graph domain, we would like nodes with similar local topology to be matched, the ideal geometric matching algorithm should also try to match points with similar local geometric structure (such as similar relative position to its neighbors). For this purpose we first introduce the concept of Earth Mover's Distance (EMD) (Rubner, Tomasi, and Guibas 2000).

Definition 3. ***(Earth Mover's Distance).*** *Given two point sets $A = \{p_1, p_2, \cdots, p_n\}$ and $B = \{q_1, q_2, \cdots, q_m\}$ in $\mathbb{R}^d$ with nonnegative weights α_i and β_j for each $p_i \in A$ and $q_j \in B$ respectively, the earth mover's distance between A and B is defined as:*

$$EMD(A, B) = \frac{\min_F \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij} \cdot \|p_i - q_j\|}{\min\{\sum_{i=1}^{n} \alpha_i, \sum_{j=1}^{m} \beta_j\}}$$

where $F = \{f_{ij}\}$ is a feasible max flow.

Despite the relatively long definition, the intuition behind EMD is easy to understand. Without loss of generality, we assume that A has a smaller total weight. We can then consider the weight α_i as the amount of "earth" that a point in A holds, and β_j as the maximum amount of "earth" that a point in B can hold. The "earth mover" wants to move all "earth" from A to B, and he wants to find a way of moving "earth" into B using the smallest amount of effort, where effort is defined as the sum of the product of the amount of earth moved and the distance that it is moved.

The intuition of EMD is that the "earth mover" should try to move more "earth" for a shorter distance. So points from A and B that are close to each other are favored. Also, since EMD is associated with an underlying flow f, a matching is naturally generated: simply match the points that have a positive flow between them (in our case, we assign a matching score that equals the flow value between them). An important property of EMD is that it is a concept based on global optimization. Thus instead of greedily matching local points that are close to each other, EMD will find a matching that is able to capture the global relationship between two point sets. All these properties of EMD makes it a suitable choice for the measure of closeness of two embedded PPI networks.

However, directly computing the EMD between two PPI networks embedded into Euclidean space will not give us any valuable information, since the embedding algorithm will only preserve the relative positions within the original graph and the generated point set may have arbitrary orientation and scale. Thus before computing the EMD between two embedded PPI networks (A and B), we need to put them in a position where points with similar local structure are actually close. To achieve this goal, we need to compute a rigid transformation for one of them, say A, to best match the position of B so that the EMD between A and B is minimized. This is called the Earth Mover's Distance under Rigid Transformation (EMDRT) problem. Before formally introducing the EMDRT problem, we need to perform a preprocessing step on A and B.

Algorithm 2 Preprocessing

```
INPUT: Point sets $A$ and $B$.
Compute average pairwise distance of points of $A$ and $B$, as $a$ and $b$;
For all point $v \in A$, scale its coordinates to $v \cdot \frac{b}{a}$
```

Because the SPE algorithm could output point sets with different scaling, and we would like to consider only relative positions of points within a point set, we do not want the matching to be influenced by the scaling of the point set. This preprocessing step ensures that point sets A and B have the same scaling.

We are now ready to introduce the last piece of the geometric part of our algorithm, the EMDRT problem.

Definition 4. ***(Earth Mover's Distance under Rigid Transformation).*** *Given two weighted point sets $A, B \in \mathbb{R}^d$, compute a rigid transformation (rotation, translation or both) τ on A, so that the EMD between the transformed point set $\tau(A)$ and B, $EMD(\tau(A), B)$ is minimized.*

There are multiple EMDRT algorithms with strength in different aspects. In this paper we choose to use the Iterative Closest Point (ICP) (Besl and McKay 1992) algorithm. It tries to find a transformation step by step to decrease its objective value. Although it may converge only to a local minimum, instead of a global minimum, ICP performs quite well in practice in most of the time, and a PTAS that finds a near global minimum will generally take a much longer running time rendering it unpractical for our problem.

Before introducing the ICP algorithm, we first need to set the weight for points in the point set A and B. In this paper we choose to let A and B have the same total weight (the same principal is also applied to the MCMF step described later). So each point in A has weight $|B|$, and each point in B has weight $|A|$.

We are now ready to introduce the ICP algorithm, which is briefly described as the following algorithm:

Algorithm 3 Iterative Closest Point

```
INPUT: Two point sets $A$, $B$ in $\mathbb{R}^d$.
For each point $v \in A$, find the closest point $u \in B$;
Find the rigid transformation that will best match $v$ to $u$ found in previous step;
Apply the rigid transformation found in previous step;
Repeat until convergence.
```

After the ICP algorithm finds a rigid transformation τ, we apply this transformation to A and compute the EMD between $\tau(A)$ and B. We record the computed flow value in a matrix $T = \{t_{ij}\}$, where t_{ij} is the flow value from node i

in A to node j in B. Based on the definition of EMD, a larger flow value between node i and j means a higher chance of them being close, since EMD tries to move large amount of "earth" along the shortest possible distance. We then use the matrix T as the topological score.

MCMF Step

The sequence similarity score is a biological measure of the sequence similarity of the two proteins represented by that two nodes, thus is considered part of the input data to the alignment algorithm. Note that in most cases (for example the NAPAbench dataset used in this paper) the similarity score is partial, meaning that not all pairs of proteins have their sequence similarity measured. In this paper in addition to the "Embedding-EMDRT" method, we also make use of the available sequence similarity score to better align the PPI networks. More specifically, we build another network flow model for the two PPI networks in the following way:

Definition 5. ***Min-cost Max-flow with Sequence Similarity Score*** *For two PPI networks* $A = (V_1, E_1)$ *and* $B = (V_2, E_2)$*, construct a flow network* $G = (V, E)$ *in the following fashion:* $V = V_1 \cup V_2 \cup \{s, t\}$*, where* s *is the source and* t *is the sink of the flow network. Connect* s *to all nodes in* V_1 *and* t *to all nodes in* V_2*, and set the edge cost to be* 0*. Set the capacity of edge between* s *and* $v \in V_1$ *to be* $|B|$*, set the capacity of edge between* t *and* $u \in V_2$ *to be* $|A|$*. For* $v \in V_1$ *and* $u \in V_2$*, connect them iff there is a sequence similarity score between them. Also set the edge cost* $c(vu)$ *to be the inverse of the similarity score between* v *and* u*. Set the edge capacity of all edges between* V_1 *and* V_2 *to be* inf*. Find a max flow* f *with minimum cost.*

In the problem definition above, since the edge cost is the inverse of sequence similarity score, an MCMF will favor those edges with large sequence similarity score. In practice, we use a scaled and rounded inverse of sequence similarity score as the edge weight, since it will speed up computation and increase precision. Min-cost max-flow is one of the most fundamental problem of computer science, and has received a significant amount of attentions over the years. In this paper we choose to use the Network Simplex (Cunningham 1976) algorithm which is very fast in practice and produces a very good result. For details on the algorithm, we refer the reader to (Cunningham 1976).

Once we have a solution f to the MCMF problem, similar to the EMDRT step, we record the flow value between nodes of A and B as $S = \{s_{ij}\}$ where s_{ij} is the flow value from node i in A to node j in B. We use this matrix S as the biological score.

The GeoAlign Algorithm

In this subsection we are ready to introduce the GeoAlign algorithm. An illustration of the geometric step is shown as Figure 1.

Experiments

We compare our algorithm with 4 popular alignment algorithms, including IsoRank, MI-GRAAL, GHOST and NETAL. We choose the value of λ and μ in GeoAlign using a 10-fold cross-validation on the NAPAbench CG data optimizing the specificity.

Algorithm 4 GeoAlign

```
INPUT: Two PPI networks, with (possibly partial) similarity scores; parameter $\lambda$, $\mu$.
Apply the SPE algorithm to both PPI Networks, with $k$-nearest-neighbor as the connectivity algorithm; Denote the two point sets that we get from SPE as $A$ and $B$.
Apply the preprocessing step to $A$ to update the coordinates of points in $A$;
Apply the ICP algorithm on $A$ and $B$ to get a rigid transformation $\tau$ so that $EMD(\tau(A), B)$ is minimized;
Compute $EMD(\tau(A), B)$ to get the topological score matrix $T = \{t_{ij}\}$, where an element $t_{ij}$ represents the topological score between node $i$ in $A$ and node $j$ in $B$;
Solve the min-cost max-flow problem to get the biological score matrix $S = \{s_{ij}\}$, where an element $s_{ij}$ represents the biological score between node $i$ in $A$ and node $j$ in $B$;
Compute the combined matching score $M = \lambda \cdot T + (1 - \lambda) \cdot S$, where $\lambda$ is the parameter controlling the relative importance of topological score over biological score;
Match node $i$ in $A$ to node $j$ in $B$, if $M_{ij} > \mu$.
```

Datasets

For synthetic data, we use the NAPAbench (Sahraeian and Yoon 2012), which is a widely accepted synthetic benchmark dataset with functional annotation of proteins. It is generated from an ancestral network using a sophisticated tree growth algorithm. The NAPAbench consists three types of network: crystal growth (CG), duplication-mutation-complementation (DMC) and duplication-with-random-mutation (DMR). The NAPAbench provides pairwise, 5-way and 8-way network data, here we are only going to use the pairwise data. Each network is grown from an ancestral network of 2000 nodes, and there are 1000 nodes generated for A, 2000 nodes generated for B. So the size of A is 3000 and the size of B is 4000. In addition to the network topology and functional annotation, the NAPAbench also provides a partial sequence similarity score between nodes of A and B that mimics the BLAST bit score.

For real world data, we use the PPI networks of *C.elegans*, *D.melanogaster*, *S.cerevisiae*, and *H.sapiens* from the IsoBase data set (Park et al. 2011). We use the BLAST bit score (Tatusova and Madden 1999) as the sequence similarity scores. We use the GO terms (Aladağ and Erten 2013) as annotations for proteins where the root GO terms that are on level higher than 5 are excluded.

Evaluation Metrics

We use the following four metrics to evaluate the performance of alignment algorithms.

Induced Conserved Structure (ICS).(Patro and Kingsford 2012) Let $G(V)$ denote the induced subgraph of G on vertices V, then the induced conserved structure score of an

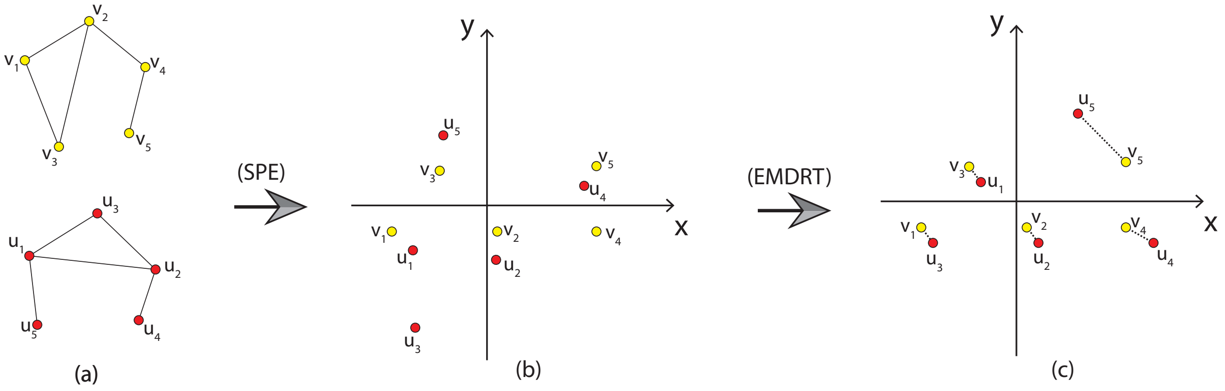

Figure 1: Illustration of the geometric step of GeoAlign. (a). The original PPI networks. (b). After applying the embedding algorithm, the PPI networks become two point sets with arbitrary orientation and position. (c). Apply EMDRT to the red point set, so that the EMD between these two point sets are minimized. The matching we get is $\{\{v_1, u_3\}, \{v_3, u_1\}, \{v_2, u_2\}, \{v_4, u_4\}, \{v_5, u_5\}\}$. Note that for the purpose of a clear demonstration, we show the matching that we get from EMDRT as a one-to-one matching. In real cases, the matching is almost always a many-to-many one.

alignment $\mathcal{M}$ from $A = (V_1, E_1)$ to $B = (V_2, E_2)$ is

$$ICS(\mathcal{M}) = \frac{|\mathcal{M}(E_1) \cap E_2|}{|E_{B(\mathcal{M}(V_1))}|}$$

This metric is based on the edge correctness (EC), where the denominator is $|E_1|$, instead of $|E_{B(\mathcal{M}(V_1))}|$. An advantage of ICS over EC is that ICS penalizes alignments that map to denser subgraphs of B. Also, $ICS = 1$ iff $\mathcal{M}$ is an isomorphism. ICS is considered a topological score, because it only takes into account the graph topology of the two PPI networks, with no functional annotation information.

Specificity (SPE).(Flannick et al. 2009)(Liao et al. 2009) Note that the alignment produced by GeoAlign is not necessarily an one-to-one mapping. It can be potentially one-to-many or many-to-many, since the indices of the highest scores in rows of the score matrix M may collide. We call a connected component $\mathcal{C}$ of the matching a *cluster*. A cluster is annotated if at least two of the proteins are annotated, and we call a cluster *correct* if all annotated proteins share the same annotation. Specificity measures the ratio of correct clusters to annotated clusters. Obviously, the higher specificity an alignment has, the more functional consistent it is.

Mean Normalized Entropy (MNE).(Flannick et al. 2009)(Liao et al. 2009) The mean normalized entropy is also a measure of the consistency of the alignment. The smaller MNE an alignment has ,the more functionally coherent it is. For a cluster $\mathcal{C}$, the normalized entropy is defined as

$$NE(\mathcal{C}) = -\frac{1}{\log d} \cdot \sum_{i=1}^{d} p_i \cdot \log p_i$$

where d is the number of annotations in $\mathcal{C}$, p_i is the fraction of proteins with annotation i. Then the mean normalized entropy is simply the average normalized entropy for all annotated clusters. By the definition of MNE, we can see that a cluster that consists of proteins with higher functional consistency will have lower normalized entropy.

Conserved Orthologous Interactions (COI). COI is defined as the ratio of the total number of interactions between all correct clusters to the total number of aligned interactions. In other words, it measures the alignment algorithm's ability to detect conserved interactions between orthologous proteins.

SPE, MNE and COI are considered biological score, since they take into account the functional annotation of each protein alongside the topological information. An alignment algorithm will be awarded higher biological score if it tries to match proteins with the same functional annotation to each other.

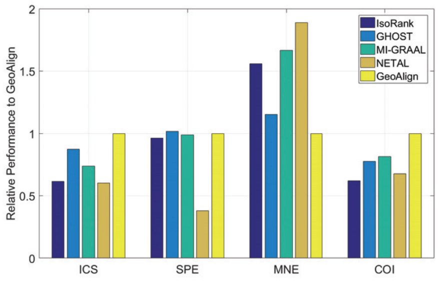

Figure 2: Average relative performance on NAPAbench dataset. Here the performance of GeoAlign is set as 1, while the performance of compared algorithms is shown as the percentage of the performance of GeoAlign.

Results on Synthetic Dataset

Results on the NAPAbench dataset are summarized in Tables 1 to 3. Note that for MNE, smaller value is better, while for other metrics, larger value is better. The best performer in each row is shown in black.

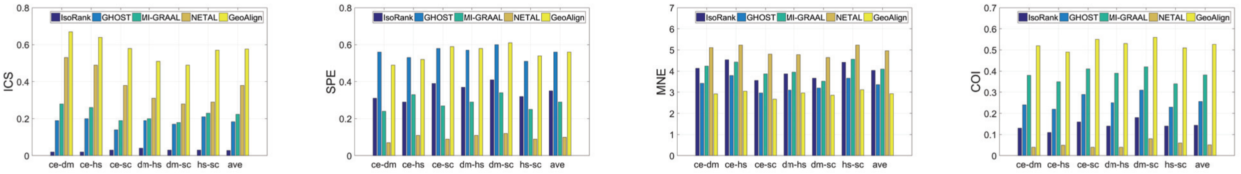

Figure 3: Comparison on real world datasets. Except MNE, all other metrics see worse performance than synthetic data. This is mainly due to noise, outlier, or incompleteness of data. GeoAlign still outperforms compared algorithms on ICS, MNE and COI.

We see that GeoAlign outperforms all other algorithms on ICS and COI, where the improvement is relatively big. For SPE and MNE the GeoAlign algorithm performs on par with the best of the compared algorithms. Since ICS is a topological metric, a high ICS score means that the alignment produced by the GeoAlign algorithm has a better topological quality. COI is a biological metric, meaning that a higher COI score is usually an evidence of the alignment being more functional consistent, that proteins that have the same functional annotation are more likely to be matched.

CG	IsoRank	GHOST	MI-GRAAL	NETAL	GeoAlign
ICS	0.58	0.81	0.76	0.52	**0.9**
SPE	0.78	**0.83**	0.8	0.21	0.82
MNE	21.54	16.34	18.23	24.32	**12.55**
COI	0.42	0.51	0.53	0.49	**0.72**

Table 1: Comparison of performance on NAPAbench CG type networks.

DMC	IsoRank	GHOST	MI-GRAAL	NETAL	GeoAlign
ICS	0.47	0.69	0.55	0.51	**0.87**
SPE	0.76	**0.81**	0.78	0.33	0.79
MNE	23.49	**16.1**	27.57	29.32	16.87
COI	0.45	0.58	0.6	0.48	**0.68**

Table 2: Comparison of performance on NAPAbench DMC type networks.

DMR	IsoRank	GHOST	MI-GRAAL	NETAL	GeoAlign
ICS	0.56	0.79	0.62	0.55	**0.85**
SPE	0.79	**0.82**	0.81	0.38	0.81
MNE	21.82	16.97	25.67	27.32	**13.45**
COI	0.44	0.55	0.59	0.46	**0.71**

Table 3: Comparison of performance on NAPAbench DMR type networks.

Figure 2 shows the average performance of the compared algorithms on the whole NAPAbench dataset. Here the performance is shown as percentage of the performance of GeoAlign, for a better comparison.

Results on Real World Dataset

We perform tests on all pair of networks (6 in total). The results, together with the average score, are shown as Figure 3. We can observe that although the general performance is worse than that on synthetic data, the proposed GeoAlign algorithm still outperforms the compared algorithms on the ICS, MNE and COI metric. The improvement on ICS is relatively significant, showing that GeoAlign is good at dealing with topological information. Also, a nearly 14% increase on COI indicates that the combination of topological and sequence similarity information and the use of global matching algorithms yield an alignment with higher functional consistency.

Conclusion and Discussion

This paper introduces a new algorithm, GeoAlign, for the global pairwise alignment of PPI networks. The most significant difference from other alignment algorithms is that GeoAlign transforms the problem from the relatively difficult graph domain to geometric domain where good global matching algorithms exist. Since the transformation preserves the topological information, the geometric matching will also serve as a good candidate for the alignment of the original PPI networks. Combining with the solution of a min-cost max-flow problem built on top of the available sequence similarity data, GeoAlign is able to produce alignment with relatively high topological quality and biological quality, on both synthetic and real world datasets.

Currently the wall clock running time of the GeoAlign algorithm is several hours. The running time is mainly bottlenecked by the embedding algorithm and flow computations. It is possible to modify the GeoAlign algorithm for different purpose. If performance and accuracy has the highest priority, we can use a PTAS for EMDRT problem, instead of the current ICP algorithm. Generally this step will be the new bottleneck of running time of the whole algorithm. If speed is desired, we can change the embedding algorithm to shorten the running time. Algorithms like spectral embedding (Luo, Wilson, and Hancock 2003) will generally run much faster than SPE, however parts of the topological information might be lost. We can also modify the MCMF step to speed it up, for example use unit edge capacity, and use sequence similarity score directly as edge weight. Then it becomes a maximum weight bipartite matching problem, where the final matching is one-to-one.

References

Aladağ, A. E., and Erten, C. 2013. Spinal: scalable protein interaction network alignment. *Bioinformatics* 29(7):917–

924.

Besl, P. J., and McKay, N. D. 1992. Method for registration of 3-d shapes. In *Robotics-DL tentative*, 586–606. International Society for Optics and Photonics.

Ciriello, G.; Mina, M.; Guzzi, P. H.; Cannataro, M.; and Guerra, C. 2012. Alignnemo: a local network alignment method to integrate homology and topology. *PloS one* 7(6):e38107.

Cunningham, W. H. 1976. A network simplex method. *Mathematical Programming* 11(1):105–116.

Ding, H., and Xu, J. 2016. Fptas for minimizing the earth movers distance under rigid transformations and related problems. *Algorithmica* 1–30.

Flannick, J.; Novak, A.; Do, C. B.; Srinivasan, B. S.; and Batzoglou, S. 2009. Automatic parameter learning for multiple local network alignment. *Journal of computational biology* 16(8):1001–1022.

Gabow, H. N., and Tarjan, R. E. 1989. Faster scaling algorithms for network problems. *SIAM Journal on Computing* 18(5):1013–1036.

Hashemifar, S., and Xu, J. 2014. Hubalign: an accurate and efficient method for global alignment of protein–protein interaction networks. *Bioinformatics* 30(17):i438–i444.

Hashemifar, S.; Huang, Q.; and Xu, J. 2016. Joint alignment of multiple protein-protein interaction networks via convex optimization. *arXiv preprint arXiv:1604.03482*.

Hu, J.; Kehr, B.; and Reinert, K. 2013. Netcoffee: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks. *Bioinformatics* btt715.

Kalaev, M.; Bafna, V.; and Sharan, R. 2008. Fast and accurate alignment of multiple protein networks. In *Annual International Conference on Research in Computational Molecular Biology*, 246–256. Springer.

Koyutürk, M.; Kim, Y.; Topkara, U.; Subramaniam, S.; Szpankowski, W.; and Grama, A. 2006. Pairwise alignment of protein interaction networks. *Journal of Computational Biology* 13(2):182–199.

Kuchaiev, O., and Pržulj, N. 2011. Integrative network alignment reveals large regions of global network similarity in yeast and human. *Bioinformatics* 27(10):1390–1396.

Liao, C.-S.; Lu, K.; Baym, M.; Singh, R.; and Berger, B. 2009. Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics* 25(12):i253–i258.

Luo, B.; Wilson, R. C.; and Hancock, E. R. 2003. Spectral embedding of graphs. *Pattern recognition* 36(10):2213–2230.

Neyshabur, B.; Khadem, A.; Hashemifar, S.; and Arab, S. S. 2013. Netal: a new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics* 29(13):1654–1662.

Park, D.; Singh, R.; Baym, M.; Liao, C.-S.; and Berger, B. 2011. Isobase: a database of functionally related proteins across ppi networks. *Nucleic acids research* 39(suppl 1):D295–D300.

Patro, R., and Kingsford, C. 2012. Global network alignment using multiscale spectral signatures. *Bioinformatics* 28(23):3105–3114.

Przulj, N. 2007. Geometric local structure in biological networks. In *Information Theory Workshop, 2007. ITW'07. IEEE*, 402–407. IEEE.

Rubner, Y.; Tomasi, C.; and Guibas, L. J. 2000. The earth mover's distance as a metric for image retrieval. *International journal of computer vision* 40(2):99–121.

Sahraeian, S. M. E., and Yoon, B.-J. 2012. A network synthesis model for generating protein interaction network families. *PloS one* 7(8):e41474.

Saraph, V., and Milenković, T. 2014. Magna: maximizing accuracy in global network alignment. *Bioinformatics* 30(20):2931–2940.

Shaw, B., and Jebara, T. 2009. Structure preserving embedding. In *Proceedings of the 26th Annual International Conference on Machine Learning*, 937–944. ACM.

Singh, R.; Xu, J.; and Berger, B. 2008. Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proceedings of the National Academy of Sciences* 105(35):12763–12768.

Tatusova, T. A., and Madden, T. L. 1999. Blast 2 sequences, a new tool for comparing protein and nucleotide sequences. *FEMS microbiology letters* 174(2):247–250.

Todor, A.; Dobra, A.; and Kahveci, T. 2013. Probabilistic biological network alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 10(1):109–121.