Protein Mover's Distance: A Geometric Framework for Solving Global Alignment of PPI Networks

Manni Liu and Hu Ding(✉)

Department of Computer Science and Engineering, Michigan State University, East Lansing, USA
{liumanni,huding}@msu.edu

Abstract. A protein-protein interaction (PPI) network is an unweighted and undirected graph representing the interactions among proteins, where each node denotes a protein and each edge connecting two nodes indicates their interaction. Given two PPI networks, finding their alignment is a fundamental problem and has many important applications in bioinformatics. However, it often needs to solve some generalized version of subgraph isomorphism problem which is challenging and NP-hard. Following our previous geometric approach [21], we propose a unified algorithmic framework for PPI networks alignment. We first define a general concept called "Protein Mover's Distance (PMD)" to evaluate the alignment of two PPI networks. PMD is similar to the well known "Earth Mover's Distance"; however, we also incorporate some other information, e.g., the functional annotation of proteins. Our algorithmic framework consists of two steps, Embedding and Matching. For the embedding step, we apply three different graph embedding techniques to preserve the topological structures of the original PPI networks. For the matching step, we compute a rigid transformation for one of the embedded PPI networks so as to minimize its PMD to the other PPI network; by using the flow values of the resulting PMD as the matching scores, we are able to obtain the desired alignment. Also, our framework can be easily extended to joint alignment of multiple PPI networks. The experimental results on two popular benchmark datasets suggest that our method outperforms existing approaches in terms of the quality of alignment.

1 Introduction

Proteins are essential parts of organisms and participate in virtually every process within cells [36]. *Protein-Protein Interaction (PPI) networks* provides effective tools for studying protein complexes and understanding their functional

The research of this work was supported in part by NSF through grant CCF-1656905 and a start-up fund from Michigan State University. Ding also wants to thank Profs. Bonnie Berger and Roded Sharan for their helpful discussions at Simons Institute, UC Berkeley.

X. Gao et al. (Eds.): COCOA 2017, Part I, LNCS 10627, pp. 56–69, 2017.
https://doi.org/10.1007/978-3-319-71150-8_5

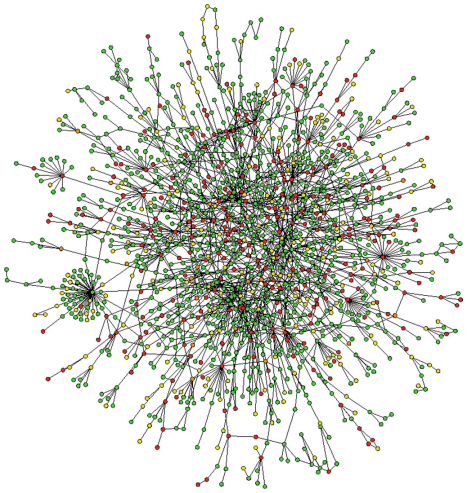


Fig. 1. An example of yeast PPI network [35].

interactions, modules, and pathways, in many cellular processes. A **PPI network** is a graph that describes the interaction of proteins, where a node represents a protein and an edge means that the two corresponding proteins interact with each other [35]. See Fig. 1.

The current research on PPI networks mainly focus on two directions: (1) knowledge discovery inside each individual network and (2) comparison and integration of different networks. The first direction includes the problems of link prediction (i.e., adding new interactions) and modules/pathways detection, while the second one often targets finding the similarity or distinction between two or more networks. Actually these two directions are closely related with each other, e.g., better knowledge discovery inside each network could lead to more accurate comparison between networks, and the integrated analysis on different networks could improve the knowledge discovery inside each individual network. In this paper, we focus on a fundamental problem in the latter direction, **PPI networks alignment**, which is often modeled as the problem of mapping two undirected graphs:

Let two undirected graphs $G_1 = (V_1, E_1)$ *and* $G_2 = (V_2, E_2)$ *denote two PPI networks. An alignment of* G_1 *and* G_2 *is to compute a mapping between* V_1 *and* V_2 *satisfying some given criteria, where the mapping could be one-to-one or many-to-many.*

Since it is usually a generalized NP-hard subgraph isomorphism problem, most of the existing algorithms on PPI networks alignment are heuristic and aimed at achieving good practical efficiency. Current research includes local and global alignment. Local alignment algorithms are designed to find isomorphic subgraphs of two or more PPI networks, where the popular ones include Mawish [15] and AlignNemo [5]. Comparing with local alignment, global alignment

can better capture the global picture of how conserved substructure motifs are organized, and consequently attracts a great deal of attentions. The well known algorithms include IsoRank [32], MI-GRAAL [17], GHOST [25], MAGNA [29], Prob [34], NETAL [23], and HubAlign [11]. For example, IsoRank defines the similarity of two nodes recursively based on the similarity of their neighbors; MI-GRAAL uses both topological and biological information, and generates the alignment by a greedy seed-and-extend approach; GHOST defines the difference of spectral signatures among the nodes and generates the alignment greedily; NETAL defines the topological similarity between the nodes in a similar way to IsoRank and tries to optimize the number of conserved edges. Moreover, some algorithms are designed to handle joint alignment of multiple PPI networks, such as IsoRankN [20], NetCoffee [13], SMETANA [28], BEAMS [2], ConvexAlign [9], and NetworkBlast-M [14].

Comparing with directly solving the problem of graph isomorphism, the aforementioned heuristic approaches can alleviate the high computational complexity to certain extent. However, they still suffer several unavoidable drawbacks. For example, their time complexities could still be relatively high (e.g., $O(n^3 \log n)$ where n is the number of vertices [10]). Moreover, the available PPI networks are often very sparse, and thus the alignment based on the local topology of each vertex is not quite reliable. One way to solve this issue is to first make use of the fact that biological networks can often be embedded into Euclidean space (due to their intrinsic nature [12,18]; recently, Cho et al. [4] propose a new algorithm for low-dimensional geometric representation of biological networks called *Diffusion Component Analysis*), and then convert the alignment problem from graph domain to geometry domain. Besides the lower computational complexity, the geometric representations of PPI networks can also remedy the issue caused by the sparse and noisy interactions of PPI networks [18].

Inspired by this observation, our previous work [21] provides a geometric embedding based algorithm "GeoAlign". Roughly speaking, GeoAlign first embeds the given two PPI networks into a Euclidean space via the method of structure preserving embedding [31], and then computes their alignment in the space.

1.1 Our Contributions

The goal of this paper is twofold. First, we follow and generalize our previous work [21] to a unified algorithmic framework for PPI networks alignment. Second, we study and compare the experimental performance of our framework with other popular methods on two benchmark datasets.

Our algorithmic framework includes two steps: **(1) embedding** and **(2) matching**. Given two PPI networks, we first use a graph embedding technique to represent them in some Euclidean space. As a consequence, each network is transformed to a point set and the local topological properties (such as the connectivity and length of shortest path between nodes) are well preserved in a geometric form. We adopt three different embedding methods, the recent popular deep learning based approach *node2vec* [8], the well studied *multi-dimensional*

scaling (MDS) [16], and *structure preserving embedding (SPE)* [31] which was used in [21] (see Sect. 2 for details). Then, we use both the geometric information and given sequence similarity scores of the proteins to establish the matching. Note that the matching should also take into account of certain transformations in Euclidean space, such as rigid transformation. To realize this idea, we propose a novel concept "**Protein Mover's Distance (PMD)**" to measure the matching cost between two PPI networks. Moreover, our framework can be naturally extended for joint alignment of multiple PPI networks.

Note: of course, the embedding method should not be limited to the aforementioned three algorithms in our general algorithmic framework, and we expect a more extensive experimental study on different embedding methods in future work.

2 Embedding Methods

In this section, we introduce three different methods for embedding PPI networks in our framework.

2.1 Node2vec

Recently, Grover and Leskovec [8] present a new algorithm called *node2vec* for feature learning. Given a graph, the key idea of node2vec is to define a novel random walk procedure to generate the neighborhood of each node (vertex) and maximize the likelihood for maintaining the interactions among the neighbors; eventually, it obtains a representation of the nodes in Euclidean space. For the sake of completeness, we briefly introduce the method below.

Let G = (V, E) be a given unweighted and undirected graph and $f : V \to \mathbb{R}^d$ be the (to be learned) mapping function from the nodes to a d-dimensional space where d is a parameter that can be specified as the input. For each node $u \in V$, node2vec defines its neighborhood $N_S(u)$ based on two classic sampling strategies, *Breadth First Sampling (BFS)* and *Depth First Sampling (DFS)*. In BFS, the neighborhood $N_S(u)$ covers the nodes which are directly connected with the source node u. Differently, DFS defines $N_S(u)$ to contain the nodes which may have indirected interactions (by depth first search) with the source node u.

Node2vec applies random walk to make a balance between BFS and DFS. For a source node u and a given positive integer l, node2vec runs the fixed l steps of random walk and the neighborhood $N_S(u)$ consists of all the passed nodes. After generating the neighborhood $N_S(u)$ for each node u, node2vec is to optimize the following objective function inspired by the Skip-gram Model [22]:

$$\max_f \sum_{u \in V} \log Prob(N_S(u)|f(u)). \tag{1}$$

With the standard assumptions of conditional independence and symmetry of feature space, the objective function (1) can be further simplified to be:

$$\max_{f} \sum_{u \in V} [-\log Z_u + \sum_{n_i \in N_S(u)} f(n_i) \times f(u)] \tag{2}$$

where $Z_u = \sum_{v \in V} \exp(f(u) \times f(v))$ (see [8] for the omitted details). Finally, the objective function (1) is optimized by stochastic gradient descent (SGD) on single hidden-layer feedforward neural networks.

2.2 Multi-dimensional Scaling

Multi-dimensional Scaling (MDS) is a widely used tool for embedding graph into Euclidean space [16]. In particular, Higham et al. [12] and Kuchaiev et al. [18] introduce the ideas based on MDS to tackle the problems of de-noising and link prediction for PPI networks.

The input of MDS is the matrix of the $n \times n$ pairwise distances (suppose the number of nodes is n in the given graph). To define the pairwise distance, [12,18] adopt the length of the shortest path between each pair of nodes in the graph (in case that the PPI network is not connected, they handle the connected components separately). Obviously, computing the whole distance matrix could be very costly if using *Dijkstra's* or other shortest path algorithms [6]. However, since PPI networks are unweighted and usually sparse, we can directly run breadth first search n times to obtain the n^2 pairwise distances, and the total running time is only $O(n^2)$ (also Higham et al. [12] set an upper bound for the distances which makes the method even more practical).

Let the obtained distance between node i and j be d_{ij} and the dimension of the desired embedding space be d. The goal of MDS is to find n points $x_i \in \mathbb{R}^d$, $i = 1, \cdots, n$, such that the distance between each pair (x_i, x_j) is roughly equal to d_{ij}. First, we generate a positive semi-definite matrix A where each

$$a_{ij} = -\frac{1}{2}(d_{ij}^2 - \frac{1}{n}\sum_{k=1}^{n} d_{ij}^2 - \frac{1}{n}\sum_{k=1}^{n} d_{kj}^2 + \frac{1}{n^2}\sum_{k=1}^{n}\sum_{l=1}^{n} d_{kl}^2). \tag{3}$$

Consequently, we know that

$$X^T X \approx A. \tag{4}$$

Further, we decompose the matrix A to be $U^T \Sigma U$ where the rows of U are the eigenvectors of A and the diagonal entries of Σ are the eigenvalues ordered decreasingly. Finally, MDS lets $\hat{X} = \sqrt{\Sigma_d} U$ be the embedding solution where Σ_d contains only the top d eigenvalues.

2.3 Structure Preserving Embedding

Given the adjacency matrix of a graph, traditional graph embedding algorithms often need to employ a spectral decomposition of the Laplacian and take the

top eigenvectors as the embedding coordinates. However, a drawback of such embedding algorithms is that they cannot efficiently preserve the topology of the input graph. To remedy this issue, Shaw and Jebara propose a novel embedding algorithm called structure preserving embedding (SPE) [31]. Different from the previous spectral embedding methods, SPE learns a new positive semi-definite kernel matrix K whose spectral decomposition can preserve the topology exactly; moreover, the problem can be modeled as a semi-definite programming with a set of linear constraints. For more detailed explanation on SPE, we refer the readers to [31].

Due to the advantage on preserving topological structure, our previous work [21] adopts SPE to embed the given PPI networks into Euclidean space for computing their alignment.

3 Protein Mover's Distance

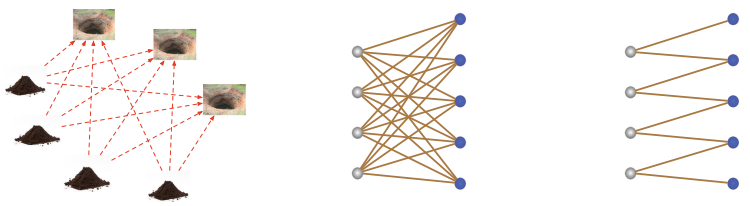
Since the given PPI networks become point sets in Euclidean space after embedding, the next question is how to measure their similarity. Actually, our idea comes from the well known concept **earth mover's distance (EMD)** in computational geometry which has been extensively studied in many areas [19,26,27].

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, define the* ***ground distance*** $D(p_i, q_j) \geq 0$ *for each pair of* p_i *and* q_j *(normally, the ground distance is simply their (squared) Euclidean distance). The EMD between* A *and* B *is:*

$$EMD(A, B) = \frac{\min_F \sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij} \cdot D(p_i, q_j)}{\min\{\sum_{i=1}^{n} \alpha_i, \sum_{j=1}^{m} \beta_j\}}, \tag{5}$$

where $F = \{f_{ij}\}$ *is a feasible flow from* A *to* B*, such that* $\forall i, j$*,* $f_{ij} \geq 0$*,* $\sum_{j=1}^{m} f_{ij} \leq \alpha_i$*,* $\sum_{i=1}^{n} f_{ij} \leq \beta_j$*, and* $\sum_{i=1}^{n} \sum_{j=1}^{m} f_{ij} = \min\{\sum_{i=1}^{n} \alpha_i, \sum_{j=1}^{m} \beta_j\}$.

Intuitively, EMD can be viewed as the minimum transportation cost between A and B, where the weights of A and B are the "supplies" and "capacities" respectively, and the cost of an edge between any pair of points from A to B is their ground distance (see Fig. 2(a)). Also, since EMD is associated with an



(a) (b) (c)

Fig. 2. (a) An illustration for earth mover's distance; (b) min-cost max flow for computing EMD; (c) the simplified min-cost max flow via FastEMD.

underlying flow F, a many-to-many matching is naturally generated via simply matching the points that have a positive flow between them. More importantly, EMD is based on a global optimization. That is, instead of greedily matching local points that are close to each other, EMD finds a matching that is able to capture the global relationship between them.

For the sake of simplicity, we also use A and B to denote the point sets, i.e., the two embedded PPI networks, respectively; each point p_i (q_j) indicates one protein. For normalization, we let each $\alpha_i = m$ and $\beta_j = n$, and thus both the total weights $\sum_{i=1}^{n} \alpha_i$ and $\sum_{j=1}^{m} \beta_j$ are equal to nm. To measure their similarity, a significant difference to EMD is that we have to consider both local topology and biological information. We introduce the following definition.

Definition 1 (Protein Mover's Distance (PMD)). *Given a parameter* $\lambda \in [0, 1]$,

$$PMD(A, B) = \lambda EMD_t(A, B) + (1 - \lambda) EMD_b(A, B), \tag{6}$$

where $EMD_t(A, B)$ *is simply the EMD between* A *and* B *with the ground distance* D_t *being the squared Euclidean distance, while* $EMD_b(A, B)$ *is the EMD between* A *and* B *with the ground distance* D_b *being some decreasing function on the given sequence similarity scores of the proteins.*

Due to the embedding procedure, we know that $EMD_t(A, B)$ reveals the similarity of local topology between A and B. Meanwhile, $EMD_b(A, B)$ shows the similarity based on biological information, where the ground distance D_b could have different forms depending on the setting in practice. In our experiment, we simply use the inverse of the similarity score as the ground distance; if the similarity score of a pair of proteins does not exist, their ground distance is $+\infty$.

We can see that the parameter λ allocates the importances of local topology and biological information in PMD. Namely, the higher (lower) λ, the more important the local topology (biological information).

4 Our Algorithms

We first introduce our algorithm for pairwise alignment of two PPI networks in Sect. 4.1, and then show how to extend the algorithm to handle multiple PPI networks in Sect. 4.2.

4.1 Two PPI Networks

After embedding, the main idea of our alignment algorithm is to compute the PMD between the two PPI networks and generate the matching between the proteins based on the flows of the PMD. For this purpose, we need to consider the following two technical issues.

(1) Registration. Note that the embedding only preserves the pairwise distances of the nodes, thus each network actually becomes a rigid structure in the space. Consequently, we need to consider the registration between A and B under rigid transformation. Before computing the PMD, we fix A and apply the widely used *Iterative Closest Point (ICP)* [3] algorithm to find an appropriate position for B. ICP algorithm is an alternating minimization procedure that each iteration fixes either the matching or the current transformation and modifies the other to minimize the difference. ICP algorithm is guaranteed to converge and performs quite well in practice.

(2) The computation of EMD. From Definition 1, we know that both EMD_t and EMD_b need to compute the EMD between A and B but with different ground distances. Actually, optimizing the objective function of EMD is a typical instance of min-cost max flow problem which can be solved by linear programming (Fig. 2(b)). However, the numbers of points (nodes) in the PPI networks A and B are often thousands which make the computation complexity of linear programming extremely high. To resolve this issue, we use the approximate algorithm FastEMD [26] instead. Roughly speaking, FastEMD deletes the flows which have large ground distances, where the intuition is that the flows with large ground distances are more likely to be small or even zero. In practice, FastEMD makes the connecting graph of EMD much more sparse (Fig. 2(c)) and thus reduces the running time significantly.

Overall, our algorithm is shown in Algorithm 1.

Algorithm 1. Pairwise alignment

Input: two PPI networks $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, three parameters $d \in \mathbb{Z}^+$, $0 \leq \lambda \leq 1$, and $\mu > 0$.
Output: An alignment between G_1 and G_2.

1. Embed G_1 and G_2 into d-dimensional Euclidean space as A and B (by node2vec, MDS, or SPE).
2. Fix A, and run ICP to registrate B to A (with a little abuse of notations, we still use B to denote the transformed B).
3. Apply FastEMD to compute $PMD(A, B) = \lambda EMD_t(A, B) + (1 - \lambda)EMD_b(A, B)$.
4. Match protein i in A to protein j in B, if the flow between them in the PMD is larger than μ.

4.2 Multiple PPI Networks

Our method in Sect. 4.1 can be easily extended to the case with multiple networks. Given N PPI networks $G_i = (V_i, E_i), i = 1, \ldots, N$, we aim to find the alignment among all of them jointly. First, we use Algorithm 1 (step 1–3) to compute the PMD between each pair of networks, and build a N-partite graph

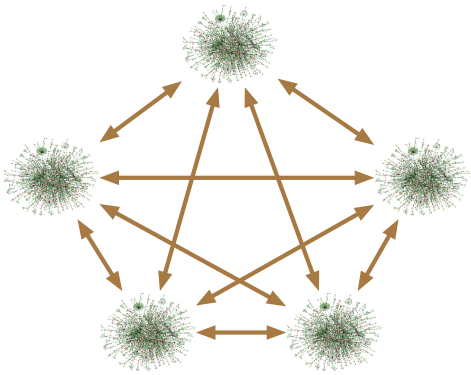


Fig. 3. Five PPI networks: we compute the PMD between each pair of networks, and build a 5-partite graph where each network is denoted as a column of vertices and the weight of each edge connecting two vertices from different columns is the corresponding value of PMD flow.

(see Fig. 3 as an example); then we apply the recent proposed convex optimization model by Hashemifar et al. [9] on the N-partite graph to find the joint alignment.

Let X_{ij} be the binary variable matrix indicating the alignment between V_i and V_j, that is, $X_{i,j}(u,v) = 1$ if $u \in V_i$ and $v \in V_j$ are aligned with each other; otherwise $X_{i,j}(u,v) = 0$. By using our obtained PMD between each pair of networks, we modify the objective function from [9] to be

$$F = \sum_{1 \le i < j \le N} \sum_{u \in V_i, v \in V_j} f_{PMD}(u,v) X_{ij}(u,v) \tag{7}$$

where $f_{PMD}(u,v)$ indicates the PMD flow from u to v. To make the optimization convex, according to [9] each binary variable matrix X_{ij} is relaxed to satisfy the following constraints: (i) X_{ii} is an identity matrix; (ii) X_{ij} is positive semi-definite. Finally, we use the alternating direction of multiplier method (ADMM) [9] to find the solution.

5 Experiments

For pairwise alignment, we compare our algorithm with IsoRank [32], MIGRAAL [17], GHOST [25], and NETAL [23]; for joint alignment of multiple networks, we compare our algorithm with IsoRankN [20], NetCoffee [13], SMETANA [28], and BEAMS [2]. In our algorithms, we try the three embedding methods node2vec, MDS, and SPE, where the algorithms are denoted as **Geo-node2vec**, **Geo-mds**, and **Geo-spe** respectively. All of the experimental results are obtained on a Windows workstation with 2.4 GHz Intel Xeon E5-2630 v3 CPU and 32 GB DDR4 2133 MHz Memory.

5.1 Datasets

First, We use the popular benchmark dataset NAPAbench [30] to test the algorithms for pairwise alignment. NAPAbench has three children datasets which are generated through crystal growth (CG), duplication-mutation-complementation (DMC), and duplication-with-random-mutation (DMR); each dataset is composed of 10 pairs of PPI networks, where each pair includes a 3000-node and a 4000-node PPI network. NAPAbench also provides the sequence similarity scores among the proteins.

To further test the algorithms for joint alignment, we use another benchmark dataset Isobase [24] which contains multiple PPI networks. Isobase is a database of functionally related orthologs developed from five major eukaryotic PPI networks; it contains five species, including H.sapiens (human), S.cerevisiae (yeast), Drosophila melanogaster (fly), Caenorhabditis elegans (worm), and Mus musculus (mouse). We use BLAST bit scores [33] as the given sequence similarity scores for Isobase. See Table 1.

Table 1. a_1: number of the proteins having interaction with other proteins; a_2: number of the proteins having BLAST bit scores with other proteins; a_3: number of interactions in the network.

	a_1	a_2	a_3
Homo sapiens (human)	10403	20313	105232
Saccharomyces cerevisiae (yeast)	5524	3764	164718
Drosophila melanogaster (fly)	7396	10336	49467
Caenorhabditis elegans (worm)	2995	10945	8639
Mus musculus (mouse)	623	21856	776

To evaluate the alignment results, we compare the obtained matchings with the annotations gene ontology (GO) terms [1]. GO terms describe the roles of proteins in terms of their associated biological process, molecular function, and cellular component (CC). We exclude CC because it only annotates a small percentage of the proteins, and moreover, the proteins with matched CC are not usually considered to be functionally similar.

5.2 Evaluation Metrics

We use the following evaluation metrics which are widely used in the previous articles to measure the alignment qualities.

1. Induced Conserved Structure (ICS). Let the two PPI networks be $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, and the resulting matching be $\mathcal{M}$. We denote the subgraph induced by $\mathcal{M}$ in G_2 as $G_2(\mathcal{M}(V_1))$ and the corresponding edge sets as $E_2(\mathcal{M}(V_1))$. Also, the set of the edges conserved in the alignment is denoted as

$\mathcal{M}(E_1, E_2)$. Then the induced conserved structure score $ICS = \frac{|\mathcal{M}(E_1,E_2)|}{|E_2(\mathcal{M}(V_1))|}$ [25]. ICS is a topological measurement, because it only takes into account the graph topology.

2. Specificity. We call each connected component 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 the annotated proteins share the same annotation. Specificity [7] measures the ratio of correct clusters to annotated clusters. Obviously, the higher Specificity an alignment has, the more functional consistent it is.

3. Mean Normalized Entropy (MNE). The mean normalized entropy [20] 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 $\mathcal{C}$ induced by the matching, the normalized entropy (NE) is defined as $NE(\mathcal{C}) = -\frac{1}{\log t} \cdot \sum_{i=1}^{t} p_i \cdot \log p_i$, where t is the number of annotations in $\mathcal{C}$ and p_i is the fraction of proteins with annotation i. Then the mean normalized entropy (MNE) is simply the average normalized entropy for all annotated clusters. We can see that a cluster that consists of proteins with higher functional consistency will have lower normalized entropy.

4. Conserved Orthologous Interactions (COI). COI is recently introduced by Hashemifar et al. [9] which only considers the total number of interactions between all pairwise correct clusters. Here we modify it to be the ratio of the total number of interactions between all pairwise correct clusters to the total number of aligned interactions. It measures the alignment algorithm's ability of detecting conserved interactions between orthologous proteins.

The latter three metrics, Specificity, MNE, and COI, are all biological measurements, since they take into account the functional annotation of each protein.

5.3 Results

In our experiments, we determine the values of λ and μ (see Algorithm 1) through optimizing Specificity score over a 10-fold cross-validation on the NAPAbench CG dataset. For simplicity, we always set the dimensionality $d = 3$ in all the embedding methods.

The average results (over 10 pairs of networks in each dataset) on pairwise alignment are shown in Table 2, where the best results are labeled in black (**for ICS, Specificity, and COI, the higher the better; for MNE, the lower the better**). Because ICS and COI are only for pairwise alignment, we use Specificity and MNE for joint alignment and the results are shown in Table 3.

We can see that Geo-spe always achieves the best for ICS, where we believe that it is due to the advantage of SPE on preserving topological structure (note that ICS is a topological measurement); for the other three evaluation metrics, Geo-node2vec often achieves the best and significantly outperforms the second best. For joint alignment, Geo-node2vec achieves the second best for Specificity which is slightly lower than the best one by NetCoffee.

Table 2. Pairwise alignment for three NAPAbench datasets CG, DMC, and DMR.

CG	IsoRank	GHOST	MI-GRAAL	NETAL	Geo-spe	Geo-node2vec	Geo-mds
ICS	0.58	0.81	0.76	0.52	**0.90**	0.72	0.66
Specificity	0.78	0.83	0.80	0.21	0.82	**0.85**	0.80
MNE	0.21	0.17	0.20	0.79	0.17	**0.15**	0.19
COI	0.42	0.51	0.53	0.49	0.72	**0.95**	0.94
DMC	IsoRank	GHOST	MI-GRAAL	NETAL	Geo-spe	Geo-node2vec	Geo-mds
ICS	0.47	0.69	0.55	0.51	**0.87**	0.56	0.50
Specificity	0.76	0.81	0.78	0.33	0.79	**0.86**	0.80
MNE	0.23	0.19	0.22	0.67	0.17	**0.14**	0.19
COI	0.45	0.58	0.60	0.48	0.68	**0.92**	0.90
DMR	IsoRank	GHOST	MI-GRAAL	NETAL	Geo-spe	Geo-node2vec	Geo-mds
ICS	0.56	0.79	0.62	0.55	**0.85**	0.62	0.57
Specificity	0.79	0.82	0.81	0.38	0.81	**0.86**	0.81
MNE	0.20	0.18	0.19	0.62	0.16	**0.14**	0.19
COI	0.44	0.55	0.59	0.46	0.71	**0.94**	0.93

Table 3. Joint alignment of the five PPI networks from Isobase

	IsoRankN	SMETANA	NetCoffee	BEAMS	Geo-spe	Geo-node2vec	Geo-mds
Specificity	0.74	0.54	**0.77**	0.73	0.73	0.75	0.71
MNE	0.83	0.99	0.95	0.81	0.81	**0.79**	0.82

6 Conclusion

In this paper, we generalize our previous work [21] and propose a unified algorithmic framework for PPI networks alignment. Different from previous methods, our framework is a geometric approach which consists of embedding and matching steps. The embedding step transforms the input PPI networks from graph domain to Euclidean space, and the matching step yields the final solution for the alignment. To efficiently solve the matching step, we define the general objective function "protein mover's distance". Moreover, our framework can be naturally extended to joint alignment of multiple PPI networks. The experimental results suggest that our method outperforms previous methods in terms of accuracy to certain extent.

To enrich the experimental study of our framework, it is deserved to explore more embedding methods instead of the three that are studied in this paper. Also, we hope that our framework can be applied to a broader range of network problems (e.g., social network) in future.

References

1. Aladağ, A.E., Erten, C.: Spinal: scalable protein interaction network alignment. Bioinformatics **29**(7), 917–924 (2013)
2. Alkan, F., Erten, C.: Beams: backbone extraction and merge strategy for the global many-to-many alignment of multiple PPI networks. Bioinformatics **30**(4), 531–539 (2013)
3. Besl, P.J., McKay, N.D.: Method for registration of 3-D shapes. In: Robotics-DL Tentative, pp. 586–606. International Society for Optics and Photonics (1992)
4. Cho, H., Berger, B., Peng, J.: Diffusion component analysis: unraveling functional topology in biological networks. In: Przytycka, T.M. (ed.) RECOMB 2015. LNCS, vol. 9029, pp. 62–64. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-16706-0_9
5. Ciriello, G., Mina, M., Guzzi, P.H., Cannataro, M., Guerra, C.: Alignnemo: a local network alignment method to integrate homology and topology. PLoS ONE **7**(6), e38107 (2012)
6. Cormen, T.H., Stein, C., Rivest, R.L., Leiserson, C.E.: Introduction to Algorithms, 2nd edn. McGraw-Hill Higher Education, New York (2001)
7. Flannick, J., Novak, A., Do, C.B., Srinivasan, B.S., Batzoglou, S.: Automatic parameter learning for multiple network alignment. In: Vingron, M., Wong, L. (eds.) RECOMB 2008. LNCS, vol. 4955, pp. 214–231. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78839-3_19
8. Grover, A., Leskovec, J.: node2vec: scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 855–864. ACM (2016)
9. Hashemifar, S., Huang, Q., Xu, J.: Joint alignment of multiple protein-protein interaction networks via convex optimization. J. Comput. Biol. **23**(11), 903–911 (2016)
10. Hashemifar, S., Ma, J., Naveed, H., Canzar, S., Xu, J.: Modulealign: module-based global alignment of protein-protein interaction networks. Bioinformatics **32**(17), 658–664 (2016)
11. Hashemifar, S., Xu, J.: HubAlign: an accurate and efficient method for global alignment of protein-protein interaction networks. Bioinformatics **30**(17), i438–i444 (2014)
12. Higham, D.J., Rasajski, M., Przulj, N.: Fitting a geometric graph to a protein-protein interaction network. Bioinformatics **24**(8), 1093–1099 (2008)
13. Hu, J., Kehr, B., Reinert, K.: NetCoffee: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks. Bioinformatics **30**(4), 540–548 (2013)
14. Kalaev, M., Bafna, V., Sharan, R.: Fast and accurate alignment of multiple protein networks. In: Vingron, M., Wong, L. (eds.) RECOMB 2008. LNCS, vol. 4955, pp. 246–256. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78839-3_21
15. Koyutürk, M., Kim, Y., Topkara, U., Subramaniam, S., Szpankowski, W., Grama, A.: Pairwise alignment of protein interaction networks. J. Comput. Biol. **13**(2), 182–199 (2006)
16. Kruskal, J.B., Wish, M.: Multidimensional Scaling, vol. 11. Sage, Newbury Park (1978)
17. Kuchaiev, O., Pržulj, N.: Integrative network alignment reveals large regions of global network similarity in yeast and human. Bioinformatics **27**(10), 1390–1396 (2011)

18. Kuchaiev, O., Rasajski, M., Higham, D.J., Przulj, N.: Geometric de-noising of protein-protein interaction networks. PLoS Comput. Biol. **5**(8), e1000454 (2009)
19. Kusner, M., Sun, Y., Kolkin, N., Weinberger, K.: From word embeddings to document distances. In: International Conference on Machine Learning, pp. 957–966 (2015)
20. Liao, C.-S., Lu, K., Baym, M., Singh, R., Berger, B.: IsoRankN: spectral methods for global alignment of multiple protein networks. Bioinformatics **25**(12), i253–i258 (2009)
21. Liu, Y., Ding, H., Chen, D., Xu, J.: Novel geometric approach for global alignment of PPI networks. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 4–9 February 2017, San Francisco, pp. 31–37 (2017)
22. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
23. Neyshabur, B., Khadem, A., Hashemifar, S., Arab, S.S.: NETAL: a new graph-based method for global alignment of protein-protein interaction networks. Bioinformatics **29**(13), 1654–1662 (2013)
24. Park, D., Singh, R., Baym, M., Liao, C.-S., Berger, B.: IsoBase: a database of functionally related proteins across PPI networks. Nucl. Acids Res. **39**(suppl 1), D295–D300 (2011)
25. Patro, R., Kingsford, C.: Global network alignment using multiscale spectral signatures. Bioinformatics **28**(23), 3105–3114 (2012)
26. Pele, O., Werman, M.: Fast and robust earth mover's distances. In: 2009 IEEE 12th International Conference on Computer Vision, pp. 460–467 (2009)
27. Rubner, Y., Tomasi, C., Guibas, L.J.: The earth mover's distance as a metric for image retrieval. Int. J. Comput. Vis. **40**(2), 99–121 (2000)
28. Sahraeian, S.M.E., Yoon, B.-J.: SMETANA: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. PLoS ONE **8**(7), e67995 (2013)
29. Saraph, V., Milenković, T.: MAGNA: maximizing accuracy in global network alignment. Bioinformatics **30**(20), 2931–2940 (2014)
30. Sayed Mohammad, E.S., Yoon, B.-J.: A network synthesis model for generating protein interaction network families. PLoS ONE **7**, e41474 (2012)
31. Shaw, B., Jebara, T.: Structure preserving embedding. In: Proceedings of the 26th Annual International Conference on Machine Learning, pp. 937–944. ACM (2009)
32. Singh, R., Xu, J., Berger, B.: Global alignment of multiple protein interaction networks with application to functional orthology detection. Proc. Natl. Acad. Sci. **105**(35), 12763–12768 (2008)
33. Tatusova, T.A., Madden, T.L.: BLAST 2 sequences, a new tool for comparing protein and nucleotide sequences. FEMS Microbiol. Lett. **174**(2), 247–250 (1999)
34. Todor, A., Dobra, A., Kahveci, T.: Probabilistic biological network alignment. IEEE/ACM Trans. Comput. Biol. Bioinform. (TCBB) **10**(1), 109–121 (2013)
35. Online Computational Biology Textbook. http://compbio.pbworks.com/w/page/16252899/Mass
36. Protein. http://www.hemostasis.com/protein/