GRASP: Graph Alignment through Spectral Signatures

Anonymous Authors

Abstract

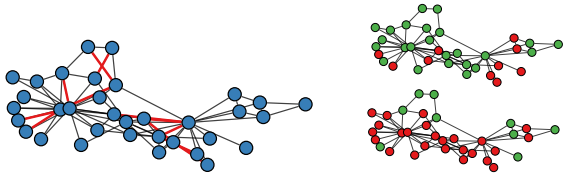
What is the best way to match nodes in two graphs? This *graph alignment* problem generalizes graph isomorphism and arises in applications from social network analysis to bioinformatics. Existing solutions usually assume that auxiliary information on known matches or node or edge attributes are available, or utilize arbitrary graph features. Such methods fare poorly in the pure form of the problem, in which only graph structures are given.

We present GRASP, a method that treats this problem as a special case of the problem of finding a mapping between functions on graphs, extending the shape-analysis concept of functional maps from the continuous to the discrete case. GRASP uses functions that capture a graph's structural characteristics, derived from eigenvectors of the graph's Laplacian matrix; it establishes a generic correspondence between eigenvector-based signature functions first, and then exploits it to derive a correspondence among nodes. Our experimental study, featuring noise levels higher than anything used in previous studies, shows that GRASP outperforms state-of-the-art methods for graph alignment across noise levels and graph types.

1 Introduction

Graphs model relationships between entities in several domains, e.g., social networks, protein interaction networks, email communication or chemical molecules. The structure of such graphs captures rich information on how people are connected, how molecules function, or how proteins interact.

At the same time, the expressive nature of graphs also implies complexity, which renders some fundamental problems hard. For instance, the *graph isomorphism* problem, which is to determine whether two graphs share the same structure is neither known to be polynomially solvable nor **NP**-complete, and has been used to define the **GI** complexity class [Kobler *et al.*, 2012]. Problems that generalize graph isomorphism occur frequently in the field of graph analytics. One of those is the **NP**-complete *subgraph isomorphism* problem; another is *graph alignment*, which aims to find the best (exact or inexact) matching among the nodes of a pair of graphs; a solution to this problem is sine qua non in tasks such as identifying users in different social networks [Kazemi *et al.*, 2015], matching objects in images by establishing feature correspondences [Schellewald and Schnörr, 2005], and comprehending protein response in the body [Klau, 2009].



(a) Karate club; Red edges removed.

(b) Alignment by GRASP (top) and REGAL (bottom).

Figure 1: With a few removed edges, REGAL [Heimann *et al.*, 2018], alignment method based on *local* features, fails to correctly align the distorted Karate club graph to the original; GRASP identifies most of nodes (correctly aligned nodes in green).

In case additional background information is available, such as node and edge attributes in the two graphs to be aligned, or existing valid *seed* matches, then the problem is solvable via supervised methods [Liu *et al.*, 2016; Chu *et al.*, 2019]. However, in case only graph structures are given, then the problem of aligning two graphs by matching structures, is at least as hard as graph isomorphism even in its approximate version [Abdulrahim, 1998].

Existing approaches to the graph alignment problem are oriented toward using a few *heuristic graph features*, such as landmarks, in order to detect a good alignment [Heimann *et al.*, 2018], *exploiting additional information* such as node attributes [Zhang and Tong, 2016] or bipartite networks [Koutra *et al.*, 2013], or optimizing objectives based only on *local connections* among nodes [Feizi *et al.*, 2019; Nassar *et al.*, 2018; Liao *et al.*, 2009]. On the other hand, the spectra of *Laplacian matrices* have been successfully employed to devise a similarity measure among graphs [Tsitsulin *et al.*, 2018]. Laplacian spectra capture important *multiscale* properties, such as local-scale ego-nets and global-scale communities. Previous approaches rooted in spectral characteristics decompose large matrices expressing all alignments among edges in two graphs [Feizi *et al.*, 2019; Nassar *et al.*, 2018; Liao *et al.*, 2009] and formulate the solution as finding the leading eigenvector of such matrices. These approaches dis-

regard most eigenvectors and consider only local edge variations. To our knowledge, the spectral properties of *Laplacian matrices* have *not yet* been utilized to any significant extent for an end-to-end graph alignment method.

We propose GRASP, short for **GR**aph **A**lignment through **SP**ectral Signatures, a principled approach towards detecting a good alignment among graphs, grounded on their spectral characteristics, i.e., eigenvalues and eigenvectors of their Laplacian matrices [Chung and Graham, 1997]. We transfer the methodology of matching among shapes based on *corresponding functions* [Ovsjanikov *et al.*, 2012] to the domain of graphs: we first extract a mapping of node-evaluated functions grounded on the graph's heat kernel, and then apply this mapping to the matching on nodes. Figure 1 shows an example alignment of the Karate club with a deteriorated version obtained by removing some edges; GRASP correctly aligns most of the nodes, while REGAL [Heimann *et al.*, 2018] based on local descriptors fails to do so.

Our contributions are as follows.

- We devise GRASP, a graph alignment algorithm based on spectral characteristics to capture multiscale interactions.
- We demonstrate the superior performance of GRASP over state-of-the-art graph alignment methods.
- We show that our method achieves better accuracy with similar run time.

2 Related Work

We discuss related work in two main categories: **restricted alignment**, which requires ground-truth mapping or other additional information, and **unrestricted alignment**, which requires neither supervision nor additional information. Table 1 gathers together previous works' characteristics.

2.1 Restricted Alignment

Supervised methods exploit pre-aligned pairs of seed nodes to construct a first alignment. *Percolation graph matching* (PGM) [Kazemi *et al.*, 2015; Yartseva and Grossglauser, 2013] propagates ground-truth alignments across the network using percolation theory.*Representation learning* approaches, such as IONE [Liu *et al.*, 2016], PALE [Man *et al.*, 2016], and DeepLink [Zhou *et al.*, 2018], learn a low-dimensional embedding of the graph nodes and map the node embeddings of one graph to another. A similar method aligns multiple networks at once [Chu *et al.*, 2019]. *Active network alignment* [Malmi *et al.*, 2017] applies active learning to elicit domain expertise for node alignments. Supervised methods may achieve good performance, but rely on the ample availability of prior knowledge in the form of seeds

Assisted methods utilize auxiliary information or structural constraints. BigAlign [Koutra *et al.*, 2013] focuses on the special cas of bipartite graphs; however, most graphs are not bipartite. FINAL [Zhang and Tong, 2016] aligns similar nodes in terms of topology and attributes GSANA [Yasar and Çatalyürek, 2018] employs a set of seed nodes, which can be given by the user or precomputed, to calculate pairwise distances used as 2D-coordinates for matching. Another variant matches *weigthed* matrices using their spectra [Umeyama, 1988]; unfortunately, that is inapplicable to the unweighted case. Overall, such methods cannot handle cases where there is no additional information other than graph structure.

Method	Unsuperv.	Spectral	Plain	Flexible	Offline Precomp.	Multiscale
Supervised	✘	✘	✔	✘	✘	✘
BigAlign	✔	✘	✔	✘	✘	✘
FINAL	✔	✘	✘	✘	✘	✘
IsoRank	✔	✘	✔	✔	✔	✘
LREA	✔	✔	✔	✘	✘	✘
REGAL	✔	✘	✔	✔	✔	✘
GRASP	✔	✔	✔	✔	✔	✔

Table 1: Related work in terms of present (✔) and absent (✘) properties. *Supervised* methods [Liu *et al.*, 2016; Chu *et al.*, 2019; Man *et al.*, 2016; Zhou *et al.*, 2018] require aligned nodes as input. *Spectral* methods [Nassar *et al.*, 2018] use spectral properties of alignment matrices. FINAL [Zhang and Tong, 2016] does not work on *plain* graph structures as it requires node attributes. REGAL [Heimann *et al.*, 2018] and IsoRank [Liao *et al.*, 2009; Singh *et al.*, 2008] are *flexible* in allowing different algorithms for alignment (e.g. bipartite matching, nearest neighbors). GRASP, REGAL, and IsoRank can benefit from an offline precomputation of graph representations (results in Figure 7). GRASP explicitly captures *multiscale* properties through the heat kernel.

2.2 Unrestricted Alignment

[Add a description for CONE-Align]

Integer-programming methods. Klau [Klau, 2009] presents a Lagrangian relaxation for the integer programming problem posed by network alignment; the resulting algorithm is polynomial, yet still impracticable for large networks.

Embedding-based methods. REGAL [Heimann *et al.*, 2018] constructs node embeddings based on the connectivity structure and node attributes, and uses the similarity between these features for node alignment; we classify REGAL as an unrestricted method since it can work without attributes.

Matrix decomposition methods. IsoRank [Singh *et al.*, 2008] aligns multiple protein-protein interaction networks aiming to maximize the overall quality across all input networks; it constructs an eigenvalue problem for every pair of input networks and extracts a global alignment across a set of networks by a k-partite matching; it uses structural properties (PageRank), but also relies on a similarity measure between nodes which in a biology-specific case builds on the similarity of the proteins; it is improved with greedy approaches [Kollias *et al.*, 2013]. Another improvement on IsoRank, IsoRankN [Liao *et al.*, 2009], performs spectral clustering on the induced graph of pairwise alignment scores; as it is based on spectral methods, IsoRankN is claimed to be both error-tolerant and computationally efficient. EigenAlign [Feizi *et al.*, 2019] formulates the problem as a Quadratic Assignment Problem that considers both matches and mismatches and solves it by *spectral decomposition* of matrices. Building thereupon, Low-Rank EigenAlign [Nassar *et al.*, 2018] solves a maximum weight bipartite matching problem on a *low-rank* version of a node-similarity matrix, hence requires memory linear in the size of the graphs. However, EigenAlign variants use the first eigenvector of a joint adjacency matrix

between the two graphs to be aligned, rather than the eigenvectors of graph Laplacians, which provides richer information. A projected power iteration version of EigenAlign, Projected Power Alignments (PPA) [Onaran and Villar, 2017], improves recovery rates.

Belief propagation methods. NetAlign [Bayati *et al.*, 2013] solves a *sparse* variant of network alignment by a message-passing algorithm.

2.3 Shape Matching

Our work is inspired by shape matching methods that employ spectral properties [Litany *et al.*, 2017; Ovsjanikov *et al.*, 2012; Kovnatsky *et al.*, 2013]. Functional maps [Ovsjanikov *et al.*, 2012] generalize the matching of points to the matching of *corresponding functions* among shapes, by revealing a common decomposition of such functions using the eigenvectors of the Laplace-Beltrami operator; the graph equivalent of that operator is a graph's Laplacian matrix. Extensions of this methods match non-isometric shapes by aligning their Laplace-Beltrami operators' eigenbases [Kovnatsky *et al.*, 2013], and match a part of a shape to another full shape [Rodolà *et al.*, 2017]; such partial matching can be done fully in the spectral domain [Litany *et al.*, 2017] without requiring spatially modeling the part of a shape.

2.4 Spectral Methods

Graph spectra are well studied in theoretical computer science [Chung and Graham, 1997] and facilitate practical problem-solving in graph analysis, image partitioning, graph search, and machine learning [Belkin and Niyogi, 2006; Belabbas and Wolfe, 2009; McSherry, 2001; Shi and Malik, 2000]. The eigenvectors of the Laplacian of a point cloud graph converge to the eigenfunctions of the Laplace-Beltrami operator on the underlying Riemannian manifold, which justifies transferring Laplace-Beltrami operator-based methods from computational geometry to similar problems in graph analysis. NetLSD [Tsitsulin *et al.*, 2018] uses this correspondence to represent graphs via Laplacian spectral signatures so as to detect similar graphs in a multi-scale fashion. Graph convolutional networks also utilize graph spectra [Defferrard *et al.*, 2016] to learn filters on the eigenvectors. Still, calculating a graph's spectrum is computationally challenging; recent work proposes an approximation via spectral moments estimated through random walks [Cohen-Steiner *et al.*, 2018]. Our work employs graph spectra, yet can rely on fast methods for diagonally dominant matrices [Koutis *et al.*, 2015].

3 Background and Problem

Graph Alignment. Consider two undirected graphs, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where V_* are node sets, $E_* \subseteq V_* \times V_*$ are edges. Without loss of generality, we assume that $|V_1| = |V_2| = n$. A graph's *adjacency matrix* $A \in \{0,1\}^{n\times n}$ is a binary matrix where $A_{ij} = 1$ if there is an edge between nodes i and j and $A_{ij} = 0$ otherwise.

Definition 1. *Given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, a graph alignment $R : V_1 \to V_2$ is an* injective *function that maps nodes of G_1 to nodes of G_2.*

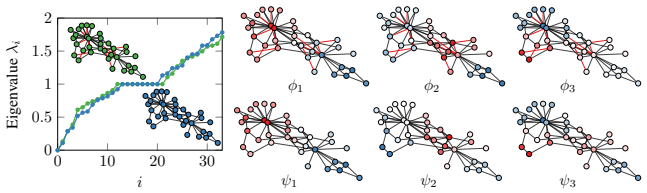
The graph alignment problem is to find such a function, which, expressed as a permutation matrix P, minimizes the difference $\|PA_1P^\top - A_2\|^2$. In case of isomorphic graphs, there exists a P such that $PA_1P^\top = A_2$, i.e., aligns the two graphs exactly. We are interested in the general, unrestricted problem case, in which there are no additional constraints on node attributes or matches known in advance. The problem is hard and not known to be in **NP**. [Cook, 1971].

We may express graph alignment in terms of a ground truth function $\tau : V_1 \to V_2$ that returns the correct alignment between the nodes V_1 in G_1 and the nodes V_2 in G_2. In the case of isomorphic graphs, this ground truth function τ is a bijection that admits an inverse mapping $\tau^{-1} : V_2 \to V_1$. The composition of the indicator function $\delta_i : V_1 \to \{0,1\}$ with τ^{-1}, $\delta_i \circ \tau^{-1} : V_2 \to \{0,1\}$ expresses the complete isomorphism among the two graphs, returning 1 if node $u \in V_2$ maps to node $i \in V_1$, 0 otherwise. By generalization, the composition $g_i = f_i \circ \tau^{-1}$ maps functions in G_2 to functions in G_1 for any family of real-valued functions $f_1, ..., f_q, f_i : V_1 \to \mathbb{R}$ and $g_1, ..., g_q, g_i : V_2 \to \mathbb{R}$ that associate a real value to each node in G_1 and G_2. This transformation among functions is called a *functional representation* of the mapping τ. In effect, finding an alignment among the nodes of two graphs corresponds to finding an alignment among functions on those nodes. We use such *functional alignments* as a shortcut to effective *node alignments*. To get there, we extend the concept of a functional map [Ovsjanikov *et al.*, 2012] from the continuous case in shape analysis to the discrete case of graph structures.

Functional maps. The operator $T_\mathcal{F} : (V_1 \times \mathbb{R}) \to (V_2 \times \mathbb{R})$ maps functions f on the nodes in G_1 to functions g on the nodes in G_2, i.e. $T_\mathcal{F}(f) = f \circ \tau^{-1} = g$. This operator is linear in the function space, i.e., $T_\mathcal{F}(c_1f_1 + c_2f_2) = (c_1f_1 + c_2f_2) \circ \tau^{-1} = c_1f_1 \circ \tau^{-1} + c_2f_2 \circ \tau^{-1} = c_1T_\mathcal{F}(f_1) + c_2T_\mathcal{F}(f_2)$. In addition, let $\phi_1, ..., \phi_n$ and $\psi_1, ..., \psi_n$ denote orthogonal bases for the space of functions on G_1's nodes, $V_1 \times \mathbb{R}$, and that on G_2's nodes, $V_2 \times \mathbb{R}$, respectively. Since those functions produce n-dimensional vectors, we can represent them as linear combinations of their basis vectors, $f = \sum_{i=1}^n a_i\phi_i$ and $g = \sum_{j=1}^n b_j\psi_j$. Then, by the linearity of $T_\mathcal{F}$,

$$T_\mathcal{F}(f) = T_\mathcal{F}\left(\sum_{i=1}^n a_i\phi_i\right) = \sum_{i=1}^n a_i T_\mathcal{F}(\phi_i) = \sum_{i=1}^n a_i \sum_{j=1}^n c_{ij}\psi_j = \sum_{j=1}^n b_j\psi_j$$

where $T_\mathcal{F}(\phi_i) = \sum_{j=1}^n c_{ij}\psi_j$. It follows that each coefficient b_j is the dot-product $\sum_{i=1}^n a_i c_{ij}$ between the coefficients $(a_1, ..., a_n)$ of functions in G_1 and the coefficients $(c_{1j}, ..., c_{nj})$ of the operator $T_\mathcal{F}$. In conclusion, in order to align real-valued functions on the nodes of two graphs, we need to find a *mapping matrix* $C \in \mathbb{R}^{n\times n}$ of coefficients among those functions. Note that such a mapping matrix C maps functions from G_1 to G_2, even when the ground-truth mapping τ is unknown. In a nutshell, GRASP obtains such a mapping matrix C for a well-chosen function and applies that C to mapping the indicator function δ from G_1 to G_2, thereby constructing a node alignment. The main question we need to answer is what orthogonal basis and functions we should use to construct our mapping matrix C. The next section answers



(a) Two graphs spectra (b) Their first 3 eigenvectors

Figure 2: We removed **red** edges from the **green** graph to obtain the **blue** graph. The eigenvalues (a) interlace; the respective eigenvectors ϕ_1, ϕ_2, ϕ_3 for **green** and ψ_1, ψ_2, ψ_3 for **blue** highlight common structures. The eigenvectors do not perfecly correspond, calling for the base alignment method of Section 4.5.

this question and builds on that answer to devise a solution based on spectral graph theory and linear algebra.

4 Solution

Here, we choose an orthonormal basis and a function, which are, in our judgement, appropriate for node alignment purposes, and define the complete pipeline of our solution.

4.1 Choice of basis: Normalized Laplacian

As a basis for representing functions as linear combinations of base functions, we use the eigenvectors of the graph's *normalized Laplacian*, i.e., the matrix $\mathcal{L} = I - D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, where D a diagonal degree matrix of node degrees $D_{ii} = \sum_{j=1}^{n} A_{ij}$ and A is the graph adjacency matrix; its eigen-decomposition is $\mathcal{L} = \Phi\Lambda\Phi^\top$, where Λ is a diagonal matrix of eigenvalues, $\{\lambda_1, \ldots, \lambda_n\}$, i.e., the graph's *spectrum*, which encodes structural information about communities, degree distribution, and diameter, and Φ is a matrix of eigenvectors, $\Phi_{\mathcal{L}} = [\phi_1 \phi_2 \ldots \phi_n]$. The eigenvectors form an orthogonal basis, which we use a standard basis. We use ϕ to indicate the eigenvectors of the Laplacian of graph G_1, and ψ to indicate those of G_2.

We consider this basis to be stuitable, since the eigenvectors of the normalized Laplacian converge to the eigenfunctions of the Laplace-Berltrami operator [Belkin and Niyogi, 2006], which measures the smoothness of continuous multi-dimensional surfaces; thus, our method extends differential geometry results to graphs.

4.2 Choice of function: Heat Kernel

The choice of functions $f_i : V_1 \to \mathbb{R}$, $g_i : V_2 \to \mathbb{R}$, is critical for our method. A poor choice would be detrimental to the results. A function of choice should have the following desirable properties:

Expressiveness. The function should express the graph's structure. For instance, a constant function returning the same value for all nodes would not yield a meaningful alignment.

Permutation-invariance. The function should not depend on the node index i; the indicator function does not have this property.

Robustness. A function is robust if it is insensitive to small perturbations in the graph. A similar property is the *multi-scale* property, which implies a function capturing both local and global characteristics (e.g., both edges and communities).

A function fulfilling these requirements is the time-parameterized *heat kernel*, which has been used for similar purposes in [Tsitsulin *et al.*, 2018]:

$$H_t = \Phi e^{-t\Lambda}\Phi^\top = \sum_{j=1}^{n} e^{-t\lambda_j}\phi_j\phi_j^\top \tag{1}$$

where $H_{t[ij]}$ measures the flow of heat from node i to node j at time t, as it diffuses from each node's neighborhood to the whole graph. We build our model functions over a sequence of time steps t using the *diagonal* of the heat kernel, which measures the heat flowing back to each node at time t.

The heat kernel enjoys the aforementioned properties [Tsitsulin *et al.*, 2018]: it expresses graph structure in a permutation-invariant manner, and is robust to small changes, as the value in the diagonal can be thought as the likelihood a node is reached by a diffusion process within time t. In the beginning of the diffusion, Equation 1 emphasises large λ, which correspond to *local* edge and ego-net properties. As time progresses, smaller eigenvalues get emphasized, reflecting *global* graph properties, such as communities. We recall that ϕ_2 is the Fiedler vector used to detect communities in spectral clustering.

We build our *corresponding functions* f_i, g_i, from the heat kernel at different time steps t, as linear combinations of the graph's Laplacian orthogonal eigenvectors $\phi_1, \ldots, \phi_n$. Specifically, let $F \in \mathbb{R}^{n\times q}$, $F = [f_1, \ldots, f_q]$ be the matrix containing the diagonals of the heat kernel of G_1, $H_t^{G_1}$, over q time[1] steps, $f_i = \sum_{j=1}^{n} e^{-t_i\lambda_j}\phi_j \odot \phi_j$, where $\odot$ denotes the element-wise vector product. Likewise, the matrix $G \in \mathbb{R}^{n\times q}, G = [g_1, \ldots, g_q]$ contains the diagonals of $H_t^{G_2}$, the heat kernel of G_2. While the q *columns* of F and G contain the same time-dependent heat-kernel-diagonal functions evaluated on the nodes of two different graphs, their n *rows* (i.e., nodes) are not aligned. We need to obtain such a node alignment.

4.3 Mapping matrix

We approximate each function f_i using only the first k eigenvectors, as done, by analogy, with eigenvectors of the Laplace-Beltrami operator on shapes [Belkin and Niyogi, 2006], and thereby calculate the corresponding function matrices F and G. F and G can be thought as coefficient matrices used to produce linear combinations, $F^\top\Phi$ and $G^\top\Psi$, of the Laplacian eigenvectors of G_1 and G_2, respectively. With a slight abuse of notation, we denote with Φ and Ψ the first k eigenvectors, hence $F^\top\Phi$ and $G^\top\Psi$ are in $\mathbb{R}^{q\times k}$. In the projection of the functions on the first k eigenvectors, we would like the corresponding functions to be equal up to a coefficient matrix $C \in \mathbb{R}^{k\times k}$. In the case of isomorphic graphs, it holds that $F^\top\Phi = G^\top\Psi C$, where C is a diagonal mapping matrix, hence:

$$\begin{bmatrix} \mathsf{diag}(g_1^\top\Psi) \\ \vdots \\ \mathsf{diag}(g_q^\top\Psi) \end{bmatrix} \begin{bmatrix} c_{11} \\ \vdots \\ c_{kk} \end{bmatrix} = \begin{bmatrix} \Phi^\top f_1 \\ \vdots \\ \Phi^\top f_q \end{bmatrix} \tag{2}$$

[1] In our experiments we select $q = 100$ values evenly spaced on the linear scale in the range [0.1, 50].

Matrix C is diagonal in the case of isomorphic graphs and deviates from a diagonal form as graphs diverge from perfect isomorphism; for the sake of simplicity, we assume a diagonal C, and obtain the diagonal entries that minimize the L_2-norm difference $\|\cdot\|_2^2$ between the left and rights side of Equation 2 using the ordinary least squares method, as in [Kovnatsky *et al.*, 2013]. In Section 4.5 we delve into the general case of non-isomorphic graphs.

4.4 Node-to-node correspondence

We consider the delta function $\delta_i(\cdot)$ as corresponding function; these functions yield an $n \times n$ identity matrix. We express such a function as a vector of coefficients, since the vector of δ_i is the ith row of the heat kernel at $t = 0$:

$$\delta_i = H_{i,t=0}^{G_1} = \sum_{j=1}^{n} \phi_{ij}\phi_j$$

The computation for delta functions on G_2 follows equivalently using Ψ in place of Φ. We may match the coefficient vectors of these corresponding indicator functions, as, ideally, for two matching nodes $v_i \in V_1$ and $v'_j \in V_2$, the coefficients of δ_i and δ_j for Φ and Ψ should be identical. In particular, the coefficients expressing δ_i as a linear combination of the first k eigenvectors are $\phi_{i1}, \ldots, \phi_{ik}$. We set $\Phi^\top$ and $C\Psi^\top$ in $\mathbb{R}^{k\times n}$ as coefficient matrices of the delta functions, aligned by C. Rows correspond to the first k Laplacian eigenvectors, while columns stand for graph nodes, rather than for time steps of heat diffusion. We need to match coefficient vectors, i.e., columns of $\Phi^\top$ and $C\Psi^\top$, to each other. This problem amounts to a *linear assignment problem*; we apply an off-the-shelf algorithm therefore, such as **nearest neighbor search** or **Jonker-Volgenant (JV)** [Jonker and Volgenant, 1987], to obtain an one-to-one matching between the columns of $\Phi^\top$ and $C\Psi^\top$, and hence an alignment of nodes in G_1 and G_2. We emphasize the *flexibility* of GRASP, as we may employ a different matching algorithm and a different transformation on the coefficient matrix, while preserving the overall framework.

4.5 Base Alignment

We have hitherto assumed that the graphs to be aligned, G_1 and G_2, are isomorphic, hence their eigenvectors correspond to each other with possible sign changes and an orthogonal and diagonal mapping matrix C exists. However, if the graphs are not isomorphic, then their eigenvectors diverge and the diagonal matrix C, which we enforce, cannot capture their relationship well. Figure 2 highlights this issue: at a high level the eigenvectors underline common structures, but they differ at the node level. In this case, we need to *align* the two eigenvector bases before we consider aligning corresponding vectors and, eventually, nodes. We express this *base alignment* [Kovnatsky *et al.*, 2013] in terms of an alignment matrix M.

Alignment matrix. We align the eigenvectors Ψ by a rotation matrix M so as transform Ψ into Φ: $\hat{\Psi} = \Psi M$. Since $\mathcal{L}\Psi = \Psi\Lambda$, finding Ψ is equivalent to the solution of the following quadratic minimization problem which penalizes elements outside of the diagonal, in order to preserve orthogonality of the basis:

$$\min_{\Psi} \mathsf{off}(\Psi^\top \mathcal{L}_2 \Psi) \text{ s.t. } \Psi^\top\Psi = I$$

where $\mathsf{off}(\cdot)$ denotes the sum of squared off-diagonal elements. Moreover, since the eigenvectors are orthonormal, $\Psi^\top\Psi = I$ and for G_2's graph Laplacian eigenvectors Λ_2, $\Psi^\top\mathcal{L}_2\Psi = \Psi^\top\Psi\Lambda_2 = \Lambda_2$, and $M^\top\Psi^\top\mathcal{L}_2\Psi M = M^\top\Lambda_2 M$. Putting the above together, our diagonalizing term is:

$$\min_{M} \mathsf{off}(M^\top\Lambda_2 M) \text{ s.t. } M^\top M = I$$

As we are minimizing over orthogonal matrices we can equivalently express the objective above as a minimization over orthogonal matrices of size $n \times n$, $S(n, n)$:

$$\min_{M\in S(n,n)} \mathsf{off}(M^\top\Lambda_2 M)$$

Coupling. The correspondence $\tau : G_1 \to G_2$ so that $\phi_i \approx \tau \circ \psi$ translates to

$$\min_{\Phi} \|F^\top\Phi - G^\top\Psi M\|_F^2$$

where F and G contain each graphs's corresponding functions. We combine the minimization terms for diagonalization and coupling, to get the following minimization problem, with regularization factor μ[2]:

$$\min_{M\in S(n,n)} \mathsf{off}(M^\top\Lambda_2 M) + \mu\|F^\top\Phi - G^\top\Psi M\|_F^2 \qquad (3)$$

Given that the eigenvectors of isomorphic graphs match each other with sign changes, we initialize M as a diagonal matrix with:

$$M_{i,i} = \begin{cases} +1 & \text{if } \|F^\top\phi_i - G^\top\psi_i\| \le \|F^\top\phi_i + G^\top\psi_i\| \\ -1 & \text{otherwise} \end{cases}$$

Equation 3 leads to a manifold optimization problem, which we solve using the trust-region methods [Absil *et al.*, 2007].

Scalability. We avoid computing all eigenvectors $n \times n$, exploiting the fact that we only need the first k eigenvectors for calculating C (see Section 4.3). So we only align the first k eigenvectors of Ψ to the first k eigenvectors of Φ, i.e $\bar{\Phi} = \hat{\Psi} = \bar{\Psi}M$ with $\bar{\Phi} = [\phi_1, \ldots, \phi_k]$ and $\bar{\Psi} = [\psi_1, \ldots, \psi_k]$. The optimization problem in Equation (3) then becomes

$$\min_{M\in S(k,k)} \mathsf{off}(M^\top\bar{\Lambda}_2 M) + \mu\|F^\top\bar{\Phi} - G^\top\bar{\Psi}M\|_F^2 \qquad (4)$$

with $\bar{\Lambda}_2 = \mathsf{diag}(\lambda_1, \ldots, \lambda_k)$.

After obtaining M, we use the eigenvectors in $\bar{\Phi}$ and the aligned eigenvectors $\hat{\Psi} = \bar{\Psi}M$ in the next step for the final alignment of nodes. In effect, our approach effectively trades off the problem of graph alignment with a proxy problem of manifold optimization, which we solve with reasonable accuracy and scalability.

[2] $\mu = 0.132$ in our experiments

4.6 Our algorithm: GRASP

Putting all together, GRASP consists of five steps. The algorithm pseudocode is described in the supplementary material.

Steps 1: Compute eigenvectors. In the first step, calculate the Laplacians $\mathcal{L}_1, \mathcal{L}_2$ of the two graphs G_1 and G_2. Then compute the eigenvectors Φ, Ψ and eigenvalues Λ_1, Λ_2 by the eigendecomposition $\mathcal{L}_1 = \Phi\Lambda_1\Phi^\top$ and $\mathcal{L}_2 = \Psi\Lambda_2\Psi^\top$.

Step 2: Compute corresponding functions. In the second step, calculate the matrices of corresponding functions $F = [f_1, \ldots, f_q]$ and $G = [g_1, \ldots, g_q]$ as diagonals of the heat kernel at time steps $[t_1, \ldots, t_q]$ with $f_i = \sum_{j=1}^{n} e^{-t_i\lambda_j}\phi_j \odot \phi_j$ and g_i equivalently using Ψ.

Step 3: Base alignment. After the corresponding functions are calculated, obtain the base alignment matrix M by minimizing equation 3. Then align the first k columns of Ψ, denoted by $\bar{\Psi}$ to the corresponding first k columns $\bar{\Phi}$ of Φ as $\hat{\Psi} = \bar{\Psi}M$.

Step 4: Calculate mapping matrix. Under the assumption that C is a diagonal matrix, calculate its diagonal elements $c_{11}, \ldots, c_{kk}$ by solving the least squares problem:

$$\min_{[c_{11},\ldots,c_{kk}]^\top} \left\| \begin{bmatrix} \mathsf{diag}(g_1^\top\bar{\Psi}) \\ \vdots \\ \mathsf{diag}(g_q^\top\bar{\Psi}) \end{bmatrix} \begin{bmatrix} c_{11} \\ \vdots \\ c_{kk} \end{bmatrix} - \begin{bmatrix} \hat{\Phi}^\top f_1 \\ \vdots \\ \hat{\Phi}^\top f_q \end{bmatrix} \right\|_2^2 \quad (5)$$

We then set $C = \mathsf{diag}(c_{11}, \ldots, c_{kk})$.

Step 5: Node alignment. Since the corresponding functions should have the same coefficients on both graphs, obtain the coefficients of the indicator functions for all nodes as $\bar{\Phi}^\top$ and $C\hat{\Psi}^\top$. In order to get the final alignment of nodes, align the columns of $\bar{\Phi}^\top$ and $C\hat{\Psi}^\top$ with a linear assignment algorithm.

4.7 Complexity analysis

The computation of the first k eigenvectors of the Laplacian matrix (Step 1), used as a base and to compute the corresponding functions, takes $\mathcal{O}(k \max\{|E_1|, |E_2|\})$ by fast methods for diagonally dominant matrices [Koutis *et al.*, 2015]. The base alignment (Step 3) needs $\mathcal{O}(k^3)$ to solve the orthogonality constraint through trust-region methods. The least-squares method (Step 4) runs in $\mathcal{O}(q\,k)$. The final matching step runs in $\mathcal{O}(n^3)$ by the JV algorithm. Overall, the $\mathcal{O}(n^3)$ time factor is dominant. In practice, as we note in Section 5.4, GRASP runs as fast as REGAL [Heimann *et al.*, 2018] on several datasets with precomputed eigendecomposition.

4.8 Connection to Differential Geometry

Our work rests on the theory on Riemannian manifolds [Gallot *et al.*, 1990] that studies continuous multidimensional surfaces, and builds on the analogy between a discrete graph's Laplacian $\mathcal{L}$ and the continuous Laplace-Beltrami operator [Tsitsulin *et al.*, 2018]. Another notable discretization of the Laplace-Beltrami operator used in shape analysis is the *cotangent scheme* [Meyer *et al.*, 2003], which includes information about the area of shape tiles and the angles among them. As such information is inapplicable in the case of graphs, we settled for the Laplacian matrix, considering a graph as a discrete form of a latent multidimensional manifold devoid of area and angle information.

5 Experiments

We run the experiments on a 40-core Intel Xeon CPU E5-2687Wv3, 3.10GHz machine with 368Gb RAM on python 3.6.9. [3] We used pymanopt [Townsend *et al.*, 2016] for solving the base alignment problem in Eq. (3).

Experimental setup. We perform our experiments on three different real-world networks, which properties can be found in Table XXX in the Supplementary Material. Following the setup described in [Heimann *et al.*, 2018], we generate permutated versions of these networks by applying a permutation matrix P to the original adjacency matrix A_{orig}, thus $A_{perm} = PA_{orig}P^\top$. We inject noise by randomly deleting edges with probability p, which ranges from 0.05 to 0.25, the latter value being higher than anything used in previous studies [Heimann *et al.*, 2018; Nassar *et al.*, 2018]. For each p, we generate 5 permutated graphs and average our accuracy results over them. We measure the alignment quality in terms of average accuracy computed as the ratio of correctly aligned nodes over the total number of nodes. Although we focus on alignment quality, we show in Section 5.4 how our method fares in terms of scalability.

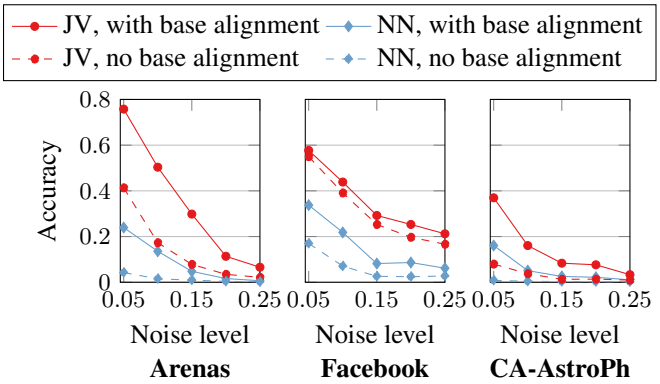


Figure 3: Accuracy of nearest neighbor and JV matching algorithms with and without aligning the eigenbases.

Baselines. We compare against the following state-of-the art baselines for *unrestriced* graph alignment.

- **REGAL** [Heimann *et al.*, 2018]: A method based on embedding vectors utilizing local structural features. In its original formulation, REGAL does not perform one-to-one alignment but allows several nodes to be matched to the same node. As the matching process is simply the task of matching the obtained embedding vectors, we modify it to fit our definition of graph alignment by applying a linear assignment algorithm to the embeddings.
- **Low Rank EigenAlign (LREA)** [Nassar *et al.*, 2018]: A spectral method which solves a one-to-one matching by optimizing a bipartite matching problem based on the minimization of edge mismatches. The method outperforms

[3] Anonimized code link: https://github.com/juhuhu/GrASp

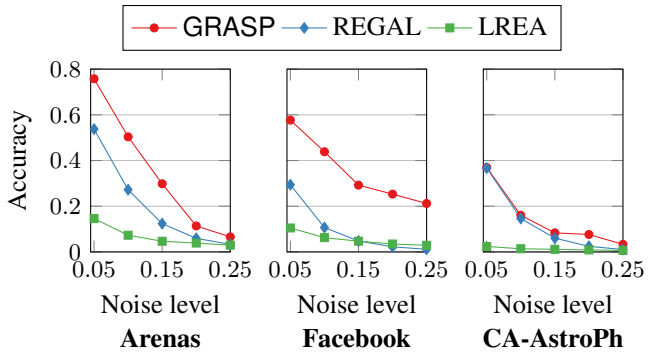


Figure 4: Accuracy compared to REGAL and LREA

all previous spectral methods like IsoRank [Singh *et al.*, 2008]; hence, it is used as the reference competitor.

We eschew a comparison with IsoRank [Singh *et al.*, 2008; Liao *et al.*, 2009] and other methods for the alignment of biological networks [**?**; El-Kebir *et al.*, 2015], since REGAL [Heimann *et al.*, 2018] and LREA [Nassar *et al.*, 2018] significantly outperform those methods.

Parameter tuning. [TODO]

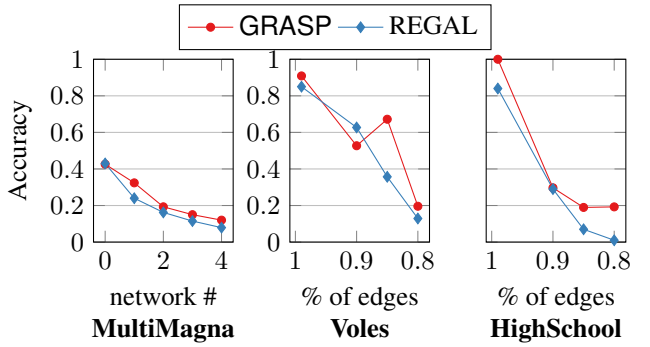


Figure 5: Accuracy compared to REGAL [Heimann *et al.*, 2018] on three real datasets.

5.1 Justifying algorithmic choices

The purpose of the experiment in Figure 3 is twofold: (i) showing how base alignment described in Section 4.5 effectively helps graph alignment in non-isomorphic graphs, and (ii) evaluating the choice of the assignment algorithm for node-to-node correspondence described in Section 4.4. Both base alignment and JV linear assignment bring a substantial advantage over the more rudimentary counterparts. The behavior of GRASP is consistent across datasets and algorithmic choices. In the following experiments, and unless otherwise stated, we settle on the variant of GRASP equipped with base alignment.

5.2 Comparison to previous methods

We compare GRASP with REGAL [Heimann *et al.*, 2018] and LREA [Nassar *et al.*, 2018] in terms of accuracy. As REGAL allows for different assignment algorithms for node-to-node correspondence, we equip REGAL with JV to supply the same advantage of GRASP. We perform sanity experiments to assess whether REGAL with JV outperforms over nearest neighbors. The results, not reported for sake of readability, confirm our hypothesis. The results in Figure 4 shows that GRASP outperforms previous methods by a large margin in Arenas and Facebook graph, achieving 76% accuracy in Arenas and 59% in Facebook with 5% noise. GRASP's behaviour is consistent despite high levels of noise. Moreover, on the complex CA-AstroPH collaboration graph that contains a large number of nodes with a large degree, our method fares as good as local methods, such as REGAL. The accuracy of GRASP increases more steeply as noise falls.

5.3 Real world networks

In addition to our experiments with permuted graphs and artificial noise, we perform experiments where we match one real world network to another unedited real world network. MultiMagna is a collection of graphs consisting of a base yeast network and five variations with different sets of edges. We match these five variations to the original, while measuring alignment accuracy, as Figure 5 presents. HighSchool and Voles are two proximity networks, given as evolving graphs. We match the latest version of these graphs with versions at time steps with 80%, 85%, 90%, and 99% of all edges. Figure 5 shows the results. Overall, we observe that the advantage of GRASP over REGAL that we observed with synthetic noise transfers well to real-world alignment problems.

5.4 Efficiency

Last, we compare the efficiency of our method on different datasets. Our method, as well as REGAL, allows precomputation of the representations prior to alignment. In our case, step 1–3 in Section 4.6 are independent of the alignment and can hence be performed in an offline phase. Figure 6 the time in seconds to compute the alignments when the corresponding functions and the eigendecompositions are not accounted for. GRASP outperforms REGAL and LREA in the largest CA-AstroPh and performs, in the other cases as well as the optimized low-rank method of LREA. Figure 7 shows the time without precomputing the eigendecomposition and the corresponding functions. In this case, REGAL does not exhibit any substantial advantage even in the smaller Arenas and Facebook graphs. GRASP attains more accurate results with a negligible increase in time. As performance was similar on real-world networks, we omit the corresponding time results.

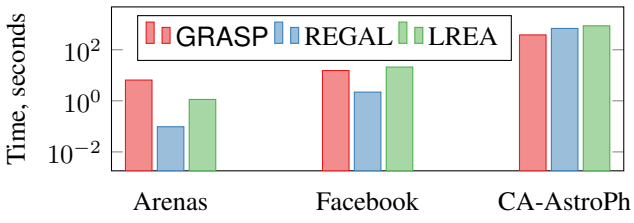


Figure 6: Alignment time *including precomputation.*

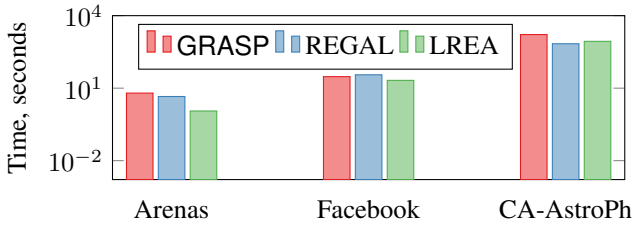


Figure 7: Alignment time *excluding precomputation.*

6 Conclusion

We proposed GRASP, a novel graph alignment method that matches graphs utilizing the eigenvectors of their Laplacian matrices. To solve this problem, we first establish a functional correspondence among the pre-aligned eigenvectors of the two graphs, extending the shape-analysis concept of functional maps, and then extract a linear assignment among matrix columns. The functional correspondences we employ capture multi-scale graph properties, and lead to a methodology that attains superior alignment quality over the state-of-the-art methods for graph alignment across noise levels and real-world graph types, with noise levels higher than anything used in previous studies. In the future, we plan to extend our method to partial correspondences among graphs, and towards flexible definitions of subgraph isomorphism, including the case of matching graphs with unequal numbers of nodes.

References

[Abdulrahim, 1998] Mohammad Abdulkader Abdulrahim. Parallel algorithms for labeled graph matching. 1998.

[Absil *et al.*, 2007] P-A Absil, Christopher G Baker, and Kyle A Gallivan. Trust-region methods on riemannian manifolds. *FoCM*, 7(3):303–330, 2007.

[Bayati *et al.*, 2013] Mohsen Bayati, David F. Gleich, Amin Saberi, and Ying Wang. Message-passing algorithms for sparse network alignment. *TKDD*, 7(1):3:1–3:31, 2013.

[Belabbas and Wolfe, 2009] Mohamed-Ali Belabbas and Patrick J Wolfe. Spectral methods in machine learning and new strategies for very large datasets. *PNAS*, 106(2):369–374, 2009.

[Belkin and Niyogi, 2006] Mikhail Belkin and Partha Niyogi. Convergence of laplacian eigenmaps. In *NeurIPS*, pages 129–136, 2006.

[Chu *et al.*, 2019] Xiaokai Chu, Xinxin Fan, Di Yao, Zhihua Zhu, Jianhui Huang, and Jingping Bi. Cross-network embedding for multi-network alignment. In *TheWebConf*, pages 273–284, 2019.

[Chung and Graham, 1997] Fan RK Chung and Fan Chung Graham. *Spectral graph theory*. Number 92. American Mathematical Soc., 1997.

[Cohen-Steiner *et al.*, 2018] David Cohen-Steiner, Weihao Kong, Christian Sohler, and Gregory Valiant. Approximating the spectrum of a graph. In *KDD*, pages 1263–1271, 2018.

[Cook, 1971] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC*, pages 151–158, 1971.

[Defferrard *et al.*, 2016] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, pages 3844–3852, 2016.

[El-Kebir *et al.*, 2015] Mohammed El-Kebir, Jaap Heringa, and Gunnar W Klau. Natalie 2.0: Sparse global network alignment as a special case of quadratic assignment. *Algorithms*, 8(4):1035–1051, 2015.

[Feizi *et al.*, 2019] Soheil Feizi, Gerald Quon, Mariana Mendoza, Muriel Medard, Manolis Kellis, and Ali Jadbabaie. Spectral alignment of graphs. *IEEE Trans. Netw. Sci.*, 2019.

[Gallot *et al.*, 1990] Sylvestre Gallot, Dominique Hulin, and Jacques Lafontaine. *Riemannian geometry*, volume 2. Springer, 1990.

[Heimann *et al.*, 2018] Mark Heimann, Haoming Shen, Tara Safavi, and Danai Koutra. REGAL: representation learning-based graph alignment. In *CIKM*, pages 117–126, 2018.

[Jonker and Volgenant, 1987] Roy Jonker and Anton Volgenant. A shortest augmenting path algorithm for dense and sparse linear assignment problems. *Computing*, 38(4):325–340, 1987.

[Kazemi *et al.*, 2015] Ehsan Kazemi, S Hamed Hassani, and Matthias Grossglauser. Growing a graph matching from a handful of seeds. *PVLDB*, 8(10):1010–1021, 2015.

[Klau, 2009] Gunnar W. Klau. A new graph-based method for pairwise global network alignment. *BMC Bioinformatics*, 10(S-1), 2009.

[Kobler *et al.*, 2012] Johannes Kobler, Uwe Schöning, and Jacobo Torán. *The graph isomorphism problem: its structural complexity*. Springer Science & Business Media, 2012.

[Kollias *et al.*, 2013] Giorgos Kollias, Madan Sathe, Shahin Mohammadi, and Ananth Grama. A fast approach to global alignment of protein-protein interaction networks. *BMC research notes*, 6(1):35, 2013.

[Koutis *et al.*, 2015] Ioannis Koutis, Alex Levin, and Richard Peng. Faster spectral sparsification and numerical algorithms for sdd matrices. *TALG*, 12(2):1–16, 2015.

[Koutra *et al.*, 2013] Danai Koutra, Hanghang Tong, and David Lubensky. BIG-ALIGN: fast bipartite graph alignment. In *ICDM*, pages 389–398, 2013.

[Kovnatsky *et al.*, 2013] Artiom Kovnatsky, Michael M Bronstein, Alexander M Bronstein, Klaus Glashoff, and Ron Kimmel. Coupled quasi-harmonic bases. In *Comput Graph Forum*, volume 32, pages 439–448, 2013.

[Liao *et al.*, 2009] Chung-Shou Liao, Kanghao Lu, Michael Baym, Rohit Singh, and Bonnie Berger. IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, 25(12):i253–i258, 2009.

[Litany *et al.*, 2017] Or Litany, Emanuele Rodolà, Alexander M. Bronstein, and Michael M. Bronstein. Fully spectral partial shape matching. *Comput. Graph. Forum*, 36(2):247–258, 2017.

[Liu *et al.*, 2016] Li Liu, William K Cheung, Xin Li, and Lejian Liao. Aligning users across social networks using network embedding. In *IJCAI*, pages 1774–1780, 2016.

[Malmi *et al.*, 2017] Eric Malmi, Aristides Gionis, and Evimaria Terzi. Active network alignment: a matching-based approach. In *CIKM*, pages 1687–1696, 2017.

[Man *et al.*, 2016] Tong Man, Huawei Shen, Shenghua Liu, Xiaolong Jin, and Xueqi Cheng. Predict anchor links across social networks via an embedding approach. In *IJCAI*, volume 16, pages 1823–1829, 2016.

[McSherry, 2001] Frank McSherry. Spectral partitioning of random graphs. In *FOCS*, pages 529–537. IEEE, 2001.

[Meyer *et al.*, 2003] Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pages 35–57. Springer, 2003.

[Nassar *et al.*, 2018] Huda Nassar, Nate Veldt, Shahin Mohammadi, Ananth Grama, and David F. Gleich. Low rank spectral network alignment. In *WWW*, pages 619–628, 2018.

[Onaran and Villar, 2017] Efe Onaran and Soledad Villar. Projected power iteration for network alignment. In *Wavelets and Sparsity XVII*, volume 10394, page 103941C. International Society for Optics and Photonics, 2017.

[Ovsjanikov *et al.*, 2012] Maks Ovsjanikov, Mirela Ben-Chen, Justin Solomon, Adrian Butscher, and Leonidas J. Guibas. Functional maps: a flexible representation of maps between shapes. *ACM Trans. Graph.*, 31(4):30:1–30:11, 2012.

[Rodolà *et al.*, 2017] Emanuele Rodolà, Luca Cosmo, Michael M. Bronstein, Andrea Torsello, and Daniel Cremers. Partial functional correspondence. *Comput. Graph. Forum*, 36(1):222–236, 2017.

[Schellewald and Schnörr, 2005] Christian Schellewald and Christoph Schnörr. Probabilistic subgraph matching based on convex relaxation. In *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 171–186. Springer, 2005.

[Shi and Malik, 2000] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *Departmental Papers (CIS)*, page 107, 2000.

[Singh *et al.*, 2008] Rohit Singh, Jinbo Xu, and Bonnie Berger. Global alignment of multiple protein interaction networks with application to functional orthology detection. *PNAS*, 105(35):12763–12768, 2008.

[Townsend *et al.*, 2016] James Townsend, Niklas Koep, and Sebastian Weichwald. Pymanopt: A python toolbox for optimization on manifolds using automatic differentiation. *The Journal of Machine Learning Research*, 17(1):4755–4759, 2016.

[Tsitsulin *et al.*, 2018] Anton Tsitsulin, Davide Mottin, Panagiotis Karras, Alexander M. Bronstein, and Emmanuel Müller. Netlsd: Hearing the shape of a graph. In *KDD*, pages 2347–2356, 2018.

[Umeyama, 1988] Shinj Umeyama. An eigendecomposition approach to weighted graph matching problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(5):695–703, 1988.

[Yartseva and Grossglauser, 2013] Lyudmila Yartseva and Matthias Grossglauser. On the performance of percolation graph matching. In *ACM COSN*, pages 119–130, 2013.

[Yasar and Çatalyürek, 2018] Abdurrahman Yasar and Ümit V Çatalyürek. An iterative global structure-assisted labeled network aligner. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2614–2623, 2018.

[Zhang and Tong, 2016] Si Zhang and Hanghang Tong. FINAL: fast attributed network alignment. In *KDD*, pages 1345–1354, 2016.

[Zhou *et al.*, 2018] Fan Zhou, Lei Liu, Kunpeng Zhang, Goce Trajcevski, Jin Wu, and Ting Zhong. Deeplink: A deep learning approach for user identity linkage. In *INFOCOM*, pages 1313–1321, 2018.