BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 16 2014, pages 2351–2359
doi:10.1093/bioinformatics/btu307

Systems biology

Advance Access publication May 2, 2014

A comparison of algorithms for the pairwise alignment of biological networks

Connor Clark* and Jugal Kalita

Department of Computer Science, University of Colorado Colorado Springs, Colorado Springs, CO 80918, USA

Associate Editor: Alfonso Valencia

ABSTRACT

Motivation: As biological inquiry produces ever more network data, such as protein–protein interaction networks, gene regulatory networks and metabolic networks, many algorithms have been proposed for the purpose of *pairwise network alignment*—finding a mapping from the nodes of one network to the nodes of another in such a way that the mapped nodes can be considered to correspond with respect to both their place in the network topology and their biological attributes. This technique is helpful in identifying previously undiscovered homologies between proteins of different species and revealing functionally similar subnetworks. In the past few years, a wealth of different aligners has been published, but few of them have been compared with one another, and no comprehensive review of these algorithms has yet appeared.

Results: We present the problem of biological network alignment, provide a guide to existing alignment algorithms and comprehensively benchmark existing algorithms on both synthetic and real-world biological data, finding dramatic differences between existing algorithms in the quality of the alignments they produce. Additionally, we find that many of these tools are inconvenient to use in practice, and there remains a need for easy-to-use cross-platform tools for performing network alignment.

Contact: cclark@uccs.edu, jkalita@uccs.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on December 2, 2013; revised on March 21, 2014; accepted on April 28, 2014

1 INTRODUCTION

Many areas of bioinformatics now use and produce network data, including protein–protein interaction (PPI) networks and gene coexpression networks, and the tools and techniques of graph theory are being brought to bear to produce new techniques for biological network analysis. Comparing two biological networks is a particularly challenging problem, as many interesting questions we might ask about these networks are computationally intractable to answer (Atias and Sharan, 2012). Here, we focus on the task of aligning two PPI networks from different species. That is, we want to find a mapping from the nodes of one network to the nodes of another, in such a way as to maximize the topological and biological similarity of the pairs of nodes that are aligned to one another. This allows for the identification of both homologous proteins as well as similar modules or pathways in the networks themselves. Successes of PPI network alignment so far include uncovering large shared subnetworks between species as diverse as *Saccharomyces cerevisiae* and *Homo sapiens*, and reconstructing phylogenetic relationships between species based solely on the amount of overlap discovered between their PPI networks (Kuchaiev and Przulj, 2011; Kuchaiev *et al.*, 2010). Most papers in the literature report promising results in creating alignments that show large regions of biological and topological similarity between the PPI networks of various species.

The ultimate goal of network alignment is to transfer knowledge of protein function from one species to another. Because sequence similarity metrics such as BLAST bit scores (Altschul *et al.*, 1990) are not conclusive evidence of similar function, the purpose of aligning two PPI networks is to supplement sequence similarity with topological information so as to identify orthologs as accurately as possible. The primary challenge in designing such an aligner is to accurately estimate the topological similarity of two nodes and to combine that with sequence similarity to produce an alignment. The aligners published so far vary widely in their approaches to doing so, and some aligners are much better at optimizing for one of these goals than the others.

Furthermore, an exact solution to the network alignment problem is unattainable. The problem of global alignment is equivalent to the subgraph isomorphism problem, which is NP-complete (Cook, 1971), so aligners settle for approximate solutions. The variety of approaches in use and the lack of a popular standard solution present a difficult situation for those who would simply like to compare some biological networks that they have produced in the course of their research. These algorithms differ greatly in the quality of the alignments they produce and even more greatly in their compute time.

In this article, we survey and benchmark network alignment algorithms that are *pairwise* and *global*. Pairwise alignment algorithms align two graphs only. They are contrasted with multiple alignment algorithms, which try to find transitive alignments between more than two input networks at a time (Flannick *et al.*, 2008; Kalaev *et al.*, 2008, 2009; Liao *et al.*, 2009; Sahraeian and Yoon, 2013; Shih and Parthasarathy, 2012). Borrowing terminology from sequence alignment, we also distinguish between *global* alignment algorithms, which attempt to find a single overall alignment from one network to another, and *local* aligners, which may output several mutually incompatible alignments for the input networks (Berg and Lässig, 2004; Flannick *et al.*, 2006; Kelley *et al.*, 2004; Koyutürk *et al.*, 2006; Liang *et al.*, 2006; Sharan *et al.*, 2005). Local alignment is more useful when we

*To whom correspondence should be addressed.

desire to identify several potential orthologs per input protein, whereas global alignment is more helpful for identifying larger conserved networks that are indicative of a common ancestor. Global alignments can also be somewhat easier to interpret, as the produced mapping is one to one. Because pairwise global aligners have been much more popular in the recent literature, and because it is unclear how to compare a global algorithm with a local algorithm, or a pairwise aligner to a multiple aligner, we evaluate pairwise global aligners only.

We present the problem of network alignment, overview the various approaches that have been proposed to solve the problem and evaluate the quality of the alignments produced by a wide range of different alignment algorithms from the first generation of global alignment tools to the most recently published techniques. Given the rapid pace at which new techniques are being published, many of these algorithms have never been directly compared with one another. Therefore, we include extensive comparisons and benchmarks in this article. We make use of a recently developed framework for testing alignment algorithms with synthetic PPI network data (Sahraeian and Yoon, 2012) as well as a real-world PPI dataset (Park *et al.*, 2011). We find great differences in the quality of the alignments produced by existing alignment programs.

2 METHODS

2.1 Pairwise alignment

Of the many proposed methods for analyzing biological networks, global network alignment is one of the most ambitious. We are given two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, whose vertices represent proteins, and the presence of the edge (u, v) in E_1 (E_2) indicates that the two proteins represented by u and v interact in G_1 (G_2). Most aligners assume, without loss of generality, that $|V_1| < |V_2|$.

The problem of pairwise alignment is to find a one-to-one function $f: V_1 \rightarrow V_2$ that maps each node in V_1 to the node in V_2 that it best matches [as a shorthand to make several equations more readable, we will treat f as a function on edges as well. In this case, $f((u, v))$ is simply a more readable shorthand for $(f(u), f(v))$]. It must be noted that some algorithms produce a *partial* function, abstaining from mapping nodes that cannot be matched well.

Most aligners decompose the process of producing a matching into two steps. First, for each pair of nodes in $V_1 \times V_2$, we compute their similarity by examining the local topology of the graph around those two nodes and by their sequence similarity, as measured by BLAST bit scores or E-values. Second, taking the similarities between these nodes as weighted edges in a bipartite graph with two sets of nodes V_1, V_2, we solve the maximum-weight bipartite matching problem to generate a mapping from V_1 to V_2. The pairwise alignment software available differs primarily in how they handle these two steps, with most innovation being focused on the first step of estimating the topological similarity of two nodes. This process is just a general schematic that applies to many aligners. Some, such as NATALIE 2.0, do not follow the two-step process and instead optimize a relaxed version of the problem directly.

We must stress that all existing aligners only *approximately* solve this problem, and they generally introduce approximations in two ways. First, they introduce a relaxed problem definition. Second, they use a heuristic algorithm to approximately solve the relaxed problem. For instance, GRAAL frames the problem as matching nodes to one another in such a way as to maximize their graphlet signature similarity (a measure of how many small subgraphs of various shapes are in both nodes' respective neighborhoods), in the hopes that maximizing this metric between all aligned pairs of nodes will produce a biologically informative alignment (Kuchaiev *et al.*, 2010). Then, a greedy matching heuristic is used that aligns the most similar pair of nodes first and then works outward, aligning their neighborhoods. When evaluating this algorithm, it is not clear how much of its performance is attributable to the metric of graphlet signature similarity, and how much is due to using a matching algorithm that prefers to map neighbors to neighbors. All we can do is evaluate how well the heuristic solution to the relaxed problem solves our original problem. If a given aligner performs poorly, it could be that the relaxed problem is a good choice, but that choice is hamstrung by a poorly designed heuristic to solve it. Without being able to swap out the parts of these aligners, however, all we can do is evaluate their final results.

2.2 Topological similarity versus domain-specific similarity measures

Broadly speaking, there are two ways to estimate the similarity of two nodes. One may assess the shape of the network around that node through a variety of metrics such as degree, eccentricity, betweenness or the more recently developed graphlet degree (Pržulj, 2007). Then, nodes that appear to be in topologically similar regions of their respective networks are considered likely matches. Sequence similarity information is also useful, and BLAST bit scores or E-values have been popular measures of node similarity. In most alignment algorithms, both topological information and sequence similarity information are needed, although there are a few notable exceptions described below.

Both topological similarity techniques and sequence similarity have advantages and disadvantages. It has been argued that overreliance on topological similarity can be misleading, as actual complexes may appear disconnected in current noisy incomplete datasets, and so sequence similarity information is essential to produce the best alignment possible (Huang *et al.*, 2012). Sequence similarity scores also have their problems, however, as the actual level of sequence similarity between two proteins that serve a similar function can vary (Chindelevitch *et al.*, 2013). Furthermore, it has been found that some aligners that rely heavily on sequence similarity tend to only produce good alignments between networks from more closely related species (Patro and Kingsford, 2012).

After an aligner has computed the similarity of each pair of nodes, it must map them together. This is the maximum-weight bipartite matching problem: for each node in G_1, we assign one node in G_2, such that no two nodes in G_1 are assigned to the same node in G_2. A simple strategy for this matching is the Hungarian algorithm, a standard algorithm that produces an optimal mapping (Chindelevitch *et al.*, 2013; *et al.*, 2010). However, because the similarity scores between nodes that we use are themselves only approximate, the $O(n^3)$ time complexity of the Hungarian algorithm is generally not worth the time, and most aligners favor faster greedy matching algorithms. Many aligners, such as GRAAL, MI-GRAAL and GHOST, use variants on a 'seed-and-extend' method, where the best matching pair of nodes are aligned first, and then nodes neighboring that pair are matched (Chindelevitch *et al.*, 2013; Kuchaiev and Przulj, 2011; Kuchaiev *et al.*, 2010; Patro and Kingsford, 2012). Additionally, several aligners, such as IsoRank and NATALIE 2.0, restrict their alignment to consider only pairs where the bit score or E-value satisfies a user-specified threshold. In these cases, the number of pairings that must be considered is restricted to be $O(V_1)$, and algorithms with better time complexity can be used (El-Kebir *et al.*, 2011; Singh *et al.*, 2008).

2.3 Test data

To perform our evaluation of existing alignment algorithms, we make use of the NAPAbench synthetic PPI network data, which was created specifically for benchmarking network alignment algorithms (Sahraeian and Yoon, 2012). These benchmark data are made using state-of-the-art

algorithms for simulating the evolution of PPI networks and can construct arbitrary phylogenetic trees with PPI networks of customizable sizes. With these synthetic data, the exact topology and orthology of the two networks are known. Because real orthology and PPI network data are constantly improving in terms of both completeness and accuracy, benchmarking on a perfect dataset gives us a good idea of the upper bound of each alignment algorithm's performance, rendering our evaluation independent of the quality of biological data available at the time it is published. Furthermore, with the known true alignment available as ground truth, we are able to get a much better idea of an alignment's quality. NAPAbench has previously only been used to benchmark several older pairwise alignment algorithms and some more recent multiple alignment algorithms (Sahraeian and Yoon, 2013). Here, we use it to test more and newer pairwise algorithms.

Because of the number of algorithms benchmarked, and the high time requirements of some, we use a subset of the standard NAPAbench dataset, consisting of nine pairwise alignment problems. This includes three problems from each of the three PPI network evolution models used to generate the synthetic network data. These are the duplication with random mutation model (Pastor-Satorras *et al.*, 2003), the duplication–mutation–complementation model (Vázquez *et al.*, 2002) and a crystal growth (CG) model that has recently been proposed by Kim and Marcotte (2008). For specifics on the sizes of the networks used in these alignments, see the Supplementary Information File.

Additionally, we use experimentally derived PPI network data from IsoBase (Park *et al.*, 2011), supplemented with sequence similarity and Gene Ontology (GO) annotation data from the Supplementary Information File of Aladag and Erten (2013). The IsoBase dataset is popular for evaluating alignment algorithms, so we make use of it to further ease comparison with alignment algorithms that may be published in the future. It is important to test alignment algorithms on real datasets, as such data are noisy and incomplete. How well an alignment algorithm handles the spurious and missing edges that occur in real data is an important part of understanding its performance. However, we do not know the true alignment between empirically derived PPI networks, so performance metrics for alignment on real datasets are inherently less informative.

2.4 Evaluating alignment quality

Several methods for evaluating the quality of an alignment algorithm's output have been proposed. When evaluating with real biological data, the true alignment between two PPI networks is unknown, so one cannot simply report the percentage of nodes mapped to their true orthologs, as we do with synthetic data. For pairwise alignment algorithms, the metric of edge correctness (EC), proposed by Kuchaiev *et al.* (2010), reports the percentage of edges in G_1 that are conserved under the produced mapping to G_2:

$$EC = \frac{|f(E_1) \cap E_2|}{|E_1|} \times 100\% \quad (1)$$

where f is the mapping produced by the alignment algorithm. EC is suggestive of a good alignment, but because it is always possible that two biologically unrelated edges have been mapped to each other, even 100% EC is not conclusive evidence of a correct alignment.

Although EC is an intuitive measure of alignment quality, a more nuanced alternative that has been proposed recently (Patro and Kingsford, 2012) is the induced conserved structure (ICS) score. This extends EC with a further intuition: if some region of G_2 is dense, then a sparse region of G_1 could be mapped to it in many different ways. We would rather align a sparse region of G_1 to a sparse region of G_2 to increase our confidence that the alignment is not merely a coincidence. The ICS score penalizes alignments that map to denser subgraphs of G_2. Let $G[V]$ denote the induced subgraph of G on the vertices V. Then, the ICS score of an alignment f from G_1 to G_2 is as follows (Patro and Kingsford, 2012):

$$ICS = \frac{|f(E_1) \cap E_2|}{|E_{G_2[f(V_1)]}|}. \quad (2)$$

An additional benefit of the ICS score is it equals 1 if and only if f is an isomorphism, whereas the same is not true for EC score. This gives us a more intuitive understanding of the upper bound of an ICS score. We report ICS scores for alignments in this article and additionally provide EC scores in the Supplementary Information File.

Another popular metric of alignment correctness is the size of the largest connected component shared by the two graphs that is found by the alignment. This largest connected shared component (LCSC) is the largest connected subgraph of G_1 that was found to also exist in G_2. A larger LCSC implies that we have found a larger amount of shared structure between the two PPI networks. Although it has been noted that current network data are woefully incomplete (Huang *et al.*, 2012), our results below show there appears to be some relationship between larger LCSC and the number of correctly mapped nodes.

Given the limitations of the above topological measures of alignment quality, measures of agreement derived from biological information are also popular. All the literature on PPI network alignment makes use of gene orthology annotations from the GO database (Ashburner *et al.*, 2000) to evaluate the accuracy of their alignments, by comparing the similarity of GO annotations between aligned proteins. Because most network alignment papers so far present entirely novel methods, most of the emphasis on evaluating the biological quality of an alignment has been focused on verifying that the alignment is plausible, by checking whether aligned nodes are biologically similar in terms of the GO. Much less work has been done so far on deciding which annotations seen in one protein should be predicted as being present in the protein to which it has been aligned. A given protein is mapped to any number of GO terms, and even true orthologs may not have the same set of GO terms, both because of incompleteness of the GO data as well as divergence of protein function. Thus, following Aladag and Erten (2013), we define GO consistency (GOC) as

$$GOC(u, v) = \frac{|GO(u) \cap GO(v)|}{|GO(u) \cup GO(v)|} \quad (3)$$

for an aligned pair of nodes $u \in V_1$ and $v \in V_2$, where $GO(u)$ denotes the GO terms associated with the protein u that are distance 5 from the root of the GO hierarchy. Limiting the set of terms in this way prevents inflated scores from counting both more- and less-specific GO terms. Because network aligners often use sequence similarity to create an alignment, we also experiment with restricting this score to only use GO terms with experimental evidence codes, which excludes GO terms that have been assigned on the basis of sequence similarity. We then report the sum of the GOC over all alignment pairs for each alignment produced. For NAPAbench's synthetic benchmark data, where true functional orthology is known, evaluation is much easier—alignments will simply be evaluated by the percentage of aligned pairs that have been aligned to true orthologs.

2.5 Alignment algorithms evaluated

Our primary focus in this article is to evaluate and compare the many alignment programs that have cropped up in the past few years. There are many algorithms that have been published since 2011 that have never been directly compared with one another. Many of these recently published network alignment papers use only the older IsoRank (Singh *et al.*, 2008) or GRAAL (Kuchaiev *et al.*, 2010) for a baseline, or are additionally compared with MI-GRAAL (Kuchaiev and Przulj, 2011). Because IsoRank and GRAAL were among the first biological network aligners created, it is unsurprising that more recent algorithms perform better. Furthermore, differences in evaluation datasets used make it difficult to

even compare algorithms transitively based on their performance relative to these older algorithms. The goal of this article is to provide direct comparisons of these algorithms so as to better inform practitioners who might want to use such alignment software in their own work.

A number of criteria determined the selection of algorithms to include. First, given how many new algorithms have been presented in the past few years, more recent algorithms are favored. Furthermore, we tend to favor algorithms that are presented as tools—those that are accompanied with publicly available software that can be used relatively easily, typically providing a command line interface. Given these restrictions, as well as time restrictions and difficulties with getting some programs to run at all, we have omitted several tools that could be competitive with the ones discussed here (Bayati *et al*., 2013; Chindelevitch *et al*., 2013; Huang *et al*., 2012; Khan *et al*., 2012; Kollias *et al*., 2013; Kpodjedo *et al*., 2014; Milenković *et al*., 2010; Phan *et al*., 2012; Tian and Samatova, 2013; Todor *et al*., 2013).

Given its historical importance, we include the pairwise algorithm IsoRank (Singh *et al*., 2008). Although virtually every alignment algorithm published since is claimed to perform better, IsoRank was the first and most widely cited algorithm used for global network alignment, and it remains a popular baseline for measuring the performance of new algorithms. With IsoRank, topological similarity between two nodes is defined recursively—two nodes are similar if their neighbors are similar. This intuition is formalized as an eigenvalue problem, and the similarity score matrix is iteratively refined using the power method for computing an eigenvalue. This is combined with sequence data to find a mapping using a seed-and-extend algorithm.

GRAAL (Kuchaiev *et al*., 2010) is the original algorithm in the GRAAL series. It is notable for being the first algorithm to use topological data exclusively to construct an alignment. GRAAL determines topological similarity by counting 'graphlets'—small induced subgraphs. For each node in the input networks, GRAAL computes how many times the node occurs in each graphlet. These counts are used to construct a graphlet degree signature, and the distance between the graphlet degree signature of two nodes is used as a measure of topological similarity between the nodes of the networks being aligned. A heuristic matching algorithm is used that first aligns the two nodes that are most similar, then works outward to align their neighbors, until all nodes in V_1 have been aligned. As MI-GRAAL is recommended by GRAAL's authors as a superior solution, this algorithm is included largely because of its historical impact on the network alignment literature.

MI-GRAAL (Kuchaiev and Przulj, 2011), from the family of GRAAL algorithms, combines several different measures of topological similarity along with sequence similarity. The user can decide which topological measures to use in a given alignment. In our benchmarks here, we use the combination of topological similarity measures that is reported to work best: graphlet degree signature distance, degree difference and clustering coefficient difference. When considering whether to align a given pair of nodes, each of these similarity measures is treated as a separate vote for or against aligning the two nodes. MI-GRAAL then uses a seed-and-extend approach to greedily build up an alignment by using the Hungarian algorithm on successive neighborhoods of already-aligned nodes. Notably, MI-GRAAL was one of the first aligners for which sequence similarity data were optional, and even when using only topological information, MI-GRAAL is reportedly able to find alignments accurate enough to reconstruct phylogenetic trees from given PPI networks (Kuchaiev and Przulj, 2011).

C-GRAAL (Memisević and Przulj, 2012) is a more recent algorithm in the GRAAL family. It differs from MI-GRAAL in that it uses only graphlet degree signature and sequence similarity to construct its alignments, and it uses a different heuristic matching algorithm. The matching algorithm works by conducting repeated seed-and-extend iterations that first align the most similar pair and then aligns neighbors of already-matched nodes. When no more unmatched neighbors exist, it finds a new seed from the most similar pair that is still unaligned and repeats the process, until all nodes are aligned. This neighbors-based approach helps to maximize the EC of the alignments produced. It is reported that C-GRAAL obtains lower EC scores than MI-GRAAL but performs better with respect to conserved functional orthology between aligned nodes and finds larger connected shared components (Memisević and Przulj, 2012).

NATALIE 2.0 is a pairwise aligner that uses both topological and sequence similarity information for alignment (El-Kebir *et al*., 2011). Unlike most aligners, which proceed in a two-stage fashion as described in Section 2.1, NATALIE 2.0 formalizes the alignment problem as a Lagrangian relaxation approach to solving an adaptation of an integer linear programming problem, which attempts to optimize topological and sequence similarity of aligned nodes. NATALIE also differs from the other aligners here in that it frames the problem as finding a *partial* function $f: V_1 \rightarrow V_2$, which allows NATALIE to leave some nodes unaligned. We found that NATALIE puts this difference to good use, abstaining from aligning nodes for which it cannot find a good alignment. NATALIE also restricts its alignment by incorporating a user-configurable cut-off for sequence similarity, below which it will not consider mapping two nodes together. This improves the execution speed considerably by allowing the use of the successive shortest paths variant of the Hungarian algorithm, which has better time complexity than the standard version. This program is one of the few available for several operating systems (see our Supplementary Information File for a table of supported operating systems).

GHOST (Patro and Kingsford, 2012) calculates spectral signatures from the Laplacian of the subgraphs around each node of the input graphs. These signatures can be saved for each graph and reused in further alignment runs, which helps to save computation time. The similarity between two nodes is then defined as the distance between the spectral signatures of the nodes. Because these are matrix operations, parallel algorithms are used to decrease GHOST's running time. For the matching stage, GHOST uses a seed-and-extend approach, where the most similar pair of nodes is aligned as the first seed, and then the neighborhood around these nodes is mapped by solving the spectral relaxation of a quadratic assignment problem, which assigns a match confidence to each pair in the neighborhood. The highest confidence match found is aligned and used as the seed for the next iteration. On real-world PPI networks, it is reported that GHOST finds a larger connected common subgraph than MI-GRAAL and does significantly better at matching nodes with shared GO annotations (Patro and Kingsford, 2012).

SPINAL (Aladag and Erten, 2013) is a rather recent pairwise alignment algorithm that claims much better performance in terms of memory usage, speed and accuracy over MI-GRAAL. It uses a two-pass matching algorithm consisting of 'coarse-grained' and 'fine-grained' steps. The coarse-grained step iteratively improves a matrix P of estimated match confidence for each pair of nodes by taking into account the confidence of matching their neighbors that was computed in the previous iteration. After P has converged, the fine-grained stage begins, which uses a seed-and-extend algorithm to construct the alignment. Additionally, on each iteration of the seed-and-extend process, local search is performed to increase the number of conserved interactions directly. The authors report significantly better performance than MI-GRAAL in both runtime and alignment quality. Furthermore, SPINAL has two distinctive modes, 1 and 2, with Mode 1 performing the coarse-grained phase and then simply performing a maximum-weight bipartite matching, whereas Mode 2 performs both the coarse- and fine-grained stages. Because neither mode is singled out as the optimal mode for the program, and they produce different results, we test both.

NETAL (Neyshabur *et al*., 2013) is another recent pairwise algorithm that boasts great improvements in execution speed over older algorithms such as MI-GRAAL. Its performance on noisy data is reportedly improved over MI-GRAAL as well, aligning nearly three times as many

nodes correctly on a network with 5% noise in its edge set (Neyshabur *et al.*, 2013). NETAL is also notable for being available for use online (http://bioinf.modares.ac.ir/software/netal/). NETAL works by recursively defining topological similarity in a manner similar to IsoRank, where nodes are similar if their neighbors are similar. Then, given this topological similarity matrix, it refines an interaction score matrix by estimating how many interactions will be conserved if the given nodes are aligned. The alignment is constructed greedily by mapping the nodes with the best interaction score together, and after each pair of nodes is aligned, the interaction score matrix is updated for selecting the next pair to align. The current version of NETAL uses topological similarity only, but the paper in which it is presented states that a version that uses sequence similarity is forthcoming (Neyshabur *et al.*, 2013).

PINALOG (Phan and Sternberg, 2012) is a pairwise aligner that identifies dense subgraphs, called communities, within the input networks to find regions of similarity between the two networks being aligned. It first finds a mapping from the communities in one graph to the communities in the other and then, for each pair of communities, matches the nodes within them. The community information is the only topological information incorporated. The similarity of two communities is determined by assessing the sum of the sequence similarity of the best alignment created between those two communities by matching their constituent nodes by sequence similarity. Like NETAL, PINALOG is available both through a Web interface (http://www.sbg.bio.ic.ac.uk/pinalog/) and as a standalone executable. PINALOG is reported to find alignments of similar EC as MI-GRAAL, but the alignments found tend to be better matches with respect to functional orthology data (Phan and Sternberg, 2012). Like most aligners, PINALOG uses both topological and sequence similarity information in constructing its alignment, but it automatically determines the trade-off between the two. PINALOG is unique in that it can also use GO annotations as input to guide the alignment process. Because no other aligner has such a feature, we do not test that functionality in our benchmarks.

Most of these programs have a variety of user-controllable parameters that can be used to tweak the behavior of the aligner. In most cases, we used the settings recommended in the tools' documentation. See our Supplementary Information File for details.

3 RESULTS AND DISCUSSION

3.1 Synthetic test results

Benchmarking with the synthetic NAPAbench data produced interesting results, showing a wide diversity in algorithm and software quality. First of all, we must note that GRAAL crashed on one of the CG problems but ran successfully for the other eight tests. When we report averages for this aligner, we average only over the alignments for which it ran to completion. Additionally, we must point out that the running time of these aligners varied greatly. For more information on software stability and running times, please see our Supplementary Information File.

ICS (Fig. 1) is the first metric we consider. NATALIE attains the best score, significantly ahead of GHOST, SPINAL and MI-GRAAL, which are otherwise the aligners with the highest ICS scores. This can partly be explained by NATALIE's alignment strategy, which leaves nodes unaligned when it cannot find a good match. This in turn reduces the size of the denominator in the ICS measure. Aside from NATALIE's high performance on this measure, the results for the other aligners show that C-GRAAL, NETAL, PINALOG and IsoRank have similar performance, but they are still weaker overall.

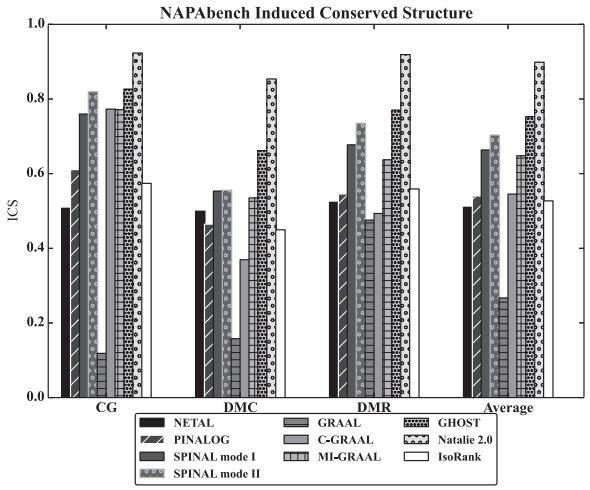

Fig. 1. ICS for all aligners with the NAPAbench benchmark data. For each aligner, we report the average for each of the network evolution models used. We also report the average over all the models. Note: in all of these bar charts, the order of aligners in the legend is the same as the order of the bars in the chart

Next, in Figure 2, we examine how many nodes of the smaller graph are included in the largest shared component found between the two graphs. This metric correlates strongly with ICS on this dataset. MI-GRAAL, SPINAL and GHOST find the largest shared subgraphs overall. C-GRAAL, NATALIE, PINALOG and NETAL, while finding similarly sized subgraphs when compared with each other, lag behind SPINAL and GHOST by a fair margin. Once again, GRAAL trails behind significantly. Notably, NATALIE is not among the top aligners for this metric. This is a result of NATALIE's alignment strategy; aligning fewer nodes overall diminishes the size of the largest shared component.

To assess the biological relevance of the alignments produced, we examine how well the aligners align nodes to orthologs (Fig. 3). Because we are using synthetic data from protein network evolution models, we know which proteins are truly orthologous and can easily calculate the percentage of aligned nodes that have been matched with their orthologs. Here, we see that NATALIE manages astounding performance, with 99% of aligned nodes mapped to true orthologs on average. However, we must reiterate that NATALIE produces partial alignments; on these alignment problems, NATALIE aligned on average 86% of the nodes in the smaller graph and left the rest unaligned. Even taking this into account, the total number of correctly aligned nodes is highest for NATALIE. SPINAL, PINALOG and GHOST are essentially tied for second place with between 75 and 80% of nodes aligned to true orthologs. IsoRank and MI-GRAAL follow slightly behind these, with C-GRAAL yet further behind, and GRAAL performing poorly. This is an extremely important metric, as it represents the ultimate goal of biological network alignment—identifying orthologous proteins.

3.2 Real-world test results

Strikingly, we find that the relative performance of these algorithms is different when tested on real PPI data from the IsoBase

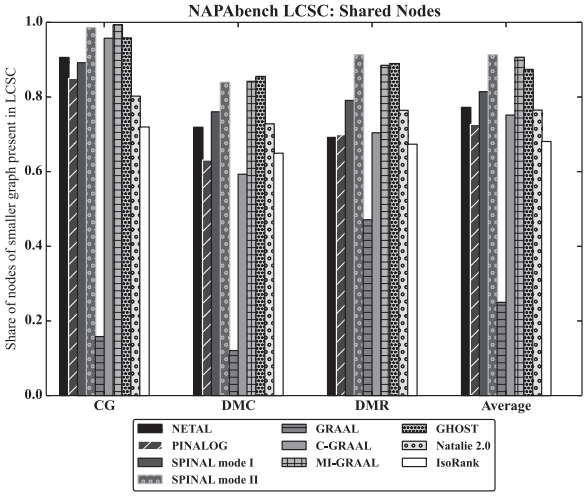

Fig. 2. Number of nodes in the largest common shared component of the aligned networks, reported as the share of the nodes in the smaller network

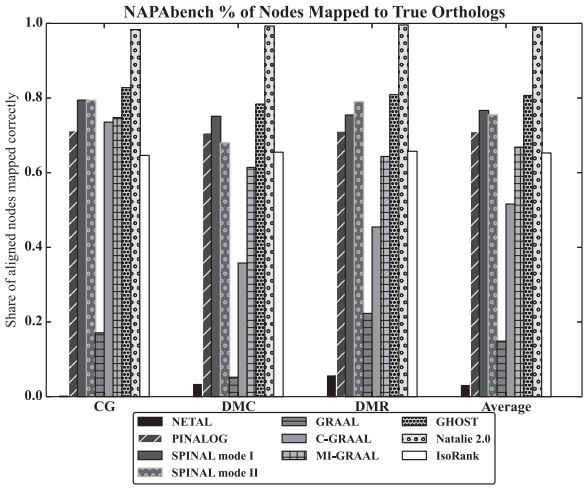

Fig. 3. Average percentage of aligned nodes correctly aligned by each aligner for each of the network evolution models

dataset (Park *et al*., 2011). In these tests, we attempt to align PPI networks from *Caenorhabditis elegans*, *Drosophila melanogaster*, *S.cerevisiae* and *H.sapiens*. For these tests, we get a much wider range of results, and poorer results overall. We suspect that much of this variance is due to the amount of noise present in experimentally derived data.

As before, we experienced difficulties with certain programs crashing. GHOST crashed on two problems. SPINAL Mode 1 crashed on two inputs, and GRAAL and MI-GRAAL crashed on three. C-GRAAL appeared to run to completion for all problems but produced no output for two. Thus, missing bars in these figures imply that the corresponding program failed to run. As before, when averages are reported, they are averaged over only those alignments that the given aligner completed successfully.

On the ICS metric, we see that NETAL outperforms the other algorithms by a fair margin, whereas NATALIE, MI-GRAAL and GRAAL perform similarly (Fig. 4). Notably, NATALIE performs much better with ICS than EC on these tests (see our Supplementary Information). Although NATALIE aligns fewer nodes, those it does align are aligned accurately, into parts of the second graph that they fit well. Oddly, given its performance elsewhere, we see GRAAL performing rather well, at times outperforming C-GRAAL, and in one case outperforming GHOST. This can be attributed to the fact that, like NETAL, GRAAL is not attempting to maximize a trade-off of both topological and sequence similarity but instead attempts to maximize topological fit only.

For the LCSC, NETAL once again leads in the percentage of both nodes and edges of the smaller graph that are found as a connected component in the larger (Fig. 5). In its best alignment, it finds that 90% of the nodes and 32% of the edges in the *D.melanogaster* network are contained in a subgraph of the *H.sapiens* network. MI-GRAAL also produces excellent LCSC scores, and SPINAL and C-GRAAL also perform well. PINALOG, SPINAL in Mode 1, GRAAL, NATALIE and IsoRank all produce alignments with LCSC sizes so small as to be negligible. It is notable that NETAL and GRAAL perform so differently here, as they both attempt to maximize topological similarity alone. Between this result and the ICS results, it seems that NETAL is the superior solution for constructing topology-only alignments.

Next, we consider GOC scores. First, we consider GOC when we include all GO terms at depth 5 from the root of their ontologies (Fig. 6). This gives us a greater number of GO annotations to work with, but because most aligners use sequence similarity to construct the alignment, this inflates scores by including GO terms that were assigned based on sequence similarity themselves. Instead of being the top performer as it was on the topological benchmarks, NETAL's performance here is the poorest. PINALOG and SPINAL mode 1 dominate the chart and IsoRank performs ably, while other alignment algorithms vary widely in performance. For the alignments it successfully produced, GHOST achieves GOC scores significantly lower than PINALOG and SPINAL mode 1, but significantly higher than SPINAL mode 2, GRAAL, MI-GRAAL, C-GRAAL, and NATALIE, which all produce relatively poor GOC scores on most of these tests. The generally lower performance on the *C.elegans* problems appears to be due to the small size and extreme sparsity of the *C.elegans* network in the IsoBase dataset. This network has only half the nodes of the next largest network and only 14.4% of the edges (see our Supplementary Information File for the number of nodes and edges in each network).

We also computed GOC scores using only GO terms with experimental evidence codes. Because, admittedly, GO terms are already transferred successfully on the basis of sequence similarity alone, excluding such terms from our set of evaluation GO terms gives us different information. If GO terms have been attributed to a protein simply due to its sequence similarity to another protein, then we would expect to see such GO terms match when we perform network alignment that uses sequence similarity information. Although it is useful to see in Figure 6 that that occurred, it is also helpful to see that experimentally verified GO annotations are also shared between aligned proteins. We found that with this restricted set of GO terms, all

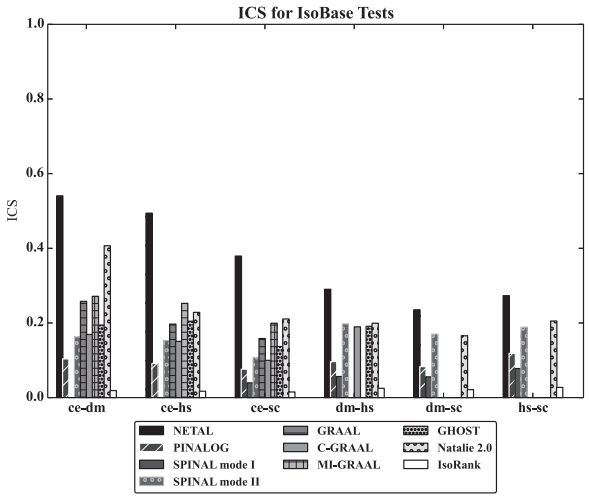

Fig. 4. ICS scores for each of the pairwise alignment tests for the IsoBase PPI networks

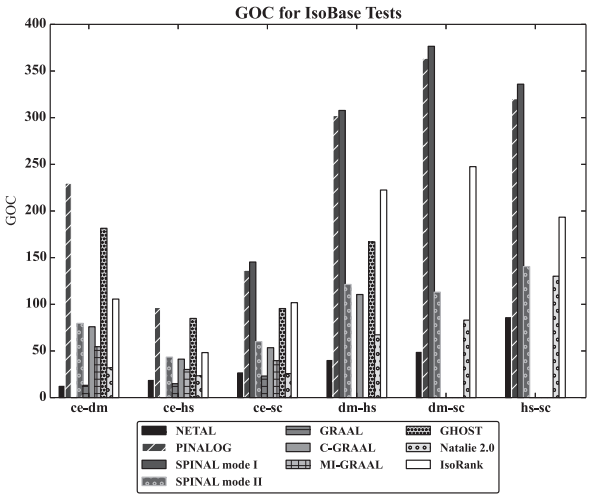

Fig. 6. GOC scores for each alignment of the IsoBase PPI networks

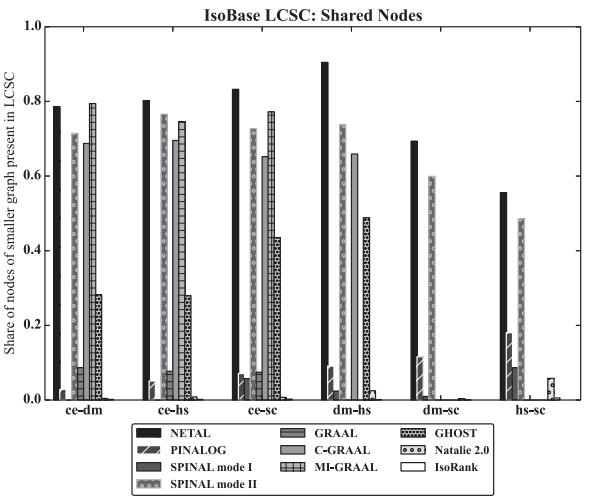

Fig. 5. Share of nodes in the smaller network in the LCSC found for each alignment

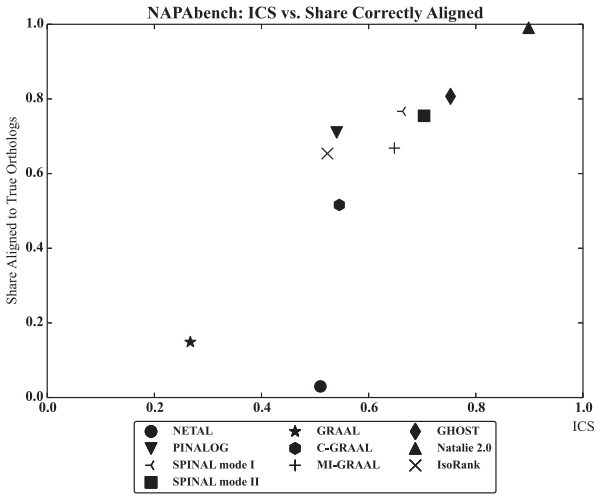

Fig. 7. Plot of average percentage of nodes correctly aligned versus integrated conserved structure score for all aligners with the NAPAbench benchmark data

GOC scores are much lower, but the relative ranking of the algorithms remains the same. These results are similar to those in the paper where GOC is introduced (Aladag and Erten, 2013). Because the relative performance of aligners is not different when counting GOC in this way, we include those results in our Supplementary Information File only.

3.3 Comparing overall performance

Looking at results from both synthetic and real-world benchmarks, we are left with a dizzying variety of metrics. As particularly observed in the real-world data, some algorithms perform extremely well with respect to one metric, while performing abysmally with respect to another. This is a consequence of the two metrics by which these aligners perform their matching. Because they use both topological similarity information as well as biological similarity information to perform their alignments, each aligner must be designed to produce a good trade-off between these two goals. Following Patro and Kingsford (2012), we plot the topological and biological quality of these aligners against each other, so as to reveal the aligners that best jointly optimize topological and biological similarity in producing their alignment.

For the NAPAbench testing data, we plot ICS against the percentage of nodes aligned to true orthologs (Fig. 7). Doing so, we can quickly pick out the best performing algorithms by looking at which ones tend toward the upper right of the graph. We see that NATALIE performs best, with GHOST and SPINAL yielding similar results to each other and lagging behind NATALIE. MI-GRAAL, PINALOG and C-GRAAL also perform reasonably well, but NETAL and GRAAL show poorer alignment results.

The results for the real-world PPI networks, on the other hand, are more ambiguous (Fig. 8). First of all, we must emphasize that this scatter plot cannot be directly compared with the

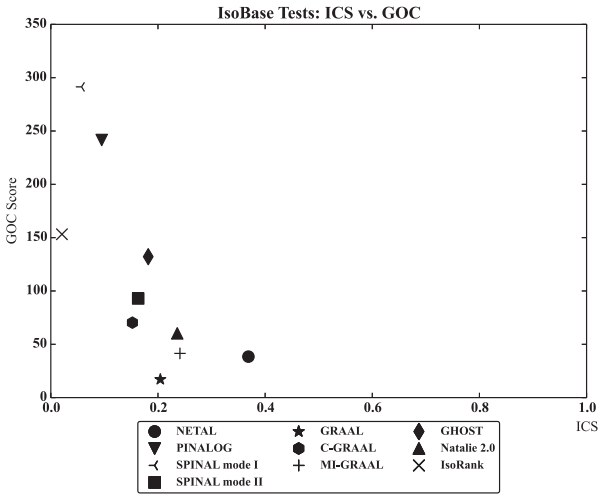

Fig. 8. Plot of the average GOC versus average integrated conserved structure score for all aligners with the IsoBase network data

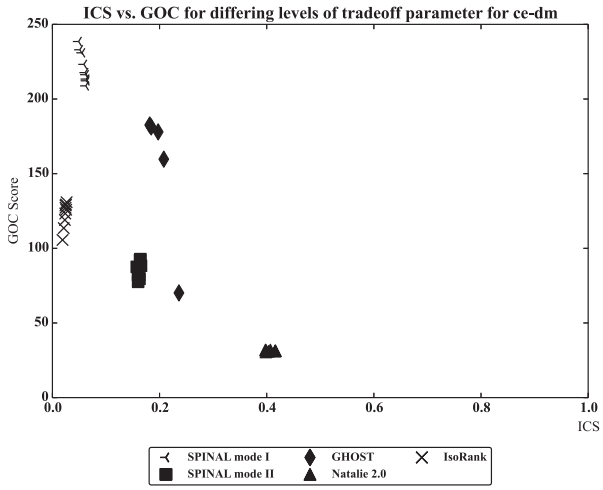

Fig. 9. Plot of the GOC versus integrated conserved structure score for aligners with user-adjustable trade-off between topological and biological similarity. We explored a range of trade-off levels for these aligners on the *C.elegans–D.melanogaster* alignment problem

NAPAbench plot because the y-axis here is our GOC score, whereas with NAPAbench, we report the percentage of nodes aligned to true orthologs. However, we can still compare these two datasets with respect to the relative performance of different aligners and examine the trade-offs between the topological and biological quality of the resulting alignments. Although SPINAL Mode 1 manages the highest GOC score, and NETAL attains the highest ICS, those that manage a decent trade-off between the two tend to perform poorly overall. If we must declare a best performer for these benchmarks, we might, hesitantly, point out GHOST, NATALIE, MI-GRAAL and SPINAL Mode 2 as aligners that find alignments with decent topological and biological similarity results. However, compared with the synthetic data of the NAPAbench test set, all these algorithms experience much greater difficulty with these PPI networks derived from experimental data.

The difference between these aligners' results on NAPAbench and IsoBase benchmarks is striking enough to deserve additional comment. All aligners perform much better on NAPAbench overall. Although the most likely difference is the fact that NAPAbench networks contain no spurious or missing edges, we must also keep in mind the possibility that the PPI network evolution models used to produce NAPAbench's networks may also contribute to the differences in performance. It is possible that the networks generated by these models are easier for existing aligners to align compared with real PPI networks. NAPAbench's networks tend to be sparser than IsoBase's as well (see the Supplementary Information File), which may also affect the results. The relative difference in aligner performance on these datasets is also curious and suggests differences in the robustness of these algorithms to noise.

Several aligners support adjusting the relative importance of topological and sequence similarity when constructing their alignment. Of the aligners we benchmarked, SPINAL, GHOST, IsoRank and NATALIE support this functionality. We ran these aligners with differing settings to the trade-off parameter repeatedly on the *C.elegans–D.melanogaster* alignment problem (Fig. 9). The results are similar overall to Figure 8. We see a rough Pareto front consisting of results from SPINAL Mode 1, GHOST and NATALIE. SPINAL Mode 2 results in lower performance than GHOST, and IsoRank is dominated by both GHOST and SPINAL Mode 1. It is important to point out that even when we vary the trade-off of these aligners through their whole range, the absolute differences in the alignments produced by a single aligner are small. We do not see any one aligner that covers a large portion of the frontier but instead see that each aligner covers a small section of it. This indicates that, to get a good idea of the variety of alignments possible between two species, it is necessary to use multiple aligners and compare their results.

4 CONCLUSION

In the last few years, many new algorithms for pairwise global network alignment have appeared, but few of them had been compared directly. By benchmarking these algorithms on synthetic and real-world data, we have shed light on the relative behavior and performance of these algorithms. The results have been surprising and show that these algorithms can behave very differently on different datasets. Several algorithms perform well, but the great differences in behavior from one test to another, and the tendency for some of the existing programs to crash, makes it difficult to recommend any one aligner. Those who want to use network alignment as part of their work should try a few of the better performing algorithms found here and see how their performance compares for the particular dataset in question. We suggest SPINAL, NATALIE and PINALOG as good aligners to try at first; they produce their alignments quickly while giving competitive results. GHOST is also an excellent performer, but its high memory requirements, slow speed and crashes render it more difficult to use. C-GRAAL and MI-GRAAL may be worthwhile in some situations, although other aligners perform better in general. All members of the GRAAL family share a bottleneck in the unpredictably slow graphlet-counting step, and this step also tends to crash. Because

NETAL does not currently use sequence similarity in constructing alignments, its uses are more limited, but it can produce much higher ICS scores and larger connected components than other aligners on a given dataset, which may be useful in understanding the relative performance of other aligners. We cannot recommend GRAAL and IsoRank, as they are bested by many aligners on both GOC and ICS scores. This is understandable, given that most of these other aligners were benchmarked against GRAAL and IsoRank when they were being designed.

We have also found a number of issues that future investigators of network alignment must focus on. First and foremost, code and documentation quality must be improved for a tool to see widespread use. Many of these programs crash often, are poorly documented or run slowly. Sensitivity to noise is also an issue. Given the drastic differences in aligner performance between the noise-free synthetic data and the noisy real-world data, it is clear that future alignment algorithms must become even more robust to such difficulties. Last of all, we have seen that many aligners perform well at yielding either good topological or good biological matches, but few do both well. This must become a higher priority in aligner design.

ACKNOWLEDGEMENTS

We would like to thank Gunnar Klau and Mohammed El-Kebir for their assistance with NATALIE, Robert Patro for his help with GHOST, Shahriar Arab for sending us the NETAL source and Rohit Singh for sending us the original pairwise IsoRank binary. We also thank four anonymous reviewers for helpful comments and suggestions.

Funding: The Biofrontiers Center at the University of Colorado, Colorado Springs.

Conflict of Interest: none declared.

REFERENCES

Aladag,A.E. and Erten,C. (2013) SPINAL: scalable protein interaction network alignment. *Bioinformatics*, **29**, 917–924.

Altschul,S.F. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Ashburner,M. *et al.* (2000) Gene Ontology: tool for the unification of biology. The gene ontology consortium. *Nat. Genet.*, **25**, 25–29.

Atias,N. and Sharan,R. (2012) Comparative analysis of protein networks: hard problems, practical solutions. *Commun. ACM*, **55**, 88–97.

Bayati,M. *et al.* (2013) Message-passing algorithms for sparse network alignment. *ACM Trans. Knowl. Discov. Data*, **7**, 3.

Berg,J. and Lässig,M. (2004) Local graph alignment and motif search in biological networks. *Proc. Natl Acad. Sci. USA*, **101**, 14689–14694.

Chindelevitch,L. *et al.* (2013) Optimizing a global alignment of protein interaction networks. *Bioinformatics*, **29**, 2765–2773.

Cook,S.A. (1971) The complexity of theorem-proving procedures. In: *Proceedings of the Third Annual ACM Symposium on Theory of Computing*. pp. 151–158.

El-Kebir,M. *et al.* (2011) Lagrangian relaxation applied to sparse global network alignment. In: *Pattern Recognition in Bioinformatics, Vol. 7036 of Lecture Notes in Computer Science*. Springer, Berlin, Heidelberg, pp. 225–236.

Flannick,J. *et al.* (2006) Graemlin: general and robust alignment of multiple large interaction networks. *Genome Res.*, **16**, 1169–1181.

Flannick,J. *et al.* (2008) Automatic parameter learning for multiple network alignment. In: *Proc. Int. Conf. Research in Computational Molecular Biology*. pp. 214–231.

Huang,Q. *et al.* (2012) CNetA: network alignment by combining biological and topological features. In: *2012 IEEE 6th International Conference on Systems Biology (ISB)*. pp. 220–225.

Kalaev,M. *et al.* (2008) NetworkBLAST: comparative analysis of protein networks. *Bioinformatics*, **24**, 594–596.

Kalaev,M. *et al.* (2009) Fast and accurate alignment of multiple protein networks. *J. Comp. Biol.*, **16**, 989–999.

Kelley,B.P. *et al.* (2004) PathBLAST: a tool for alignment of protein interaction networks. *Nucleic Acids Res.*, **32**, W83–W88.

Khan,A.M. *et al.* (2012) A multithreaded algorithm for network alignment via approximate matching. In: *2012 International Conference for High Performance Computing, Networking, Storage and Analysis*. pp. 1–11.

Kim,W.K. and Marcotte,E.M. (2008) Age-dependent evolution of the yeast protein interaction network suggests a limited role of gene duplication and divergence. *PLoS Comput. Biol.*, **4**, e1000232.

Kollias,G. *et al.* (2013) A fast approach to global alignment of protein-protein interaction networks. *BMC Res. Notes*, **6**, 35.

Koyutürk,M. *et al.* (2006) Pairwise alignment of protein interaction networks. *J. Comp. Biol.*, **13**, 182–199.

Kpodjedo,S. *et al.* (2014) Using local similarity measures to efficiently address approximate graph matching. *Discrete Appl. Math.*, **164**, 161–177.

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

Kuchaiev,O. *et al.* (2010) Topological network alignment uncovers biological function and phylogeny. *J. R. Soc. Interface*, **7**, 1341–1354.

Liang,Z. *et al.* (2006) Comparison of protein interaction networks reveals species conservation and divergence. *BMC Bioinformatics*, **7**, 457.

Liao,C.-S. *et al.* (2009) IsoRankN: spectral methods for global alignment of multiple protein networks. *Bioinformatics*, **25**, i253–i258.

Memisević,V. and Przulj,N. (2012) C-GRAAL: common-neighbors-based global GRAph ALignment of biological networks. *Integr. Biol.*, **4**, 734–743.

Milenković,T. *et al.* (2010) Optimal network alignment with graphlet degree vectors. *Cancer Inform.*, **9**, 121.

Neyshabur,B. *et al.* (2013) NETAL: a new graph-based method for global alignment of protein–protein interaction networks. *Bioinformatics*, **29**, 1654–1662.

Park,D. *et al.* (2011) IsoBase: a database of functionally related proteins across PPI networks. *Nucleic Acids Res.*, **39** (**Suppl. 1**), D295–D300.

Pastor-Satorras,R. *et al.* (2003) Evolving protein interaction networks through gene duplication. *J. Theor. Biol.*, **222**, 199–210.

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

Phan,H.T. *et al.* (2012) Aligning protein-protein interaction networks using random neural networks. In: *2012 IEEE International Conference on Bioinformatics and Biomedicine*. pp. 1–6.

Phan,H.T. and Sternberg,M.J. (2012) PINALOG: a novel approach to align protein interaction networks—implications for complex detection and function prediction. *Bioinformatics*, **28**, 1239–1245.

Pržulj,N. (2007) Biological network comparison using graphlet degree distribution. *Bioinformatics*, **23**, e177–e183.

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

Sahraeian,S.M.E. and Yoon,B.-J. (2013) SMETANA: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. *PLoS One*, **8**, e67995.

Sharan,R. *et al.* (2005) Conserved patterns of protein interaction in multiple species. *Proc. Natl Acad. Sci. USA*, **102**, 1974–1979.

Shih,Y.-K. and Parthasarathy,S. (2012) Scalable global alignment for multiple biological networks. *BMC Bioinformatics*, **13** (**Suppl. 3**), S11.

Singh,R. *et al.* (2008) Global alignment of multiple protein interaction networks with application to functional orthology detection. *Proc. Natl Acad. Sci. U.S.A.*, **105** (35), 12763–12768.

Tian,W. and Samatova,N.F. (2013) Global alignment of pairwise protein interaction networks for maximal common conserved patterns. *Int. J. Genomics*, **2013**, 670623.

Todor,A. *et al.* (2013) Probabilistic biological network alignment. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, **10**, 109–121.

Vázquez,A. *et al.* (2002) Modeling of protein interaction networks. *Complexus*, **1**, 38–44.