Bioinformatics, 31(12), 2015, 1988–1998
doi: 10.1093/bioinformatics/btv063
Advance Access Publication Date: 9 February 2015
Original Paper

Systems biology

A multiobjective memetic algorithm for PPI network alignment

Connor Clark* and Jugal Kalita

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

*To whom correspondence should be addressed.

Associate Editor: Igor Jurisica

Received on July 28, 2014; revised on December 10, 2014; accepted on January 27, 2015

Abstract

Motivation: There recently has been great interest in aligning protein–protein interaction (PPI) networks to identify potentially orthologous proteins between species. It is thought that the topological information contained in these networks will yield better orthology predictions than sequence similarity alone. Recent work has found that existing aligners have difficulty making use of both topological and sequence similarity when aligning, with either one or the other being better matched. This can be at least partially attributed to the fact that existing aligners try to combine these two potentially conflicting objectives into a single objective.
Results: We present *Optnetalign*, a multiobjective memetic algorithm for the problem of PPI network alignment that uses extremely efficient swap-based local search, mutation and crossover operations to create a population of alignments. This algorithm optimizes the conflicting goals of topological and sequence similarity using the concept of Pareto dominance, exploring the tradeoff between the two objectives as it runs. This allows us to produce many high-quality candidate alignments in a single run. Our algorithm produces alignments that are much better compromises between topological and biological match quality than previous work, while better characterizing the diversity of possible good alignments between two networks. Our aligner's results have several interesting implications for future research on alignment evaluation, the design of network alignment objectives and the interpretation of alignment results.
Availability and Implementation: The C++ source code to our program, along with compilation and usage instructions, is available at https://github.com/crclark/optnetaligncpp/
Contact: connor.r.clark@gmail.com
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

As the sizes of known protein–protein interaction (PPI) networks grow, so does interest in analyzing them. One of the more ambitious efforts in this area is aligning the PPI networks of two different species, with the goal of identifying orthologous proteins as well as shared pathways and complexes that hint at the PPI network of a common ancestor. There has already been a good deal of progress in this area, with past work reporting success in finding large shared subnetworks between *Saccharomyces cerevisiae* and *Homo sapiens*, as well as success in reconstructing phylogeny based on network overlap discovered by an aligner (Kuchaiev and Przulj, 2011; Kuchaiev *et al.*, 2010).

Despite these successes, PPI network alignment is a young research area. The incomplete, noisy nature of existing PPI networks makes alignment difficult, with existing aligners performing dramatically better on noise-free synthetic networks compared with noisy real-world networks (Clark and Kalita, 2014). Furthermore, it has been found that the two objectives of biological and topological fit that these aligners optimize conflict to a larger extent than previously realized. Different aligners construct alignments that are dramatically different, with some optimizing topological fit at the expense of sequence similarity, while others do the opposite. While some of these aligners provide a user-controlled parameter for adjusting between optimizing one objective over the other, even when this parameter is adjusted as widely as possible, existing aligners only output alignments in small, non-overlapping regions of objective space that are often quite distant from each other (Clark and Kalita, 2014; Patro and Kingsford, 2012). Lastly, these aligners tend to produce very different alignments, sometimes agreeing on as few as 5% of their total aligned pairs (Patro and Kingsford, 2012), which makes it necessary for the user to run many different aligners to get an idea of the many alignments possible. The difficulties with the conflict between topological and biological fit, and the disparate results of existing aligners, show that the problem of PPI network alignment is far from solved.

To this end, we introduce a multiobjective memetic algorithm, Optnetalign (*Opt*imizing *Net*work *Align*er), for performing network alignment. Unlike most existing aligners which use custom-built estimators of node similarity which don't always have a clear connection to final alignment quality, we use a multiobjective memetic algorithm (Deb, 2001), which can produce a large number of diverse, high quality alignments in a single run. Our approach is able to better optimize both the topological and sequence similarity of the proteins aligned, and instead of arbitrarily emitting a single alignment at one point in objective space, Optnetalign produces a Pareto front of alignments that characterize the wide variety of possible tradeoffs between the conflicting objectives of network alignment. This tends to produce a wider variety of alignments than using all previously created aligners together, and these alignments tend to be comparable to or of better quality than those produced by previous aligners.

2 Methods

2.1 PPI networks and the goal of alignment

A PPI network is a graphical representation of all the proteins in a given organism, where the nodes of the network represent proteins, and a link between two proteins indicates that they interact to perform some biological function. Protein interaction is measured through a number of different experimental methods, with the most common being coimmunoprecipitation (Aebersold and Mann, 2003). A large amount of effort has gone into collecting and curating the results of many protein interaction experiments, and the results are stored in databases such as DIP (Xenarios *et al.*, 2002), Isobase (Park *et al.*, 2011), BIOGrid (Chatr-aryamontri *et al.*, 2013) and STRING (Franceschini *et al.*, 2013). A related research effort in this area is the Gene Ontology (GO), a large database annotating genes and proteins with information about their function, the biological processes in which they participate and the cellular components in which they are found (Ashburner *et al.*, 2000). The GO database is constantly being updated and revised as new discoveries are made. An important area of research is the transfer of GO annotations from proteins in one species to orthologous proteins in another. Some species have more complete GO annotation data than others, so it is desirable to find a way to accurately and automatically predict GO annotations for unannotated proteins in less-studied species.

PPI network alignment is a way to use both the sequence similarity between proteins, as well as the topology of PPI networks, to find likely orthologous proteins between species. The goal is to create algorithms that can automatically find likely subnetworks of two species' PPI networks that are common to the two species, and from there form hypotheses about protein function in species for which less information exists. A single run of an alignment program can generate thousands of candidate ortholog pairs.

2.2 Pairwise global alignment

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 an edge (u, v) in E_1 (or E_2) indicates that the two proteins represented by u and v interact in G_1 (or G_2). We assume, without loss of generality, that $|V_1| \leq |V_2|$.

The problem of pairwise global alignment is to find a one-to-one function (In the definitions that follow, we will employ a shorthand where f is also applied to edges. In such cases $f((u, v))$ is shorthand for $(f(u), f(v))$. This makes several definitions significantly easier to read.) $f : V_1 \rightarrow V_2$ that maps each node in V_1 to the node it best matches in V_2, according to the topology of the network around the node and the sequence similarity of the proteins represented by the nodes. The various methods for determining the 'best' match will be described further below. The resulting alignment f shows us a subnetwork that the two networks have in common. The problem of global alignment is equivalent to the subgraph isomorphism problem, which is NP-Complete (Cook, 1971), so aligners produce approximate solutions using a variety of techniques that will be discussed below.

2.3 Evaluating alignment quality

Once an alignment has been constructed, we must measure the extent to which it has satisfied the two objectives of maximizing both the topological fit between the two networks and the number of likely orthologs that have been matched. These objectives are intended to measure the extent to which we have identified overlapping pathways and orthology relationships respectively. Three closely related metrics for topological fit have been used in the global network alignment literature, which present different ways of evaluating how well one network has been matched to the other: EC (Kuchaiev *et al.*, 2010), ICS (Patro and Kingsford, 2012) and S^3 (Saraph and Milenković, 2014). Of these, EC is the most universally used, while ICS and S^3 are newer refinements of EC. These three metrics are ratios taking values between 0 and 1, with higher values being better. They all share the same numerator, which represents the number of edges that have been conserved by the alignment f.

$$\text{conserved}(f) = |f(E_1) \cap E_2| \tag{1}$$

They differ, however, in their denominators. EC simply divides by $|E_1|$, showing us how many of the edges in the first network have been successfully matched. ICS refines this by instead dividing by $|E_{G_2[f(V_1)]}|$, the number of edges in the induced subgraph of G_2 that f maps to. This effectively penalizes alignments from sparse graph regions to dense graph regions. S^3 is a further refinement of ICS, which penalizes dense-to-sparse as well as sparse-to-dense alignments by dividing by $|E_1| + |E_{G_2[f(V_1)]}| - |f(E_1) \cap E_2|$. While we previously evaluated existing algorithms using ICS (Clark and Kalita, 2014), we switch to S^3 in this article because, in contrast to the aligners we evaluated previously, Optnetalign is able to 'cheat' when optimizing ICS by minimizing the denominator without increasing the numerator, and MAGNA exhibits this behavior as well (Saraph and Milenković, 2014). S^3 properly penalizes this behavior.

We also use a measure of agreement derived from biological knowledge. The true alignment between two PPI networks is at least partially unknown, so one cannot simply report the percentage of nodes mapped to their true orthologs, except when using synthetic benchmark data. However, we can at least verify that the proteins mapped to one another are similar in function. All the literature on PPI network alignment makes use of GO annotations to evaluate the accuracy of their alignments, by comparing the similarity of GO annotations between aligned proteins. Several papers simply report the percentage of aligned pairs that share more than k GO annotations (Kuchaiev and Przulj, 2011; Memišević and Pržulj, 2012; Neyshabur *et al.*, 2013), while others use ontological similarity methods (Guzzi and Mina, 2014; Patro and Kingsford, 2012). We use a Jaccard index, introduced by Aladag and Erten (2013). In the context of network alignment, this measure is called GO consistency, and is defined as

$$GOC(u,v) = \frac{|GO(u) \cap GO(v)|}{|GO(u) \cup GO(v)|} \quad (2)$$

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. We restrict this to GO terms at distance five from the root of the ontology, so as not to confuse the analysis by using GO terms of differing specificity. We then report the sum of the GO consistency over all alignment pairs for each alignment produced.

To further ease comparison with existing aligners, we evaluate on the same dataset of Isobase networks as we used in our previous work (Clark and Kalita, 2014), which combines PPI information from DIP (Xenarios *et al.*, 2002), BIOGrid (Chatr-aryamontri *et al.*, 2013), and the Human Protein Reference Database (Prasad *et al.*, 2009), three curated sources of PPI data derived from a variety of experimental methods (for more information on the Isobase networks, see our Supplementary Information File). The only differences are that we now use the S^3 metric instead of ICS, and, since the synthetic dataset NAPAbench (Sahraeian and Yoon, 2012) did not exhibit the same tradeoff between ICS and biological similarity that is exhibited in the real-world Isobase networks (Clark and Kalita, 2014), we evaluate primarily on the Isobase networks, using the NAPAbench data to verify the correctness of our alignments, since the true alignments for NAPAbench are known.

Both topological similarity techniques and sequence similarity have advantages and disadvantages. It has been argued that over-reliance on topological similarity can be misleading, since 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, since the actual level of sequence similarity between two proteins that serve a similar function can vary (Chindelevitch *et al.*, 2013). Previous work has found (Clark and Kalita, 2014; Patro and Kingsford, 2012) that existing aligners uncover a tradeoff between topological fit and biological fit when constructing alignments—it is not possible to jointly maximize both. This tradeoff can be quite dramatic. For example, on the *Caenorhabditis elegans*–*Drosophila melanogaster* alignment problem, NETAL (Neyshabur *et al.*, 2013) achieves an S^3 of 0.41 and a GOC of 12, while PINALOG (Phan and Sternberg, 2012) manages a S^3 of 0.065 and a GOC of 230. When existing aligners are all plotted for one of these problems (as in Fig. 3 later), we see a distinctive curve of tradeoffs between these two objectives. We also find that these aligners produce alignments in non-overlapping regions of the objective space, even for those aligners that provide a parameter for controlling the tradeoff between the two objectives. As we explain further below, ours is the only aligner that was designed to explicitly explore the possible tradeoffs between these two objectives, and to output many alignments for further analysis.

2.4 Existing alignment algorithms

One of the first algorithms for global network alignment was IsoRank (Singh *et al.*, 2008). This was then extended into a version that could align many networks simultaneously, called IsoRankN (Liao *et al.*, 2009). These two work by computing a similarity measure similar to Google's PageRank algorithm, and this is combined with BLAST bit scores to find a mapping. Another notable set of aligners is the GRAAL family (Kuchaiev and Przulj, 2011; Kuchaiev *et al.*, 2010; Memišević and Pržulj, 2012; Milenković *et al.*, 2010). These work by counting small induced subgraphs called 'graphlets' to compute pairwise node similarity. They differ mainly in what additional similarity measures they use, and in their matching stages. NATALIE 2.0 frames alignment as a relaxation of an integer linear programming problem (El-Kebir *et al.*, 2011). GHOST uses methods from spectral graph theory to estimate node similarity (Patro and Kingsford, 2012). PINALOG recursively maps networks together by first matching dense subgraphs, and then matching the nodes within those subgraphs (Phan and Sternberg, 2012). NETAL uses topological data exclusively, iteratively computing match confidence as the expected number of conserved edges (Neyshabur *et al.*, 2013). SPINAL uses both coarse- and fine-grained matching steps to iteratively improve an alignment (Aladag and Erten, 2013). PISwap (Chindelevitch *et al.*, 2013) first creates a maximum-weight match by bit score, and then improves the topological fit of the alignment through swap-based local search. MAGNA (Saraph and Milenković, 2014) is the first aligner to use a genetic algorithm, and uses a crossover-only design.

Most aligners can be categorized as belonging to one of two classes. The first class, which has been much more popular, is what we call *two-stage* alignment. Such aligners first compute some estimate of the similarity of each pair of nodes in $V_1 \times V_2$ using some topological similarity metric and (in most cases) BLAST bit scores or E-values, and then use this similarity matrix to create a maximum-weight bipartite matching between the nodes of V_1 and V_2. A simple strategy for this matching is the Hungarian algorithm, which produces an optimal matching (Chindelevitch *et al.*, 2013; Milenković *et al.*, 2010). However, because the similarity scores used 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 (Kuchaiev *et al.*, 2010), MI-GRAAL (Kuchaiev and Przulj, 2011), and GHOST (Patro and Kingsford, 2012), use variants on a 'seed-and-extend' method, where the most similar pair of nodes is aligned first, and then nodes neighboring that pair are matched.

The second, and much smaller class, that is sometimes used in network alignment are *search-based* aligners. These aligners use heuristic or metaheuristic algorithms to continuously refine a single alignment or a population of alignments. With such algorithms, the alignment objectives that these aligners are trying to optimize are computed continuously *as the alignment is constructed*, instead of being measured afterward as they are in the two-stage aligners. Only a handful of search-based aligners exist. These are SIM-T (Kpodjedo *et al.*, 2014), which uses tabu search, where 'moves' are additions and removals of aligned pairs from the alignment; PISwap (Chindelevitch *et al.*, 2013), which uses the 3-opt heuristic to interchange the assignments of three aligned pairs at each step; and MAGNA (Saraph and Milenković, 2014), which uses a genetic algorithm that attempts to maximize either EC, ICS or S^3. SPINAL (Aladag and Erten, 2013) also performs some amount of swap-based hill climbing in its 'fine-grained' matching stage. Outside of the biological network alignment literature, the use of metaheuristic search for graph matching has been researched as well, but these approaches are less relevant as they only optimize topological fit and are applied to graphs much smaller than those seen in bioinformatics (Barecke and Detyniecki, 2007; Cicirello and Smith, 2000; Cross *et al.*, 1997; Lipets *et al.*, 2009).

Optnetalign falls into the search-based category, because we believe the two-stage paradigm has a number of severe weaknesses. First of all, two-stage aligners typically introduce both a novel way of computing pairwise node similarity along with a new matching algorithm. Little work exists on mixing-and-matching approaches to the two stages to discover whether the first or second stage is more responsible for the results of a given aligner. To our knowledge, the only works attempting this compared only MI-GRAAL to IsoRankN (Milenković *et al.*, 2013) and MI-GRAAL to GHOST (Crawford *et al.*, 2014). Since two-stage algorithms do not track alignment quality as they construct their alignments, it is very difficult to understand what portions of a two-stage algorithm are responsible for the quality of the results. This makes it difficult to create a new algorithm of this type. As Saraph and Milenković (2014) recently observed, these two-stage aligners, while hoping to optimize metrics such as EC and some metric of GO term agreement, actually optimize ad-hoc similarity scores that have no clearly established relation to standard alignment evaluation metrics. The existing literature provides little explanation of why or whether such similarity scores would be expected to work. In contrast, search-based aligners focus on maximizing the actual alignment objectives that are currently accepted in the alignment literature.

A problem shared by all existing aligners is that they do not explicitly approach the problem of network alignment as a multiobjective one. It has become increasingly clear that there is a tradeoff between our alignment objectives, as we discussed above. However, the best any existing aligner does to address this problem is providing users with a parameter α, which gives the user rough control over the weight of biological or topological similarity in constructing an alignment, with some equation similar to

$$sim(u,v) = \alpha \times t(u,v) + (1-\alpha) \times b(u,v) \tag{3}$$

where $sim(u, v)$ is the aligner's estimated overall similarity between nodes u and v and $t(u, v)$ and $b(u, v)$ are the aligner's estimated topological and sequence similarity of the two nodes respectively. This is insufficient. Not only does this approach requires trial-and-error experiments on the part of the user, and complete recalculation of the alignment each time α is adjusted, but existing aligners only cover a narrow section of the objective space, even when varying α as widely as possible. Optnetalign produces a variety of solutions more diverse than existing aligners combined, and does so in a single run without extra effort on the part of the user.

2.5 Multiobjective optimization

Metaheuristics are a family of optimization techniques for problems that cannot be framed in a way amenable to standard optimization techniques, such as when objective functions are discontinuous, not differentiable or not functions of real numbers. In PPI network alignment, our goal is to maximize several objectives that are a function of a network alignment—the problem must first be relaxed for standard optimization techniques to work (El-Kebir *et al.*, 2011). Metaheuristics are thus a natural fit for network alignment. Genetic algorithms are a particular family of metaheuristics inspired by biological evolution (Michalewicz and Fogel, 2004). Initially, a population of candidate solutions are generated at random. These solutions are ranked by their fitness, and the best of them are allowed to 'breed' to produce a new population of solutions. The new population undergoes mutation with some small probability, and then the process begins again, until an acceptable solution has been found (Floreano and Mattiussi, 2008). This approach is applicable to many problems, since it only requires a way of scoring the 'fitness' of candidate solutions, and a data structure representing solutions for which we can define a useful crossover operation (for combining two existing solutions) and a mutation operation (for randomly modifying solutions). Memetic algorithms are a generalization of standard genetic algorithms that additionally incorporate a local search heuristic to further improve their results (Blum *et al.*, 2011). Genetic algorithms and local search heuristics are complementary—the former explores large spaces quickly through crossover and mutation, while the latter is particularly effective at refining solutions that have been found.

While genetic algorithms have been used since the 1970's (e.g. Holland, 1975), variants capable of efficiently optimizing multiobjective problems have only become a focus of research more recently, with algorithms such as NSGA-II (Deb *et al.*, 2002) and SPEA2 (Zitzler *et al.*, 2001) becoming immensely popular. Other local search approaches to multiobjective optimization have had great success, as well (Corne *et al.*, 2000; Czyżżak and Jaszkiewicz, 1998; Knowles and Corne, 2000). Multiobjective memetic algorithms have also been developed in recent years (Knowles and Corne, 2005; Knowles *et al.*, 2001).

The key difference between single objective metaheuristics such as MAGNA (Saraph and Milenković, 2014) and multiobjective metaheuristics such as Optnetalign is that the latter can optimize several objectives, even if they conflict, and then output a representative set of solutions that are optimal tradeoffs between the conflicting objectives. Standard genetic algorithms represent the fitness of a solution as a single number, often by creating a linear combination of multiple objectives using an equation similar to Equation 3. Then, solution x is preferred to solution y if fitness$(x) >$ fitness(y). However, if x has a high topological fit and a low biological fit, while y has a low topological fit and a high biological fit, it may be that neither solution is clearly better. This information is lost using the approach of Equation 3. Multiobjective optimization replaces the greater-than relation with the subtler relation of *Pareto dominance*, which operates on a vector of fitness values. We now introduce several new terms used in multiobjective optimization (we refer the reader to e.g. Deb, 2001 or Zhou *et al.*, 2011 for a more thorough description of these concepts than space permits here). The first is *Pareto optimality*: a solution to a multiobjective optimization problem is Pareto optimal if further increasing one of the objectives would require decreasing one or more of the other objectives. Such solutions represent optimal tradeoffs between the given objectives. The second concept is *Pareto dominance*: solution x dominates solution y if x is at least as good as y with respect to all objectives, and strictly better than y with respect to at least one objective. We prefer a solution x to a solution y if x Pareto dominates y, and we are indifferent between x and y if neither dominates the other. This definition leads to the possibility that a set of solutions may be non-dominated with respect to each other. Multiobjective genetic algorithms work by selecting non-dominated members of the population for reproduction at every generation, in the hopes of approximating the set of Pareto optimal solutions. This approximate Pareto optimal set will, in turn, allow us to characterize the *Pareto front* of the objective space—the boundary between attainable and unattainable solutions. Multiobjective genetic algorithms output a wide variety of representative solutions from the Pareto optimal set. Since these solutions are all Pareto optimal, we cannot *a priori* prefer one of them over another, and a human decision maker must use some other information or subjective preference to decide which of these solutions to keep. In the case of PPI network alignment, this means that our algorithm produces a number of alignments with different tradeoffs between topological and biological fit. These alignments can then be studied further to better understand the relationship between the two networks under consideration.

2.6 Our algorithm

We present a multiobjective memetic algorithm that discovers a representative set of non-dominated alignments using crossover, mutation and swap-based hill climbing. Our algorithm maintains a population of non-dominated alignments and attempts to improve them. The algorithm initializes with randomly-constructed alignments. On each iteration, Optnetalign selects random members of the population, performs swap-based crossover and mutation, followed by swap-based hill climbing. We randomly select either hill climbing that improves one objective while ignoring the others, or hill climbing that only makes moves that do not worsen any objective. The resulting alignment is then placed in the population if it is not Pareto dominated by any other population member, and alignments it Pareto dominates are removed from the population. This continues until a user-specified time limit is reached, after which the program outputs all non-dominated solutions found. Generally, we find that the algorithm converges after about 10 h of runtime.

Following other genetic algorithms for network alignment, we adopt a permutation encoding for our alignments (Barecke and Detyniecki, 2007; Saraph and Milenković, 2014). In this encoding, we label the n nodes of each network with the integers $\{0, 1, ...n-2, n-1\}$. Since we assume $|V_2| \geq |V_1|$, we add 'dummy' nodes to V_1 so that $|V_1| = |V_2|$. Then our alignment f can be stored in memory as an array a such that if $f(u) = v$, then $a_u = v$. All indices $i \geq |V_1|$ of a are ignored. This representation simplifies searching through the space of alignments to searching through the space of permutations of integers in the interval $[0, |V_2|)$, allowing us to adapt permutation-based search operators that have been well-studied in genetic and local search algorithms for the traveling salesman problem (Goldberg, 1989).

We adopt the Uniform Partially Matched Crossover (UPMX) operator introduced in a genetic algorithm for a related largest common subgraph problem (Cicirello and Smith, 2000). We compared this to several other standard and custom-built crossover operators and found that it tended to give the best performance. UPMX works as follows: for two permutations a and b, for each index i with probability *cxswappb*, swap the value of a_i and b_i. Unless $a_i = b_i$, the new values of a_i and b_i after the swap will occur in their respective arrays in two places. To fix this, we also swap a_j and b_k, where j and k are the indices of the duplicate values in their respective arrays. An example of this swap process for index $i = 0$ follows.

1. Initial arrays: $a = \{1, 3, 2\}$ and $b = \{2, 1, 3\}$.
2. Swap the first elements of a and b: $a = \{2, 3, 2\}$ and $b = \{1, 1, 3\}$.
3. a has a duplicate at $j = 2$ and b has a duplicate at $k = 1$. Swap a_j and b_k: $a = \{2, 3, 1\}$ and $b = \{1, 2, 3\}$.

For mutation, we also adopt a simple swap-based scheme. For a permutation a, for each index i with probability *mutswappb*, we randomly select an index $j \neq i$ and swap a_i with a_j.

We also make use of two variants on a simple hill climbing algorithm based on swaps. The first variant, *hillClimbNonDominated*, performs repeated random swaps, undoing a swap if it worsens any of the objective functions. The second variant, *hillClimbOneObj*, takes an objective name as an argument, and only undoes a swap if the given objective is worsened, ignoring any worsening of other objectives. We use the former hill climbing variant as a core function to improve our population of alignments, and use the latter to increase the variance of our population in objective space, which helps us to find a better set of representatives of the approximated Pareto front.

It is important to note that our crossover, mutation and hill climbing functions all use swapping as their basic operation. Since the only way permutations are modified is by swapping, this allows us to implement our fitness evaluations very efficiently—we only compute the change in the user-specified objectives at each swap. This allows us to evaluate millions of swaps per second on a desktop CPU. This speed is such that a simplified version of our program, which employs only hill climbing to create a single alignment, produces a single excellent alignment in seconds. The reasoning behind Optnetalign is to leverage this speed to produce hundreds of alignments in the same amount of time that several other aligners take to produce a single alignment, since waiting hours for a result is already expected by users of network alignment software.

Our population is stored in an archive data structure that automatically discards dominated solutions when better solutions are inserted. This gives us a data structure that stores only the non-dominated solutions that have been found so far. When the archive exceeds its user-defined size limit, it shrinks itself to the maximum size by discarding the most crowded solutions, using the crowded comparison operator first introduced in NSGA-II (Deb *et al.*, 2002). This helps to ensure that a diverse variety of solutions are maintained in the population at all times. This archiving strategy is similar to the multiobjective local search algorithm PAES (Knowles and Corne, 2000).

To increase the performance of our algorithm, it was designed from the beginning to be amenable to parallelization. The user specifies how many threads the program is to use, and each thread executes Algorithm 1 until a user-specified time limit has been reached. The threads all share the same archive, which is the only point of communication between the threads. This solution is highly scalable, and is able to fully utilize 16 processor cores in our tests. The overall structure of the algorithm is very straightforward, and reminiscent of other hybrid genetic algorithms in the literature; e.g. (Nguyen *et al.*, 2007). One quirk of the algorithm is that we first decide probabilistically whether to perform *hillclimbOneObj* or *hillClimbNonDominated*, but then afterwards, perform another round of *hillClimbNonDominated*. This ensures that, when *hillclimbOneObj* is chosen, the alignment is not left in a dominated state that could easily be improved to a non-dominated one.

Algorithm 1 Per-Thread Loop

```
while time limit not reached do
    $(parent1, parent2) \leftarrow$ two random members of $Archive$
    $child \leftarrow$ initialize new alignment
    with probability $cxrate$ do
        $child \leftarrow crossover(cxswappb, parent1, parent2)$
    end
    with probability $mutrate$ do
        $child \leftarrow mutate(mutswappb, child)$
    end
    with probability $oneobjrate$ do
        $randObj \leftarrow$ randomly-chosen objective
        $child \leftarrow hillclimbOneObj(randObj, niters, child)$
        else do
            $child \leftarrow hillclimbNonDominated(niters, child)$
        end
        $child \leftarrow hillclimbNonDominated(niters, child)$
        $insert(child, Archive)$
    end while
```

The user-adjustable parameters of our algorithm include the time limit, the rate of crossover (*cxrate*) and mutation (*mutrate*), the probability of performing a swap at each index for crossover (*cxswappb*) and mutation (*mutswappb*), the probability that single-objective hill climbing will be used instead of non-dominated hill climbing (*oneobjrate*), and the number of iterations of hill climbing to perform in each loop (*niters*). We find that over the course of the algorithm's execution, the optimal rate of crossover, mutation and one-objective hill climbing seem to vary. We thus provide the option of automatically adjusting those rates to their rates of success at producing non-dominated solutions over the course of the algorithm's execution. Other, more sophisticated control schemes exist in the genetic algorithms literature (Eiben *et al.*, 1999; Eiben and Smit, 2011), but this simple heuristic appears to work quite well. We find that $cxswappb = 0.5$, $mutswappb = 0.0001$, and $hillclimbiters = 10000$ are good settings for the remaining parameters. It is important to emphasize that we set the number of hill climbing iterations quite high, so that computation time is dominated by hill climbing. We find that hill climbing is best used as the primary means for obtaining results, with crossover and mutation mostly playing a secondary role to prevent premature convergence.

3 Results and discussion

As discussed above, we report our results aligning the networks included in Isobase (Park *et al.*, 2011). For each alignment problem, we run our algorithm for 12 h. We try two different sets of objectives. In the first experiment, we set Optnetalign to maximize S^3 as well as the sum of the BLAST bit scores of proteins that have been aligned, leaving GOC as an external measure of alignment quality. In the second experiment, we optimize S^3 and GOC directly. The first experiment is most directly comparable to existing aligners, which try to optimize the sum of the bit score of all pairs of nodes that have been aligned (i.e. a maximum-weight matching), while the second is an investigation of how well we can optimize the overlap of GO terms directly, which helps us obtain an approximate upper bound on the extent to which both topological fit and biological function can be jointly optimized. For each alignment problem, we set Optnetalign's maximum population parameter to output an approximate Pareto front containing at most 200 alignments.

Since Optnetalign outputs more candidate alignments than the other aligners we evaluate against combined, we report the average, minimum and maximum of S^3 and GOC of the output alignments for each problem. In our experiments, these numbers are stable from run to run. We also compare the Pareto front of alignments our algorithm produces to all the alignments produced by the aligners that have a user-configurable tradeoff parameter between topological and biological similarity for the *C.elegans*–*D.melanogaster* alignment problem.

We also make some changes in the aligners we compare with, when compared with our previous benchmark paper. Since our alignment algorithm is somewhat similar to PISwap and MAGNA, we add these two aligners to the evaluation. We also remove IsoRank (Singh *et al.*, 2008), GRAAL (Kuchaiev *et al.*, 2010), MI-GRAAL (Kuchaiev and Przulj, 2011) and NATALIE (El-Kebir *et al.*, 2011) from consideration, since they were Pareto dominated by other existing aligners on this dataset.

To ensure that GO annotations transferred by sequence similarity were not skewing our results, we evaluated using only experimentally-verified GO terms. For results using all GO terms, see our Supplementary Information File.

3.1 First experiment benchmark results

Here we report the results of optimizing S^3 and the sum of bit scores of aligned pairs. We report the same data in tabular form in our Supplementary Information File. Optnetalign matches or outperforms all existing aligners on S^3, and outperforms all but one aligner on GOC. As noted above, we report the minimum, average and maximum of these scores. These metrics are almost perfectly inversely correlated, so for all of these experiments, the alignment Optnetalign produces with the highest S^3 has the lowest GOC, and vice versa.

We first consider our performance on S^3 in Figure 1. We must note that several aligners crashed on several experiments. This is denoted by a missing bar in these instances. The maximum S^3 found by Optnetalign on each problem instance is comparable to or exceeds that of NETAL, which was previously the best performer on this dataset. However, as shown in Figure 2, our topologically best alignments have much higher GOC scores than NETAL's. Even our alignments with the poorest topological quality tend to perform comparably to or better than many existing aligners.

With respect to GOC scores, our best alignments are all a distant second place behind PISwap. However, because PISwap works by first maximizing sequence similarity with the Hungarian algorithm, and then is limited to performing swaps that don't worsen the sequence similarity matching, its topological fit is extremely low, with S^3 scores generally an order of magnitude lower than any other aligner. Thus, PISwap's alignments constitute an extreme point on the Pareto front. On its topologically worst performance, PISwap only finds 91 conserved edges on the *C.elegans* to *S.cerevisiae* alignment problem, out of the 4495 possible. Our best GOC scores, which outperform all aligners except PISwap, come from alignments that have non-trivial S^3 scores that are in some cases, such as the *S.cerevisiae* to *H.sapiens* problem, competitive with existing algorithms.

We also compare the Pareto front found by Optnetalign on the *C.elegans* to *D.melanogaster* problem to the range of tradeoffs produced by existing aligners that expose a tradeoff parameter to the user for balancing topological and biological similarity. The results in Figure 3 show that Optnetalign is able to find a much wider Pareto front than all previous aligners. Previously, it was not nearly as clear that such a wide range of results was possible. The alignments Optnetalign produced for this problem also varied greatly in the pairs of nodes they chose to align. The least similar pair of networks had only 8.7% of their aligned pairs in common. The maximum was 73%, and the average was 36%. This diversity in aligned pairs is comparable to the diversity in all the alignments produced by previous aligners.

3.2 Second experiment benchmark results

Here we report the results we obtain when we set Optnetalign to optimize S^3 and GOC directly. Since ours is the only aligner among those tested that can optimize GO term consistency instead of bit score sum, all other aligners are still optimizing bit score sum, when

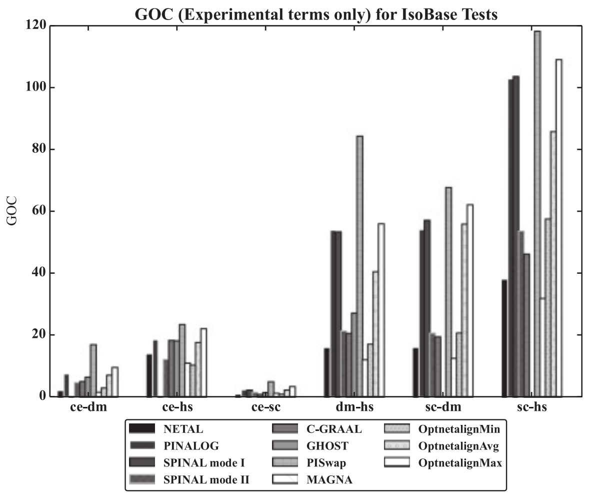

Fig. 1. S^3 score for various aligners and our own for the Isobase dataset, when Optnetalign is set to optimize S^3 and sum of bit scores. Note that the bars in the chart are in the same order as in the legend

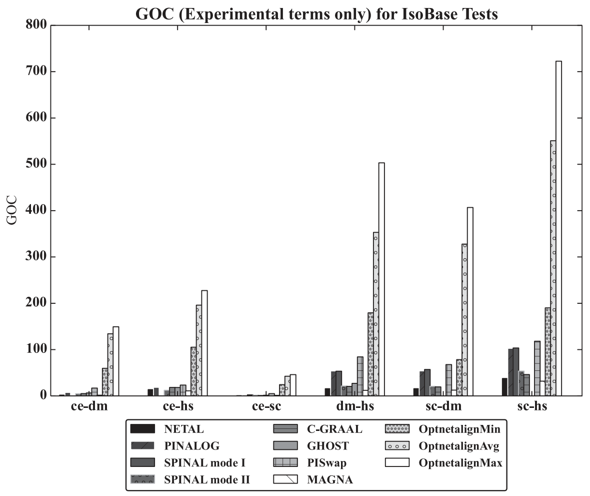

Fig. 2. GOC for various aligners and our own for the Isobase dataset, when Optnetalign is set to optimize S^3 and sum of bit scores. Only GO terms with experimental evidence codes are used

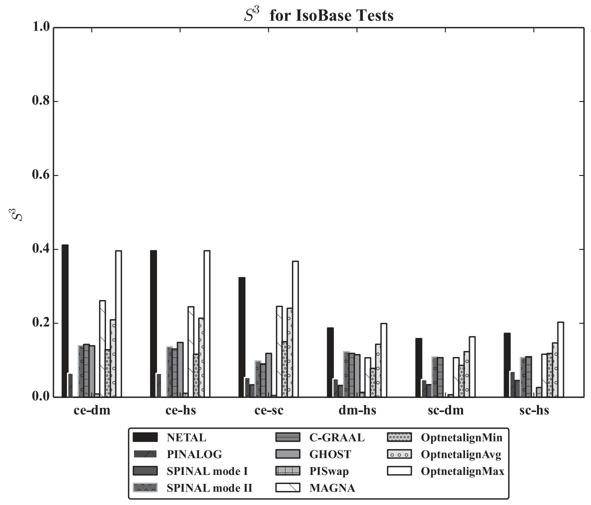

Fig. 3. Pareto fronts found by various aligners with adjustable tradeoff parameters and our own for the *C.elegans* to *D.melanogaster* alignment problem, when Optnetalign is set to optimize S^3 and sum of bit scores. The scatter plot here is visually indistinguishable from equivalent plots using other networks from this dataset. We use *C.elegans* to *D.melanogaster* in this figure because SPINAL and GHOST tended to crash on other problems, giving incomplete data

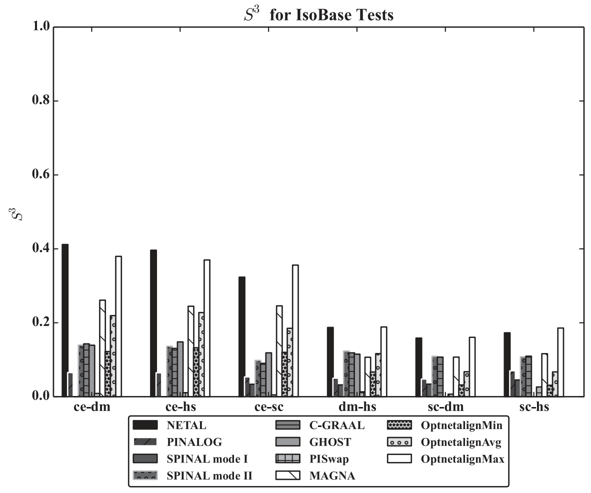

Fig. 4. S^3 score for various aligners and our own for the Isobase dataset, when Optnetalign is set to optimize S^3 and GOC. Note that the bars in the chart are in the same order as in the legend

applicable, so all of the scores in Figures 4 and 5 are the same as in Figures 1 and 2, except for Optnetalign's.

In Figure 4, we see that our S^3 results are much the same as before with the best alignments, with only slightly lower scores. The average scores are somewhat lower now, especially on the latter 3 alignments, where the average no longer outperforms most aligners. The minimum scores are very similar to those in Figure 1, as well.

The primary difference when optimizing GOC directly, instead of its imperfect proxy of bit score sum, is that we are now able to obtain GOC scores that outperform PISwap on several problems. While on the *C.elegans* to *D.melanogaster* problem, PISwap still strongly outperforms all other aligners, Optnetalign is actually able to outperform PISwap on the latter three alignment problems, and performs much more strongly on the other problems involving *C.elegans*. These results show that apparently better results can be obtained by optimizing GOC directly, and we recommend that future aligners also include this functionality.

3.3 Synthetic benchmark results

While the preceding benchmarks show that Optnetalign obtains excellent performance on real-world data, the true alignment for such networks is unknown. Therefore, we additionally benchmark against the NAPAbench network alignment benchmark dataset (Sahraeian and Yoon, 2012) to evaluate *node correctness*—the share of nodes in the smaller network that have been correctly mapped to orthologous nodes in the second. On these synthetic tests, Optnetalign achieves performance similar to, but slightly worse than, the best-performing existing aligners on this dataset, PISwap and GHOST, with Optnetalign's best-scoring alignments achieving node correctness levels between about 77 and 80%. On this dataset, the alignments with highest node correctness are the alignments on the high topological side of the Pareto front found; biological

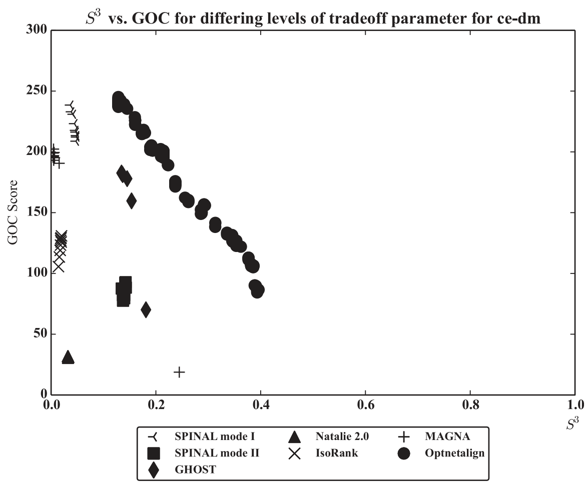

Fig. 5. GOC for various aligners and our own for the Isobase dataset, when Optnetalign is set to optimize S^3 and sum of GOC

similarity is less helpful, though still necessary, to create correct alignments. For more details, see our Supplementary Information File.

3.4 Execution times

The benchmarked aligners differed greatly in their execution times. Unfortunately, since not all aligners were available for the same platform, we had to execute many of them on virtual machines, which prevented us from performing a precise comparison of their relative execution times. Instead, we report here 'ballpark' execution times, expressed in orders of magnitude. Like several of the best-performing aligners, Optnetalign requires several hours to obtain good results. Since Optnetalign is a metaheuristic, it can conceivably keep trying to improve its alignments indefinitely. However, after between 10 and 12 h, it usually failed to find any further improvement. For this reason, we ran all tests of Optnetalign with a time limit of 12 h. Comparison here is not entirely straightforward, however. Optnetalign uses efficient swap-based operators that can produce alignments extremely quickly. These operators allow Optnetalign to produce hundreds of mutually non-dominated alignments in the course of one run. In contrast, other aligners with adjustable tradeoff parameters only produce one alignment per run, and so must be run many times to produce many alignments. When the user wants to produce many alignments at different tradeoffs, Optnetalign can save the user a significant amount of time. Additionally, if the user wants only one alignment, a simplified version of Optnetalign is available that can produce a single alignment in seconds or minutes, depending on the size of the networks.

Aligner	*Time*
NETAL	seconds
PINALOG	minutes
SPINAL	minutes
GRAAL	hours to days
C-GRAAL	hours to days
GHOST	hours to days
IsoRank	minutes
PISwap	minutes
MAGNA	hours to days
Optnetalign	hours to days

Table 1. Correlation matrix for various objectives on the *S.cerevisiae* to *H.sapiens* alignment problem

	ICS	S^3	EC	GOC	Bit score sum	LCCS size
ICS	1.0	−0.132	−0.338	0.079	−0.103	−0.42
S^3		1.0	0.957	−0.964	−0.848	0.939
EC			1.0	−0.948	−0.796	0.987
GOC				1.0	0.81	−0.91
Bit score sum					1.0	−0.759
LCCS size						1.0

Unlike the other objectives, the number of nodes in the largest common connected subgraph (LCCS) was not set as an optimization objective in our program; it was computed afterwards

3.5 Comparing objectives

The speed of Optnetalign, its ability to handle any number of objectives, and the large number of diverse alignments it can create in a reasonable amount of time, allow us to perform novel analyses that explore the inherent tradeoffs between many objectives, as well as examine how different objectives are correlated. This can give us insight into the nature of the network alignment problem, and help us identify deficiencies in current alignment evaluation metrics. We demonstrate this by performing another run of the *S.cerevisiae* to *H.sapiens* problem, in which we set the aligner to optimize EC, ICS, S^3, sum of bit scores and GOC. This time, we set the maximum archive size very high, to allow the aligner to output as many non-dominated alignments as possible in a 12 h run. This resulted in 571 unique, non-dominated alignments. The minimum agreement on aligned pairs for these alignments was 1%, the maximum was 92% and the average was 8.3%. This lack of agreement is consistent with the lack of agreement between existing aligners (Patro and Kingsford, 2012), so this confirms that our aligner produces alignments as diverse as using many different previously-published aligners together.

We present a correlation matrix for our objectives in Table 1. EC and S^3 are strongly positively correlated, but, as we noted above, since Optnetalign can minimize the *ICS* denominator without increasing the number of conserved edges, ICS is negatively correlated with the other two topological metrics. We also include the commonly used metric of the number of nodes in the largest connected common subgraph (LCCS) here (see e.g. Kuchaiev and Przulj, 2011 for more information), which Optnetalign cannot optimize directly. To demonstrate how Optnetalign allows us to compare the behavior of different alignment metrics, the relationship between ICS and GOC is illustrated in Figure 3 of our Supplementary Information. To better understand ICS inflation, we also present a plot of EC against ICS in Figure 4 of our Supplementary Information.

This experiment also demonstrates the potential utility of optimizing for GO term consistency instead of bit score sum. In Figure 6 in our Supplementary Information, we show that, while bit score sum and GOC generally correlate, the highest GOC scores we obtain have bit scores sums that are less than half of the highest bit score sum Optnetalign finds. This reinforces the fact that, in some cases, proteins of similar function can have significant differences in their sequences, and it shows the value of optimizing GOC directly.

This large set of alignments again highlights the inherent conflict between optimizing measures of topological fit and measures of biological fit simultaneously. In Figure 5 of our Supplementary Information, we show S^3 plotted against GOC. Even though most of the alignments dramatically outperform those of previous aligners, we still see a distinct tradeoff between S^3 and GOC. Since previous aligners varied dramatically in performance while only outputting a single alignment, there was very little evidence that such a wide Pareto front between these two objectives was obtainable. Previous aligners which provided a user-controlled tradeoff parameter between these topological and biological similarity scores framed the parameter as a small tweak, and in most cases, adjusting that parameter through its entire range, as we did in producing Figure 3, explores only a small portion of objective space. Therefore, it was not clearly understood that these two objectives conflict so sharply.

3.6 Potential directions for future work

Our results here show some weaknesses in existing network alignment research that open up many possibilities for future work. Perhaps the most glaring question, given these results, is that of the tradeoff between biological and topological similarity. The assumption underlying the entire premise of global network alignment is that different species share large regions of their PPI networks, which they inherited from a common ancestor. Our results show that finding that overlap is not as easy as was previously thought. We can either align two networks in a way that matches many neighbors in G_1 to many neighbors in G_2, *or* we can align two networks in such a way that proteins with similar GO annotations are matched together. It does not appear from our results that it is possible to do both well. There are a number of possible reasons for this that should be better explored in future work. First, as we noted previously (Clark and Kalita, 2014), this tradeoff is much less dramatic in synthetic networks. It's possible that currently known protein networks have too many false positive and false negative edges to be aligned reliably, and the false positive edges mislead alignment algorithms. If this is the case, we should probably prefer alignments with higher GOC, even if they have a lower S^3. Second, it is possible that our current best topological metrics are insufficient and mislead Optnetalign. These metrics all essentially count how many neighbors in G_1 have been matched to neighbors in G_2, but they have no way of ensuring neighbors of neighbors are matched to neighbors of neighbors, or that other large structures between networks are matched properly. One proposed metric that partially overcomes this is to report the size the largest connected common subgraph, but we have found that this metric correlates so strongly with metrics like S^3 that it does not provide us with much additional helpful information. This makes sense intuitively: conserving one more edge could join two previously disconnected common subgraphs and essentially double the score. Equivalently, reporting the size of the largest common connected subgraph doesn't tell us anything about the distribution of the sizes of the remaining connected subgraphs found. A few moderately large ones may be much more helpful than one very large one, especially if the species are more distantly related. Our algorithm performs well even when optimizing many objectives in one run, and, as we show above, this can be used to critically evaluate different candidate objectives.

It would also be possible to use this aligner to evaluate existing aligners' approximate topological similarity metrics, such as GHOST's (Patro and Kingsford, 2012) spectral graph signatures or graphlet degree signatures (Pržulj, 2007; Milenković and Pržulj, 2008). The relative performance of these similarity metrics has always been unclear, though some preliminary work has been done on the problem (Crawford and Milenković, 2014; Crawford *et al.*, 2014; Milenković *et al.*, 2013). With metaheuristic multiobjective alignment, we could optimize many of these metrics at once and analyze many alignments to determine which ones best predict or conflict with the objectives of network alignment. It's even possible that some of these similarity measures, when paired with Optnetalign, could produce better results than optimizing S^3 or GOC directly.

An important direction for future work is to find ways to summarize or condense the large number of possible alignments between two PPI networks. The original goal of pairwise global alignment was to produce the one best alignment given our objectives. Our work here has shown that currently studied alignment objectives do not uniquely specify a best alignment. Instead, there are many diverse alignments along a wide Pareto front, and these alignments may agree on as few as 1% of their aligned pairs while still appearing good with respect to at least one objective. While this was already known, our multiobjective aligner makes this problem much more obvious. This creates difficulties for a biologist who would like to use the output of such an algorithm. An obvious next step would be to find ways to summarize the results of these alignments so that someone who wants to use an alignment can either pick the alignment that best fits their particular needs, or produce a composite alignment that summarizes the most interesting alignments found. We have done some preliminary experiments with creating *partial* alignments, using the number of aligned pairs of the alignment as an objective. This could be used to produce small, local alignments and large, global alignments in one run of the program as well, which would unify the two related, but heretofore separate, techniques of local and global network alignment. This has similarities to NetAligner (Pache and Aloy, 2012; Pache *et al.*, 2012), which is another tool capable of both local and global alignment. There are many other possibilities for producing summary or approximate alignments, as well. The recent publication of DualAligner (Seah *et al.*, 2014) is a good step in this direction.

4 Conclusion

We have shown that Optnetalign can outperform or perform competitively with the best previously published aligners. Furthermore, since it is a multiobjective algorithm, it manages to better balance the tradeoff between the conflicting objectives of biological and topological fit. Its ability to produce a large number of alignments in one run allows us to explore the problem of network alignment more thoroughly, and reveals the strengths and weaknesses of current evaluation metrics. We expect Optnetalign will be useful not only in aligning particular PPI networks to one another, but also in evaluating and experimenting with new alignment objectives, as well. The large number of highly diverse alignments Optnetalign can produce allows us to thoroughly understand the sometimes complex relationship between alignment objectives, and also shows promise in developing new techniques to combine many alignments into summary alignments, to produce many-to-many alignments, or to create local alignments that find small-scale regions of similarity between PPI networks. Perhaps most importantly, instead of running many different programs dozens of times to get a wide variety of possible alignments to analyze, users of alignment software need only run Optnetalign once.

Acknowledgement

The authors thank Vikram Saraph for helping them get MAGNA working, Cheng-Yu Ma for quickly diagnosing and fixing a bug in PISwap that the authors were having difficulty with, and Rob Patro for insightful discussion.

Funding

This work was supported by a grant from the Biofrontiers Center at the University of Colorado, Colorado Springs.

Conflict of Interest: none declared.

References

Aebersold,R. and Mann,M. (2003) Mass spectrometry-based proteomics. *Nature,* **422**, 198–207.

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

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

Barecke,T. and Detyniecki,M. (2007) Memetic algorithms for inexact graph matching. In: *IEEE Congress on Evolutionary Computation, 2007. CEC 2007.* pp. 4238–4245, IEEE, Singapore.

Blum,C. *et al.* (2011) Hybrid metaheuristics in combinatorial optimization: a survey. *Appl. Soft Comput.*, **11**, 4135–4151.

Chatr-aryamontri,A. *et al.* (2013) The biogrid interaction database: 2013 update. *Nucleic Acids Res.*, **41**, D816–D823.

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

Cicirello,V.A. and Smith,S.F. (2000) Modeling GA performance for control parameter optimization. In: *Genetic and Evolutionary Computation Conference,* pp. 235–242, Las Vegas.

Clark,C. and Kalita,J. (2014) A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics,* **30**, 2351–2359.

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, Shaker Heights, Ohio.

Corne,D.W. *et al.* (2000) The pareto envelope-based selection algorithm for multiobjective optimization. In: *Parallel Problem Solving from Nature VI Conference,* pp. 839–848, Springer, Paris.

Crawford,J. *et al.* (2014) Fair evaluation of global network aligners. *arXiv preprint arXiv:1407.4824.*

Crawford,J. and Milenković,T. (2014) Great: graphlet edge-based network alignment. *arXiv preprint arXiv:1410.5103.*

Cross,A.D. *et al.* (1997) Inexact graph matching using genetic search. *Pattern Recogn.,* **30**, 953–970.

Czyzżak,P. and Jaszkiewicz,A. (1998) Pareto simulated annealing—a metaheuristic technique for multiple-objective combinatorial optimization. *J. Multi-Criteria Decis. Anal.,* **7**, 34–47.

Deb,K. (2001) *Multi-objective Optimization Using Evolutionary Algorithms.* Wiley, England.

Deb,K. *et al.* (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.,* **6**, 182–197.

Eiben,A.E. *et al.* (1999) Parameter control in evolutionary algorithms. *IEEE Trans. Evol. Comput.,* **3**, 124–141.

Eiben,A.E. and Smit,S.K. (2011) Parameter tuning for configuring and analyzing evolutionary algorithms. *Swarm Evol. Comput.,* **1**, 19–31.

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

Floreano,D. and Mattiussi,C. (2008) *Bio-inspired Artificial Intelligence: Theories, Methods, and Technologies.* MIT press, Cambridge, MA.

Franceschini,A. *et al.* (2013) STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Res.*, **41**, D808–815.

Goldberg,D.E. (1989) *Genetic Algorithms in Search Optimization and Machine Learning*, Boston, MA.

Guzzi,P. and Mina,M. (2014) Computational Biology and Bioinformatics. *IEEE/ACM Transactions*, **11**, 561–572.

Holland,J.H. (1975) *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence.* University of Michigan Press, Ann Arbor, MI.

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, Xi'an, China.

Knowles,J. and Corne,D. (2005) Memetic algorithms for multiobjective optimization: issues, methods and prospects. In: *Recent Advances in Memetic Algorithms,* pp. 313–352. Springer.

Knowles,J.D. and Corne,D.W. (2000) Approximating the nondominated front using the pareto archived evolution strategy. *Evol. Comput.,* **8**, 149–172.

Knowles,J.D. *et al.* (2001) Reducing local optima in single-objective problems by multi-objectivization. In: *Evolutionary Multi-Criterion Optimization,* pp. 269–283. Springer.

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

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

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

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

Lipets,V. *et al.* (2009) Subsea: an efficient heuristic algorithm for subgraph isomorphism. *Data Mining Knowl. Dis.,* **19**, 320–350.

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

Michalewicz,Z. and Fogel,D.B. (2004) *How to solve it: modern heuristics.* Springer, Berlin Heidelberg.

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

Milenković,T. *et al.* (2013) Global network alignment in the context of aging. In: *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics,* p. 23. ACM.

Milenković,T. and Pržulj,N. (2008) Uncovering biological network function via graphlet degree signatures. *Cancer Inf.*, **6**, 257.

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

Nguyen,H.D. *et al.* (2007) Implementation of an effective hybrid ga for large-scale traveling salesman problems. *IEEE Trans. Syst. Man Cybern. Part B,* **37**, 92–99.

Pache,R.A. and Aloy,P. (2012) A novel framework for the comparative analysis of biological networks. *PLoS One,* **7**, e31220.

Pache,R.A. *et al.* (2012) Netaligner—a network alignment server to compare complexes, pathways and whole interactomes. *Nucleic Acids Res.*, **40**(Web Server issue), W157–W161.

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

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

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.

Prasad,T.K. *et al.* (2009) Human protein reference database—2009 update. *Nucleic Acids Res.,* **37**, D767–D772.

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.

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

Seah,B.-S. *et al.* (2014) Dualaligner: A dual alignment-based strategy to align protein interaction networks. *Bioinformatics,* **30**, 2619–2626.

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

Xenarios,I. *et al.* (2002) Dip, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.*, **30**, 303–305.

Zhou,A. *et al.* (2011) Multiobjective evolutionary algorithms: a survey of the state of the art. *Swarm and Evolutionary Computation,* **1**, 32–49.

Zitzler,E. *et al.* (2001) SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: Giannakoglou,K. C. *et al* (eds). *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems (EUROGEN 2001)*, pp. 95–100, International Center for Numerical Methods in Engineering (CIMNE).