BIOINFORMATICS

Vol. 00 no. 00 2015
Pages 1–9

SANA: Simulated Annealing far outperforms many other search algorithms for biological network alignment

Nil Mamano and Wayne B. Hayes*
Department of Computer Science, University of California, Irvine CA 92697-3435, USA

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Every alignment algorithm consists of two orthogonal components: an objective function M measuring the quality an alignment, and a search algorithm that explores the space of alignments looking for ones scoring well according to M. We introduce a new search algorithm called SANA (Simulated Annealing Network Aligner) and apply it to protein-protein interaction networks using S^3 as the the topological measure. Compared against 12 recent algorithms, SANA produces 5–10 *times* as many correct node mappings as the others when the correct answer is known. We expose an anti-correlation in many existing aligners between their ability to produce good topological vs. functional similarity scores, whereas SANA usually outscores other methods in both measures. If given the perfect objective function encoding the identity mapping, SANA quickly converges to the perfect solution while many other algorithms falter. We observe that when aligning networks with a known mapping and optimizing only S^3, SANA creates alignments that are not perfect and yet whose S^3 scores match that of the perfect alignment. We call this phenomenon *saturation of the topological score*. Saturation implies that a measure's correlation with alignment correctness falters before the perfect alignment is reached. This, combined with SANA's ability to produce the perfect alignment if given the perfect objective function, suggests that better objective functions may lead to dramatically better alignments. We conclude that future work should focus on finding better objective functions, and offer SANA as the search algorithm of choice.

Software available at http://sana.ics.uci.edu.

Contact: whayes@uci.edu

1 INTRODUCTION

1.1 Context

Network alignment is the task of finding the best way to "fit" one network inside another. It has applications in several areas, including ontology matching (Li *et al.*, 2009), pattern recognition (Zaslavskiy *et al.*, 2009), language processing (Bayati *et al.*, 2009), and social networks (Zhang and Tang, 2013). Thus, the specific goal of network alignment depends on the context. We focus on a particular application from the computational biology domain: the alignment of protein-protein interaction (PPI) networks. In a PPI network, nodes represent proteins from a given organism, and edges connect proteins that interact physically. These kinds of interactions are discovered through high throughput experimental methods such as yeast two-hybrid screening (Ito *et al.*, 2000) or protein complex purification via mass-spectrometry (Krogan *et al.*, 2006).

PPI network alignment has many interesting applications. It can serve to transfer functional information across species (Kuchaiev *et al.*, 2010), which, in turn, has been used to offer insights on the mechanisms of human diseases (Uetz *et al.*, 2006), or the process of aging in humans (Milenković *et al.*, 2013).

Network alignment can be classified as local or global. The former aims to align small regions accurately (Kelley *et al.*, 2004). Consequently, it often fails to find large conserved connected subgraphs in different networks. By contrast, global network alignment aims to generate one-to-one node mappings between two networks. By aligning entire networks, it overcomes the shortcomings of local network alignment. For this reason, the majority of recent research has focused on global network alignment (see section 1.2). There are also methods that allow alignments between more than two networks (Liao *et al.*, 2009; Hu *et al.*, 2014; Gligorijević *et al.*, 2015; Vijayan and Milenkovic, 2016; Alkan and Erten, 2014).

We focus on pairwise, one-to-one global network alignment. Its goal is to find a injective mapping from the proteins of the smaller network to proteins in the larger network. Ideally, we would like to find the most biologically relevant mapping: aligned proteins in both networks should be functionally and/or homologically related, in the sense that they used to be the *same* protein in the species which was the common ancestor of the species of both PPI networks. Since proteins may have multiple descendants, such a mapping may not be one-to-one. For now we ignore this complication and view global pairwise 1-to-1 network alignment as a convenient approximation, as do almost all of the algorithms we compare against (Table 1).

1.2 Previous work

Current PPI network alignment methods use a combination of sequence and topological information to align similar proteins. Biological information includes *a priori* knowledge about the proteins, such as amino acid sequences. On the other hand, topological information is extracted exclusively from the structure of the PPI network. Since the goal is to obtain biologically relevant alignments, early methods focused on biological information (usually sequence). However, as our understanding of topology-function relationships (Kuchaiev *et al.*, 2010; Davis *et al.*, 2015) has improved, topology information has gradually shifted to a central role. For instance, it has been shown recently that topological

[*] to whom correspondence should be addressed

information is more important than sequence information for uncovering functionally conserved interactions (Malod-Dognin and Pržulj, 2015). Topological knowledge can be extracted in many forms. For example, the wiring patterns in the vicinity of homologous proteins in different networks tend to be similar. This information is well captured by graphlets (Pržulj *et al.*, 2004), which generalize the concept of the degree of a node.

There are a wide variety of network alignment methods. This diversity is motivated by the inherent computational complexity of network alignment: topologically speaking, one would like to find an alignment that maximizes the number of preserved interactions, i.e., interactions between proteins that are mapped to proteins that also interact. However, this problem is *NP-hard* because it is a generalization of *subgraph isomorphism*, which is *NP-complete* (Cook, 1971). This means that no efficient algorithm is known. Thus, practical methods must rely on approximation and heuristic techniques; and when it comes to heuristic algorithms, the possibilities are endless but there is no obvious "best option".

In the biological network domain, work over the past few years has included IsoRank (Singh *et al.*, 2008), the family of GRAAL algorithms (GRAAL (Kuchaiev *et al.*, 2010), H-GRAAL (Milenković *et al.*, 2010), C-GRAAL (Memisevic and Pržulj, 2012), MI-GRAAL (Kuchaiev and Pržulj, 2011), L-GRAAL (Malod-Dognin and Pržulj, 2015)), NATALIE (Klau, 2009; El-Kebir *et al.*, 2011), GHOST (Patro and Kingsford, 2012), NETAL (Neyshabur *et al.*, 2013), SPINAL (Aladağ and Erten, 2013), PISwap (Chindelevitch *et al.*, 2013) MAGNA (Saraph and Milenković, 2014) and its successor MAGNA++ (Vijayan *et al.*, 2015), GREAT (Crawford and Milenković, 2015), WAVE (Sun *et al.*, 2015), HubAlign (Hashemifar and Xu, 2014), OptNetalign (Clark and Kalita, 2015), SPINAL (Aladağ and Erten, 2013), GEDEVO (Ibragimov *et al.*, 2013) and CytoGEDEVO (Malek *et al.*, 2016). Several recent surveys exist (Clark and Kalita, 2014; Elmsallati *et al.*, 2015; Faisal *et al.*, 2015).

In general, each method defines an *objective function* or *measure* over alignments that can be viewed as a *score*, and then proposes a *search algorithm* that searches through the enormous space of possible alignments in an attempt to maximize the objective function. Some measures such as sequence similarity or graphlet similarity (Kuchaiev *et al.*, 2010) are defined over pairs of proteins instead of whole alignments. They can be generalized to whole alignments by taking the average similarity score among all pairs of aligned proteins. These measures are called *local measures*, because the contribution of each mapping is independent of the others. On the other hand, global measures aim to evaluate the alignment from a global perspective, and not on a node-by-node basis.

While every method has its own objective function, the alignments are compared according to a set of *target* measures that have been established as the most important (see section 3.1). However, even target measures are of heuristic nature, because except when we align a network with itself, the *correct* mapping is unknown. A good objective function should guide the search algorithm to alignments that score well in all the target measures. Sometimes the objective function can be one of the target measures (Saraph and Milenković, 2014). However, since target measures are usually global measures, the search algorithms for local measures described above are in general not applicable.

1.3 Our contributions

We present SANA (*Simulated Annealing Network Aligner*), a search algorithm based on Simulated Annealing (Kirkpatrick *et al.*, 1983; Černý, 1985), a metaheuristic search algorithm with a rich history of successful applications to many optimization problems across a wide variety of domains. It shares many characteristics with OptNetAlign (Clark and Kalita, 2015), MAGNA and MAGNA++ (Saraph and Milenković, 2014; Vijayan *et al.*, 2015). All are random search algorithms that can be used to directly optimize any objective function, and all can start with any alignment and then improve it.[1] However, as we shall see below, SANA significantly outperforms them, achieving better results while utilizing dramatically less RAM and CPU.

2 METHODOLOGY

2.1 Main idea

Annealing is a process used in metallurgy to create crystals. A crystal is a highly structured, low-energy state of a material. This state can only be reached if the material is cooled on a very specific temperature schedule, during which the material can settle into the lowest energy state which forms the most perfect crystal.

Simulated annealing is a metaheuristic algorithm, which means that it is not tailored to any specific optimization problem. It can be applied to any optimization problem as long as the necessary elements are defined: the *solutions*, the *objective function* and the *neighbor relationship*. The analogy to annealing goes as follows: A solution is like a state of the material. In our case, a solution is an alignment. The objective function is analogous to the energy of the material. While in metallurgy the goal is a state with lowest energy, in simulated annealing the best solution optimizes some arbitrary objective function of our choice, and we choose either to minimize or maximize it. When atoms move due to high temperature, the state of the material changes slightly from moment to moment. In order to simulate this, we need a neighbor relationship that indicates which solutions are close to each other. Then, we can change a solution for a neighbor solution. For instance, we can say two alignments are neighbors if they only vary in one or two mappings of individual pairs of aligned nodes.

If we take a random alignment, called the *initial solution*, we can improve it by looking at its neighbor solutions and choosing the best one. If we repeat this process, we will quickly reach a *local optimum* which we can no longer improve. However, since the energy landscape is unlikely to be monotonic everywhere, this local optimum is unlikely to be the global optimum. To avoid this pitfall, simulated annealing introduces the ability to allow worse solutions to be selected with some probability, analogous to how high-temperature materials have enough energy to move freely through different states. As the temperature decreases, the ability to escape local optimum decreases. If the temperature schedule is chosen correctly, then the solution will tend towards a global optimum. In the limit of an infinitely slow temperature decrease, the global optimum is reached with probability 1 (Mitra *et al.*, 1985).

2.2 SANA algorithm

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs (networks) with $|V_1| \leq |V_2|$. A pairwise global alignment a (from this point simply alignment) from G_1 to G_2 is an injective function from V_1 to V_2. An

[1] While we were developing SANA, two other network alignment algorithms have appeared based on simulated annealing (Hu *et al.*, 2014; Larsen *et al.*, 2016). However, both perform multiple network alignment and so are inappropriate for comparison to SANA, which currently only aligns two networks at a time.

Algorithm 1 SANA

```
input: $G_1, G_2, f, a_0, t_{max}$
output: $a$, which attempts to maximize $f(a)$
1: Let $a = a_0, i = 0$
2: while $t_{exec} < t_{max}$ do
3:     $a' \leftarrow$ random neighbor(a)
4:     $\Delta f \leftarrow f(a') - f(a)$
5:     if $\Delta f \geq 0$ then $a \leftarrow a'$
6:     else
7:         $a \leftarrow a'$ with probability $P(\Delta f, T(i))$
8:     $i \leftarrow i + 1$
9: return $a$
```

objective function f is a function from the set of all alignments to the closed range [0, 1].

The basic scheme of SANA is shown in Algorithm 1. The input consists of the two networks G_1, G_2, an objective function f, a starting alignment (in the absence of one, SANA generates a random alignment), and a maximum execution time t_{max}. Since SANA is a generic search algorithm, f can be any objective function. The output is an alignment a that aims to optimize f—in our case, we *maximize* various topological and biological similarity measures rather than minimizing energy. The general form we use is

$$\text{score}(a) = (1 - \alpha)T(a) + \alpha S(a) \tag{1}$$

where $T(a)$ is some measure of topological similarity, and $S(a)$ is a measure of sequence similarity of an alignment a. For topology we usually optimize S^3, the *Symmetric Substructure Score* (Saraph and Milenković, 2014) although we can also directly optimize "importance" (Hashemifar and Xu, 2014) and various forms of *weighted edge coverage* (Sun *et al.*, 2015; Malod-Dognin and Pržulj, 2015). For sequence similarity, we use normalized BLAST bit-scores (Camacho *et al.*, 2009). If $s(u_1, u_2)$ is the BLAST bit-score of proteins u_1 and u_2,

$$S(a) = \sum_{u_1 \in V_1} \frac{s(u_1, a(u_1))}{\max_{v_1 \in V_1, v_2 \in V_2} s(v_1, v_2)}.$$

2.2.1 Temperature Schedule An important element of simulated annealing is the *temperature schedule*, which determines the decline of the probability to accept a worse solution as the algorithm advances. The *temperature* $T(i)$ is a control parameter which depends on the current iteration i. It is commonly defined as $T(i) = k \cdot e^{-\lambda \cdot i}$, where k and λ are empirically determined constants greater than zero (Kirkpatrick *et al.*, 1983). The temperature is at its highest point at iteration 0 where $T(0) = k$, and approaches 0 asymptotically. The constant λ determines how fast the temperature approaches 0. We use an automated temperature schedule which we will describe in a future paper.

We define two types of neighbors among alignments: *change* and *swap* neighbors. Change neighbors differ only in one mapping, which has the same origin in G_1 but different destinations in G_2. Swap neighbors differ in exactly two mappings, which have the same sources but their images are exchanged (see Supplement for figure). Together, the two types of neighbors allow SANA to explore the solution space completely: through a sequence of neighbors, it is possible to go from any alignment to any other alignment. In SANA when a random neighbor is generated (line 3 of Algorithm 1) the probability of choosing each type of neighbor is proportional to its *branching factor*, i.e., the number of different neighbors of a of that type. This way, all neighbors are equally likely. The idea of using swaps to improve the alignment is not new; it has been used before with other local search algorithms (Chindelevitch *et al.*, 2013; Saraph and Milenković, 2014).

In the algorithm, Δf denotes the increment in the objective function between the new and the current solution. The probability to accept a worse solution is $P(\Delta f, T(i)) = e^{\Delta f / T(i)}$ (Kirkpatrick *et al.*, 1983). This probability decreases when the difference between $f(a)$ and $f(a')$ increases, and it also decreases as the temperature decreases.

2.2.2 Time complexity and incremental evaluation of $f(a)$ A crucial ingredient in the speed and success of SANA is *incremental evaluation* of measures: each move is a relatively small change in the alignment and thus can be computed quickly; in turn, to maintain speed, we must be able to quickly evaluate whether the move was a good or a bad one. For this we require that each measure can be easily updated *incrementally*. For example, for any local measure, we simply subtract the value of the node pair(s) that were aligned before the move, and then add in the value of post-move pair(s), both of which are constant-time operations. For any edge-based measure such as EC, CS, or S^3, the time to evaluate the new-vs-old values of the objective function are bounded by the degrees of the nodes involved in the change or swap operation, and so the amortized cost is proportional to the average degree of the networks being aligned, which is typically small compared to the number of nodes. (A more detailed analysis appears in the Supplement.) Incremental evaluation allows SANA to perform millions of moves per second even on enormous networks, allowing billions of alignments to be compared in a matter of minutes. Note that OptNetAlign (Clark and Kalita, 2015) also performs such incremental evalutation; however, we were unable to make OptNetAlign give answers as good as SANA's even after extensive consultation with its author (Connor Clark, personal communication).

Since SANA (and OptNetAlign) can perform millions of small moves per second, we effectively perform a more fine-grained and comprehensive search of the alignment space than other methods. As an example comparison, MAGNA++ creates each new alignment in its population by combining existing alignments. Such new alignments are difficult to evaluate incrementally, and so each new alignment in MAGNA's population must be evaluated from scratch, which is far more expensive than an incremental evaluation. More importantly, however, generating an alignment from two existing "parent" alignments is effectively making a large "leap" through the search space, without the opportunity to evaluate all the individual steps to get there. Thus one loses the ability to make fine-grained decisions along the way. We hypothesize that this ability to perform fine-grained search and decision may partly explain SANA's success over other search algorithms.

Theoretical work has shown that, if i is the iteration number, then a temperature schedule scaled as $k/\ln(i)$ converges to the optimal solution with probability 1 in the limit $i \to \infty$ if the initial temperature k is "large enough" (Geman and Geman, 1984). However, in practice this temperature schedule is too slow. The schedule or k/i has been shown to work well in many problems (Szu and Hartley, 1987), although in practice the fastest convergence appears to come from an exponentially decaying temperature schedule (Ingber, 1989).

3 RESULTS

3.1 Alignment evaluation

Assuming an alignment a, and nodes $u_1, v_1 \in V_1$ and $u_2, v_2 \in V_2$, we describe the following *topological measures*. **Edge Coverage (EC)**[2], which is simply the fraction of edges in G_1 that are aligned to edges in G_2. Formally, let $E_a = \{(u_1, v_1) \in E_1 \mid (a(u_1), a(v_1)) \in E_2\}$ denote the set of edges in G_1 that cover edges in G_2 in alignment a. Then, the EC of a is $EC(a) = |E_a|/|E_1|$. EC has the drawback that it can be high if G_2 is edge-dense because, for example, if G_2 is a clique then *any* alignment has EC=1. **Induced Conserved Structure (ICS)**: the ratio of aligned edges to induced edges, $|E_a|/|\hat{E}_a|$; like EC, it is also assymmetric and we include it only for consistency with prevous studies. **Symmetric Substructure Score** (S^3) which was invented precisely to overcome

[2] aka "edge conservation" or "edge correctness", but see Supplement.

Table 1. List of methods compared in this paper, along with approximate RAM and CPU requirements for aligning the networks used in this paper. RAM "small" means no significant storage beyond the networks themselves. The rows are ordered roughly by publication date. QAP=Quadratic Assignment Problem.

Name	Year	≈CPU	≈RAM	Objective, Search Method, & Comments
PI-SWAP	2010	minutes-hours	many GB	sequence+EC; 2- and 3-swap; designed to post-process existing alignments
GHOST	2012	days-years	100 GB	Multiscale Spectral Signatures; seed+extend+QAP
NETAL	2013	seconds-minutes	small	sequence + neighbor similarity matrix; greedy
SPINAL	2013	minutes-days	few GB	incremental hierarchical matching of neighborhood bipartite graphs; iterative refinement
HubAlign	2014	seconds-minutes	small	"importance"; greedy
WAVE	2014	seconds-minutes	small	weighted edge conservation (WEC); greedy
MAGNA(++)	2014	hours-days	small	S3 (or any objective); evolutionary search
CytoGEDEVO	2015	hours-days	few GB	Graph Edit Distance; evolutionary search
NATALIE 2.0	2015	hours-days	many GB	EC+sequence; Restricted QAP
OptNetAlign	2015	hours-days	few GB	S3 (or any objective); Memetic Search
L-GRAAL	2015	hours	few GB	graphlets + WEC; Lagrangian Relaxation
SANA	2016	minutes	small	S3 (or any objective); Simulated Annealing

the asymmetries of EC and ICS (Saraph and Milenković, 2014). Let $\hat{E}_a = \{(u',v') \in E_2 \mid \exists u,v \in V_1 \wedge a(u) = u' \wedge a(v) = v'\}$ denote the set of edges of the subgraph of G_2 induced by the nodes in the alignment. Then, S^3 is defined as: $S^3(a) = \frac{|E_a|}{|E_1|+|\hat{E}_a|-|E_a|}$. **Node Correctness (NC)**, the fraction of nodes correctly aligned when the mapping is known. NC is not usually known except in synthetic cases such as aligning a network to itself, or if the network(s) have been synthetically constructed with a known node mapping. **Largest Common Connected Subgraph (LCCS)**: The *common subgraph* of an alignment a between G_1 and G_2 is the subgraph of G_1 that remains when considering only edges covered by the alignment: $CS_a = (V_1, E_a)$. A good alignment has a common subgraph with large connected regions. Let $CC_l = (V_l, E_l)$ be the largest connected component of CS_a. LCCS measures the size of CC_l as the geometric mean of (i) the fraction of nodes in CC_l, $|V_l|/|V_1|$, and (ii) the fraction of edges in CC_l, $|E_l|/\min(|E_1|, |\hat{E}_a|)$. See Kuchaiev *et al.* (2010) and Saraph and Milenković (2014).

To assess *functional similarity*, we measure common Gene Ontology (GO) terms (Ashburner *et al.*, 2000) using Resnik maximum best-match semantic similarity (Resnik, 1995) according to the Python FastSemSim package.[3] In the Supplementary material we also show SANA's superiority in several other measures including mapping proteins with a probable common ancestor, and in recovering known homologs between species (both of which are stringent measures of functional similarity).

3.2 Datasets

We use three different datasets from several previous studies, outlined in Table 2: ***BioGRID*** comprises eight manually curated networks (Chatr-aryamontri *et al.*, 2013) (v3.2.101, June 2013). Like other authors, we align the six networks with lowest edges count, but to show SANA can easily handle it, we also align the two largest networks (SC and HS). ***Yeast2 and Human1*** (Collins *et al.*, 2007; Radivojac *et al.*, 2008): an old pair that has become a bit of a *de facto* standard (Kuchaiev *et al.*, 2010; Kuchaiev and Pržulj, 2011; Neyshabur *et al.*, 2013; Saraph and Milenković, 2014).

Table 2. List of networks grouped by datasets, with the corresponding identifiers used in the plots.

Network	Identifier	Proteins	Edges
Rattus norvegicus	RN	1,657	2,330
Schizosaccharomyces pombe	SP	1,911	4,711
Caenorhabditis elegans	CE	3,134	5,428
Mus musculus	MM	4,370	9,116
Arabidopsis thaliana	AT	5,897	13,381
Drosophila melanogaster	DM	7,937	34,753
Saccharomyces cerevisiae	SC	5,831	77,149
Homo Sapiens	HS	13,276	110,528
Yeast2	Y2	2,390	16,127
Human1	H1	9,141	41,456
Yeast	Y0	1,004	8,323
Yeast (+5% noise)	Y5	1,004	8,739
Yeast (+10% noise)	Y10	1,004	9,155
Yeast (+15% noise)	Y15	1,004	9,571
Yeast (+20% noise)	Y20	1,004	9,987
Yeast (+25% noise)	Y25	1,004	10,403

Noisy yeast: six variations of yeast (Collins *et al.*, 2007), all having the same set of nodes but different numbers of edges. The first has 8,323 edges, while the others add 5, 10, 15, 20, and 25% "lower-confidence" interactions. We align the first network against each noisy variant. Since the underlying network is the same, we know the true node mapping. This dataset has been used by many previous authors (Saraph and Milenković, 2014; Patro and Kingsford, 2012; Sun *et al.*, 2015; Crawford and Milenković, 2015).

Different size yeast networks: Finally, we note that Noisy Yeast, Yeast2, and *S. cerevisiae* (SC) are all the same species, with PPI networks of 1004, 2390, and 5831 nodes respectively, and thus they share a known node mapping. We thus align Y0 to both Yeast2 and SC. To our knowledge this the first time anybody has aligned real networks of different sizes that have a known node mapping.

3.3 Compared methods and parameters

Table 1 summarizes the methods we compare against. For every method that allows it, we vary the α of Eq. 1 from 0 to 0.9 in steps

[3] `https://pypi.python.org/pypi/fastsemsim`

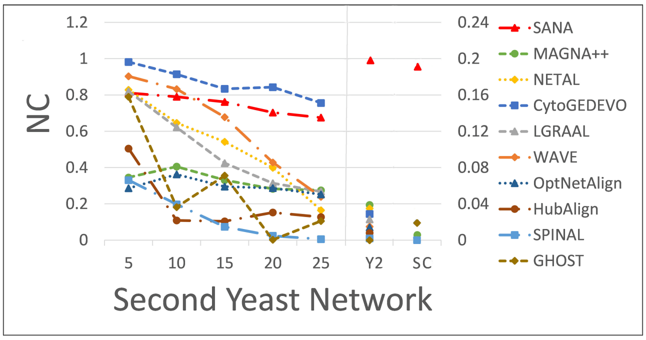

Fig. 1. NC (Node Correctness) scores for aligning the "clean" Collins Yeast dataset Y0 to other Yeast networks (see Supplemental info for EC, ICS, S^3, and LCCS), optimizing only topology—S^3 in the case of all random search algorithms (SANA, MAGNA++, and OptNetAlign), otherwise the native topological properties of the other algorithms. The first network is always Y0 (0% noise, 1004 nodes); the second network is depicted along the x-axis, with the values 5-25 representing the 1004-node networks Y5 through Y25. In these cases, CytoGEDEVO, based on Graph Edit Distance, does very well in NC, even beating SANA. However, since real PPI networks rarely have the same number of nodes, aligning equal-sized networks is both unrealistic and far too easy. The last two columns of the x-axis depict aligning Y0 (1004 nodes) with Y2 (2390 nodes) and SC (5831 nodes). In both cases SANA gets an NC of just under 20% (using the scale on the right-hand vertical axis), while other aligners, including CytoGEDEVO, drop to an NC less than 4% for Y2 and less than 2% for SC; in the latter case only GHOST and MAGNA++ score above 0. The legend naming the algorithms is ordered top-to-bottom according to the ordering of scores in the Y2 column. These last 2 columns paint a very sobering picture, demonstarting that *no* algorithm yet devised does well when the networks are of different sizes, though SANA (optimizing S^3) towers above the crowd. (We note that some algorithms, in particular MAGNA++, can get better NC using other objective functions such as graphlets. We leave the discussion of other objective functions to future work, and only emphasize here that SANA gets much better NC score when optimizing exactly the same objective as MAGNA and OptNetAlign.)

of 0.1; $\alpha = 0$ represents a topological only alignment; we avoid $\alpha = 1$ since an alignment devoid of a topological measure defeats the purpose of a network alignment. Care was taken to ensure that each algorithm was given the kind of sequence comparison measure it expects, eg BLAST e-values, or bitscores, etc., and that the direction of α was correct since some algorithms use $\alpha = 0$ to mean "topology only" and some use $\alpha = 1$ for that meaning. We allow generous CPU and RAM allocations for algorithms that require it—for example some algorithms required up to 100G of RAM and months of CPU (in parallel on a multi-core machine) on the largest networks. SANA in comparison was never given more than 30 minutes on a single core (frequently surpassing other algorithms during its first minute of execution, sometimes within *seconds*), and uses little memory over the storage of the networks themselves. The Supplementary section has a detailed table of run times.

3.4 Topology-only comparisons

In this section we compare the topological quality of our alignments (thus $\alpha = 0$ in Equation (1)). Aligners that allow it (SANA, MAGNA++, and OptNetAlign) are assigned the objective function that maximizes S^3; other alignment algorithms such as GHOST and GEDEVO implicitly optimize something similar to EC.

3.4.1 Alignments with known mapping In Figure 1 we compare all the aligners aligning various forms of the Yeast (*S.Cerevisiae*) network (Collins *et al.*, 2007; Chatr-aryamontri *et al.*, 2013). In all cases the true node mapping is known. When the networks are the same size, SANA beats all other algorithms except CytoGEDEVO in Node Correctness (NC). However, when aligning networks of different sizes with a known node mapping, SANA significantly outperforms all other algorithms, including CytoGEDEVO—to our knowledge the first time aligners have been compared in this way.

Table 3 demonstrates that while optimizing S^3, SANA achieves a score close to the S^3 score of the *perfect* (NC=1) alignment, usually within about 1%. That is, SANA does *the best that can possibly be done* under the constraint of optimizing only S^3. This also tells us that it is impossible to distinguish the perfect alignment from the one SANA found, using only S^3. There are, in fact, a huge number of different alignments that have the same S^3 score as the perfect alignment, all with NC much lower than 1. This point is made even more stark when we align the old "human1" network (H1) with the new one (HS) from BioGRID (not plotted). In this case SANA again matches the S^3 score of the perfect H1-HS alignment, but acheives an NC score of *zero*—along with every other aligner. Since SANA is a random search, it produces innumerable different alignments each with a near-optimal S^3 score—every one of them with an NC of effectively 0 (occasionally it correctly aligns 1 or 2 pairs by chance). These results did not change even when we tried various graphlet measures (Kuchaiev *et al.*, 2010; Malod-Dognin and Pržulj, 2015). Clearly, better objective functions are needed in order to improve alignments.

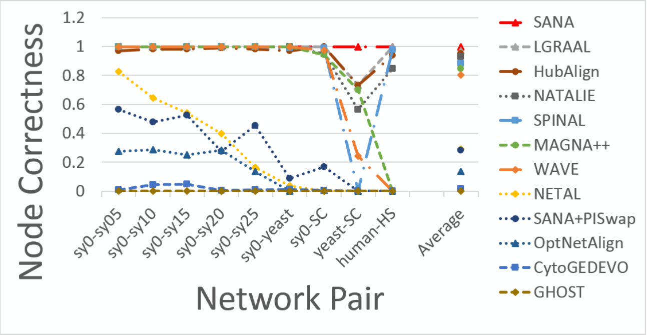

Fig. 2. Performance when given *perfect* information We provide the correct answer as input to each algorithm as a binary matrix of "sequence" similarities encoding the identity mapping, and set $\alpha = 0.9$ in Equation (1). This heavily weighs the correct answer (90%) but allows each algorithm a bit of leeway to use its native topological objective at a 10% weighting. We then run the algorithm and measure the node correctness of its output alignment. Any score significantly less than 1 suggests that the search component of the algorithm is sub-optimal, perhaps getting caught in local optima. Getting *consistently* much less than 1 suggests a significant deficiency in the search algorithm. The results did not change significantly with other values of α, including $\alpha = 1$.

One may also ask, "How well can a search algorithm do if given the perfect objective function?" All the algorithms we test have the option of providing a similarity matrix for pairs of nodes (u_i, v_j) for $u_i \in G_1$ and $v_j \in G_2$ (usually used to provide sequence similarities). To provide *perfect* information, we fill this matrix

2nd network	nodes	edges	S^3 when NC=1	SANA: S^3	NC	MAGNA++: S^3	NC	OptNetAlign: S^3	NC
syeast05	1004	8739	0.952	0.940	0.8108	0.721	0.346	0.665	0.286
syeast10	1004	9155	0.909	0.907	0.7898	0.730	0.405	0.690	0.362
syeast15	1004	9571	0.869	0.861	0.7620	0.675	0.331	0.656	0.294
syeast20	1004	9987	0.833	0.821	0.6922	0.621	0.283	0.659	0.287
syeast25	1004	10403	0.800	0.786	0.6753	0.611	0.275	0.604	0.252
Yeast2	2390	16127	0.611	0.601	0.198	0.440	0.039	0.630	0.014
SC	5831	77149	0.501	0.500	0.189	0.364	0.009	0.601	0.000

Table 3. "Saturation" of the S^3 score. We compare the S^3 score of the perfect (NC=1) alignment to the S^3 score of alignments found by SANA and other algorithms that also optimize S^3, when aligning syeast0 (1004 nodes, 8323 edges) to other yeast networks. We see that SANA gets within a about 1.5% of the S^3 score of the perfect alignment, while having NC $\ll 1$, thus demonstrating that significantly less-than-perfect alignments exist with virtually the same S^3 score as the perfect one. Meanwhile the other aligners achieve significantly lower scores.

with 0s and 1s encoding the perfect (identity) mapping, and then set α close to 1 in Equation (1) so that the scoring function is heavily weighted towards this information. If a search algorithm fails to consistently produce a node correctness close to 1 in this case, it suggests the search algorithm is inoptimal, possibly getting stuck in local minima. Figure 2 demonstrates that while many of the algorithms do reasonably well in this case, some of them do surprisingly badly. SANA is the only search algorithm that produces the correct answer (NC=1) in every case tested.[4]

3.4.2 Topology-only comparisons of BioGRID networks with unknown mapping The Supplementary material contains figures demonstrating that SANA significantly outperforms all other methods in EC, S^3, ICS, and LCCS, when optimizing only topology. In particular, it also outperforms the other random search methods MAGNA++ and OptNetAlign, when all three are set to optimize the same objective function. Futhermore, it does so in only minutes using a single core compared to the other algorithms, which require hours or days of multi-core CPU time.

3.5 GO-term functional similarity comparisons

Figure 1 and our discussion in §3.4.1 strongly suggest that all current biological network alignments should be viewed with suspicion, since nobody can acheive a good NC score in the case of aligning networks of different sizes, even though all other measures can look good. Nonetheless, it is customary to perform a functional similarity comparison of aligners, so we now do this.

Figure 3 shows a subset of results (full set in Supplementary material) of how SANA compares to the other methods in terms of topological and functional quality of the alignments, when non-zero values of α from Equation 1 are allowed. Since Resnik semantic similarity of GO terms (Resnik, 1995) correlates strongly with sequence similarity (see Supplementary info), we depict only the former of the two in our plots. Since some GO terms are inferred from sequence similarity, Figure 3 allows only experimentally-verified GO terms in the computation of the Resnik similarity.

As can be seen, SANA significantly outperforms all other aligners in S^3 score, for every pair of networks tested. The Supplementary material shows that SANA also uniformly outperforms all aligners for all network pairs in the topolological measures EC and ICS; in LCCS its dominance is not universal, but we argue in the Supplemental section that this is not important. In terms of the functional quality of the alignments, SANA is still on top but by a less wide margin. Figure 3 shows only the Resnik maximum semantic similarity across all experimentally verified GO terms, while the Supplementary info also plots sequence, and the Resnik GO term scores for biological process, cellular component, and molecular function. While SPINAL sometimes matches or slightly beats SANA in functional similarity, it does so at a high cost to the topological quality of its alignments (Figure 3). As pointed out by Meng *et al.* (2015), local alignments tend to perform better than global alignments in functional similarity score, and SANA is no exception to this rule. We hypothesize that this a consequence of most global aligners forcing a globally 1-to-1 mapping on the alignment, thus resulting in suboptimal placement of proteins that may legitimately claim a right to map to more than one location in the other network. Until this restriction is lifted (a likely nontrivial software enhancement), local alignments are likely to continue to have enhanced functional similarity scores.

The most striking part of Figure 3 is the comparison of the *ordering* of the aligners, best-to-worst, between the two sides of the figure. Except for SANA, every other aligner shows a tendency to either be able to produce high topological scores, or high functional similarity scores, but only SANA does both. This tendency of being able to score well in only one side of the figure is consistent no matter which topological score, or which biological relevance score, is used. (See Supplement for more examples.) The dominance of SANA becomes obvious if one creates a score which is the product of topology and functional similarity, since every other algorithm has its score reduced significantly by at least one of the two; this is depicted in Figure 4. It has been observed previously that there is a trade-off between the competing objectives of maximizing topological quality and maximizing sequence and/or functional similarityPatro and Kingsford (2012); Crawford *et al.* (2015); Meng *et al.* (2015); Clark and Kalita (2015). However, to our knowledge this is the first time that it has been shown that many algorithms are not capable of fully leveraging the trade-off to both ends of the spectrum.

Since Figure 3 represents the best value of the plotted score across many values of α, it is appropriate to discuss which values of α are represented in the Figure. In general most algorithms (including SANA) produce better topological scores when the α parameter is near the endpoint where topology is most heavily weighted, and vice versa. What is surprising is that (unlike SANA) most algorithms

[4] Note this test says nothing about the quality of the objective functions introduced by each of the papers we compare against, it only shows that the search algorithms they use are sub-optimal.

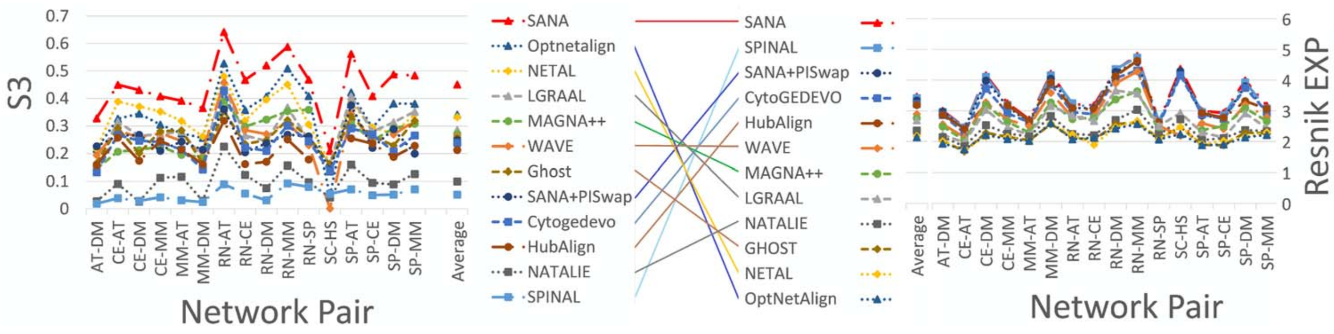

Fig. 3. Algorithms other than SANA produce alignments with a strong anti-correlation between topological quality and functional similarity. Both figures show the maximum stated score—S^3 (left) or Resnik Max GO term semantic similarity using only experimental terms (right)—of each aligner across each pair of BioGRID networks, maximized over the value of α in Equation 1. The legend listing the algorithms is sorted from best-to-worst on each side according to the "Average" column, and the criss-cross of colored lines in the middle show that any algorithm (other than SANA) that performs well on one side performs poorly on the other. Note in particular that the best on each side (other than SANA) is the worst on the other. We see a similar juxtaposition for any (topology, biology) pair of explicit measures.

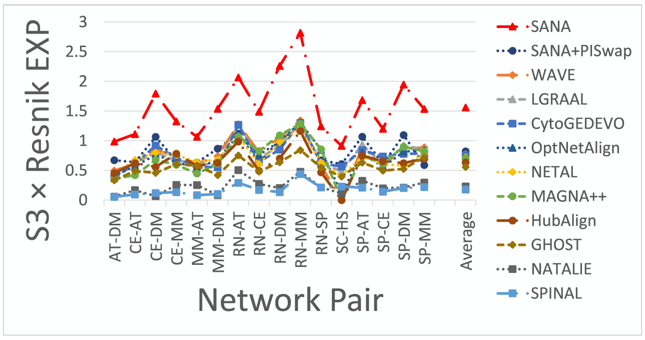

Fig. 4. Since most algorithms are incapable of produding both good topological scores and good functional scores simultaneously, here we plot the *product* of S^3 and the Resnik (Experimental) semantic similarity across all algorithms, for all pairs of networks. Using this measure, it is clear that SANA significantly outperforms all other algorithms in its ability to produce alignments both with high topological score, and high funnctional similarity.

produce good results at only one endpoint, not both. One would expect that if an algorithm produces good results at one endpoint of α, it would progressively get better results in the other measure as α moves towards that end of the spectrum, but this is not observed to be a very strong effect in any case except SANA's.

4 DISCUSSION

Like MAGNA++ and OptNetAlign, SANA is a random search algorithm. Such algorithms can be designed to optimize *any* function. We want to emphasize that such algorithms clearly separate the *search* part of the method, from the *objective function* being optimized. Such is not the case for many other algorithms. For example, any seed-and-extend approach (Kuchaiev *et al.*, 2010; Kuchaiev and Pržulj, 2011; Aladağ and Erten, 2013; Hashemifar and Xu, 2014) will implicitly enforce connectivity to the common subgraph due to the very nature of the "extend" part of the algorithm, which is only allowed to follow edges out of the partly-aligned structure that is being built. The resulting implicit topology is probably ill-defined and not well understood, and may depend upon minute details of the algorithm and its implementation. No such dependency exists for a random search. In a random search, one must think very carefully about *exactly* what one wants to optimize, including all elements of topology and biology. Given how badly existing algorithms can reproduce Node Correctness in differently sized networks as depicted in the last two columns of Figure 1, it is probably a very good thing to be forced to think about exactly *what* one is trying to optimize.

The catch is the objective function must *explicitly* list *all* the properties you want your solution to have. For example, we programmed the "importance" objective function from HubAlign (Hashemifar and Xu, 2014) directly into SANA and produced aligments that scored better in importance than the alignments produced by HubAlign itself; unfortunately, they compared terribly against HubAlign in terms of EC, S^3, and LCCS. This is because HubAlign uses a seed-and-extend approach, which enforces topological connectivity in the alignment. Without an *explicit* connectivity criterion in our objective function, SANA's alignments produced pairs that had *very* high "importance similarity", without regard to connectivity. Simply adding a tiny amount of S^3 into our objective function solved the problem—but it forced us to *think* about what we wanted in our alignments. As long as you know what you want (and even if you don't), SANA will give it to you, quickly and effectively.

It is clear that when SANA explicitly optimizes a particular objective function (such as EC or S^3), it does so better than many existing methods, and it does so far more quickly, using far less RAM, than existing algorithms. (See Tables 1 and 3 above, plus Suppl. Table 1.)

What is less clear is whether SANA's success depends on the search algorithm, or the fact that we used S^3 (Saraph and Milenković, 2014) as our topological objective function. Certainly SANA's speed is also a factor (§2.2.2). Far more detail on this question will be addressed in a forthcoming paper where we will program various objective functions into SANA and see how its alignments compare to those of the aligners supplying the objective functions. Assuming that SANA compares favorably in these cases, we believe that most further research into biological network alignment should focus on creating better objective functions. We humbly submit that, unless something better comes along (and the saturation depicted in Table 3 suggests it may be unlikely that anything significantly better exists), SANA may be the best currently available choice among search algorithms.

ACKNOWLEDGMENTS

NM was supported by the Balsells fellowship. We are grateful to Tijana Milenković, Vikram Saraph, Noël Malod-Dognin, and Oleksii Kuchaiev for directions on how to perform sequence and GO similarity analysis. Viet Ly wrote the Resnik comparison scripts based on FastSemSim. Charlotte Deane suggested using NC as the perfect objective function. The following students made contributions to programming, comparing to other methods, and data acquisition and analysis: Dillon Kanne, Evan Palmer, Pasha Khosravi, Daniel Weng Lawrence, Yvonne Kaire, Alyssa Marie Tan, Po-Chien Chung, Huy Ba Ngo, Heyang Chen, Zhennan Fu, Chifeng Wen, Chad Thompson, Trenton Coleman, Yuki Sawa, Arshia Dolatabadi, Austin Hwang.

REFERENCES

Aladağ,A.E. and Erten,C. (2013) Spinal: scalable protein interaction network alignment. *Bioinformatics,* **29** (7), 917–924.

Alkan,F. and Erten,C. (2014) Beams: backbone extraction and merge strategy for the global many-to-many alignment of multiple ppi networks. *Bioinformatics,* **30** (4), 531–539.

Ashburner,M., Ball,C.A., Blake,J.A., Botstein,D., Butler,H., Cherry,J.M., Davis,A.P., Dolinski,K., Dwight,S.S., Eppig,J.T., Harris,M.A., Hill,D.P., Issel-Tarver,L., Kasarskis,A., Lewis,S., Matese,J.C., Richardson,J.E., Ringwald,M., Rubin,G.M. and Sherlock,G. (2000) Gene Ontology: tool for the unification of biology. *Nature Genetics,* **25** (1), 25–29.

Bayati,M., Gerritsen,M., Gleich,D.F., Saberi,A. and Wang,Y. (2009) Algorithms for large, sparse network alignment problems. In *Proceedings of the 9th IEEE International Conference on Data Mining* pp. 705–710.

Camacho,C., Coulouris,G., Avagyan,V., Ma,N., Papadopoulos,J.S., Bealer,K. and Madden,T.L. (2009) Blast+: architecture and applications. *BMC Bioinformatics,* **10**, 421.

Chatr-aryamontri,A., Breitkreutz,B.J., Heinicke,S., Boucher,L., Winter,A., Stark,C., Nixon,J., Ramage,L., Kolas,N., ODonnell,L., Reguly,T., Breitkreutz,A., Sellam,A., Chen,D., Chang,C., Rust,J., Livstone,M., Oughtred,R., Dolinski,K. and Tyers,M. (2013) The biogrid interaction database: 2013 update. *Nucleic Acids Research,* **41** (D1), D816–D823.

Chindelevitch,L., Ma,C.Y., Liao,C.S. and Berger,B. (2013) Optimizing a global alignment of protein interaction networks. *Bioinformatics,* **29** (21), 2765–2773.

Clark,C. and Kalita,J. (2014) A comparison of algorithms for the pairwise alignment of biological networks. *Bioinformatics,* **30** (16), 2351–2359.

Clark,C. and Kalita,J. (2015) A multiobjective memetic algorithm for ppi network alignment. *Bioinformatics,* **31** (12), 1988–1998.

Collins,S.R., Kemmeren,P., Zhao,X.C., Greenblatt,J.F., Spencer,F., Holstege,F.C.P., Weissman,J.S. and Krogan,N.J. (2007) Toward a comprehensive atlas of the physical interactome of saccharomyces cerevisiae. *Molecular and Cellular Proteomics,* **6** (3), 439–450.

Cook,S.A. (1971) The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing* STOC '71 pp. 151–158 ACM, New York, NY, USA.

Crawford,J. and Milenković,T. (2015) Great: graphlet edge-based network alignment. In *Bioinformatics and Biomedicine (BIBM), 2015 IEEE International Conference on* pp. 220–227 IEEE.

Crawford,J., Sun,Y. and Milenković,T. (2015) Fair evaluation of global network aligners. *Algorithms for Molecular Biology,* **10** (1), 1.

Davis,D., Yaveroğlu,O.N., Malod-Dognin,N., Stojmirovic,A. and Pržulj,N. (2015) Topology-function conservation in protein-protein interaction networks. *Bioinformatics,* .

El-Kebir,M., Heringa,J. and Klau,G.W. (2011) Lagrangian relaxation applied to sparse global network alignment. In *IAPR International Conference on Pattern Recognition in Bioinformatics* pp. 225–236 Springer.

Elmsallati,A., Clark,C. and Kalita,J. (2015) Global alignment of protein-protein interaction networks: a survey. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on,* **PP** (99), 1–1.

Faisal,F.E., Meng,L., Crawford,J. and Milenković,T. (2015) The post-genomic era of biological network alignment. *EURASIP Journal on Bioinformatics and Systems Biology,* **2015** (1), 1.

Geman,S. and Geman,D. (1984) Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence,* **PAMI-6** (6), 721–741.

Gligorijević,V., Malod-Dognin,N. and Pržulj,N. (2015) Fuse: multiple network alignment via data fusion. *Bioinformatics,* , btv731.

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

Hu,J., Kehr,B. and Reinert,K. (2014) NetCoffee: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks. *Bioinformatics,* **30** (4), 540–548.

Ibragimov,R., Malek,M., Guo,J. and Baumbach,J. (2013) Gedevo: an evolutionary graph edit distance algorithm for biological network alignment. In *OASIcs-OpenAccess Series in Informatics* vol. 34, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.

Ingber,L. (1989) Very fast simulated re-annealing. *Mathematical and computer modelling,* **12** (8), 967–973.

Ito,T., Tashiro,K., Muta,S., Ozawa,R., Chiba,T., Nishizawa,M., Yamamoto,K., Kuhara,S. and Sakaki,Y. (2000) Toward a proteinprotein interaction map of the budding yeast: a comprehensive system to examine two-hybrid interactions in all possible combinations between the yeast proteins. *Proceedings of the National Academy of Sciences,* **97** (3), 1143–1147.

Kelley,B.P., Yuan,B., Lewitter,F., Sharan,R., Stockwell,B.R. and Ideker,T. (2004) Pathblast: a tool for alignment of protein interaction networks. *Nucleic Acids Res,* **32**, 83–88.

Kirkpatrick,S., Gelatt,C.D. and Vecchi,M.P. (1983) Optimization by Simulated Annealing. *Science,* **220**, 671–680.

Klau,G. (2009) A new graph-based method for pairwise global network alignment. *BMC Bioinformatics,* **10** (Suppl 1), S59.

Krogan,N., Cagney,G., Yu,H., Zhong,G. and Guo,X. (2006) Global landscape of protein complexes in the yeast saccharomyces cerevisiae. *Nature,* **440**, 637–643.

Kuchaiev,O., Milenković,T., Memišević,V., Hayes,W. and Pržulj,N. (2010) Topological network alignment uncovers biological function and phylogeny. *Journal of The Royal Society Interface,* **7** (50), 1341–1354.

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

Larsen,S.J., Alkærsig,F.G., Ditzel,H.J., Jurisica,I., Alcaraz,N. and Baumbach,J. (2016) A simulated annealing algorithm for maximum common edge subgraph detection in biological networks. In *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference* pp. 341–348 ACM.

Li,J., Tang,J., Li,Y. and Luo,Q. (2009) Rimom: A dynamic multistrategy ontology alignment framework. *IEEE Trans. Knowl. Data Eng.,* **21** (8), 1218–1232.

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

Malek,M., Ibragimov,R., Albrecht,M. and Baumbach,J. (2016) Cytogedevoglobal alignment of biological networks with cytoscape. *Bioinformatics,* **32** (8), 1259–1261.

Malod-Dognin,N. and Pržulj,N. (2015) L-graal: lagrangian graphlet-based network aligner. *Bioinformatics,* .

Memisevic,V. and Pržulj,N. (2012) C-graal: common-neighbors-based global graph alignment of biological networks. *Integr. Biol.,* **4**, 734–743.

Meng,L., Striegel,A. and Milenkovic,T. (2015). Local versus global biological network alignment.

Milenković,T., Ng,W.L., Hayes,W. and Pržulj,N. (2010) Optimal network alignment with graphlet degree vectors. *Cancer Informatics,* **9**, 121–137.

Milenković,T., Zhao,H. and Faisal,F.E. (2013) Global network alignment in the context of aging. In *Proceedings of the International Conference on Bioinformatics, Computational Biology and Biomedical Informatics* BCB'13 pp. 23:23–23:32 ACM, New York, NY, USA.

Mitra,D., Romeo,F. and Sangiovanni-Vincentelli,A. (1985) Convergence and finite-time behavior of simulated annealing. In *Decision and Control, 1985 24th IEEE*

Conference on pp. 761–767 IEEE.

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

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

Pržulj,N., Corneil,D.G. and Jurisica,I. (2004) Modeling interactome: scale-free or geometric? *Bioinformatics,* **20** (18), 3508–3515.

Radivojac,P., Peng,K., Clark,W.T., Peters,B.J., Mohan,A., Boyle,S.M. and Mooney,S.D. (2008) An integrated approach to inferring genedisease associations in humans. *Proteins: Structure, Function, and Bioinformatics,* **72** (3), 1030–1037.

Resnik,P. (1995) Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1* IJCAI'95 pp. 448–453 Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

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

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

Sun,Y., Crawford,J., Tang,J. and Milenkovič,T. (2015) Simultaneous optimization of both node and edge conservation in network alignment via WAVE. In *Algorithms in Bioinformatics*, (Pop,M. and Touzet,H., eds), vol. 9289, of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg pp. 16–39.

Szu,H. and Hartley,R. (1987) Fast simulated annealing. *Physics letters A,* **122** (3-4), 157–162.

Uetz,P., Dong,Y.A., Zeretzke,C., Atzler,C., Baiker,A., Berger,B., Rajagopala,S.V., Roupelieva,M., Rose,D., Fossum,E. and Haas,J. (2006) Herpesviral protein networks and their interaction with the human proteome. *Science,* **311** (5758), 239–42.

Černý,V. (1985) Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm. *Journal of Optimization Theory and Applications,* **45** (1), 41–51.

Vijayan,V. and Milenkovic,T. (2016). Multiple network alignment via multimagna++.

Vijayan,V., Saraph,V. and Milenković,T. (2015) Magna++: maximizing accuracy in global network alignment via both node and edge conservation. *Bioinformatics,* , btv161.

Zaslavskiy,M., Bach,F. and Vert,J.P. (2009) A path following algorithm for the graph matching problem. *IEEE Trans. Pattern Anal. Mach. Intell.,* **31** (12), 2227–2242.

Zhang,Y. and Tang,J. (2013) Social network integration: towards constructing the social graph. *CoRR,* **abs/1311.2670**.