NetCoffee: a fast and accurate global alignment approach to identify functionally conserved proteins in multiple networks

Jialu Hu1,*, Birte Kehr1,2 and Knut Reinert1

1Department of Mathematics and Computer Science, Freie Universität Berlin, Takustrasse 9, 14195 Berlin, Germany and 2Max Planck Institute for Molecular Genetics, Ihnestrasse 63-73, 14195 Berlin, Germany

ABSTRACT

Motivation: Owing to recent advancements in high-throughput technologies, protein–protein interaction networks of more and more species become available in public databases. The question of how to identify functionally conserved proteins across species attracts a lot of attention in computational biology. Network alignments provide a systematic way to solve this problem. However, most existing alignment tools encounter limitations in tackling this problem. Therefore, the demand for faster and more efficient alignment tools is growing.

Results: We present a fast and accurate algorithm, NetCoffee, which allows to find a global alignment of multiple protein–protein interaction networks. NetCoffee searches for a global alignment by maximizing a target function using simulated annealing on a set of weighted bipartite graphs that are constructed using a triplet approach similar to T-Coffee. To assess its performance, NetCoffee was applied to four real datasets. Our results suggest that NetCoffee remedies several limitations of previous algorithms, outperforms all existing alignment tools in terms of speed and nevertheless identifies biologically meaningful alignments.

Availability: The source code and data are freely available for download under the GNU GPL v3 license at https://code.google.com/p/netcoffee/

Contact: Jialu.Hu@fu-berlin.de

Supplementary information: Supplementary data are available at Bioinformatics online.

Received on May 27, 2013; revised on November 8, 2013; accepted on December 7, 2013

1 INTRODUCTION

Discovering functionally conserved proteins across different species is a fundamental task in comparative systems biology (Park et al., 2011; Tatusov et al., 1997). With the development of high-throughput technologies such as mass spectrometry (Ho, 2002), microarrays (Lashkari et al., 1997), yeast two-hybrid assays (Ito et al., 2001) and next-generation sequencing, a tremendous amount of genomic, proteomics and protein interaction data has been generated and became available in public databases (Szklarczyk et al., 2011; Uniprot Consortium, 2007). This comprehensive experimental data provide a basis for analyses that aim at discovering conservation of protein function among different species, i.e. functional orthologs (FO). Orthologs are genes/proteins of different species that have descended from the same gene in a common ancestor. The FO are orthologs whose functions are conserved in different species.

It is often assumed that two proteins with similar sequences or similar structure have similar functions, and conversely that functionally related proteins have similar sequences. Based on this assumption, a number of approaches that use sequence similarity have been developed, e.g. reciprocal-best-BLAST-hits (RBH), for predicting FO. This resulted in several orthologs databases such as the Clusters of Orthologous Groups (Tatusov et al., 2000), Inparanoid (O’Brien et al., 2005) and OrthoDB (Waterhouse et al., 2011). However, high sequence similarity does not necessarily indicate functional conservation. Because functional sites of proteins are usually only one or several small parts of the whole sequence, two proteins can have a highly significant overall similarity even though all functional sites are completely different (Brutlag, 2008). To overcome this problem, network alignment approaches (Bandyopadhyay et al., 2006; Liao et al., 2009; Shih and Parthasarathy, 2012) have been proposed that supplement sequence-based algorithms with information from protein–protein interaction (PPI) networks.

Network alignment approaches can be generally classified into pairwise or multiple and into local or global approaches. Pairwise approaches align two networks and multiple approaches three and more networks. Local alignment approaches detect conserved subnetworks of two (pairwise local alignment) or more (multiple local alignment) networks. These conserved subnetworks are usually independent high-scoring local regions of the compared networks, each implying a putative functional module such as a protein complex (Sharan, 2005) or a metabolic pathway (Kelley, 2003; Kelley et al., 2004). Global alignment approaches determine an overall alignment between the input networks (Singh et al., 2008). The resulting alignments can be used to transfer annotation from characterized to uncharacterized proteins. Typically, pairwise global alignment algorithms attempt to provide only a one-to-one node mapping between proteins of compared networks (El-Kebir et al., 2011; Kuchaiev and Prulj, 2011), whereas multiple global alignment algorithms attempt to provide a many-to-many node mapping (Flannick et al., 2009; Liao et al., 2009). A many-to-many node mapping allows to find a set of functionally similar proteins that have descended from the same protein in a common ancestor based on four types of evolutionary events: protein deletion, protein duplication, protein mutation and paralog mutation (Flannick et al., 2009). For more information on the difference between local and global alignment, see the Supplementary Material.

*To whom correspondence should be addressed.
Many pairwise local and pairwise global alignment tools have been developed in the last decade. Among pairwise local alignment tools are PathBlast (Kelley et al., 2004), MoWish (Koyutürk et al., 2005), NetworkBlast (Kalavä et al., 2008), NetAligner (Pache et al., 2012), PIALOG (Phan and Sternberg, 2012) and SIPPINAL (Aladag and Erten, 2013). A review by Sharan et al. (Sharan and Ideker, 2006) extensively discusses some of these methods. Well-known pairwise global alignment tools are IsoRank (Singh et al., 2007), PLSwap (Chindelevitch et al., 2010), MI-GRAAL (Kuchaiev and Prulj, 2011), Natalie 2.0 (El-Kebir et al., 2011) and GHOST (Patro and Kingsford, 2012).

With the increasing availability of PPI networks, the demand for local and global alignment tools of multiple networks has risen. Several multiple alignment tools have been developed, most notably the multiple global alignment tools Graemlin 2.0 (Flannick et al., 2009), IsoRank-N (Liao et al., 2009) and SMETA-N (Sahraei and Yoon, 2013), and the multiple local alignment tool NetworkBlast-M (Kalavä et al., 2009). However, these tools have some limitations. Graemlin 2.0 requires a dataset of known alignments to learn its many network-dependent parameters and a phylogenetic tree, which means it can not be applied to species without known alignments or without a phylogenetic tree. NetworkBlast-M does not work on networks containing protein nodes with large vertex degree, such as the yeast network in our test datasets. Both IsoRank-N and SMETA-N need a lot of computing time for aligning six or more species. Thus, there is a demand for tools that can deal with many networks in a more efficient way.

As a remedy for these limitations, we present a fast and accurate tool NetCoffee, which addresses the problem of global alignment of multiple networks. The algorithm implemented in NetCoffee has four main steps: (i) building the PPI networks and a library of bipartite graphs, (ii) assigning an integrated weight to each edge in the bipartite graphs using a triplet extension approach similar to T-Coffee, (iii) building a search space that consists of candidate edges (protein pairs) and (iv) simulated annealing (SA) with a large number of iterations of a Metropolis Scheme to maximize a scoring function for global alignments. We ran NetCoffee on four datasets consisting of up to six PPI networks. Our results show that NetCoffee overcomes the limitations of existing algorithms, outperforms all existing alignment tools in terms of speed and nevertheless identifies a biologically meaningful alignment.

## 2 METHODS

### 2.1 Definitions and notation

Let \( \{G_1, G_2, \cdots, G_k\} \) represent a set of \( k \geq 3 \) PPI networks. Each network \( G_i = (V_i, E_i) \) is an unweighted graph, where \( V_i \) is a set of nodes representing proteins and \( E_i \) a set of binary interactions appearing in the networks. We refer to elements of \( E_i \) as interactions to distinguish them from edges in a different type of graph below. Let \( V = \bigcup_{i=1}^{k} V_i \) be the union of all proteins. A match set \( \theta \) is a subset of \( V \). By definition, a global alignment of the \( k \) networks is a node mapping that consists of a set of mutually disjoint match sets, \( \{\Theta^1, \Theta^2, \cdots, \Theta^m\} \) with \( \Theta^i \cap \Theta^j = \emptyset, \forall i, j, i \neq j \). A match set can contain more than one node from each network.

Like a match set, a \( k \)-spine is a subset of \( V \) but contains exactly one protein from each network. In addition, two different \( k \)-spines can share nodes. In contrast to global alignments, a local alignment of \( k \) networks is a set of independent high-scoring local node mappings, each node mapping consisting of a set of \( k \)-spines (Kalavä et al., 2009).

### 2.2 Generating a bipartite graph library

Given \( k \) species and their corresponding PPI networks, we build a bipartite graph library, which contains a graph \( B_2 = (V_2 \cup V_3, E_2) \) for each pair of input networks \( G_1 \) and \( G_2 \), \( i \leq j, i, j \in \{1, 2, \cdots, k\} \). We use the term edges to refer to elements in \( E_2 \). To determine the sets \( E_2 \), we perform an all-against-all sequence comparison with the program BLASTP (Altschul et al., 1997) for each pair of species, including pairs of the same species like human-human. Then, the set of \( \binom{k}{2} \) bipartite graphs can be constructed by simply joining protein pairs \( v_1 \in V_1, v_2 \in V_1 \) that have an \( e \)-value \( \leq 10^{-7} \) by edges \( \{v_1, v_2\} \in E_2 \). In bipartite graphs \( B_2 \) of the same species, we add only edges for pairs of two distinct proteins \( v_1 \neq v_2 \) to \( E_2 \). This allows us to construct match sets that might reflect duplication events within a species and hence exhibit functional relation within a species.

### 2.3 Integration of two conservation measures

To search for a biologically meaningful alignment, we developed a linear scoring model that assigns a weight to each edge of the bipartite graphs. The development of the scoring model was intuitively guided by two basic assumptions: (i) functionally conserved proteins are likely to have sequence similarity and (ii) interactions among orthologous proteins are likely to be conserved across species. Likewise, our scoring model consists of two independent parts for sequence and topology similarity. Given an edge \( e = (v_1, v_2) \), we use \( S_{(v_1, v_2)} \) to denote a normalized sequence score and \( S_{(v_1, v_2)} \) to denote a normalized topology score for proteins \( v_1 \) and \( v_2 \). A combined score for the edge \( e \) is calculated with \( S(v_1, v_2) = aS_{(v_1, v_2)} + (1-a)S_{(v_1, v_2)} \), where \( a \) is a user-defined parameter controlling how much of the topology score contributes to \( S(v_1, v_2) \).

To compute the sequence-based score \( S_{(v_1, v_2)} \) for a pair of proteins \( v_1 \) and \( v_2 \), we adopt a previously introduced log-ratio scoring function that uses distributions of \( e \)-values in two models, the homology model \( H \) and the null model \( N \) (Flannick et al., 2006). The null model includes all pairs of proteins from the input networks, whereas the homology model includes only pairs of proteins with an \( e \)-value \( \leq 10^{-7} \). Given the distributions of \( e \)-values in these two models, we calculate the probabilities to observe the \( e \)-value \( e_{v_1, v_2} \) of the two proteins \( v_1 \) and \( v_2 \) in the two models, \( Pr(e_{v_1, v_2} | H) \) and \( Pr(e_{v_1, v_2} | N) \). Our normalized sequence score is the log-ratio \( y_{v_1, v_2} = \log \frac{Pr(e_{v_1, v_2} | H)}{Pr(e_{v_1, v_2} | N)} \) of these probabilities scaled to the range from 0 to 1 with the minimal observed log-ratio \( y_{\text{min}} \) and maximal observed log-ratio \( y_{\text{max}} \) of all protein pairs in the \( H \) model:

\[
S_{(v_1, v_2)} = \frac{y_{v_1, v_2} - y_{\text{min}}}{y_{\text{max}} - y_{\text{min}}}.
\]

To compute the topology-based score \( S_{(v_1, v_2)} \), we use a triplet approach that bears similarities to the concept of overlapping weights (Morgenstern, 1999) and T-Coffee’s consistency approach (Notredame et al., 2000) in multiple sequence alignment. Our approach is an incremental process with the final score reflecting the likelihood of a pair of proteins being topologically conserved. Initially, we set the topology-based scores of all edges in the \( \binom{k}{2} \) bipartite graphs of two different species to zero. After this initialization, each of the edges has an equal \( \alpha \)-value in these two models, we calculate the probabilities to observe the \( e \)-value \( e_{v_1, v_2} \) of the two proteins \( v_1 \) and \( v_2 \) in the two models, \( Pr(e_{v_1, v_2} | H) \) and \( Pr(e_{v_1, v_2} | N) \). Our normalized sequence score is the log-ratio \( y_{v_1, v_2} = \log \frac{Pr(e_{v_1, v_2} | H)}{Pr(e_{v_1, v_2} | N)} \) of these probabilities scaled to the range from 0 to 1 with the minimal observed log-ratio \( y_{\text{min}} \) and maximal observed log-ratio \( y_{\text{max}} \) of all protein pairs in the \( H \) model:

\[
S_{(v_1, v_2)} = \frac{y_{v_1, v_2} - y_{\text{min}}}{y_{\text{max}} - y_{\text{min}}}.
\]
three different PPI networks. A set of three nodes that are mutually connected by edges is a triplet match, e.g. \( \{A_1, A_2, A_3\} \) in Figure 1b. In the process of reweighing, we consider all pairs of triplet matches that are connected by conserved interactions in all three networks, such as the edges in line with fine dots in Figure 1b–c, and increase the score of each edge in the pair of triplet matches by one. In graphs of the same species, we set the topology-based score of all edges to zero.

All edge scores of the two species 1 and 2 are illustrated in Figure 1f. As an example, the overall topology-based score for the two proteins \( B_1 \) and \( B_2 \) in Figure 1f is five, which is explained as follows: In Figure 1b the conserved interaction between \( \{B_1, C_1\} \) and \( \{B_2, C_2\} \) is confirmed by \( \{B_1, C_1\} \), and hence the triplet matches \( \{B_1, B_2, B_1\} \) and \( \{C_1, C_2, C_1\} \) are completely connected by interaction edges contributing one to the score. In Figure 1b, the triplet matches \( \{A_1, A_2, A_3\} \) and \( \{B_1, B_2, B_1\} \) do not contribute because of the missing interaction edge \( \{A_1, B_3\} \). In Figure 1c, the four combinations of triplet matches \( \{B_1, B_2, B_3\}, \{A_1, A_2, A_4\}, \{B_1, B_2, A_1\}, \{A_1, D_2, C_3\} \) and \( \{B_1, B_2, B_3\}, \{A_1, D_2, C_3\} \) contribute four to the score. For more details for this example including the other weighted bipartite graphs and pseudo-code for the triplet comparison see the Supplementary Material.

After this process, each edge of the bipartite graphs has been assigned a topology-based score, which we normalize to the range between 0 and 1. However, the distribution of these scores is extremely non-uniform, as the connectivity of biological networks follows a power-law distribution (Barabasi and Albert, 1999). A few edges between hub-nodes have a topology score close to the maximal score, and many others are close to zero. For example, \(-90\%\) of the protein pairs have a normalized topology score between 0 and 0.1 in our dataset 2. In contrast, \(95\%\) of the protein pairs has a normalized sequence score between 0.6 and 1. This implies that small normalized topology scores might still be statistically significant and indicate a high probability of functional relatedness. A large number of protein pairs have a small normalized topology score because the maximal score is large. Therefore, we lift the small scores up using a power-law redistribution to make sure that the topology score has a reasonable impact on the whole alignment score (for details see the Supplementary Material). This concludes the computation of the edge scores \( S(v_i, v_j) \), where each score now reflects sequence similarity and topology conservation.

### 2.4 Alignment algorithm

Our algorithm for aligning multiple networks first collects candidate edges from the \( (\cdot, \cdot) \) bipartite graphs. Subsequently, the algorithm combines some of these candidate edges to a global alignment with the meta-heuristic method SA (Kirkpatrick et al., 1983). The collection of candidate edges reduces the computational complexity while retaining the sensitivity and specificity of the algorithm in praxis.

#### 2.4.1 Collection of candidate edges

We have given \( (\cdot, \cdot) \) weighted bipartite graphs, \( (\cdot) \) of which formed by proteins from two different species. The weights of all edges in \( B_{ij}, i<j \) reflect the likelihoods of the edges to be a true match of the global alignment, including information about sequence and topology conservation. We use a maximum weighted matching algorithm, namely, Edmond’s Algorithm (Galil, 1983), to find a one-to-one node mapping in each of the \( (\cdot) \) bipartite graphs and collect the matching edges as candidate edges. Furthermore, we collect protein pairs of the same species with scores higher than a threshold \( \alpha = \eta (1 - \alpha) \). The parameter \( \eta \) is user-defined and enables our method to identify match sets formed by proteins of one species. The term \((1 - \alpha)\) accounts for the fact that the topology score of these edges is always 0. We obtain a collection of candidate edges, denoted as \( \Omega \).

#### 2.4.2 Multiple alignment

To find a multiple global alignment \( A \in \Omega \), we define the scoring function \( \Phi(A) = \sum_{i \in A} f(\theta) \), where \( f(\theta) \) is the score of a match set \( \theta = \{v_1, v_2, \ldots, v_n\} \). The score of \( \theta \) is calculated with the function \( f(\theta) = \sum_{(v_i, v_j) \in \theta} S(v_i, v_j) \delta_{ij} \), where \( \delta_{ij} = 1 \) if \( v_i, v_j \in V \), otherwise \( \delta_{ij} = 0 \).

Let \( I \) be the collection of all possible global alignments. Then, the problem of multiple global alignment can be modeled as an optimization problem \( \max_{A \in I} \Phi(A) \). We use an SA approach to approximate the highest-scoring alignment. Unlike the strategy of progressive alignment (Flannick et al., 2006), which successively aligns closest pairs of networks and constructs a new network alignment, our SA approach starts with an empty alignment of all networks and runs a large number of iterations of a Metropolis scheme (Metropolis et al., 1953) to maximize \( \Phi(A) \).
Algorithm 1 Simulated annealing algorithm

Input: Matching edges \( \Omega \), \( K \), \( T_{\text{min}} \), \( T_{\text{max}} \), \( s \)

Output: A solution \( x^* \) with a set of mutually disjoint match sets

1: \( x = \emptyset, T_0 = T_{\text{max}}, l = 1; \)
2: while \( i \leq K \) do
3: \( n = 0; \)
4: \( T_{l} = T_{p} \left( \frac{\exp(-x)}{\exp(-x)} \right) \)
5: while \( n < N \) do
6: draw arbitrary sample \( \xi \in \Omega \) from uniform distribution;
7: \( x' = \text{updateState}(x, \xi) \);
8: \( \Delta \Phi = \Phi(x') - \Phi(x); \)
9: if \( \Delta \Phi > 0 \) then
10: \( x = x' \);
11: else \( \text{rand}(0, 1) < \exp(\Delta \Phi/(sT_l)) \)
12: \( x = x' \);
13: end if
14: \( n = n + 1; \)
15: end while
16: \( i = i + 1; \)
17: end while
18: \( x^* = x; \)
19: return \( x^* \);

The pseudo-code of the SA approach is given in Algorithm 1. Let \( x \in I \) be a feasible solution (a set of mutually disjoint match-sets) for the problem and \( \Phi(x) \) the alignment score of \( x \). At the beginning of the algorithm, we initialize our alignment \( x \) with \( \emptyset \) and set a temperature parameter \( T_0 \) to its maximum. In the following annealing phase, we decrease the temperature and repeatedly perturb the current solution \( x \) with a Metropolis scheme using \( \pi \propto \exp(-\Phi(x)/(sT_l)) \) as the equilibrium distribution. Parameters \( s, K, N, T_{\text{min}} \), and \( T_{\text{max}} \) control the SA. The \( \text{updateState} \) \((x, \xi)\) updates the current alignment with an arbitrary sample \( \xi \in \Omega \). It runs into four possible scenarios. Let \( u \notin x \) indicate that \( \forall v \in x, u \notin v \), and \( u \in x \) indicate that \( \exists v \in x \) such that \( u \notin v \). Then, the scenarios are: (i) \( u \in x \) and \( u \notin x \); (ii) \( u \notin x \) and \( v \in x \); (iii) \( u \in x \) and \( v \notin x \); and (iv) \( u \in x \) and \( v \in x \), but \( u \) and \( v \) are not in the same match set. In the first scenario, \( u \) and \( v \) are moved to the current alignment \( x \) as a new match set. In the other scenarios, \( u \) and \( v \) are moved to the same match set of the alignment \( x \) in two possible ways, called combination and substitute. Details are described in the Supplementary Material. We continue this process until the ‘temperature’ \( T_l \) decreases to \( T_{\text{min}} \).

### 2.5 Complexity analysis

We assume that the number of proteins in the largest PPI network is \( n \), and the number of input networks is \( k \). The pseudo-code of the triplet approach (see Supplementary Material) has a complexity of \( \Omega(n^2) \). Suppose there is a bipartite graph, \( B_i = \langle V_i, \cup V_i, E_i \rangle \), the running time complexity of Edmond’s Algorithm on \( B_i \) is \( O(V_i + U_i) \log |E_i| \). Therefore, the collection of candidate edges \( x \) is \( \Omega(n \log n) \) time.

The convergence time of SA has been a widely studied question in the last two decades. We assume \( \Delta = \max(\Phi(x) - \Phi(x)) \), where \( \Delta \) is a neighbor state of state \( x \). As shown by the proof in Rajasekaran (1990), SA converges (at any temperature) in time \( 2\beta(\exp(\Delta/sT_l)) \), where \( D \) is the diameter, \( d \) is the degree of the underlying Markov chain and \( \beta \) is defined by the convergence probability \( 1 - 2^{-d} \). Theoretically, \( D \) and \( d \) are hard to calculate. However, in practice, the complexity of SA only depends on two parameters of the cooling scheme, \( K \) and \( N \). From Algorithm 1, we can easily find that the complexity is \( \Omega(K \cdot N) \), which is independent of the number of compared species \( k \). To sum up, practically, \( \text{NetCoffee} \) can be used with multiple networks and has a favorable time complexity. Our results show that the alignment score converges rapidly in our experiments (see Supplementary Fig. S5).

### 3 RESULTS AND DISCUSSION

#### 3.1 Test datasets

We have evaluated \( \text{NetCoffee} \) on three datasets of up to five eukaryotic species and one dataset of six microbes as shown in Table 1. The five eukaryotic species include Homo sapiens (human), Mus musculus (mouse), Drosophila melanogaster (fruit fly), Caenorhabditis elegans (nematode) and Saccharomyces cerevisiae (yeast). The six microbes include Escherichia coli, Salmonella typhimurium, Vibrio cholerae, Campylobacter jejuni NCTC 11168, Helicobacter pylori 2695 and Caulobacter crescentus.

To build the five eukaryotic networks of dataset 0, 1 and 2, we collected all experimentally determined interactions from the public database IntAct (Kerrien et al., 2012). In addition, we collected the reference proteome sets of the five species from UniProtKB/Swiss-Prot release 2012_07 (Uniprot Consortium, 2007), which are used for all-against-all sequence comparisons. To make sure the proteins in our networks are non-redundant and well-annotated, we discarded interactions between proteins that are not in the reference proteome sets. The number of proteins and interactions of these PPI networks are given in Table 1. Dataset 3 is the same dataset used in the original publication of Graemlin 2.0 (Flannick et al., 2009).

For analyzing the biological quality of the alignments, gene ontology (GO) information was collected from UniProt-GOA (Camon et al., 2004) (downloaded on Jan. 8, 2013) to annotate proteins with the three basic types of ontologies: biological process (BP), molecular function (MF) and cellular component (CC). To exclude unreliable function annotations, GO annotations with evidence codes IEA (inferred from electronic annotation) and ISS (inferred from sequence or structural similarity) were discarded.

#### 3.2 Experimental setup

We have implemented \( \text{NetCoffee} \) in C++ using the LEMON Graph Library (Dezső et al., 2011) version 1.2.3. The implementation supports multicore parallelism for the triplet comparison. We ran \( \text{NetCoffee} \) on all four datasets and tuned its SA parameters such that the SA process converges to a stable score (see Supplementary Fig. S5). The default values are now \( s = 0.005 \), \( K = 100 \), \( N = 2000 \), \( T_{\text{min}} = 10 \), \( T_{\text{max}} = 10 \) and \( \eta = 1.0 \).

To compare \( \text{NetCoffee} \) with the state-of-the-art algorithm IsoRank-N, we executed IsoRank-N on the same datasets with recommended parameters: \( K = 20 \), thresh = \( 10^{-4} \), maxvdecen = \( 10^6 \). Additionally, NetworkBlast-M, Graemlin 2.0 and SMETANA were included in our assessment. However, NetworkBlast-M is unable to work on dataset 0, 2 and 3 for two reasons. First, the yeast network has proteins with up to 3276 interactions, which is prohibitive for NetworkBlast-M. Second, NetworkBlast-M requires e-values as a protein similarity measure, but dataset 3 provides only bitscores. Furthermore, we ran Graemlin 2.0 only on dataset 3 because it needs additional training data (i.e. known alignments for the compared species) to learn its parameters. Because Graemlin 2.0 identifies match sets whose proteins are from a single species, we set \( \eta = 0.7 \) for dataset 3 to allow a fair comparison with Graemlin 2.0.
We input the networks of the species in the same order for all programs, namely, the order from Table 1. Only the results of IsoRank-N depend on the order of input species. All experiments mentioned in the following parts were carried out on the same machine, an Intel(R) Xeon(R) CPU X5550 with 2.67GHz.

### 3.3 Performance comparison

We demonstrate the quality of our alignments in terms of coverage and consistency and assess the performance of our method by measuring running times. Coverage, which serves as a proxy for sensitivity, indicates the amount of input data the algorithm can explain. Consistency, which serves as a proxy for specificity, measures the functional similarity of proteins in each match set. Coverage can be easily achieved by sacrificing consistency and vice versa. The running time demonstrates the ability of NetCoffee to deal with large datasets. Intuitively, the goal is to find a global alignment that has a good consistency while explaining as many proteins as possible (i.e. high coverage) in reasonably short time. We first look at differences the programs exhibit in coverage and then investigate the consistency of the match sets with three measures. Next, we compare running times and, finally, demonstrate how much NetCoffee benefits from the integration of similarity and topology score by addressing the influence of the parameter $\alpha$.

#### 3.3.1 Coverage

For each program, we calculated the percentage of proteins (PPV) in the whole set of proteins that are covered by the alignment as the coverage (see Table 2). In comparison with IsoRank-N and NetworkBlast-M, the coverage of NetCoffee is significantly higher. For instance, the PPV of NetCoffee is up to 41.8% for dataset 1, whereas it is only 31.1% for IsoRank-N and 16.1% for NetworkBlast-M. The lower coverage of these two alignment tools can be explained by the facts that NetworkBlast-M is a local aligner and, thus, considers only conserved modules; IsoRank-N aligns proteins of at least three species into match sets and does not report match sets of proteins from only two species (These match sets can be recognized by running the pairwise aligner IsoRank on each pair of species.) (see an example in Supplementary Table S1). In comparison with Graemlin 2.0, NetCoffee also has a slightly higher PPV value except for the extreme case of $\alpha = 1$. When $\alpha = 1$, sequence scores of all pairs of proteins are set to 0 in NetCoffee. As a result, all protein pairs from a single species are excluded from the collection of candidate edges and consequently from the alignment. Hence, the coverage drops to 69.7% for dataset 3. In comparison with SMETANA, the coverage of NetCoffee is similar. NetCoffee achieves a lower PPV for dataset 0, 1 and 2, but a higher PPV for dataset 3. Concerning the number of match sets, IsoRank-N identifies more match sets formed by proteins from three of the compared species, and both Graemlin 2.0 and SMETANA find more match sets for dataset 3 than NetCoffee except for $\alpha = 1$ (see Supplementary Table S1).

#### 3.3.2 Consistency

An alignment tool that achieves a high coverage is not necessarily better than others. For example, a random global alignment may cover all proteins but aligns many unrelated proteins. Hence, we now address the performance of the alignment tools in terms of consistency. Consistency demonstrates the biological significance of predicted match sets.

As a first consistency measure, we computed the mean entropy and the mean normalized entropy of the predicted match sets in the alignments of each algorithm. We calculated the entropy of a match set with the same method as in IsoRank-N according to its GO annotations. A match set has lower entropy if its GO annotations are more functionally coherent. From Table 2, we can see that the entropy of NetCoffee is considerably lower than that of IsoRank-N and NetworkBlast-M no matter which $\alpha$ was used, whereas at the same time having a high coverage. Additionally, the entropy of NetCoffee is lower than that of SMETANA on all datasets except for dataset 3. In comparison with Graemlin 2.0, NetCoffee achieves nearly identical entropy results for dataset 3, whereas being considerably faster. The results for $\alpha = 0$ and $\alpha = 1$ demonstrate that both of our two conservation measures can favorably predict the functional relatedness between protein pairs.

Dataset 3 exhibits an interesting trade-off using the $\alpha$ parameter in terms of coverage and consistency. For $\alpha = 1$, NetCoffee has the lowest entropy, however, at the cost of a much lower coverage. Decreasing $\alpha$ improves the coverage while deteriorating the entropy measure. This behavior is less pronounced for the other datasets. However, it shows that the $\alpha$ parameter can be used for having a specificity versus sensitivity trade-off.

### Table 1. The number of proteins and PPI of four datasets that consist of the PPI networks from 11 species

<table>
<thead>
<tr>
<th>Species</th>
<th>Proteins</th>
<th>Interactions</th>
<th>Dataset 0</th>
<th>Dataset 1</th>
<th>Dataset 2</th>
<th>Dataset 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>H.sapiens</td>
<td>8777</td>
<td>28366</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>M.musculus</td>
<td>1531</td>
<td>1626</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>D.melanogaster</td>
<td>1534</td>
<td>2664</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>C.elegans</td>
<td>767</td>
<td>915</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>S.cerevisiae</td>
<td>5739</td>
<td>36226</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>E.coli</td>
<td>4719</td>
<td>169636</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>V.cholerae</td>
<td>3044</td>
<td>76341</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>C.jejuni 11168</td>
<td>1424</td>
<td>76913</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>H.pylori 26695</td>
<td>1206</td>
<td>48430</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>C.crescentus</td>
<td>3022</td>
<td>52302</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>S.typhimurium</td>
<td>4326</td>
<td>151118</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Second, we assessed consistency by three elaborate semantic similarity measures introduced in Schlicker et al. (2006, 2007): \(BPscore\), \(MFscore\) and \(rfunSim\). Unlike many existing approaches (El-Kebir et al., 2011; Kuchaiev and Prulj, 2011) that simply evaluate functional similarity by counting the number of common GO terms involved proteins, \(BPscore\) and \(MFscore\) assess the functional similarity of two proteins by exploiting BP and MF annotations with the GO hierarchy tree. The measure \(rfunSim\) is a combination of \(BPscore\) and \(MFscore\) (for details see the Supplementary Material). We report the arithmetic mean of the similarity scores of all involved protein pairs as the functional consistency of a match set. For instance, given a match set \(\theta = (v_1, v_2, \ldots, v_n)\), the functional consistency of \(\theta\) with respect to the BP annotation is defined as
\[
BPscore(\theta) = \frac{\sum_{i,j} BPscore(v_i, v_j)}{|\theta|}, \quad i,j \in \{1, 2, \ldots, |\theta|\}.
\]

Analogously, we can calculate \(MFscore\) and \(rfunSim\). All three scores range from 0 to 1, which translates into an increasing degree of functional similarity. We calculated the scores using the functional similarity search tool (FSST) (Schlicker et al., 2007). To avoid skipping too many meaningful match sets, match sets that contain <40% uncharacterized proteins were also taken into consideration. We separately compared match sets that contain proteins from 3, 4 and 5 species. And the distribution of match sets in each category can be seen in Supplementary Table S1.

We compared the consistency of NetCoffee with that of IsoRank-N (see Fig. 2) and SMETANA (see Supplementary Fig. S6) on their alignments of dataset 2. As shown in Figure 2a-c, when \(\alpha > 0\), the \(BPscore\) of NetCoffee is higher than that of IsoRank-N, and the \(MFscore\) and \(rfunSim\) are roughly the same. More importantly, the advantage of NetCoffee expands when \(i\) (i.e. the number of species) increases to 4, as shown in Figure 2d-f, although it identifies more match sets. NetCoffee shows significant improvements with regard to the \(BPscore\), \(MFscore\) and \(rfunSim\) except for the case of \(\alpha = 0\). When \(\alpha = 0\), IsoRank-N reaches its highest point. However, we do not recommend to use \(\alpha = 0\) for IsoRank-N, as its coverage drops to only 21.9%. For \(i=5\) illustrated in Figure 2g-i, IsoRank-N improves the quality of match sets in terms of \(BPscore\). The two algorithms are comparable in terms of \(MFscore\) and \(rfunSim\). However, NetCoffee identifies ~3–8 times more match sets than IsoRank-N (see Supplementary Table S1). Compared with the alignment of SMETANA, match sets identified by NetCoffee have lower semantic scores for \(i=3\) but roughly the same scores for \(i=4\) and \(i=5\) (see Supplementary Fig. S6).

Finally, we measured the consistency by computing the percentage of qualified match sets the algorithms identified. As demonstrated in Schlicker et al. (2006), almost 60% of protein pairs in the Inparanoid Orthologs (IO) dataset has an \(MFscore\) >0.8 and 65% has a \(BPscore\) >0.6. Therefore, we regard those match sets that have an \(MFscore\) >0.8 or a \(BPscore\) >0.6 as qualified match sets, i.e. functionally related proteins. With these thresholds, ~45% of the match sets recognized by NetCoffee are qualified match sets (see Supplementary Fig. S7), which is significantly more than those identified by IsoRank-N (~25%) and more than those identified by SMETANA (~42%). Visualizations of the GO trees for each qualified match set [drawn using the package GO::TermFinder](https://academic.oup.com/bioinformatics/article/30/4/540/204711)
3.3.3 Running time Table 2 demonstrates that our method is robust to the parameter \( \alpha \) in terms of running time. The running time of \textit{IsoRank-N}, however, increases dramatically when \( \alpha \) grows. Specifically, \textit{NetCoffee} is \( \sim 1-3 \) orders of magnitude faster than \textit{IsoRank-N}, 37 times faster than \textit{NetworkBlast-M}, 82 times faster than \textit{Graemlin 2.0} (including training time) and 2-26 times faster than \textit{SMETANA}. We chose to report the results achieved with multiple cores (i.e., eight cores) because they are the real running time for \textit{NetCoffee}. \textit{NetCoffee} is still faster than its competitors even on a single core except for \textit{SMETANA} (see Supplementary Table S2).

3.3.4 Influence of the parameter \( \alpha \) To figure out how much the alignment tools benefit from the topology and sequence score, we ran both \textit{NetCoffee} and \textit{IsoRank-N} with various \( \alpha \) values. If \( \alpha = 0 \), the global alignment is constructed only based on sequence score, and if \( \alpha = 1 \), only based on topology score.

Table 2 and Figure 2 demonstrate that \textit{NetCoffee} is robust to the parameter \( \alpha \) in terms of coverage, consistency and speed, and that the \( \alpha \) parameter can be used for having a specificity versus sensitivity trade-off. Both the topology and the sequence score favorably predict functional relatedness between protein pairs. However, using either sequence score or topology score alone is not favorable for the coverage of \textit{IsoRank-N} as shown in Table 2. Furthermore, the alignment quality and the computing time depend on \( \alpha \). Table 2 suggests that the performance of \textit{IsoRank-N} tends to be best at a value of \( \alpha = 0.3 \).

4 CONCLUSION

We have proposed a fast and accurate algorithm for global alignment of multiple networks. It overcomes several limitations of existing tools by aligning multiple networks without additional
training data, finding a global alignment of six species within several minutes and scaling to networks with tens of thousands of proteins and interactions. Further, it is the first alignment tool that can run with multiple cores in parallel.

We rigorously combine protein sequence similarity and network topology similarity into a suitable scoring scheme for multiple networks, adapting a successful technique from multiple sequence alignment. This allows us to model the problem as a combinatorial optimization problem, which we solve with SA. On PPI networks of five eukaryotic species, such as human, mouse, fruit fly, nematode and yeast, our implementation NetCoffee successfully finds a global alignment covering $\sim50\%$ of the proteins; and $\sim45\%$ of the match sets are qualified.

We compared NetCoffee with four existing tools, three of which fail to run on at least one of the three test datasets in our benchmark. The results indicate that NetCoffee outperforms the state-of-the-art algorithm IsoRank-N in terms of coverage and consistency, and at the same time is $\sim1-3$ orders of magnitude faster. Compared with NetworkBlast-M, Graemlin 2.0 and SMETANA, NetCoffee not only overcomes their limitations but also retains the quality of alignments in terms of both coverage and consistency.

This suggests that NetCoffee provides substantial improvements to global network alignment and that the research community working on function annotation and phylogenetic analysis can benefit from it. Further, its application is not restricted to PPI networks. It could also be extended to other types of complex networks, such as Scientific Collaboration Networks (SCN) and World Wide Web Networks (WWW).

ACKNOWLEDGEMENT

The authors are grateful to Gunnar Klau and Mohammed El-Kebir for helpful discussions. They thank Roded Sharan and Maxim Kalaye for their helpful support regarding NetworkBlast-M. They also wish to thank Jason Flannick and Tony Novak for their help with the dataset used in Graemlin 2.0.

Funding: China Scholarship Council (CSC).

Conflict of Interest: none declared.

REFERENCES


