SiPAN

simultaneous prediction and alignment of protein–protein interaction networks

Alkan, Ferhat; Erten, Cesim

Published in:
Bioinformatics

DOI:
10.1093/bioinformatics/btv160

Publication date:
2015

Document version
Publisher's PDF, also known as Version of record

Bioinformatics, 31(14), 2015, 2356–2363
doi: 10.1093/bioinformatics/btv160
Advance Access Publication Date: 18 March 2015
Original Paper

Systems biology

SiPAN: simultaneous prediction and alignment of protein–protein interaction networks

Ferhat Alkan[1,2,3] **and Cesim Erten**[3,*]

[1]Center for Non-Coding RNA in Technology and Health, [2]Department of Veterinary Clinical and Animal Sciences, University of Copenhagen, Grønnegardsvej 3, DK-1870 Frederiksberg, Denmark and [3]Department of Computer Engineering, Kadir Has University, Cibali, Istanbul 34083, Turkey

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on November 18, 2014; revised on February 5, 2015; accepted on March 14, 2015

Abstract

Motivation: Network prediction as applied to protein–protein interaction (PPI) networks has received considerable attention within the last decade. Because of the limitations of experimental techniques for interaction detection and network construction, several computational methods for PPI network reconstruction and growth have been suggested. Such methods usually limit the scope of study to a single network, employing data based on genomic context, structure, domain, sequence information or existing network topology. Incorporating multiple species network data for network reconstruction and growth entails the design of novel models encompassing both network reconstruction and network alignment, since the goal of network alignment is to provide functionally orthologous proteins from multiple networks and such orthology information can be used in guiding interolog transfers. However, such an approach raises the classical chicken or egg problem; alignment methods assume error-free networks, whereas network prediction via orthology works affectively if the functionally orthologous proteins are determined with high precision. Thus to resolve this intertwinement, we propose a framework to handle both problems simultaneously, that of *SImultaneous Prediction and Alignment of Networks (SiPAN)*.

Results: We present an algorithm that solves the SiPAN problem in accordance with its simultaneous nature. Bearing the same name as the defined problem itself, the SiPAN algorithm employs state-of-the-art alignment and topology-based interaction confidence construction algorithms, which are used as benchmark methods for comparison purposes as well. To demonstrate the effectiveness of the proposed network reconstruction via SiPAN, we consider two scenarios; one that preserves the network sizes and the other where the network sizes are increased. Through extensive tests on real-world biological data, we show that the network qualities of SiPAN reconstructions are as good as those of original networks and in some cases SiPAN networks are even better, especially for the former scenario. An alternative state-of-the-art network reconstruction algorithm random walk with resistance produces networks considerably worse than the original networks and those reproduced via SiPAN in both cases.

Availability and implementation: Freely available at http://webprs.khas.edu.tr/∼cesim/SiPAN.tar.gz.

Contact: cesim@khas.edu.tr

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

1 Introduction

In protein–protein interaction (PPI) networks, nodes represent the proteins and the edges correspond to interactions between pairs of proteins. The PPI networks are quite central in understanding cell regulatory mechanisms, extracting protein functions and in disease diagnosis. Several high-throughput techniques gave rise to extraction of large-scale PPI networks for many organisms (Finley and Brent, 1994). However, experimental prediction of PPI networks is usually time-consuming and expensive. Thus, many approaches based on a wide range of computational techniques have been suggested for the problem of PPI *network prediction* (Aebersold and Mann, 2003; Goh and Cohen, 2002; Marcotte *et al.*, 1999). Existing computational methods vary depending on the type of information they employ for predictions. These include methods based on genomic context, structure, domain or sequence information; see Skrabanek *et al.* (2008); Xia *et al.* (2010) for useful reviews. A crucial problem with the predicted networks is that the extracted set of interactions may provide erroneous results in terms of false positives and false negatives. Therefore, several methods have been developed for network reconstructions that are based solely on network topology (Cannistraci *et al.*, 2013; Lei and Ruan, 2013). Such methods usually rely on making new interaction predictions or identifying spurious interactions by examining node neighborhoods of a given network.

Parallel to the growth in produced experimental data several problem formulations and computational methods have been developed for the comparative analysis of genome-wide PPI networks as well. In particular, biological network alignment problem has been of particular interest. In general terms, given two or more PPI networks from different species, the *network alignment* problem is to align the nodes of the networks or subnetworks within them. Functional orthology is an important application that serves as the main motivation to study the alignment problems as part of a comparative analysis of PPI networks; a successful alignment could provide a basis for deciding the proteins that have similar functions across species. Such information may further be used in predicting functions of proteins with unknown functions or in verifying those with known functions (Singh *et al.*, 2008), in detecting common orthologous pathways between species (Kelley *et al.*, 2003) or in reconstructing the evolutionary dynamics of various species (Kuchaiev and Pržulj, 2011). Network alignment algorithms incorporate the interaction data and the evolutionary relationships represented possibly in the form of sequence data. Based on the assumption that the interactions among functionally orthologous proteins should be conserved across species, such an incorporation is usually achieved by aligning proteins so that both the sequence similarities of aligned proteins and the number of conserved interactions are large (Chindelevitch *et al.*, 2010; Kuchaiev and Pržulj, 2011; Memišević and Pržulj, 2012; Singh *et al.*, 2008).

The problems of network prediction and network alignment are intertwined. The alignment methods make use of the predicted interaction networks to produce output alignments with large interaction conservation and sequence similarities. On the other hand, orthology detection maybe used in network prediction/reconstruction (Izarzugaza *et al.*, 2008; Lee *et al.*, 2008; Pache and Aloy, 2014). However, such an intertwinement creates a so-called chicken or egg problem; the alignment methods produce the orthologies based on the assumption that the interaction networks are correct and the interaction prediction detected from network alignments assume the proteins or the functional complexes are aligned correctly. This intertwinement is especially evident with the topology-based network prediction/reconstruction methods, which usually rely on a definition of 'similarity' between node pairs. The similarity might be a measure of direct neighborhood similarity, such as sharing many common neighbors, or some indirect neighborhood similarity, usually described in some form of a graph-theoretical distance between nodes. Likewise, one of the objectives of many alignment methods is to provide alignments where the neighborhoods of aligned nodes are also similar in the sense that nodes in the neighborhoods are also aligned, so as to provide large interaction conservation.

Thus, rather than considering the two problems independently, we propose a new problem formulation encompassing both components simultaneously, that of *simultaneous prediction and alignment of networks (SiPAN)*. To the best of our knowledge, this is the first study providing a model for capturing the simultaneous nature of the problems of network reconstruction and alignment. Employing data from relevant databases, we design two experimental settings; one where the network densities are preserved and one where network densities are increased. We show that the network qualities of SiPAN reconstructions are as good as those of original networks, and in some cases, SiPAN networks are even better, especially for the former scenario. An alternative state-of-the-art topology-based network reconstruction algorithm random walk with resistance (RWS) produces networks considerably worse than the original networks and those reproduced via SiPAN in both cases under most of the quality metrics.

2 Methods

Let G_1 $G_1(\mathrm{V}_1, E_1)$ and G_2 $G_2(\mathrm{V}_2, E_2)$ be two undirected graphs corresponding to the pair of input PPI networks belonging to two species. Here V_1, V_2 denote the node sets, whereas E_1, E_2 denote the edge sets of the graphs. Below we describe the general framework for the SiPANs and the SiPAN algorithm employed within this framework.

2.1 General framework

The general framework is described in pseudo-code in Algorithm 1. We start with initial input interaction networks G_1, G_2. The algorithm first constructs an alignment A of G_1, G_2. Then it iteratively computes the interaction confidence matrices T_1, T_2 and employing A, T_1, T_2 updates the networks G_1, G_2 and the alignment A via SiPAN. Here, the alignment A is a set of one-to-one mappings (u, u'), such that $u \in V_1, u' \in V_2$. For ease of description, we assume $A(u) = u'$ or $A(u') = u$ both denote the same mapping. The general framework is depicted on a toy example in Figure 1. After iteration $i = 1$, the networks G_1, G_2 are reconstructed according to the SiPAN algorithm. Changing the networks gives rise to a new alignment and new interaction confidence scores through which the next

Algorithm 1. ***Simultaneous Prediction and Alignment Frame work***

```
Input: $G_1(V_1, E_1), G_2(V_2, E_2), BL(V_1, V_2), k$
Output: Updated $G1$, $G2$, Alignment $A$
$A = Alignment(G_1, G_2, BL(V_1, V_2))$
for iteration $i = 1$ to k do
    $T_1 = Interaction\_Confidence\_Matrix(G_1)$
    $T_2 = Interaction\_Confidence\_Matrix(G_2)$
    $\prec G_1, G_2, A \succ = SiPAN(G_1, G_2, A, T_1, T_2, BL(V_1, V_2))$
end for
```

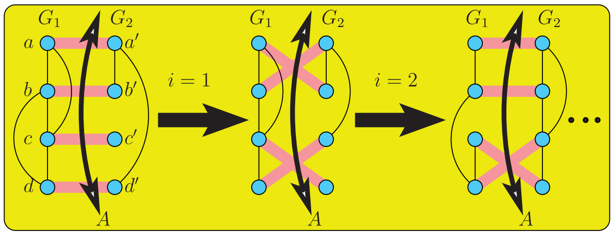

Fig. 1. A depiction of the general SiPAN framework on a toy example

iteration of the SiPAN network reconstruction is guided and so on and so forth.

We employ the SPINAL algorithm to produce the alignment A of G_1, G_2 (Aladağ and Erten, 2013). This choice for the alignment component is due to SPINAL's scalability and the quality of its output alignments. A recent survey of global network aligners indicates that SPINAL is among the top performers as far as biological significance of the output alignments are concerned (Clark and Kalita, 2014). The algorithm employs G_1, G_2 and the BLAST bit scores of every pair of nodes, one from V_1 the other from V_2, denoted with $BL(V_1, V_2)$, to produce an alignment. It consists of two phases; a coarse-grained alignment score estimations phase and a fine-grained conflict resolution and improvement phase. Both phases make use of the construction of neighborhood bipartite graphs and a set of contributors as a common primitive. Employing these concepts within iterative local improvement heuristics constitute the backbone of the algorithm. In terms of scalability, it runs much faster and provides more accurate results than most of the alternative state-of-the-art methods when tested on real-world biological data (Aladağ and Erten, 2013; Clark and Kalita, 2014). With respect to the network prediction component within the general SiPAN framework, we make use of the *RWS* algorithm of Lei and Ruan (2013). It is one of the several topology-based network prediction/correction algorithms (Fang *et al.*, 2013; Fouss *et al.*, 2007; Kuchaiev *et al.*, 2009; Tong *et al.*, 2006). It is reported that RWS provides network corrections with higher functional relevance than the rest of the heuristics as far as biological significance measures using multiple information sources, including gene ontology annotations, gene expression data, protein complexes, list of essential genes and conservation between species are concerned (Lei and Ruan, 2013). Given an input PPI network, all topology-based network correction algorithms produce an interaction confidence matrix or the so-called topological similarity matrix as output. T_1 in Algorithm 1 is a $|V_1| \times |V_1|$ matrix of real values as extracted from RWS, where entry (u, v) denotes the confidence assigned to the interaction between u, v. T_2 is defined similarly on V_2 of G_2.

2.2 Algorithm SiPAN

The main novelty of the proposed approach lies in how SiPAN updates the networks based on the current network topologies, the provided alignment of the networks and the interaction confidence score matrices. The core SiPAN algorithm is described in pseudocode in Algorithm 2 and 3. An overall depiction of the important steps of the algorithm can be found in Figure 2.

2.2.1 Significant non-conservations

Given a pair of mappings $(u, u'), (v, v') \in A$, we say that the pair induces a *non-conservation*, if $(u, v) \in E_1, (u', v') \notin E_2$ or $(u, v) \notin E_1, (u', v') \in E_2$. To *resolve* a non-conservation of the first type, we can either delete the edge from E_1 or insert the missing edge into E_2. The non-conservation of the second type can be resolved analogously. The general objective of the algorithm is to resolve all *significant* non-conservations arising from alignment A, taking into account the interaction confidence score matrices of the networks.

Algorithm 2. ***SiPAN***

```
Input: $G_1(V_1, E_1), G_2(V_2, E_2), A, T_1, T_2, BL(V_1, V_2)$
Output: Updated $\prec G_1, G_2, A \succ$

for $x = 1, 2$ do
    Construct *candidate* set $C_x$
    Sort $C_x$ with respect to scores in $T_x$
end for
Compute *breakpoints* $p_1, p_2$
*Resolve_Indels*$(C_1, C_2, p_1, p_2)$
// *Update networks and the alignment*
for $x = 1, 2$ do
    Commit the best $\beta_x \times |D_x^{p_x}|$ deletions in $D_x^{p_x}$
    Commit the best $\beta_x \times |I_x^{p_x}|$ insertions in $I_x^{p_x}$
end for
$A = Alignment(G_1, G_2, BL(V_1, V_2))$
```

Algorithm 3. ***Resolve_Indels***

```
Input: $C_1, C_2, p_1, p_2$
Output: Updated $\prec C_1, C_2, p_1, p_2 \succ$

Construct priority queue $Q$ of all *indels*
while $Q$ not empty do
    Remove $\prec (u, v), (u', v') \succ$ from $Q$
    if $\prec (u, v), (u', v') \succ$ not an *indel* then
        continue
    end if
    if $w(u, v) < w(u', v')$ then
        Remove $(u', v')$ from $C_2$
        Recompute $p_2, I_2^{p_2}, D_2^{p_2}$
    else
        Remove $(u, v)$ from $C_1$
        Recompute $p_1, I_1^{p_1}, D_1^{p_1}$
    end if
    Insert new *indels*, if any, to $Q$
end while
```

We start by constructing *candidate* sets, C_1, C_2. Every set includes pairs of nodes from the same network that lead to a non-conservation with respect to the current alignment A. More specifically, for $x, y = 1, 2$ and $x, y = 2, 1$ we define C_x as the union of $\{(u, v) | (u, v) \notin E_x \wedge (A(u), A(v)) \in E_y\}$ and $\{(u, v) | (u, v) \in E_x \wedge (A(u), A(v)) \notin E_y\}$. The former set in the union corresponds to the possible insertions of interactions into G_x, whereas the latter corresponds to the deletions of interactions from G_x. The candidate set of a network represents the set of operations that resolves all non-conservations of the alignment assuming no changes in the other network. However, since usually both networks may contain false positives/negatives, committing all updates on a single network may lead to erroneous network corrections. Thus, the main difficulty is to consider both networks simultaneously to extract an overall set of appropriate updates. Informally, the goal is

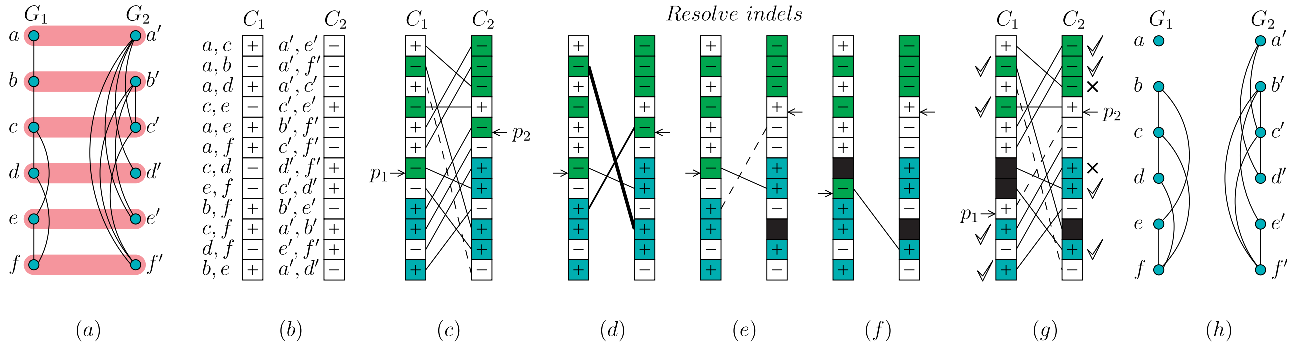

Fig. 2. A depiction of the important steps and concepts involved in SiPAN. **(a)** Input graphs G_1, G_2 and their current alignment. The aligned pairs of nodes are shown in ellipses. **(b)** Candidate lists C_1, C_2. The indices of the candidate lists are in increasing order from top to bottom. The pair of nodes corresponding to each operation is written next to each entry. The minus signs in the candidate lists indicate the deletion operations, whereas the plus signs indicate the insertions. The insertions/deletions in C_1, C_2 are sorted with respect to their interaction confidence scores. Edges $(b, c) \in E_1$ and $(b', c') \in E_2$ are conserved, thus they do not have a corresponding entry in the candidate lists. **(c)** The breakpoints p_1, p_2 when $\alpha_1 = \alpha_2 = 1$. In this setting $|D_1^{p_1}| = |I_1^{p_1}| = 3$ and $|D_2^{p_2}| = |I_2^{p_2}| = 4$. The shaded squares above p_1, p_2 indicate the operations in $D_1^{p_1}, D_2^{p_2}$, the shaded squares below p_1, p_2 indicate those in $I_1^{p_1}, I_2^{p_2}$, respectively. The non-conservations are shown with line segments between C_1, C_2. All non-conservations are significant, except the one depicted with the dashed line segment. **(d)** Indels among all the non-conservations are depicted with line segments. The thicknesses of the segments indicate the priorities of the indels. The indel shown with the thickest segment has a weight of $\frac{2}{12} \times \frac{3}{12} = \frac{1}{2}$, which provides the highest priority, and should be resolved first. **(e)** For the processed indel the weight of the deletion operation is smaller than that of the insertion; $\frac{1}{6}$ versus $\frac{1}{4}$. Thus, the insertion is removed from C_2 and p_2 is shifted up which further removes an existing indel, the one shown with the dashed line segment. **(f)** The only remaining indel of d is resolved by removing the deletion operation since the weight of the insertion is smaller. The breakpoint p_1 is shifted down. This creates a new indel shown with the line segment. **(g)** The new indel of e is resolved by removing the deletion operation since the weight of the insertion is smaller. The breakpoint p_1 is shifted down. No further indels are left. The non-conservations shown with the dashed line segments are not significant. With $\beta_1 = 1$, $\beta_2 = \frac{2}{3}$, we commit all the check-marked operations in C_1, C_2 attached with significant non-conservations depicted with solid segments; two deletions, two insertions in G_1 and two deletions and two insertions in G_2. The deletion and insertion operations in C_2 with cross marks next to them are not committed. **(h)** New G_1 and G_2

to select a subset of operations from each candidate set C_1, C_2, such that the selected insertions have higher interaction confidences measured by RWS than those of the selected deletions.

To this end, we first sort the candidate lists in ascending order with respect to the interaction confidence values; see Figure 2a. For each sorted candidate list, we find a *breakpoint* index that separates the set of possible insertions/deletions on each network such that all final deletions will have indices smaller than or equal to the breakpoint index, whereas all insertions will have indices larger than the breakpoint. For an index i, let D_x^i represent the list of deletions with indices smaller than or equal to i and I_x^i represent the list of insertions with indices larger than i. Since for a given network of interactions the ratio of false negatives to false positives is not necessarily one, we determine the breakpoints p_1, p_2 based on user defined parameters α_1, α_2. For $x = 1$, 2 let p_x be the index, such that $\alpha_x \times |D_x^{p_x}| = |I_x^{p_x}|$. A non-conservation $(u, v) \in E_1, (u', v') \notin E_2$ or $(u, v) \notin E_1, (u', v') \in E_2$ is *significant* if $(u, v) \in D_1^{p_1} \cup I_1^{p_1}$ or $(u', v') \in D_2^{p_2} \cup I_2^{p_2}$. In other words, the breakpoints determine the borders between undesired insertions/deletions in the candidate sets; the insertions below the index (those with interaction confidence values less than that recorded at breakpoint) and the deletions above the index (those with interaction confidence values larger than that recorded at breakpoint) are not significant and shall never be committed; see Figure 2b for a depiction of breakpoint selection and the construction of the sets $D_x^{p_x}, I_x^{p_x}$.

2.2.2 Resolving indels

We note that simply committing all the insertions or deletions defined by the breakpoints may not resolve all significant non-conservations. For the mappings $(u, u'), (v, v') \in A$, and the non-conserved edge $(u, v) \in E_1, (u', v') \notin E_2$, if $(u, v) \in D_1^{p_1}$ and $(u', v') \in I_2^{p_2}$ then deleting (u, v) from E_1 and inserting (u', v') into E_2 simply reverses the direction of non-conservation which still exists. Similarly, if $(u, v) \in I_1^{p_1}$ and $(u', v') \in D_2^{p_2}$, then committing both operations still retains the non-conservation. We call all such tuples of node pairs $\prec (u, v), (u', v') \succ$ *indels*. The goal is to *resolve* each indel by selecting the insertion or the deletion operation defined by it.

While resolving the indels, a factor that affects the overall quality of all the resolutions is the order with which the indels are processed. This is due to the fact that as the indels are resolved, the breakpoints may change which further may eliminate some of the existing indels or introduce new indels. The indels with higher *priority* should be resolved so that they are not eliminated by future indel resolutions. To this aim, we construct a priority queue Q that includes all current indels. Let *in* denote the index of (u, v) in the sorted candidate list C_1 and let *in′* denote the index of (u', v') in the sorted candidate list C_2. An indel $\prec (u, v), (u', v') \succ$ has a weight of $w(u, v) \times w(u', v')$, where $w(u, v) = \frac{in+1}{|C_1|}$ and $w(u', v') = \frac{|C_2|-in'}{|C_2|}$ if $(u, v) \in D_1^{p_1}$ and $(u', v') \in I_2^{p_2}$, whereas $w(u, v) = \frac{|C_1|-in}{|C_1|}$ and $w(u', v') = \frac{in'+1}{|C_2|}$ if $(u, v) \in I_1^{p_1}$ and $(u', v') \in D_2^{p_2}$; see Figure 2c. With this definition, the weight of a pair representing a deletion is proportional to its interaction confidence score, whereas that of a pair representing an insertion is inverse proportional. Intuitively, a small indel weight corresponds to the deletion of an interaction with small confidence score and the insertion of a missing interaction with large confidence score and thus corresponds to an indel with higher priority.

We iteratively remove the indel with the smallest weight from Q and process it by updating necessary data structures until Q is empty. With the provided weight definitions, no matter what the operation is, be it insertion or deletion, the smaller the weight, the higher the confidence we have in committing it. Thus, processing an indel first involves selecting the operation with larger weight and removing it from its candidate list, C_x; the selected operation shall not be committed. This will change the breakpoint p_x. We recompute p_x and $D_x^{p_x}$, $I_x^{p_x}$ using the formula $\alpha_x \times |D_x^{p_x}| = |I_x^{p_x}|$. Changing $D_x^{p_x}, I_x^{p_x}$ may cause the removal of an indel; see Figure 2d. Therefore, at the beginning of each iteration, we actually first make sure that the indel under consideration still satisfies the properties of an indel at line 6 of Algorithm 3. This is achieved by checking whether $(u, v) \in D_1^{p_1} \cup I_1^{p_1}$ and $(u', v') \in D_2^{p_2} \cup I_2^{p_2}$. In addition to possible indel removals, the changes in $D_x^{p_x}, I_x^{p_x}$ may also create new indels not already in Q which are simply inserted into Q; see Figure 2e. It should be noted that throughout the whole process of indel resolution, indices of all the operations remain unchanged.

2.2.3 Updating the networks and the alignment

Once all the indels are resolved, all the remaining operations in $D_x^{p_x}, I_x^{p_x}$, for $x = 1, 2$, can be committed simultaneously to resolve all significant non-conservations. Note that resulting insertions and deletions with indices close to the breakpoints p_x may have very close interaction confidence scores, which may be considered high as far as deletions are concerned and may be considered low with respect to insertions. Thus, rather than blindly committing all operations, we introduce user-defined parameters β_x, for $x = 1, 2$. The committed deletions are the best $\beta_x \times |D_x^{p_x}|$ of all the deletions in $D_x^{p_x}$, that is those with the smallest $\beta_x \times |D_x^{p_x}|$ indices, and the committed insertions are those with the largest $\beta_x \times |I_x^{p_x}|$ indices among all insertions in $I_x^{p_x}$; see Figure 2f. This finalizes the network prediction component of SiPAN. We then align the updated networks and the new alignment becomes input to the next iteration of SiPAN.

3 Discussion of results

The SiPAN algorithm has been implemented in C++ using the LEDA library (Mehlhorn and Naher, 1999). The source code, executables, useful scripts for evaluations and all the input data are freely available as part of the Supplementary Material.

We present an evaluation of the SiPAN framework applied on the PPI networks of four extensively studied species: *Caenorhabditis elegans, Drosophila melanogaster, Homo sapiens* and *Saccharomyces cerevisiae*. As input data, SiPAN requires a pair of these PPI networks and the pairwise sequence similarity scores of involved proteins. All these data are retrieved from the IsoBase database (Park *et al.*, 2011), which is employed in the performance evaluations of many of the PPI network alignment algorithms including IsoRank (Singh *et al.*, 2008), IsoRankN (Liao *et al.*, 2009), SMETANA (Sahraeian and Yoon, 2013), and BEAMS (Alkan and Erten, 2014). The *C.elegans* network has 19 756 proteins and 4884 interactions, the *D.melanogaster* network has 14 098 proteins and 25 054 interactions, the *H.sapiens* network has 22 369 proteins and 55 168 interactions, the *S.cerevisiae* network has 6659 proteins and 82 932 interactions and in total, there are 62 882 proteins and 168 038 interactions. Pairwise sequence similarity scores correspond to the BLAST Bit values of the protein sequences retrieved from Ensembl (Hubbard *et al.*, 2009).

Since the general SiPAN framework aims at simultaneously reconstructing the given pair of PPI networks and at the same time aligning them, we provide performance evaluation results with respect to both problems. The discussion of results regarding the latter are presented in the Supplementary Document due to space restrictions. The benchmark method for network reconstructions is RWS (Lei and Ruan, 2013), whereas for network alignments, we employ SPINAL (Aladağ and Erten, 2013). In addition to being state-of-the-art methods for the corresponding problems of network reconstruction and alignment, this choice of the benchmark methods also allows us to examine the overall improvement brought forth by the general SiPAN framework as the proposed algorithm makes use of both methods. For all the computational experiments, the parameter k of Algorithm 1 is set to 5, that is the general SiPAN framework is iterated five times. The parameters β_1, β_2 are both set to 0.5, i.e. 50% of the proposed insertion/deletion operations are committed at each iteration of SiPAN.

3.1 Evaluation metrics

The criteria for performance evaluations are based on two databases: the Gene Ontology (GO) database (Ashburner *et al.*, 2000) and the STRING database (Franceschini *et al.*, 2013). The GO database annotates proteins from several species with appropriate GO categories organized as a directed acyclic graph (DAG) (Ashburner *et al.*, 2000). To standardize the GO annotations of proteins, similar to the evaluation methods of (Aladağ and Erten, 2013; Liao *et al.*, 2009; Singh *et al.*, 2008), we restrict the protein annotations to level 5 of the GO DAG by ignoring the higher-level annotations and replacing the deeper-level category annotations with their ancestors at the restricted level. Note that considering other levels of the GO DAG, we obtain almost the same results as those of level 5. All evaluations restricted to levels 4 and 6 of the GO DAG are provided in the Supplementary Document. A 'pair' is *annotated* if both of its proteins are annotated by some GO categories. Note that a pair represents a pair of interacting proteins, in the case of network reconstruction evaluations and a pair of aligned proteins from the aligned PPI networks, in the case of network alignment evaluations. Only the annotated pairs are considered for the GO-based evaluations. An annotated pair is considered *consistent* if both of its proteins share at least one common standard GO annotation.

As part of the GO-based evaluations, we employ five metrics. *Sensitivity* represents the number of consistent pairs, whereas *specificity* denotes the ratio of number of consistent pairs to the number of annotated pairs. We employ three additional metrics to measure how the GO terms are distributed over the defined pairs. We define the *GO distribution sensitivity* as the number of proteins deemed consistent due to an average standard GO term and *GO distribution specificity* normalizes it with the number of annotations a GO term provides. More specifically, for a given GO category i, let GO_i denote the set of pairs containing proteins both of which are annotated with i and let P_i denote the set of all proteins annotated with i. The GO distribution sensitivity is defined as $\frac{\sum_{\forall i} 2 \times |\mathrm{GO}_i|}{n}$, where n denotes the number of standard GO terms. The GO distribution specificity is defined as $\frac{\sum_{\forall i} 2 \times |\mathrm{GO}_i| / |P_i|}{n}$. Finally, for an annotated pair x, let $\mathrm{GO}_{\mathrm{int}}(x)$ and $\mathrm{GO}_{\mathrm{uni}}(x)$ indicate, respectively, the intersection set of GO annotations of proteins in x and the union set of GO annotations of all the proteins in x. The *GO consistency* score, *GOC*, is defined as the sum of $|\mathrm{GO}_{\mathrm{int}}| / |\mathrm{GO}_{\mathrm{uni}}|$ over all annotated pairs.

Regarding the evaluations based on the STRING database, we make use of the metrics *neighborhood, fusion, coexpression, experimental, database, textmining* as defined in von Mering *et al.* (2005). Given a PPI, the STRING database provides for each metric a confidence score for the interaction. For a PPI network, for each metric, we compute an average of the scores over all interactions. Note that the STRING-based assessments apply only to the network reconstruction evaluations and not to the evaluations of network alignment.

Apart from the biological performance evaluations employing GO and STRING databases, one final evaluation metric we employ is the *edge coverage*, a performance metric especially employed in network alignment studies. Given a specific alignment of G_1, G_2, the edge coverage on G_1 measures the ratio of conserved edges in the alignment to the number of edges in G_1. Employing a fixed alignment algorithm, comparing the edge coverage values of the

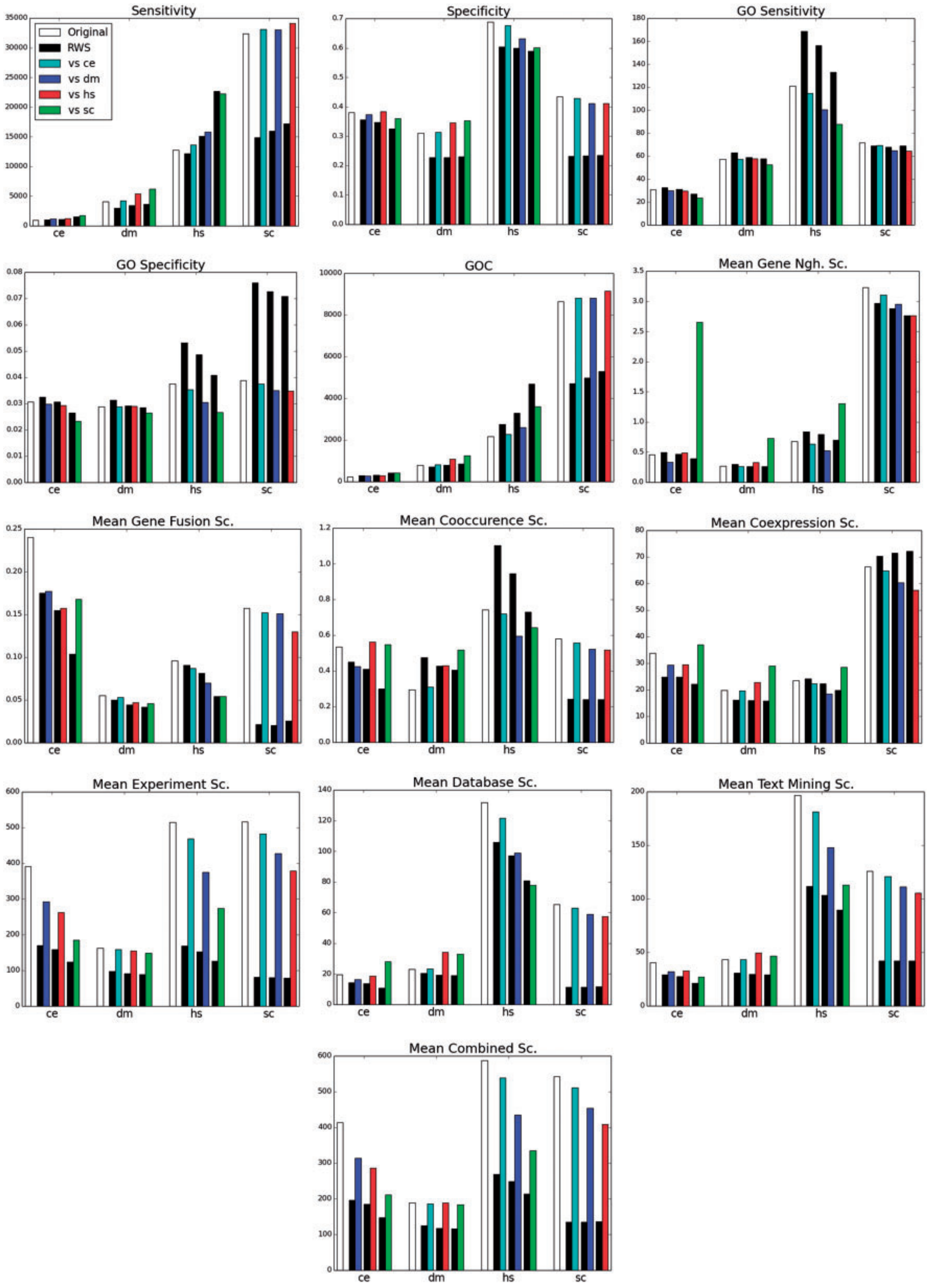

Fig. 3. Bar charts of reconstructed network qualities assuming increased network sizes

reconstructed networks provides insight regarding the degrees of evolutionary conservation of the reconstructions. The SiPAN reconstructed networks provide quite large edge coverage values when compared with both original and the RWS-reconstructed networks; see Supplementary Document for further details.

3.2 Network reconstruction quality

We provide two types of evaluations, one where the network sizes are preserved and the other where the network sizes are increased. The latter is especially important as the PPI networks have large false-negative rates, usually larger than the corresponding false-positive rates (Huang *et al.*, 2007). The evaluations presented here are restricted to comparing the networks with respect to defined metrics. The networks are comprised of the original networks and those produced by RWS and SiPAN. However, when considering the reconstruction algorithms yet another useful evaluation consists of comparing the set of insertions to the set of deletions of a specific reconstruction with respect to the performance metrics. We implemented such evaluations on RWS and SiPAN reconstructions. All such evaluations are presented in the Supplementary Document. Additionally, as SiPAN framework depends on iterative reconstructions, analyzing the change in the number of modifications committed iteration after iteration provides an intuition regarding the convergence characteristics of the algorithm. Details regarding these evaluations can also be found in the Supplementary Document.

3.2.1 Preserving network sizes

Because of to space restrictions, we only present a brief summary of results regarding this scenario here. Evaluation plots and a detailed discussion of further results can be found in the Supplementary Document. To preserve the network sizes, the parameters α_1, α_2 are set to 1 for the SiPAN algorithm. Out of 52 evaluation instances in total, in 10 of them, the scores of RWS are the best, in 13 of them the original network scores are the best and in 28 of them SiPAN results are the best. The scores of SiPAN and those of the original are equal in 1 instance. We note that for the case of preserving network sizes, a proposed reconstructed network can be considered successful if its evaluation results are somewhat similar to those of the original network. When comparing the results of RWS networks against those of the original networks, the scores of RWS are higher in 17 instances, whereas the original network scores are better in 35 instances. On the other hand, SiPAN networks provide better scores than the originals for most of the instances; for 15 instances, the original scores are better than those of SiPAN's, for 34 instances, those of SiPAN networks are better than those of the originals and for 4 instances, the scores are tied. Notably, the sensitivity and the specificity results provided by the RWS networks are worse than both those of the original and of the SiPAN networks for all the species.

3.2.2 Increasing network sizes

Because of the large false-negative rates of PPI networks, the main objective in network reconstruction is to increase network densities while preserving network qualities. Figure 3 provides evaluation results when the number of interactions is increased. For SiPAN, such an increase is achieved through the α parameter. Note that, intuitively, the α parameter for a network represents the ratio of false negatives to the false positives and should thus be set accordingly by the user. We employed α values inversely proportional to network densities; for a network $G_x(V_x, E_x)$ under evaluation, the corresponding α parameter is set to $\frac{|V_x|^2}{|E_x|\times 250}$. On the other hand, RWS, rather than producing a reconstructed network explicitly, provides an interaction confidence matrix where entry (u, v) denotes the confidence of the interaction between u, v as computed by the algorithm. A new network of size d is then reconstructed by introducing an edge for each of the top scoring d pairs in the matrix. In the bar chart of Figure 3, for a subset of bars marked with X on the x-coordinate, the leftmost bar (white) indicates the score of the original network X. The rest consist of three pairs of bars. For each pair, the right bar indicates the score of X after applying SiPAN on X paired with Y, where Y is *C.elegans, D.melanogaster, H.sapiens and S.cerevisiae* going from left to right over the pairs, assuming $X \neq Y$. The left bar of the pair represents the score of applying RWS on X, with the constraint that the reconstructed network contains as many edges as the output of SiPAN reconstruction of X. Going from left to right, the sizes of the reconstructed *C.elegans* networks are 6530, 7405 and 11 247, those of the *D.melanogaster* networks are 25 644, 29 155 and 30 866, those of the *H.sapiens* networks are 60 151, 75 361 and 114 900 and finally, those of the *S.cerevisiae* networks are 85 652, 91 277 and 97 520. For a given network X, although the same α parameter is employed, the size of the reconstructed network X grows more as the paired network Y gets denser. This is a feature of SiPAN; as Y becomes denser more edges in X are non-conserved, which further increases the number of edges eligible for non-conservation resolution. Intuitively, this corresponds to transferring more information from Y to X, as the information content of Y increases.

For the *C.elegans* network, the reconstructed network qualities of SiPAN are better than those of RWS for almost all measures. The measures *GO distribution sensitivity* and *GO Distribution specificity* are the exceptions. In fact, for all the species instances, RWS achieves better scores than SiPAN for these two measures. However, it should be noted that, unlike the rest of the GO-based measures, these two measures do not directly provide a correctness measure for the reconstructed network itself. They rather provide evaluations from the opposite direction; assuming the reconstructed networks are correct, they can be considered as metrics for evaluating how well defined the employed set of GO categories are. Nevertheless, to be consistent with the previous network alignment studies, we provide the two scores as indirect measures of network reconstruction quality. In the *C.elegans* network, SiPAN provides better sensitivity and specificity than RWS and the GO consistency scores of both algorithms are close, although in two instances those of RWS are slightly better. Regarding the STRING-based evaluations SiPAN provides better scores than RWS for almost all metrics. This is especially evident for the *combined score* as computed by STRING. Similar arguments hold for the *D.melanogaster* as well; SiPAN networks have better quality in terms of the GO-based evaluation scores of sensitivity, specificity and GO consistency. The STRING-based evaluation results are again better for the SiPAN networks in almost all cases. For the *H.sapiens* network, the superiority of SiPAN over RWS is not as evident as the other species. The specificity results of SiPAN are better than those of RWS in all three instances, whereas for sensitivity, RWS is better than SiPAN in two out of three. The GO consistency results of RWS are also better in each case. With respect to the STRING-based evaluations, the scores of SiPAN are better in almost half of the metrics, whereas those of RWS are better in the other half. Nevertheless, the scores of SiPAN reconstructions provide higher scores than those of RWS for the *combined score* metric for all three instances. The superiority of SiPAN over RWS is at its best when the reconstruction is applied on the *S.cerevisiae* network. SiPAN networks provide considerably larger scores than those of RWS for almost all the metrics under all three instances. It is important to note that SiPAN networks contain more interactions than the corresponding original networks, in some cases almost three times denser networks are reconstructed. Nevertheless, even if we only consider the metrics employing normalizations with respect to the network sizes, such as specificity and all the STRING-based metrics, the scores of SiPAN networks are close to those of the original networks. This further emphasizes the success of SiPAN in achieving the main reconstruction objective, that is that of growing the network with as little loss in network quality as possible.

4 Conclusion

We proposed a problem formulation that considers the network reconstruction and the network alignment problems simultaneously. We then provided an algorithm, SiPAN, that solves the problems in accordance with their simultaneous nature. Through extensive evaluations, we showed that the SiPAN algorithm outperforms the state-of-the-art algorithms, RWS and SPINAL, each proposed respectively to solve the problems of network reconstruction and network alignment independently. It should be noted that SiPAN algorithm employs two separate algorithms: a topology-based network reconstruction algorithm, RWS, and a network alignment algorithm, SPINAL. However, as SiPAN can be considered as a general framework to solve these two problems simultaneously, it is possible to replace RWS with an alternative topology-based network reconstruction algorithm (as long as it provides a scoring matrix representing confidences assigned to interactions as output) and SPINAL with an alternative global one-to-one network alignment algorithm. Employing pairs of alternative algorithms for each component within the general SiPAN framework and determining the best performing pair is part of future work. Employment of alternative reconstruction and alignment methods might indirectly affect yet another important issue, that of desirable parameter selection. Our experiments indicate that the SiPAN framework performs best when the algorithm is iterated many times and very small changes on the networks are committed at each iteration. This corresponds to choosing a large value for k and small values for β_1, β_2. Currently, the number of iterations is set to 5, β_1, β_2 to 0.5. Even with these settings, each execution requires a considerable amount of time, up to several days. The major bottleneck in the algorithm is the reconstruction component, that is the RWS algorithm. Thus, employing an alternative reconstruction algorithm or reimplementing the RWS algorithm with the running time performance issue in mind might allow the SiPAN algorithm provide even higher quality network reconstructions.

Funding

This work was, in part, supported by The Scientific and Technological Research Council of Turkey (TUBITAK) [112E137].

Conflict of Interest: none declared.

References

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

Aladağ,A.E. and Erten,C. (2013) Spinal: scalable protein interaction network alignment. *Bioinformatics*, **29**, 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**, 531–539.

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

Cannistraci,C.V. *et al.* (2013) Minimum curvilinearity to enhance topological prediction of protein interactions by network embedding. *Bioinformatics*, **29**, i199–i209.

Chindelevitch,L. *et al.* (2010) Local optimization for global alignment of protein interaction networks. In: *Pacific Symposium on Biocomputing*, Hawaii, USA, pp. 123–132.

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

Fang,Y. *et al.* (2013) Global geometric affinity for revealing high fidelity protein interaction network. *PLoS One*, **6**, e19349.

Finley,R.L. and Brent,R. (1994) Interaction mating reveals binary and ternary connections between drosophila cell cycle regulators. *Proc. Natl Acad. Sci. USA*, **91**, 12980–12984.

Fouss,F. *et al.* (2007) Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Trans. Knowl. Data Eng.*, **19**, 355–369.

Franceschini,A. *et al.* (2013) STRING v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Res.*, **41**(Database Issue), 808–815.

Goh,C.S. and Cohen,F.E. (2002) Co-evolutionary analysis reveals insights into protein-protein interactions. *J. Mol. Biol.*, **324**, 177–192.

Huang,H. *et al.* (2007) Where have all the interactions gone? Estimating the coverage of two-hybrid protein interaction maps. *PLoS Comput. Biol.*, **3**, e214.

Hubbard,T. *et al.* (2009) Ensembl 2009. *Nucleic Acids Res.*, **37**(Database Issue), 690–697.

Izarzugaza,J.M. *et al.* (2008) Enhancing the prediction of protein pairings between interacting families using orthology information. *BMC Bioinformatics*, **9**, 35.

Kelley,B.P. *et al.* (2003) Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc. Natl Acad. Sci. USA*, **100**, 11394–11399.

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

Kuchaiev,O. *et al.* (2009) Geometric de-noising of protein-protein interaction networks. *PLoS Comput. Biol.*, **5**, e1000454.

Lee,S.-A. *et al.* (2008) Ortholog-based protein-protein interaction prediction and its application to inter-species interactions. *BMC Bioinformatics*, **9**, S11.

Lei,C. and Ruan,J. (2013) A novel link prediction algorithm for reconstructing protein-protein interaction networks by topological similarity. *Bioinformatics*, **29**, 355–364.

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

Marcotte,E.M. *et al.* (1999) Detecting protein function and protein-protein interactions from genome sequences. *Science*, **285**, 751–753.

Mehlhorn,K. and Naher,S. (1999) *Leda: A Platform for Combinatorial and Geometric Computing*. Cambridge University Press, Cambridge.

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

Pache,R. and Aloy,P. (2014) Increasing the precision of orthology-based complex prediction through network alignment. *Peer J.*, **2**, e413.

Park,D. *et al.* (2011) Isobase: a database of functionally related proteins across PPI networks. *Nucleic Acids Res.*, **39**(Database Issue), 295–300.

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

Singh,R. *et al.* (2008) Global alignment of multiple protein interaction networks. In: *Proceedings of Pacific Symposium on Biocomputing*, pp. 303–314.

Skrabanek,L. *et al.* (2008) Computational prediction of protein-protein interactions. *Mol. Biotechnol.*, **38**, 1–17.

Tong,H. *et al.* (2006) Fast random walk with restart and its applications. In: *Proceedings of the Sixth International Conference on Data Mining,* ICDM '06, IEEE Computer Society, Washington, DC, USA, pp. 613–622.

von Mering,C. *et al.* (2005) String: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Res.*, **33**(Database Issue), 433–437.

Xia,J. *et al.* (2010) Computational methods for the prediction of protein-protein interactions. *Protein Pept. Lett.*, **9**, 1069–1078.