**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

*Citation for published version (APA):*
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 intertwinedment, 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 orthogonal pathways between species (Kelley et al., 2003) or in reconstituting 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 orthogonal 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 may be used in network prediction/reconstruction (Izarzugaza et al., 2008; Lee et al., 2008; Pache and Aloy, 2014). However, such an intertwined 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 $G_1, G_2(V_1, E_1)$ and $G_2, G_2(V_2, E_2)$ 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 Framework

1: Input: $G_1(V_1, E_1), G_2(V_2, E_2), BL(V_1, V_2), k$
2: Output: Updated $G_1, G_2$, Alignment $A$
3: $A = \text{Alignment}(G_1, G_2, BL(V_1, V_2))$
4: for iteration $i = 1$ to $k$ do
5: $T_1 = \text{Interaction Confidence Matrix}(G_1)$
6: $T_2 = \text{Interaction Confidence Matrix}(G_2)$
7: $< G_1, G_2, A > = \text{SiPAN}(G_1, G_2, A, T_1, T_2, BL(V_1, V_2))$
8: end for
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 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 \((u, v) \in E_1\) and \((u', v') \notin E_2\) or \((u, v) \notin E_1\) and \((u', v') \in E_2\). 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.

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 \(x \neq y = 2, 1\) we define \(C_x\) as the union of \((u, v) | (u, v) \notin E_x \land (A(u), A(v)) \in E_y\) and \((u, v) | (u, v) \in E_x \land (A(u), A(v)) \notin E_y\). 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

**Algorithm 2. SiPAN**

1: Input: \(G_1(V_1, E_1), G_2(V_2, E_2), A, T_1, T_2, BL(V_1, V_2)\)  
2: Output: Updated \(\langle G_1, G_2, A \rangle\)  
3: for \(x = 1, 2\) do  
4: Construct candidate set \(C_x\)  
5: Sort \(C_x\) with respect to scores in \(T_x\)  
6: end for  
7: Compute breakpoints \(p_1, p_2\)  
8: Resolve Indels\((C_1, G_2, p_1, p_2)\)  
9: // Update networks and the alignment  
10: for \(x = 1, 2\) do  
11: Commit the best \(B_x \times |D|_x\) deletions in \(D_0\)  
12: Commit the best \(B_x \times |I|_x\) insertions in \(I_0\)  
13: end for  
14: \(A = \text{Alignment}(G_1, G_2, BL(V_1, V_2))\)

**Algorithm 3. Resolve Indels**

1: Input: \(C_1, C_2, p_1, p_2\)  
2: Output: Updated \(\langle C_1, C_2, p_1, p_2 \rangle\)  
3: Construct priority queue \(Q\) of all indels  
4: while \(Q\) not empty do  
5: Remove \(<u, v, (u', v') >\) from \(Q\)  
6: if \(<u, v, (u', v') >\) not an indel then  
7: continue  
8: end if  
9: if \(w(u, v) < w(u', v')\) then  
10: Remove \((u', v')\) from \(C_2\)  
11: Recompute \(p_2, I_0^2, D_0^2\)  
12: else  
13: Remove \((u, v)\) from \(C_1\)  
14: Recompute \(p_1, I_0^1, D_0^1\)  
15: end if  
16: Insert new indels, if any, to \(Q\)  
17: end while

---

Fig. 1. A depiction of the general SiPAN framework on a toy example
We note that simply committing all the insertions or deletions in the candidate sets will not lead to a sequence of final deletions with indices smaller than or equal to the breakpoint.

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 \( D^i_v \) represent the list of deletions with indices smaller than or equal to i and \( I^i_v \) 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 \( s_1, s_2 \). For \( s = 1, 2 \) let \( p_s \) be the index, such that \( s_s \times |D^2_p| = |I^1_v| \). 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 \( (u, v) \in D^1_v \cup I^1_v \) or \( (u', v') \in D^2_v \cup I^2_v \). 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 larger 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 \( D^1_v, I^1_v \).

### 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 \( (u, v) \in D^1_v \) and \( (u', v') \in I^1_v \) 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 \( (u, v) \in I^1_v \) and \( (u', v') \in D^2_v \), then committing both operations still retains the non-conservation. We call all such tuples of node pairs \( (u, v), (u', v') \rightarrow \text{indel} \). 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 \( m \) denote the index of \( (u, v) \) in the sorted candidate list \( C_1 \) and let \( m' \) denote the index of \( (u', v') \) in the sorted candidate list \( C_2 \). An indel \( (u, v), (u', v') \rightarrow \text{indel} \) has a weight of \( w(u, v) \times w(u', v') \), where \( w(u, v) = \frac{s_1}{c_{u,v}} \) and \( w(u', v') = \frac{s_2}{c_{u',v'}} \) if \( (u, v) \in D^1_v \) and \( (u', v') \in I^1_v \), whereas \( w(u, v) = \frac{s_1}{c_{u,v}} \) and \( w(u', v') = \frac{s_2}{c_{u',v'}} \) if \( (u, v) \in I^1_v \) and \( (u', v') \in D^1_v \); 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, \( C_u \); the selected operation shall not be committed. This will change the breakpoint \( p_s \). We recompute \( p_s \) and \( D^1_v, I^1_v \) using the formula \( s_s \times |D^s_v| = |I^s_v| \). Changing \( D^1_v, I^1_v \) 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 \((u, v) \in D_p^x \cup I_p^x\) and \((u', v') \in D_p^y \cup I_p^y\). In addition to possible indel removals, the changes in \(D_p^y \cup I_p^y\) 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 \(D_p^x \cup I_p^x\), for \(x = 1, 2, \ldots\), 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 \(\beta_x\) for \(x = 1, 2\). The committed deletions are the best \(\beta_x \times |D_p^x|\) of all the deletions in \(D_p^x\), that is those with the smallest \(\beta_x \times |D_p^x|\) indices, and the committed insertions are those with the largest \(\beta_x \times |I_p^x|\) indices among all insertions in \(I_p^x\); 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 used in the performance evaluations of many of the PPI network alignment algorithms including IsoRank (Singh et al., 2008), IsoRankN (Liao et al., 2009), SMETANA (Sahraei and Yoon, 2013), and BEAMS (Alakan and Erten, 2014). The C. elegans network has 19 756 proteins and 4884 interactions, the D. melanogaster network has 14 098 proteins and 23 054 interactions, the H. sapiens network has 22 369 proteins and 55 168 interactions, the S. cerevisiae network has 66 59 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 Ensemble (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 (Aladan 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 \(\beta_1, \beta_2\) 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 (Aladan 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 \(G_O\) denote the set of pairs containing proteins both of which are annotated with \(i \) and let \(P\), denote the set of all proteins annotated with \(i\). The GO distribution sensitivity is defined as \(\frac{2 \cdot |G_O|}{n}\), where \(n\) denotes the number of standard GO terms. The GO distribution specificity is defined as \(\frac{\sum_{i=1}^{n} 2 \cdot |G_O(i)|}{|P|}\). Finally, for an annotated pair \(x\) let \(G_{ann}(x)\) and \(G_{ann}(x)\) 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 \(G_{ann}(x)/G_{ann}(x)\) 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
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

Fig. 3. Bar charts of reconstructed network qualities assuming increased network sizes
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 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 $a_i$, $a_2$ 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 $x$ parameter. Note that, intuitively, the $x$ 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 $x$ values inversely proportional to network densities; for a network $G_i(V_i, E_i)$ under evaluation, the corresponding $x$ parameter is set to $\frac{\sqrt{\text{density}}}{\text{number of interactions}}$. 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 $x$ 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 $\beta_1$, $\beta_2$. Currently, the number of iterations is set to 5, $\beta_1$, $\beta_2$ 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