BIOINFORMATICS ORIGINAL PAPER Vol. 30 no. 10 2014, pages 1449–1455
doi:10.1093/bioinformatics/btu043

Systems biology

Advance Access publication January 27, 2014

Network orientation via shortest paths

Dana Silverbush and Roded Sharan*

The Balavatnik School of Computer Science, Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel

Associate Editor: Igor Jurisica

ABSTRACT

The graph orientation problem calls for orienting the edges of a graph so as to maximize the number of pre-specified source–target vertex pairs that admit a directed path from the source to the target. Most algorithmic approaches to this problem share a common preprocessing step, in which the input graph is reduced to a tree by repeatedly contracting its cycles. Although this reduction is valid from an algorithmic perspective, the assignment of directions to the edges of the contracted cycles becomes arbitrary, and the connecting source–target paths may be arbitrarily long. In the context of biological networks, the connection of vertex pairs via shortest paths is highly motivated, leading to the following problem variant: given a graph and a collection of source–target vertex pairs, assign directions to the edges so as to maximize the number of pairs that are connected by a shortest (in the original graph) directed path. This problem is NP-complete and hard to approximate to within sub-polynomial factors. Here we provide a first polynomial-size integer linear program formulation for this problem, which allows its exact solution in seconds on current networks. We apply our algorithm to orient protein–protein interaction networks in yeast and compare it with two state-of-the-art algorithms. We find that our algorithm outperforms previous approaches and can orient considerable parts of the network, thus revealing its structure and function.

Availability and implementation: The source code is available at www.cs.tau.ac.il/~roded/shortest.zip.

Contact: roded@post.tau.ac.il

Received on June 4, 2013; revised on January 11, 2014; accepted on January 21, 2014

1 INTRODUCTION

Protein–protein interactions (PPIs) form the skeleton of signal transduction in the cell. Although many of these interactions carry directed signaling information, current PPI measurement technologies, such as yeast two hybrid (Fields, 2005) and co-immunoprecipitation (Gavin *et al.*, 2002), reveal the presence of a signal flow but not of its directionality. Identifying this directionality is fundamental to our understanding of how signaling networks function. Although some interactions are naturally directed, such as kinase–substrate and phosphatase–substrate interactions (KPIs), the directions of the vast majority of PPIs remain unknown.

From a theoretical point of view, graph orientation problems have been studied by several authors. In the following, we emphasize algorithmic results pertaining to mixed graphs (i.e. graphs with both directed and undirected edges), which are the focus of this work. Arkin and Hassin (2002) showed that the problem of orienting a mixed graph so that it admits directed paths for a given set of source–target vertex pairs is NP-complete. Elberfeld *et al.* (2013) showed that the corresponding maximization problem is NP-hard to approximate to within a factor of 7/8, and provided a sublinear approximation for it. Recently, Blokh *et al.* (2013) studied the problem of orienting the edges of an undirected graph so as to maximize the number of pre-specified source–target pairs that admit a directed shortest path between them (i.e. a directed path whose length is equal to the distance between them in the unoriented graph). They showed that for an undirected graph with a vertex set of size n, edge set of size m and p source-target pairs the problem is NP-hard to approximate within factors of $O(p^{1-\epsilon})$ and $O(m^{1/3-\epsilon})$ for any fixed $\epsilon>0$, and provided an approximation algorithm with a factor of $O(\max\{n,p\})^{1/\sqrt{2}}$. For the k-length-bounded orientations in a weighted undirected graph, which aims to maximize the weight of all satisfied paths between sources and targets with length at most k, Gitter *et al.* (2011) showed that while the resulting problem is NP-hard, it can still be approximated to within a factor $O(2^k/k)$.

In the biological domain, previous work to infer interaction directionality has mostly used unsupervised methods, with the exception of Vinayagam *et al.* (2011), which represents each directed interaction using topological features and trains a classifier to predict the directions of unseen interactions. Specifically, previous unsupervised methods relied on information from perturbation experiments, in which a gene is perturbed (*cause*) and as a result other genes change their expression levels (*effects*), to guide the inference (Yeang *et al.*, 2004). They assumed that for an effect to take place, there must be a directed path in the network from the causal gene to the affected gene. As there are many paths that can link two proteins in the interaction network, previous solution methods relied on either length-bounded paths (Gitter *et al.*, 2011; Ourfali *et al.*, 2007; Yeang *et al.*, 2004) or parameterized and integer programming techniques (Dorn *et al.*, 2011; Medvedovsky *et al.*, 2008; Silverbush *et al.*, 2011). The former are limited by the length of the considered paths (at most five). The latter, while producing highly precise predictions, are limited in their coverage, as they start by contracting cycles of the input network, eliminating the vast majority of the interactions (>92% on the network) from further consideration.

It is likely that biological responses are controlled by relatively short signaling cascades (Gitter *et al.*, 2011; Navlakha *et al.*, 2012; Vinayagam *et al.*, 2011); however, in a large-scale network, enumerating all possible paths between two vertices can still be computationally intractable, even when considering only paths of limited length. In this article, we propose an orientation method that overcomes the limitations of previous approaches by focusing attention on shortest paths, which allows the efficient

*To whom correspondence should be addressed.

representation of all such paths and avoids the contraction problem. The method aims to orient the edges of an input graph G so that a maximum number of input source–target pairs admit a directed source-to-target path whose length is the shortest possible (i.e. equal to distance between the two vertices in G). Our algorithm builds on a first efficient (for realistic instances) integer linear program (ILP) formulation of the problem, allowing the computation of optimal solutions in seconds on current networks. We applied our method to large-scale datasets of yeast physical interactions and evaluated it using cross-validation experiments. Our method outperformed two previous state-of-the-art orientation methods (Gitter *et al*., 2011; Silverbush *et al*., 2011) by a significant margin.

2 MATERIALS AND METHODS

2.1 Problem definition and the shortest-paths graph

We focus on simple graphs with no loops or parallel edges. A mixed graph is a triple $G = (V, E_U, E_D)$ that consists of a set of vertices V, a set of undirected edges $E_U \subseteq \{e \subseteq V : |e| = 2\}$ and a set of directed edges $E_D \subseteq V \times V$. A path of length ℓ in G is a sequence of distinct vertices $\langle v_1, \ldots, v_{\ell+1}\rangle$ such that $(v_i, v_{i+1}) \in E_U \cup E_D$ for every $1 \leq i \leq \ell$. We denote the distance between vertices s and t in G by $d_G(s,t)$. An *orientation* of G is a directed graph $\vec{G}$ on the same vertex set V whose edge set contains all the directed edges of G and a single directed instance of every undirected edge, but nothing more. We say that a pair of vertices (s,t) is *satisfied* by an orientation $\vec{G}$ when the latter graph contains at least one directed path from s to t whose length is $d_G(s,t)$, i.e. equal to the length of a shortest s to t path in G. The *Maximum Shortest-path Orientation (MSPO)* problem is defined as follows:

Input: A mixed graph $G = (V, E_U, E_D)$ with non-negative edge weights $w(e)$ for every $e \in E_U \cup E_D$, and a collection of source–target vertex pairs $P = \{(s_1, t_1), \ldots, (s_k, t_k)\}$.

Objective: An orientation of G that satisfies (via shortest paths) a maximum number of pairs from P.

MSPO was shown to be NP-hard in Blokh *et al*. (2013). Here we provide a polynomial size ILP for it that allows solving it to optimality in seconds on current networks. For ease of presentation, we focus on unweighted graphs but the algorithm can be easily generalized to the weighted case.

A key component of the algorithm is an efficient representation of shortest paths. For an ordered pair (s,t) of vertices, we define their *shortest paths graph* $G_{(s,t)} = (V, E_{(s,t)})$ to be a directed graph on V, which consists of all edges that reside on a shortest path from s to t in G. $G_{(s,t)}$ can be efficiently constructed by using breadth-first searches from s and t (the latter, after reversing all the directed edges). Now for every edge (u,v) such that u is reachable from s and such that t is reachable from v, we include it in $G_{(s,t)}$ if and only if $d_G(s,t) = d_G(s,u) + d_G(v,t) + 1$. It is easy to see that each shortest path in G is a shortest path in $G_{(s,t)}$. Furthermore, each path in $G_{(s,t)}$ is a shortest path in G by definition of its edges.

2.2 The integer program

The ILP formulation is based on checking pair connectivities via flow-based computations. The basic observation is that a pair $(s,t) \in P$ is satisfied by a given orientation if and only if t is reachable from s in $G_{(s,t)}$. The latter condition can be rephrased as allowing us to send one unit of flow from s to t. Our ILP consists of a set of binary orientation variables describing the edge orientations, flow variables describing the flow on edges of $G_{(s,t)}$ for every $(s,t) \in P$ and binary closure variables describing reachability of every pair in P.

Formally, the variables are as follows:

$$\left\{o_{(u,v)}, o_{(v,u)} \in \{0,1\} | (u,v) \in E_U \cup E_D\right\} \tag{1}$$

$$\left\{f^{(s,t)}_{(u,v)} \in [0,1] | (s,t) \in P \wedge (u,v) \in E_{(s,t)}\right\} \tag{2}$$

$$\left\{c_{(s,t)} \in \{0,1\} | (s,t) \in P\right\} \tag{3}$$

The orientation variables in (1) are used to encode orientations of the edges: an assignment of 1 to $o_{(u,v)}$ means that the undirected edge $\{u,v\}$ is oriented from u to v. We set $o_{(u,v)} = 1$ and $o_{(v,u)} = 0$ for every directed edge in E_D. The flow variables in (2) are used to measure the existence of flow on every edge of $G_{(s,t)}$ for every pair $(s,t) \in P$. Unlike traditional flow problems, the amount of flow does not concern us, but rather its existence. The closure variables in (3) are used to represent which vertex pairs of the graph are satisfied: an assignment of 1 to $c_{(s,t)}$ will imply that the orientation satisfies (s,t).

The objective is as follows:

$$\text{maximize} \sum_{(s,t)\in P} c_{(s,t)} \tag{4}$$

The constraints are as follows:

$$o_{(u,v)} + o_{(v,u)} = 1 \quad \text{for all } (u,v) \in E_U \tag{5}$$

$$o_{(u,v)} = 1, o_{(v,u)} = 0 \quad \text{for all } (u,v) \in E_D \tag{6}$$

$$f^{(s,t)}_{(u,v)} \leq o_{(u,v)} \quad \text{for all } (s,t) \in P,\ (u,v) \in E_{(s,t)} \tag{7}$$

$$f^{(s,t)}_{(u,v)} \leq \sum_{w:(w,u)\in E_{(s,t)}} f^{(s,t)}_{(w,u)} \tag{8}$$

$$\begin{aligned} &\text{for all } (s,t) \in P,\ (u,v) \in E_{(s,t)}, u \neq s \\ &c_{(s,t)} \leq \sum_{(w,t)\in E_{(s,t)}} f^{(s,t)}_{(w,t)} \quad \text{for all } (s,t) \in P \end{aligned} \tag{9}$$

Constraints in (5) ensure that each undirected edge is oriented in exactly one direction. In case an interaction is unlikely to be directed (e.g. a co-complex interaction), the constraints in (5) can be modified to allow both directions. As we show in the sequel, the assignment of confidence scores to edge orientations allows us to automatically refrain from assigning directions to the vast majority of such interactions. Constraints in (6) ensure that the chosen orientations are consistent with the directed edges of G. Constraints in (7–9) ensure that for every pair $(s,t) \in P$ the closure variable $c_{(s,t)}$ can be set to 1 only if there is a flow from s to t in $G_{(s,t)}$. In detail, the constraints in (7) ensure that the flow respect the edge directions; the constraints in (8) ensure that no edge carries a flow from a vertex u if there is no incoming flow to u and the constraints in (9) ensure that the pair $(s,t) \in P$ is not satisfied if there is no flow from s to t in $G_{(s,t)}$. The overall size of the ILP is $O(|P|\,|E_D \cup E_U|)$.

2.3 A more efficient formulation

The above ILP formulation can be made more efficient by observing that in a biological knockout experiment one measures simultaneously all the effects of a certain knockout (cause), and thus, many pairs in P share the same source. We show below how to unify all the flow computations of a given source, thus significantly reducing the number of variables and subsequently the time of solving the ILP.

Let $S(P)$ be the set of source vertices in P and denote by $M_s = \{(s,t_1), \ldots, (s,t_l)\}$ the set of pairs with source s. Let $G_s = \bigcup_{i=1}^{l} G_{(s,t_i)}$ be the union of all shortest path graphs with s as a

source. Denote its set of edges by E_s. We introduce the following updates to the flow variables and constraints of the program:

$$\left\{f^s_{(u,v)} \in \mathcal{R}^+ | s \in S(P) \text{ and } (u,v) \in E_s\right\} \tag{10}$$

$$f^s_{(u,v)} \leq o_{(u,v)} \quad \text{for all } s \in S(P), (u,v) \in E_s \tag{11}$$

$$f^s_{(u,v)} \leq \sum_{(w,u)\in E_s} f^s_{(w,u)} \tag{12}$$

$$\text{for all } s \in S(P), (u,v) \in E_s, u \neq s$$
$$c_{(s,t)} \leq \sum_{(w,t)\in E_s} f^s_{(w,t)} \quad \text{for all } (s,t) \in P \tag{13}$$

Clearly, any (s,t) path identified by the previous ILP can be used in this formulation as well. Thus, correctness of this formulation depends on showing that any (s,t) path it identifies must be a shortest path. Suppose to the contrary that there exists an (s,t) path $\mathcal{L} \in G_s$ such that $\mathcal{L} \notin G_{(s,t)}$. Let p be the maximal prefix of $\mathcal{L}$ that is a prefix of some shortest path in $G_{(s,t)}$ and let (u,v) be the first edge on $\mathcal{L}$ that is not included in p (i.e. p diverges at u). As (u,v) belongs to some $G_{(s,t')}$ (by definition of G_s), p is also a prefix of a shortest (s,t') path that traverses (u,v). By our assumption, (u,v) is not on a shortest path to t and, hence, the same property holds for all the remaining edges on $\mathcal{L}$ including the last one — (w,t). This results in a contradiction, as the presence of (w,t) in G_s implies it must be on a shortest (s,t) path.

2.4 Computing orientation confidence

In principle, there could be many optimal solutions to the orientation problem. Hence, some of the edges may be arbitrarily oriented in the sense that both of their directions can be used in some optimal solutions. Let s_{opt} be the value of an optimal solution. To compute a measure of confidence in a given orientation of an edge $e = (v,w)$, we rerun the ILP while forcing e to carry the opposite orientation (w,v). We set its *confidence value* to $c_e = s_{\text{opt}} - s_e$, where s_e is the maximum number of satisfied pairs for the modified instance. If $c_e > 0$, then the direction of e is the same in all optimal solutions; thus, we keep c_e as the edge's confidence. If $c_e = 0$ and $e = (v,w)$ is on a shortest s to t path, then there are two cases to consider: (i) the opposite edge is on a shortest s′ to t′ path; in this case an orientation of e may be arbitrary and both (v,w) and (w,v) will be assigned with a confidence of 0. (ii) The opposite edge (w,v) does not participate in any shortest path; thus, there must be some parallel path from s to t that does not visit e, allowing it to be oppositely oriented without altering the objective. In this case, e is assigned a positive confidence according to its contribution to satisfying pairs. Precisely, each pair p that $e = (v,w)$ is used to satisfy contributes $1/n_p$ to its confidence, where n_p is the number of vertices $u \in G_p$ such that $d(s,u) = d(s,w)$ (reflecting the number of alternatives to using e). Now a cutoff may be defined, and an edge e is said to be oriented with *confidence* if and only if its confidence exceeds the cutoff.

2.5 Iterative expansion

In an application of our orientation algorithm, edges that are assigned zero confidence remain undirected. To expand our orientation to include some of those edges, we run several iterations of the algorithm. In each iteration, the directions of edges that were confidently oriented in the previous iteration are held fixed, while the criteria for satisfying the source-target pairs are modified to allow additional orientations. Specifically, let $G_i = (V_i, E_{U_i}, E_{D_i})$ be the input graph at iteration i. For a given source-target pair (s,t), let $E_i(s,t) \subseteq E_{D_i}$ be the set of all directed edges occurring on directed paths from s to t in G_i. Then we define (s,t) to be satisfied by a given orientation of G_i if and only if it admits a directed path under this orientation such that: (i) the path does not intersect $E_i(s,t)$; and (ii) its length is equal to the length of a shortest s-to-t path in $(V_i, E_{U_i}, E_{D_i} \backslash E_i(s,t))$. The algorithm terminates when no new edges are oriented with confidence.

3 RESULTS

3.1 Data acquisition and integration

We gathered physical interactions and cause–effect pair information for *Yeast Saccharomyces cerevisiae* from different sources. We used the PPI dataset 'Y2H-union' from Yu *et al.* (2008), which contains 2930 highly reliable undirected interactions between 2018 proteins. Protein–DNA interactions (PDI), which are directed by nature, were taken from MacIsaac *et al.* (2006), an update of which can be found at (http://fraenkel.mit.edu/improved_map/). We used the collection of PDIs with $P < 0.001$ conserved over at least two other yeast species, which consists of 4113 unique PDIs spanning 2079 proteins. KPIs were collected from Breitkreutz *et al.* (2010) by taking the directed kinase–substrate interactions out of their dataset. This results in 1361 KPIs among 802 proteins. We used a set of 14427 knockout pairs between 2870 genes and proteins from (Hu *et al.*, 2007) by taking their set of unrefined and unfiltered knockout pairs and filtering all pairs with $P < 0.001$.

We integrated the data to obtain a physical network of undirected and directed interactions. We removed self loops and parallel interactions; for the latter, whenever both a directed and an undirected edge were present between the same pair of vertices, we maintained the former only. Pairs of edges that are directed in opposite directions were integrated as an undirected edge. The resulting physical network spans 3686 proteins, 2655 PPIs, 4091 PDIs and 1359 KPIs.

3.2 Application and performance evaluation

We implemented our algorithm, which we call SHORTEST, in C++ using BOOST C++ libraries (version number 1.43.0) and the commercial IBM ILOG CPLEX optimizer (version number 12.5) to solve ILPs.

To evaluate the orientations suggested by our algorithm, we ran the algorithm in a cross-validation setting, hiding the directions of the larger subset of known directed interactions, the PDIs. This subset was considered as undirected *test edges*. Guided by the set of knockout pairs, our program computes orientations for all undirected edges, including the test edges.

We tested our algorithm using the efficient version of Section 2.3, taking advantage of the fact that the knockout pairs were derived from a small set of shared sources. When using 10 expansion rounds, the algorithm oriented with confidence 3379 (82.6%) of the test edges, orienting correctly 2283 (67.6%) of them (hypergeometric $P < 2.7^{-159}$). When restricting the confidence cutoff to 2 ($c_e \geq 2$) and preforming a single iteration (i.e. no expansion rounds), the algorithm oriented with confidence 902 (22%) of the test edges with confidence, orienting correctly 714 (79.2%) of them (hypergeometric $P = 0$). To further evaluate our algorithm, we define precision and recall as defined in Vinayagam *et al.* (2011): we considered each interaction as two different instances, where the interaction from A to B is defined twice, as $A \rightarrow B$ representing its positive instance, and $B \rightarrow A$ representing its negative instance. If an orientation $A \rightarrow B$

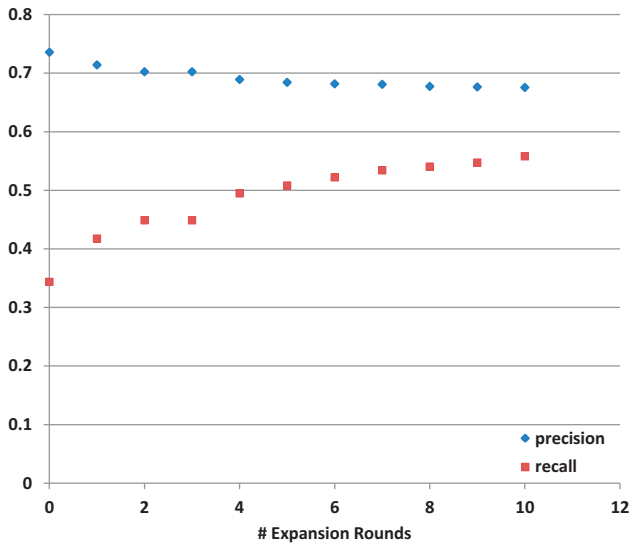

Fig. 1. Performance as a function of the number of expansion rounds used. The confidence cutoff in these experiments is set to 0

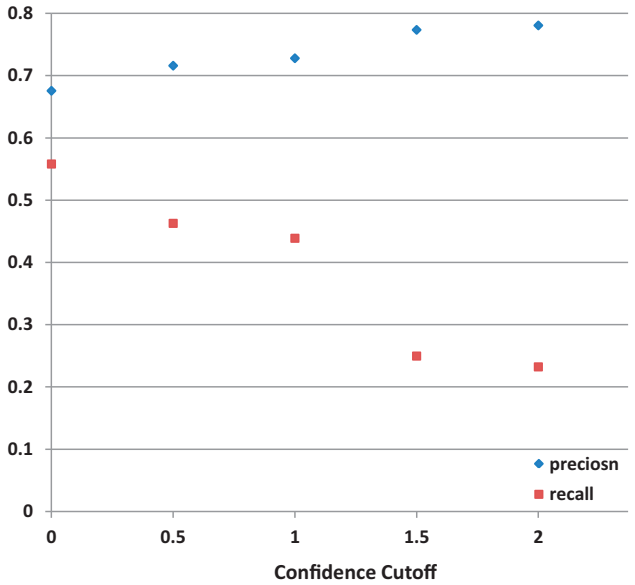

Fig. 2. Performance as a function of the confidence cutoff used. The number of expansion rounds for all experiments is 10 (11 iterations)

exceeds the confidence cutoff then it is classified as positive. As expected, a higher confidence cutoff yields a higher precision [TP/(TP + FP)] and lower recall [TP/(TP + FN)]. More rounds of expansion increase recall and lower the precision. The performance of the algorithm for different choices of a confidence cutoff and number of expansion rounds is summarized in Figures 1 and 2.

As discussed earlier, co-complex interactions are not likely to have a preferred orientation. To test how frequently our algorithm assigns directions to such interactions, we evaluated our results against a list of 393 known complexes as annotated by the Gene Ontology (GO) (Ashburner *et al.*, 2000) and downloaded from Saccharomyces Genome Database (SGD) (Cherry *et al.*, 2012) (June 2011). We found that 9% (720) of the interactions in the network lie within known complexes. Using a confidence cutoff of 0, 50% of all interactions in the network are assigned a direction, but only 3.4% of those (139 interactions) lie within known complexes. This number is significantly small compared with the random expectation (hypergeometric $P < 1.17^{-70}$) testifying the quality of our predictions.

To study the effect of the amount of cause–effect pairs on the orientation, we applied the algorithm with increasing portions (chosen at random) of pairs. As evident from Figure 3, the more pairs the higher are the measured recall and precision, supporting our use of the cause–effect pairs to guide the orientation. A high percentage (~93%) of the knockout pairs are satisfied throughout the experiments. Our results seem more robust to variations in the percentage of directed interactions in the input network. Even when eliminating the KPIs, an F-measure of 0.26 was attained (cutoff = 0, no expansion rounds).

To evaluate the scalability of the method, we downloaded the full set of PPIs from BioGrid (Stark *et al.*, 2006) (October 2013), containing 89 512 unique interactions. Although the preprocessing time increased to 5 min, the ILP solution was still obtained in under a second. The orientation obtained had high quality: 2061 (50%) of the 4091 PDIS were oriented with confidence, and 1691

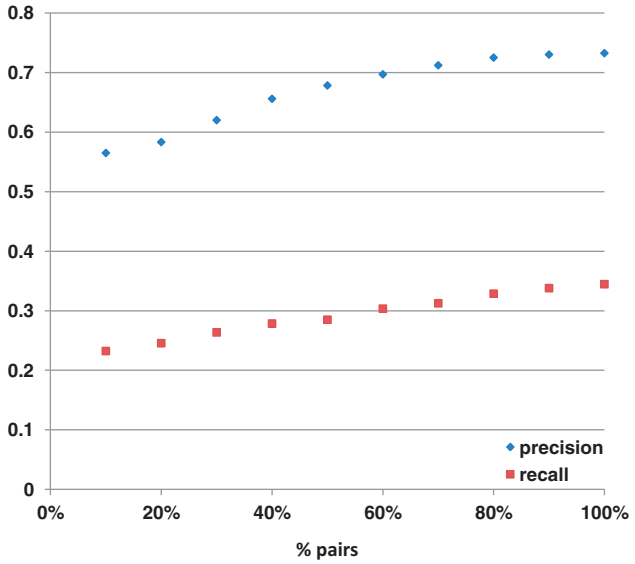

Fig. 3. Performance as a function of the percentage of cause-effect pairs guiding the orientation. The confidence cutoff in these experiments is set to 0 and there are no expansion rounds

(82%) of the orientations were accurate (cutoff = 0, no expansion rounds). Expectedly, using a higher confidence cutoff yielded higher precision values (e.g. 90% for a cutoff of 1).

3.3 Comparison to previous work

We compared our approach with two previous state-of-the-art methods: the MIXED algorithm (Silverbush *et al.*, 2011) and the random orientation followed by local search algorithm of (Gitter *et al.*, 2011). Below we provide precision, recall and F-measure values for each approach, where the latter combines both

precision and recall into a single value of their harmonic mean: $\text{F-measure} = 2 * \left(\frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}}\right)$.

The results of the comparison to MIXED are shown in Figure 4, where SHORTEST was applied in its maximum coverage setting. The 8-fold recall increase by SHORTEST is dramatic, albeit at the price of lower precision. To investigate the precision differences further, we compared both algorithms performances over the subset oriented by both. In this subset, both approaches inferred identical orientation, achieving 94.8% precision over this evidently easier subset.

We used an analogous procedure to infer the directions of the smaller subset of KPIS, hiding its directions while keeping the PDIS as a directed subset and guided by the same set of knockout pairs as in the previous experiment. We were able to orient with confidence 1077 (79.2%) of them, orienting correctly 600 (56%) with an F-measure of 0.49. In comparison, the MIXED approach oriented 52 (3.8%) of the edges, orienting correctly 46 (88.4%) with an F-measure of 0.065. As before, within the easier subset of edges oriented by both methods, a much higher precision of 85.4% was obtained.

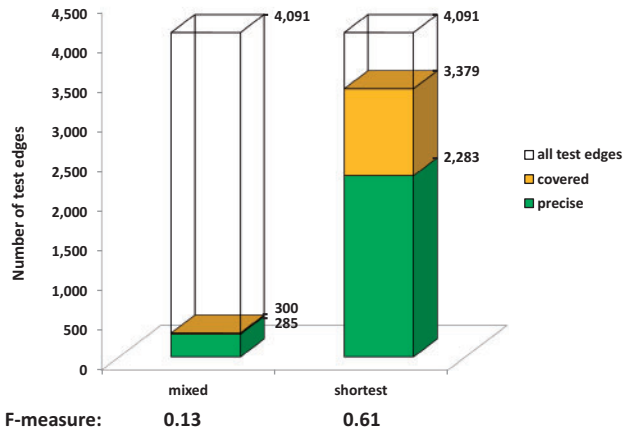

Fig. 4. Performance comparison of MIXED and SHORTEST

To compare with the method of (Gitter *et al*., 2011), we used the same data and network used in their article (A.Gitter, private communication). The unweighted PPI network was extracted from (Stark *et al*., 2006) spanning 3446 proteins and 10 945 undirected interactions. To guide our orientation, we used the same source–target pair set manually chosen by Gitter *et al*. (2011). A test set was extracted from KEGG (Kanehisa and Goto, 2000) and the Science Signaling Database of Cell Signaling. It contains 91 directed interactions among 69 proteins. Gitter *et al*. (2011) oriented 2447 interactions of the initial network. Their orientation oriented 55 of the 91 tested directed interactions (60.4%), of which 37 orientations were accurate (67.3%). Applying our algorithm in its maximum recall setting, resulted in orienting 5458 interactions among 2221 proteins, >2-fold increase compared with Gitter *et al*. (2011). On the test set, our method oriented with confidence 79 interactions (86.8%), of which 61 orientations were accurate (77.2%), providing both higher recall (67.0% versus 40.7%), higher precision (77.2% versus 67.3%) and achieving a higher F-measure of 0.71 compared with 0.51. The comparison is summarized in Tables 1 and 2 and depicted in Figures 5 and 6.

As pointed out by Gitter *et al*. (2011), scalability is an important issue for methods analyzing high-throughput datasets, especially because current data are incomplete and networks for other organisms may be larger than those for yeast. Thus Gitter *et al*. (2011) had examined their running time using the different algorithms suggested in their paper. We compared our running time when using the same dataset. Running our algorithm one iteration only (which includes running the ILP for each test edge to determine confidence), reaching next to identical recall as Gitter *et al*. (2011) and much higher precision, took 8 s on average, faster than all algorithms used in Gitter *et al*. (2011).

Running our approach when allowing the maximum number of expansion rounds up to exhaustion (reaching five iterations in total), reaching higher recall and higher precision, took 75 s,

Table 1. A comparison of GITTER and SHORTEST for different numbers of expansion rounds. The best score for each measure is highlighted

Performance measure	$\text{SHORTEST}_{\#0}$	$\text{SHORTEST}_{\#1}$	$\text{SHORTEST}_{\#2}$	$\text{SHORTEST}_{\#3}$	$\text{SHORTEST}_{\#\infty}$	GITTER
Precision (%)	**81.6**	79.0	79.5	76.9	77.2	67.3
Recall (%)	44.0	53.8	63.7	65.9	**67.0**	40.7
F-measure	0.57	0.64	0.71	0.71	**0.72**	0.51

Table 2. A comparison of GITTER and SHORTEST for different confidence cutoffs. The best score for each measure is highlighted

Performance measure	$\text{SHORTEST}_{\text{conf}>0}$	$\text{SHORTEST}_{\text{conf}\geq 1}$	$\text{SHORTEST}_{\text{conf}\geq 2}$	$\text{SHORTEST}_{\text{conf}\geq 3}$	GITTER
Precision (%)	77.2	74.6	82.0	**90.9**	67.3
Recall (%)	**67.0**	58.2	45.1	33.0	40.7
F-measure	**0.72**	0.65	0.58	0.48	0.51

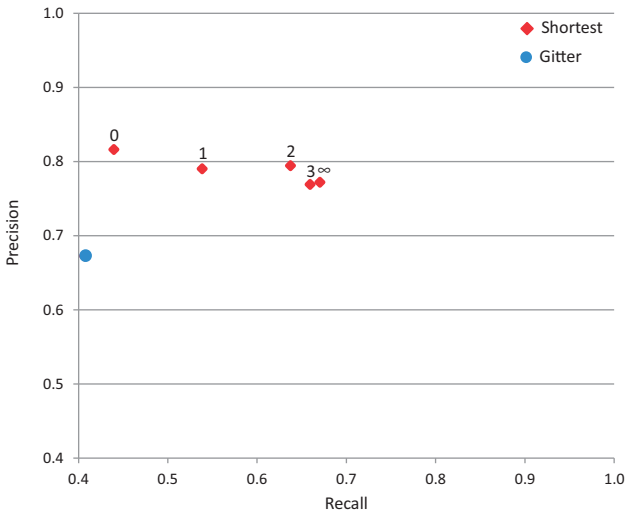

Fig. 5. Performance comparison of GITTER and SHORTEST using different number of expansion rounds. Labels denote the number of expansion rounds used by SHORTEST

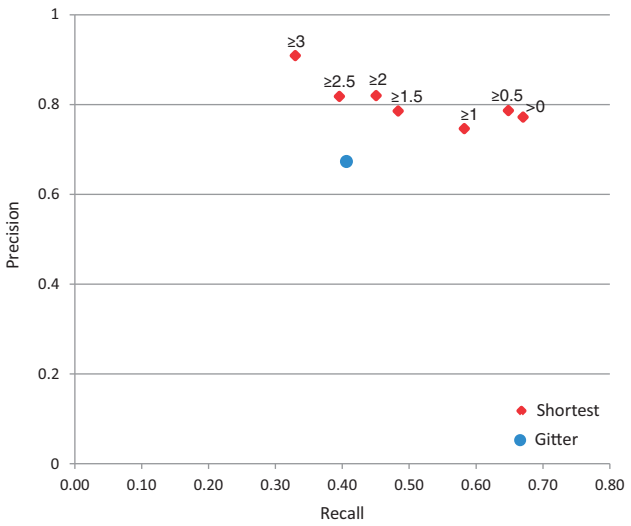

Fig. 6. Performance comparison of GITTER and SHORTEST using different confidence cutoff. Labels denote the confidence cutoff used by SHORTEST

challenged only by the random approach. Running time is presented in Table 3.

4 DISCUSSION

The orientation of a network is key to understanding its function. Here we have presented the SHORTEST approach, which allows us for the first time to confidently orient the majority of the edges in a network. The most natural use of such an orientation is in enhancing methods for pathway inference. Specifically, current pathway inference algorithms, like Scott *et al.* (2006), receive as input an undirected PPI network and search for likely paths that start and end at specific proteins.

Table 3. Running time comparison

SHORTEST		Gitter et al.		
No Expansion	Unlimited Expansion	RANDOM	MIN-SAT	MAX-CSP
8.0	75.0	16.2	2742.5	10806.7

Note: Algorithm run times in seconds. The full dataset was used, with 256 source–target pairs.

The search space of such methods can be greatly reduced by the orientation information.

To test the potential utility of our orientation method in a pathway inference context, we checked its power in filtering candidate pathways by their agreement with the predicted orientations. To this end, we inspected the source–target pairs connected by more than one possible pathway within the networks in Section 3.3: (i) In the orientation instance because of Gitter *et al.* (2011), there are 740 possible shortest paths from source to target (6.2 paths per source–target pair). However, when filtering these paths against the confident orientation predictions, only 589 (4.9 on average per pair) remain. (ii) In the orientation instance because of Silverbush *et al.* (2011) there are 46 782 shortest paths from source to target (11.9 on average per pair), of which only 28 273 (7.2 on average per pair) agree with the confident orientations. Thus, the use of our method can potentially reduce the search space by up to 40%.

ACKNOWLEDGEMENTS

The authors would like to thank Dr. Anthony Gitter for his help in the comparison to Gitter *et al.* (2011).

Funding: R.S. was supported by a research grant from the Israel Science Foundation (grant no. 241/11). This study was supported in part by a fellowship from the Edmond J. Safra Center for Bioinformatics at Tel-Aviv University.

Conflict of Interest: none declared.

REFERENCES

Arkin,E.M. and Hassin,R. (2002) A note on orientations of mixed graphs. *Discrete Appl. Math.*, **116**, 271–278.

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

Blokh,D. *et al.* (2013) The approximability of shortest pathbased graph orientations of protein–protein interaction networks. *J. Comput. Biol.*, **20**, 945–957.

Breitkreutz,A. *et al.* (2010) A global protein kinase and phosphatase interaction network in yeast. *Science*, **328**, 1043–1046.

Cherry,J.M. *et al.* (2012) *Saccharomyces* genome database: the genomics resource of budding yeast. *Nucleic Acids Res.*, **40**, D700–D705.

Dorn,B. *et al.* (2011) Exploiting bounded signal flow for graph orientation based on cause–effect pairs. *Algorithms Mol. Biol.*, **6**, 21.

Elberfeld,M. *et al.* (2013) Approximation algorithms for orienting mixed graphs. *Theor. Comput. Sci.*, **483**, 96–103.

Fields,S. (2005) High-throughput two-hybrid analysis. *FEBS J.*, **272**, 5391–5399.

Gavin,A. *et al.* (2002) Functional organization of the yeast proteome by systematic analysis of protein complexes. *Nature*, **415**, 141–147.

Gitter,A. *et al.* (2011) Discovering pathways by orienting edges in protein interaction networks. *Nucleic Acids Res.*, **39**, e22.
Hu,Z. *et al.* (2007) Genetic reconstruction of a functional transcriptional regulatory network. *Nat. Genet.*, **39**, 683–687.
Kanehisa,M. and Goto,S. (2000) Kegg: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **28**, 27–30.
MacIsaac,K. *et al.* (2006) An improved map of conserved regulatory sites for saccharomyces cerevisiae. *BMC Bioinformatics*, **7**, 113.
Medvedovsky,A. *et al.* (2008) An algorithm for orienting graphs based on cause-effect pairs and its applications to orienting protein networks. In: Crandall,K. and Lagergren,J. (eds) *Algorithms in Bioinformatics. Lecture Notes in Computer Science*. Vol. 5251, Springer, Berlin, Heidelberg, pp. 222–232.
Navlakha,S. *et al.* (2012) A Network-based approach for predicting missing pathway interactions. *PLoS Comput. Biol.*, **8**, e1002640.
Ourfali,O. *et al.* (2007) SPINE: a framework for signaling-regulatory pathway inference from cause-effect experiments. *Bioinformatics*, **23**, i359–i366.
Scott,J. *et al.* (2006) Efficient algorithms for detecting signaling pathways in protein interaction networks. *J. Comput. Biol.*, **13**, 133–144.
Silverbush,D. *et al.* (2011) Optimally orienting physical networks. *J. Computat. Biol.*, **18**, 1437–1448.
Stark,C. *et al.* (2006) Biogrid: a general repository for interaction datasets. *Nucleic Acids Res.*, **34** (**Suppl. 1**), D535–D539.
Vinayagam,A. *et al.* (2011) A directed protein interaction network for investigating intracellular signal transduction. *Sci. Signal.*, **4**, rs8.
Yeang,C.-H. *et al.* (2004) Physical network models. *J. Comput. Biol.*, **11**, 243–262.
Yu,H. *et al.* (2008) High-Quality binary protein interaction map of the yeast interactome network. *Science*, **322**, 104–110.