Sign assignment problems on protein networks

Shay Houri and Roded Sharan

Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel.
{shayhour,roded}@post.tau.ac.il.

Abstract. In a maximum sign assignment problem one is given an undirected graph and a set of signed source-target vertex pairs. The goal is to assign signs to the graph's edges so that a maximum number of pairs admit a source-to-target path whose aggregate sign (product of its edge signs) equals the pair's sign. This problem arises in the annotation of physical interaction networks with activation/repression signs. It is known to be NP-complete and most previous approaches to tackle it were limited to considering very short paths in the network. Here we provide a sign assignment algorithm that solves the problem to optimality by reformulating it as an integer program. We apply our algorithm to sign physical interactions in yeast and measure our performance using edges whose activation/repression signs are known. We find that our algorithm achieves high accuracy (89%), outperforming a state-of-the-art method by a significant margin.

Key words: network annotation, protein-protein interaction, activation, repression, integer linear program.

1 Introduction

A holy grail of biological research is obtaining a working model of the cell. Protein-protein interactions (PPIs) form the skeleton of signal processing circuitry. Despite their importance, current models of PPI networks are mostly topological, lacking the underlying logic of those circuits. In this paper we tackle combinatorial problems arising in the annotation of PPI networks with signs of activation/repression. Constructing such an annotation is a key step in deciphering the logic of those networks.

Current technologies for PPI mapping do not provide information on the direction of signal flow or the activation/repression effect of the measured interactions. Such information can be gained from additional, indirect data such as gene expression knockout experiments. The latter pinpoint pairs of genes such that the deletion of a gene (cause) leads to a change in the expression level of the other gene (effect) with a certain sign (down- or up-regulation). Yeang et al. [7] were the first to suggest a computational framework for annotating networks with directions and signs given cause-effect data. Later work by Ourfali et al. [4] formulated the sign assignment problem as an integer programming problem, aiming to maximize the expected number of cause-effect pairs that can be explained. The main caveat of these works is the need to consider an exponential space of paths in the network during the optimization process, necessitating heuristic approaches that focus on very short paths (with at most 3 edges) and miss information contained in longer ones. Finally, Peleg et al. [5] gave an algorithm that assigns signs to nodes in a manner that is independent of a PPI network; these signs are then used in order to predict edge signs in the network. Their algorithm was shown to compare favorably to the previous sign assignment algorithms of [7, 4].

Here we revisit the problem of assigning signs to the edges of the PPI network so as to best explain a given cause-effect data set. Following previous work, we consider a cause-effect pair of a certain sign to be *explained* by an assignment if the network contains a path from the cause to the effect whose aggregate sign (product of the signs along its edges) is opposite (due to the knockout) to the sign of the pair. Our goal is to assign signs to the edges of the network so that a maximum number of cause-effect pairs can be explained. We study the resulting Maximum Sign Assignment (MSA) problem, which was shown to be NP-hard in [5]. We provide a polynomial 0.878-approximation algorithm for it and for a constrained variant where some of the edges are pre-assigned with signs. We further provide an integer programming formulation of the problem that allows us to solve it to optimality on current networks. We apply our algorithm to annotate a network of physical interactions in yeast, obtaining high success rates against known sign data. In comparison to our previous state-of-the-art method by Peleg et al. [5] the current approach attains significantly higher accuracy levels (89% vs. 68-73%).

2 Preliminaries

Let $G = (V, E)$ be an undirected graph, representing a protein network, with a set V of vertices and a set E of edges. We assume that each edge $e \in E$ is associated with a sign $s(e) \in \{-1, +1\}$ (which may be unknown) describing its activation/repression effect. For a path P, we define its sign as the product of the signs of its edges (assuming their signs are known), i.e., $s(P) = \prod_{e \in P} s(e)$. Our goal is to infer the edge signs from knockout data which can be summarized as triples of a knockout gene u, an affected gene v and the effect sign s. Our assumption, following [7], is that each such triple can be explained by a path in G from u to v whose sign is $-s$ (and due to the knockout the observed effect is s). The problem is formally stated as follows:

Definition 1 (Maximum Sign Assignment (MSA)). *Given an undirected graph $G = (V, E)$ and a collection of signed vertex pairs $\{(u, v, s)\}_{u,v \in V, s \in \{+1,-1\}}$, assign signs to the edges of G so that a maximum number of signed pairs (u, v, s) admit a path whose sign is s.*

In [5] it is shown that MSA is NP-complete and hard to approximate to within a factor of 11/12. In the next section we shall develop approximation algorithms for the problem and a constrained variant of it.

We assume that the input graph $G = (V, E)$ is connected, otherwise one can operate independently on each of its connected components. We note that any cycle in G can be contracted without affecting the optimum solution. This follows from the observation that by assigning -1 to one of its edges and +1 to the remaining edges, every pair of vertices that admit a path that visits this cycle can be satisfied. Thus, we may assume, w.l.o.g., that G is a tree. Given a sign assignment to the edges of G, we say that two vertices admit a *positive path* (resp., negative path) between them if there exists a path connecting the vertices whose sign is +1 (resp., -1).

3 An approximation algorithm for MSA

We approximate the problem by reducing it into a MAX-E2-LIN2 problem. In MAX-E2-LIN2 the input consists of a set of linear equations over Z_2 with at most two variables per equation; the goal is to find a solution that maximizes the number of satisfied equations. MAX-E2-LIN2 is known to be hard to approximate to within 11/12. On the positive side, it can be 0.878-approximated using the semi-definite programming approach of Geomans and Williamson [2].

Given an instance of MSA, we reduce it by representing every signed pair (u, v, s) as a linear equation over Z_2: $x_u \oplus x_v = f(s)$, where $f(s) = (1 - s)/2$.

Theorem 1. *The reduction is approximation preserving.*

Proof. We prove that there exists a solution of the MAX-E2-LIN2 instance that satisfies k equations iff there is a solution to the MSA instance that satisfies k of the pairs. Let X be a solution to the MAX-E2-LIN2 instance which satisfies k of the input equations. X induces a partition of V into two parts: $\{v \in V : X_v = 1\}$ and $\{v \in V : X_v = 0\}$. Vertices that did not participate in the equations are arbitrarily assigned to one of the parts. Now, we can sign the edges as follows: edges that cross between the two parts are assigned a minus sign; the rest are assigned a plus sign. It is easy to check that under this assignment (and since G is connected) every pair (u, v, s) whose corresponding equation was satisfied admits a path of sign s.

Conversely, let S be a sign assignment which satisfies k pairs. Take an arbitrary vertex v and set $X_v = 0$. For any other vertex $u \in V$ that has a positive path to v, set $X_u = 0$; otherwise, u must have a negative path to v and we set $X_u = 1$. Since G is a tree, every two vertices have a unique path between then, so its sign is well defined. It is again easy to check that the proposed assignment satisfies at least k equations (corresponding to the k satisfied pairs).

Corollary 1. *MSA can be 0.878-approximated.*

4 Dealing with assignment constraints

The discussion thus far assumed that any assignment is legal. In practice, some edge signs are known in advance. Under such constraints the problem cannot be reduced to a tree anymore, since some cycles cannot be contracted without affecting the optimum solution. For a given sign assignment, a pair of vertices are called *ambiguous* if they admit both a positive and a negative path between them. In the following we call a graph *strongly signed* if there exists an assignment to its yet unsigned edges such that every pair of vertices is ambiguous.

Lemma 1. *A cycle is strongly signed iff it admits an assignment whose aggregate sign is -1.*

Proof. If a cycle is strongly signed then by definition the product of signs along its edges is -1, as every two vertices on the cycle have exactly two paths connecting them with one being negatively-signed and the other being positively-signed.

Conversely, if the product of edge signs along the cycle is -1 then every two vertices on the cycle are ambiguous.

Our algorithm for the constrained case relies on decomposing the input graph to its blocks. Recall that a *block* is a 2-vertex connected component. A block may contain a single edge, or else it contains a cycle. Any block of size at least 3 admits an *open ear decomposition* of its edges $(P_0, \ldots, P_k)$. P_0 may be any cycle of the block. Each $P_i, i > 0$ is a path whose two end-points are distinct and included in previous ears $P_j, j < i$.

Lemma 2. *A block G of size at least 3 is strongly signed iff it admits an assignment such that the aggregate sign of some cycle in G is -1.*

Proof. If G is strongly signed then pick any two ambiguous vertices and their connecting positive and negative paths span a cycle as desired.

We prove the opposite direction by induction on the number of ears in the decomposition of G. The base case trivially holds as the first ear is a cycle. Suppose we constructed a partial decomposition $P_0, \ldots, P_{k-1}$, where P_0 is the strongly signed cycle, and let P_k be a new ear to be added to the decomposition. By the induction hypothesis, every pair of vertices in $P_0 \cup \ldots \cup P_{k-1}$ is ambiguous. By this property, it is also trivial to see that every pair of vertices in P_k are ambiguous – simply use as one path the path connecting them in P_k and as the other path a path of opposite sign that visits the previous ears. Such a path exists by the ambiguity of the end-points of P_k. Finally, consider two vertices $u \in P_0 \cup \ldots \cup P_{k-1}$ and $v \in P_k$. Let w be one of the end-nodes in P_k such that $w \neq u$. Since u and w are ambiguous the claim follows.

Since every unsigned edge in a block (of size at least 3) is on some cycle and can be used to force the aggregate sign of this cycle, we get the following corollary:

Corollary 2. *A block of size at least 3 that is not strongly signed must be completely pre-assigned with signs to its edges. Moreover, every pair of vertices in the block is unambiguous.*

Given an input graph G, we can build a tree-like decomposition of G into its blocks and cut-vertices. In this decomposition, the tree vertices are the blocks and cut vertices of G; the tree edges connect cut vertices to the blocks that contain them. Every path in this tree can be translated to a path in G. If a block in G is strongly signed, all input pairs whose connecting path visits this block can be satisfied and the block can be contracted. Every other block must be a single edge or pre-assigned with signs. In particular, if we consider any input pair whose connecting path visits a pre-assigned block, then any sub-path used by the connecting path through the block has a pre-defined sign (since the corresponding pair of cut vertices used are unambiguous). By multiplying the sign of every input pair by the pre-defined signs of the sub-paths through such pre-assigned blocks along its connecting path, we can contract all pre-assigned blocks. This contraction reduces G into a tree and thus we can apply the approximation algorithm of the previous section to it.

Corollary 3. *Constrained-MSA can be 0.878-approximated.*

4.1 An ILP formulation

As we have seen, both the constrained and unconstrained versions of MSA can be reduced to solving a system of linear equations over Z_2. The latter problem can be easily translated to an integer linear program (ILP) formulation and solved to optimality using an industrial solver such as CPLEX. Let $x_{i_1} \oplus x_{i_2} = r_i$ denote the i-th equation in the reduced instance. The translation is done by defining auxiliary variables y_i denoting whether equation i is satisfied. The exact formulation is:

$$\begin{array}{lll} \max & \sum_i y_i & \\ \text{s.t.} & y_i = (x_{i_1} + x_{i_2} + r_i + 1 - 2\gamma_i) & \forall i \\ & y_i, x_{i_1}, x_{i_2}, \gamma_i \in \{0,1\} & \forall i \end{array}$$

where the γ_i-s are auxiliary variables that allow computing the parity of x_{i_1}, x_{i_2} and r_i.

5 Experimental Results

5.1 Data acquisition and integration

We gathered yeast physical interactions, including PPIs, protein-DNA interactions (PDIs) and kinase-substrate and phosphatase-substrate interactions (KPIs), from different sources. We used the PPI data set "Y2H-union" from Yu et al. [8], which contains 2,930 highly reliable undirected interactions among 2,018 proteins. The PDI data were taken from MacIsaac et al. [3]. We used the collection of PDIs with $p < 0.001$ conserved over at least two other yeast species, which consists of 4,095 unique PDIs spanning 2,079 proteins. The KPIs were collected from Breitkreutz et al. [1] and included 1,361 KPIs among 802 proteins. We complemented the interaction data by information on cause-effect pairs. To this end, a set of 110,487 knockout pairs among 6,228 proteins was taken from Reimand et al. [6]. We integrated the data to obtain a physical network of 3,659 proteins, 2,649 PPIs, 4,095 PDIs and 1,361 KPIs, which spans 52,650 of the cause-effect pairs.

For validation purposes we collected sign information on the PDIs and KPIs in our data set. For a given PDI, we assumed that it is positive (resp., negative) if the transcription-factor involved is an activator (resp., repressor). We retrieved activator-repressor information from the gene ontology (GO:0045893 for activators and GO:0045892 for repressors), obtaining signs for 1,938 PDIs. For a given KPI, we assumed that it is positive if one of the two interacting proteins was a kinase and the other was not a phosphatase; we assumed that it is negative if one of the proteins was a phosphatase and the other was not a kinase. Overall, we estimated signs for 1,148 KPIs. Note that the signs of kinase-phosphatase edges remain undecided.

5.2 Evaluation procedure

The algorithm can assign signs to two types of edges: (i) an edge within a strongly signed block when it is the *last* unsigned edge of the block; (ii) and an edge "participating" in the ILP. The former assignments are very scarce in our setting as typically most of the edges reside in a single huge block. The latter assignments are sensitive to the exact solution chosen by the ILP. To ensure that we focus on non-arbitrary assignments, we test our confidence in each edge assignment in the following way: for an edge e that is assigned sign s by the ILP, we rerun the ILP while forcing the sign of e to be $-s$. If the resulting objective (number of satisfied pairs) is equal to the original one, we view the assignment as arbitrary; otherwise, the new objective is smaller than the optimal one and we say that the assignment of e is *confident*. Finally, we focus the evaluation on those confidently assigned edges.

5.3 Results

We applied our algorithm in three main settings and evaluated its results (see Table 1). First, we used the entire unsigned network. In this case there are no "last" edges whose signs are determined. Out of 167 edges participating in the ILP, the assignment to 127 of them was confident. 56 of the 127 edges were PDIs with known signs; none was a KPI with a known sign (with only 3 KPIs among the 167 participating edges and the rest in strongly signed blocks). The algorithm predicted correctly the signs of 50 of the 56 edges, yielding a success rate of 89.3%.

Next, we reran the algorithm while using the known signs of the KPIs and focusing the evaluation on the PDIs. While the additional information slightly increased the number of confidently assigned edges (to 135), the intersection with the known PDIs remained the same (56) and again 50 of these were predicted correctly.

Finally, we reran the algorithm on two smaller network instances: one containing PPIs and PDIs but no KPIs, and the other containing PDIs only. As the network gets smaller, the strongly signed components cover less edges and, thus, a larger fraction of the edges participate in the ILP and are assigned with signs. Indeed, in the first application (PPIs and PDIs) more confident sign assignments were made (191) compared to the original application. Out of 64 PDIs with known signs that were confidently assigned, 57 were correctly predicted, yielding a success rate of 89%. When operating on the PDI network only, the largest number of confident sign assignments were made (263). Out of 79 PDIs with known signs that were confidently signed, 70 were correctly predicted, yielding a success rate of 89%.

Notably, in all cases reported above the percent of positive PDIs among the confidently signed ones ranged from 68-72%, significantly lower than the success rate of the algorithm. We further compared our results to the state-of-the-art approach of Peleg et al. [5]. The approach of Peleg et al. works in two stages. In the first stage, the cause-effect pair information is used to split the vertices into groups so that signs of pairs within the same group tend to be positive while signs of pairs crossing groups tend to be negative. Notably, this step is independent of a physical network. The second stage uses the group information to annotate the network's edges with signs. Precisely, the sign of an edge is determined based on the majority sign of pairs that come from the same groups as the edge's endpoints. This implies that in the setting we have used it suffices to apply the method of Peleg et al. once to the entire knockout data, use it to annotate the edges of the integrated network, and then evaluate its performance with respect to each one of the test networks. The results are summarized in Table 1 and show a marked advantage to our approach with success rates higher by 16-22% across the different test cases.

Network	Size	#Assignments	#Confident	#Known	%Success	%Success of [5]
KPI+PDI+PPI	8,105	167	127	56	**89.3**	67.9
KPI (signed)+ PDI+PPI	8,105	164	135	56	**89.3**	67.9
PDI+PPI	6,798	228	191	64	**89.1**	70.3
PDI	4,095	290	263	79	**88.6**	73.4

Table 1. Assessment of results and comparison to the network-free approach of Peleg et al. [5]. The best result in each row appears in bold.

6 Conclusions

We provided an ILP based algorithm to predict edge signs in signaling-regulatory networks. Our algorithm can account for known signs and evaluate the confidence of the assignment. In application to real data it exhibits high success rates of 89%, outperforming the state-of-the-art method of Peleg et al. [5] by a significant margin.

While the annotation results are promising, a current limitation of the algorithm is the low percent of edges that are confidently assigned. One way to tackle this problem is by combining sign prediction with direction prediction which could potentially eliminate many degrees of freedom in the solution and yield a higher percent of annotated edges.

Acknowledgments

We thank Richard Karp for fruitful discussions about this work. RS was supported by a research grant from the Israel Science Foundation (grant no. 241/11).

References

1. A. Breitkreutz, H. Choi, J. R. Sharom, L. Boucher, V. Neduva, B. Larsen, Z. Lin, B. Breitkreutz, C. Stark, G. Liu, J. Ahn, D. Dewar-Darch, T. Reguly, X. Tang, R. Almeida, Z. S. Qin, T. Pawson, A. Gingras, A. I. Nesvizhskii, and M. Tyers. A global protein kinase and phosphatase interaction network in yeast. *Science*, 328(5981):1043–1046, May 2010.
2. M. Goemans and D. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *J. ACM*, 42:1115–1145, 1995.
3. K. MacIsaac, T. Wang, D. B. Gordon, D. Gifford, G. Stormo, and E. Fraenkel. An improved map of conserved regulatory sites for saccharomyces cerevisiae. *BMC Bioinformatics*, 7(1):113, 2006.
4. O. Ourfali, T. Shlomi, T. Ideker, E. Ruppin, and R. Sharan. SPINE: a framework for signaling-regulatory pathway inference from cause-effect experiments. *Bioinformatics*, 23(13):i359–i366, 2007.
5. T. Peleg, N. Yosef, E. Ruppin, and R. Sharan. Network-free inference of knockout effects in yeast. *PLoS Computational Biology*, 6(1):e1000635, 2010. PMID: 20066032.
6. J. Reimand, J. M. Vaquerizas, A. E. Todd, J. Vilo, and N. M. Luscombe. Comprehensive reanalysis of transcription factor knockout expression data in saccharomyces cerevisiae reveals many new targets. *Nucleic Acids Research*, 38(14):4768–4777, 2010.
7. C. Yeang, T. Ideker, and T. Jaakkola. Physical network models. *Journal of Computational Biology*, 11(2-3):243–262, 2004.
8. H. Yu, P. Braun, M. A. Yildirim, I. Lemmens, K. Venkatesan, J. Sahalie, T. Hirozane-Kishikawa, F. Gebreab, N. Li, N. Simonis, T. Hao, J. Rual, A. Dricot, A. Vazquez, R. R. Murray, C. Simon, L. Tardivo, S. Tam, N. Svrzikapa, C. Fan, A. de Smet, A. Motyl, M. E. Hudson, J. Park, X. Xin, M. E. Cusick, T. Moore, C. Boone, M. Snyder, F. P. Roth, A. Barabasi, J. Tavernier, D. E. Hill, and M. Vidal. High-Quality binary protein interaction map of the yeast interactome network. *Science*, 322(5898):104–110, Oct. 2008.