BIOINFORMATICS

Vol. 23 ISMB/ECCB 2007, pages i359–i366
doi:10.1093/bioinformatics/btm170

SPINE: a framework for signaling-regulatory pathway inference from cause-effect experiments

Oved Ourfali[1], Tomer Shlomi[1], Trey Ideker[3], Eytan Ruppin[1,2] and Roded Sharan[1,*]

[1]School of Computer Science, [2]School of Medicine, Tel-Aviv University, Tel-Aviv, Israel and [3]Department of Bioengineering, University of California, San Diego, CA 92093, USA

ABSTRACT

Motivation: The complex program of gene expression allows the cell to cope with changing genetic, developmental and environmental conditions. The accumulating large-scale measurements of gene knockout effects and molecular interactions allow us to begin to uncover regulatory and signaling pathways within the cell that connect causal to affected genes on a network of physical interactions.

Results: We present a novel framework, SPINE, for Signaling-regulatory Pathway INferencE. The framework aims at explaining gene expression experiments in which a gene is knocked out and as a result multiple genes change their expression levels. To this end, an integrated network of protein–protein and protein-DNA interactions is constructed, and signaling pathways connecting the causal gene to the affected genes are searched for in this network. The reconstruction problem is translated into that of assigning an activation/repression attribute with each protein so as to explain (in expectation) a maximum number of the knockout effects observed. We provide an integer programming formulation for the latter problem and solve it using a commercial solver.

We validate the method by applying it to a yeast subnetwork that is involved in mating. In cross-validation tests, SPINE obtains very high accuracy in predicting knockout effects (99%). Next, we apply SPINE to the entire yeast network to predict protein effects and reconstruct signaling and regulatory pathways. Overall, we are able to infer 861 paths with confidence and assign effects to 183 genes. The predicted effects are found to be in high agreement with current biological knowledge.

Availability: The algorithm and data are available at http://cs.tau.ac.il/~roded/SPINE.html

Contact: roded@post.tau.ac.il

1 INTRODUCTION

High-throughput technologies are routinely used to map molecular interaction within the cell. These include chromatin immuno-precipitation experiments (Lee *et al.*, 2002) for measuring protein–DNA interactions, and yeast two-hybrid assays (Fields and Song, 1989) and co-immunoprecipitation screens (Gavin *et al.*, 2002) for measuring protein–protein interactions (PPIs). The resulting maps promise to shed light on the regulatory mechanisms that allow the propagation of certain signals within the cell. In particular, we aim at explaining gene expression experiments in which a gene is knocked out and as a result multiple genes change their expression levels.

A series of works concerned the inference of functional interactions in small signaling-regulatory networks from perturbation experiments (Andree *et al.*, 2005; Kholodenko *et al.*, 2002; Yalamanchili *et al.*, 2006). The problem of explaining knockout experiments using a physical network was introduced by Yeang *et al.* (2004). The authors looked at a specific setting of the problem where the objective is to infer for each edge of the network a direction, specifying the direction in which information flows through that interaction, and a sign, representing the regulatory effect of the interaction (activation or repression). The annotated network can then be used for pathway inference, although this inference problem was not explicitly treated by Yeang *et al.*

To tackle the network annotation problem, Yeang *et al.* assumed that explanatory pathways should satisfy several constraints. In particular, they should be directed from the knockout gene to the affected gene, and the aggregate sign along the pathway's edges should complement the sign of the knockout effect. They devised a probabilistic framework for the annotation task, in which the different constraints are translated into potential functions (for edges, paths and knockouts), and the objective is to maximize the product of all potentials. A follow up work (Yeang *et al.*, 2005) devised methods for validation and refinement of the network model. A similar model was used by Yeang and Vingron (2006) to integrate metabolic data with regulatory interactions and gene-knockout data. Workman *et al.* (2006) used the same model to infer the regulatory pathways that control the response to DNA damage.

While Yeang *et al.* provided an elegant formulation of the annotation problem, their solution method suffers from several shortcomings: (i) The objective function does not distinguish between situations in which few knockout pairs are explained by many pathways each, and those in which many pairs are explained by few pathways each. Clearly, the latter should be preferable. (ii) The components in the objective function are multiplied, although they are dependent on one another, as also mentioned in (Yeang *et al.*, 2004). (iii) The maximization procedure does not guarantee reaching a global optimum.

Here, we propose a novel combinatorial model for the inference problem. Our optimization goal is to maximize the expected number of explained cause-effect pairs. We reformulate the problem as an integer programming problem and apply a commercial solver to it. Moreover, we provide a measure of

*To whom correspondence should be addressed.

confidence in the predictions made, which quantifies the change in the function being optimized when the prediction is flipped.

We validate the method by applying it to a yeast subnetwork involved in mating. In cross-validation tests, SPINE obtains very high accuracy in predicting knockout effects (99%). Next, we apply SPINE to the entire yeast network to predict protein effects and reconstruct signaling and regulatory pathways. Overall, we are able to infer 861 paths with confidence and assign effects to 183 genes. The predicted effects are found to be in high agreement with current biological knowledge. Our results compare favorably to those attained by Yeang *et al.* (2004) in both the small-scale and the large-scale applications.

2 METHODS

2.1 Data description

The input data consist of a directed network of physical interactions $G = (V, E)$ and a collection X of cause-effect pairs of genes. The latter is commonly obtained using knockout experiments in which a gene is perturbed (*cause*) and the expression levels of all other genes are monitored. Those genes that significantly change their expression levels in response to the knockout are said to be *affected* by it. All other genes are assumed to be non-affected. In the context of knockout experiments, we refer to a cause-effect pair also as a *knockout pair*.

The nodes in the network represent genes and their protein products (no distinction is made between those two). The network has two types of edges: protein–DNA interactions and PPIs. Protein–DNA interactions are naturally represented as edges directed from a transcription factor to a regulated gene. Each PPI is represented using two oppositely directed edges. In addition to a direction, every interaction in the network is assigned a positive *reliability* value, representing the probability that the interaction is true; and a binary *sign*, representing its effect (0-activation or 1-repression). The former is computed in a pre-processing step (see below), while the latter is to be inferred by the model. In the following, we denote the sign and reliability of each interaction $e \in E$ by $sgn(e)$ and $r(e)$, respectively. We denote the set of protein–DNA interactions by $E_R \subseteq E$. Finally, we denote by $e(s, t)$ the complement of the observed effect on gene t when knocking out gene s. $e(s, t)$ is assumed to represent the effect of s on t in wild type.

2.2 Explanatory pathways

Yeang *et al.* (2004) defined certain basic rules for paths that *explain* knockout pairs. We use the same rules, and define a *k-explanatory* of a knockout pair $(s, t) \in X$ as an ordered set of vertices $\pi = (s = p_1, p_2 \ldots p_{k+1} = t)$ that satisfies the following conditions:

(1) π is a path: $\forall_{1 \le i \le k}(p_i, p_{i+1}) \in E$.

(2) π is a simple path: $\forall_{i \ne j}(p_i \ne p_j)$.

(3) The last edge in the path is a protein–DNA edge: $(p_k, p_{k+1}) \in E_R$.

(4) If an intermediate gene along the path has been knocked out, then it should also exhibit an effect on the target protein: $\forall_{1 \le i \le k}(\exists_{p \in V}(p_i, p) \in X) \rightarrow ((p_i, p_{k+1}) \in X)$.

For a path π, let $s(\pi)$ and $t(\pi)$ denote its source and target, respectively. Let N_π be an indicator variable denoting whether π explains the pair $(s(\pi), t(\pi))$. An explanatory path π is said to be *consistent* if in addition:

(1) The aggregate sign of π is equal to $e(s(\pi), t(\pi))$. Formally: $\oplus_{e \in \pi}(sgn(e)) = e(s(\pi), t(\pi))$, where $\oplus$ is the xor operator (addition modulo 2).

(2) Every proper suffix γ of π that connects another knockout pair is consistent with respect to it.

Note that we focus here on simple paths, as a protein is likely to play a single role in a given pathway. Further note that the consistency requirement, imposed here on suffixes, is adequate only for sub-paths that end with a protein–DNA edge, as the effect is observed at the transcription level.

These definitions can be easily generalized to accommodate for already known edge signs. For ease of presentation, we concentrate in the following on the case where all edge signs are unknown. In particular, if we denote by M_π an indicator variable specifying whether an explanatory path π is consistent, then

$$M_\pi \equiv (\oplus_{e \in \pi} sgn(e) = e(s(\pi), t(\pi)) \wedge (\forall_{\gamma \in \text{suffix}(\pi)}(N_\gamma \rightarrow M_\gamma)) \quad (1)$$

Yeang *et al.* (2004) considered in their work the inference of edge signs only. Another possibility is to assign signs to nodes rather than edges. In this case, all the edges emanating from a node carry its sign. This may be less accurate but would yield many fewer variables and hence avoid overfitting problems and computational bottlenecks. Indeed, most proteins in current databases are classified as either activators or repressors solely, so this relaxation seems to be in line with current biological knowledge. In the rest of the article, we focus on this latter variant, and use the edge variant mainly for comparison purposes.

2.3 Optimization criterion

Intuitively, the goal of the algorithm is to infer regulatory pathways in the network that provide a consistent explanation for the input set of knockout pairs. In practice, the pathways are dependent and there could be more than one pathway explaining a given pair. Hence, rather than trying to pinpoint a single pathway per pair, we aim at assigning signs to the interactions in the network so that the data can be best explained. There are several definitions to what a 'best' explanation is. A parsimonious definition would seek a sign assignment so that a maximum number of pairs have at least one consistent path. Note that an implicit assumption here is that the effect propagates through a single pathway, and the existence of other, possibly inconsistent pathways does not contradict the existence of the effect. However, the parsimonious criterion ignores the information on edge reliabilities and could, e.g. prefer solutions in which many pairs are explained by very low probability paths. Thus, we define our optimization problem as that of finding an assignment that will maximize the expected number of pairs that have at least one consistent path. We give a formal definition of this criterion below.

Define the probability of a path as the product of the reliabilities of its edges:

$$p(\pi) = \prod_{e \in \pi} r(e) \quad (2)$$

We say that a knockout pair (s, t) is *explained* or *satisfied* if there exists at least one explanatory path that is consistent with it. We associate an indicator variable $K_{s,t}$ with this event.

2.3.1 Expectation calculation We wish to find a setting of the node or edge signs that maximizes the expected number of satisfied knockout pairs, which is given by:

$$E\left(\sum_{(s,t) \in X} K_{s,t}\right) = \sum_{(s,t) \in X} E(K_{s,t}) = \sum_{(s,t) \in X} p(K_{s,t} = 1) \quad (3)$$

Given a collection Π of explanatory paths for a knockout pair (s, t), we denote by M_Π an indicator variable specifying whether all paths in Π are consistent with this pair. Clearly, $M_\Pi \equiv \wedge_{\pi \in \Pi} M_\pi$.

The probability that all the paths exist, $p(\Pi)$, is equal to the probability that all the edges in Π exist. Denoting the latter set of edges by E_Π we have:

$$p(\Pi) = \prod_{e \in E_\Pi} r(e) \tag{4}$$

For a knockout pair $(s,t) \in X$, denote by Π the collection of explanatory paths for it, with $|\Pi| = n$. To compute the probability that at least one of the paths exist, we must take into account the dependencies among them due to edge intersections. Let P^i_Π denote the collection of all sets $S \subseteq \Pi$ of cardinality i. Using the inclusion–exclusion principle, the probability that at least one of the paths in Π is consistent is:

$$p(K_{s,t} = 1) = \sum_{1 \le i \le n} (-1)^{i-1} \sum_{\Gamma \in P^i_\Pi} p(\Gamma) M_\Gamma \tag{5}$$

As mentioned earlier, in the objective function proposed by Yeang *et al.* the components are multiplied, although they are dependent on one another. Here, path dependencies are explicitly treated by the model Equation (5). Notably, there are two additional differences, not mentioned above, between our model and that of Yeang *et al.*: (i) no requirement is placed in Yeang *et al.* (2004) on the consistency of suffixes of an explanatory path. (ii) SPINE models PPI edges as potentially bi-directional, while in Yeang *et al.* (2004) they are forced to have a single direction. As we shall see below, in practice, most of the edges attain a single direction also under our model.

2.4 Algorithmic approach

We tackle the optimization problem using an integer program reformulation of it, to which we apply the commercial *CPLEX*™ solver. Below, we give the integer programming formulation and the full SPINE algorithm. A diagram that describes the SPINE's input, main phases and output, appears in Figure 1.

2.4.1 Integer programming formulation We reformulate the network annotation problem as an integer linear program. The objective function is simply $\sum_{(s,t) \in X} p(K_{s,t} = 1)$. The constraints are of two types:

(1) Path constraints: determining the consistency of explanatory paths.

(2) Knockout pair constraints: determining the expectation value of knockout pairs.

To obtain a linear program, we convert the Boolean equations defining the variables of the model, as well as the objective function, into linear forms.

Denote by q_π the longest suffix of π which is an explanatory path. Thus, we express M_π linearly using two constraints as follows:

(1) Consistent path sign:

$$M_\pi \le \sum_{e \in \pi} sgn(e) - 2d_\pi + 1 - e(s(\pi), t(\pi)) \le 1 \tag{6}$$

where $0 \le d_\pi \le \lceil \frac{|\pi|}{2} \rceil$ is a dummy variable, used to express the parity of the repressor signs.

(2) Consistent suffix:

$$M_\pi - M_{q_\pi} \le 0 \tag{7}$$

Note that this constraint is used only in case there is indeed a suffix which is an explanatory path.

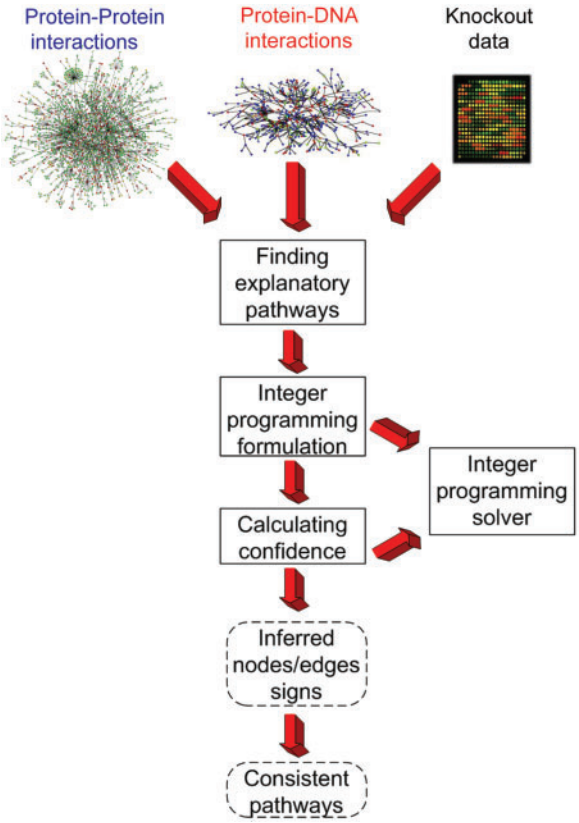

Fig. 1. The SPINE computational pipeline. Top: the input, consisting of a PPI network, a protein–DNA network and gene-expression data from knockout experiments. Middle: the main analysis stages. Bottom: the output, consisting of inferred signs for proteins/interactions and consistent explanatory pathways induced by them.

As for Equation (5), the challenge is to express M_Γ, an indicator for a set of paths, in a linear form. This can be done using the following constraint:

$$0 \le \sum_{\gamma \in \Gamma} 2M_\gamma - 2|\Gamma| M_\Gamma \le 2|\Gamma| - 1 \tag{8}$$

2.4.2 Confidence assignment SPINE tries to optimize the expected number of explained knockout pairs. For a given instance of the problem, let $O_{\max}$ denote the optimum value of the objective function. Clearly, there may be more than one configuration of protein signs that attains this value. Hence, for each protein v, we provide a *confidence value* C_v measuring how confident we are in its sign assignment. This value is calculated by running SPINE on a modified instance, in which the sign of the protein is flipped, and recording the difference between the new value of the objective function and $O_{\max}$. We declare v to be *confident* if $C_v > \epsilon$, for a pre-specified $\epsilon \ge 0$. We consider a knockout pair to be *confidently explained* if there is a consistent explanatory path for it whose proteins are all confident.

Yeang *et al.* (2004) did not define a confidence measure. However, they distinguished between variable assignments that uniquely maximize the objective function, and those that do not. In their model, once all the unique assignments (e.g. of signs) have been determined, one variable is fixed arbitrarily, and the inference procedure is iteratively applied until all the variables are determined. Hence, in contrast to SPINE, some of the inferred assignments may be arbitrary.

$SPIN(G, X, k)$
For i in $1 \ldots k$ **loop**
  $(solution, O_{max}) = Optimize(G, X, i)$
  C = MeasureConfidence$(solution, O_{max})$
  Foreach variable $v \in solution$ **loop**
    If $C_v > \epsilon$ **then**
      Update the sign of v in G
    End if
  End loop
End loop

$Optimize(G, X, k)$
Find explanatory paths of length $\leq k$
Foreach unexplained knockout pair $x \in X$ **loop**
  Foreach explanatory path π for x **loop**
    Construct linear equations for π
  End loop
  Construct linear equations for the constraints of x
  Construct linear equations that calculate $p(K_x = 1)$
End loop
Maximize the expected number of explained knockout pairs subject to the constraints
Return the optimal solution and its value

MeasureConfidence$(Solution, O_{max})$
Foreach variable $v \in Solution$ **loop**
  O_v = maximum objective given $sgn(v) = (1 - sgn(v))$
  $C_v = O_{max}$ - O_v
End loop

Fig. 2. The SPINE algorithm. Top: the main procedure; middle: preprocessing and optimization; bottom: confidence computation.

2.5 A speedup heuristic

To deal with large integer programs, we infer the model variables using multi-level runs. At each level $1 \leq l \leq k$, we start from the confident variables inferred on previous levels, and search for a setting of the remaining variables of the model that maximizes the expected number of explained knockout pairs using explanatory paths of length $\leq l$. This heuristic is based on the assumption that short, confident pathways are more likely than long ones.

We use $\epsilon = 0$ and consider only confident predictions, as non-confident predictions can be arbitrarily flipped without changing the overall value of the objective function. The full SPINE algorithm is summarized in Figure 2. Note that this algorithm solves both the node- and the edge-variant of the problem.

2.6 A toy example

Figure 3a gives an example of a simple input network. The nodes $\{A, B, C, D, E, F, G\}$ represent proteins, the solid edges represent protein–DNA interactions (for convenience, no PPIs are used) and the dashed edges represent knockout pairs along with the required knockout effect. The reliability of each edge appears to its side.

The explanatory paths in this example (with their probabilities in parentheses) are:

- $\pi_1 = A \rightarrow B \rightarrow C \rightarrow D$ (0.9).
- $\pi_2 = B \rightarrow C \rightarrow D$ (1).
- $\pi_3 = A \rightarrow G \rightarrow D$ (0.45).
- $\pi_4 = A \rightarrow E \rightarrow F$ (0.8).

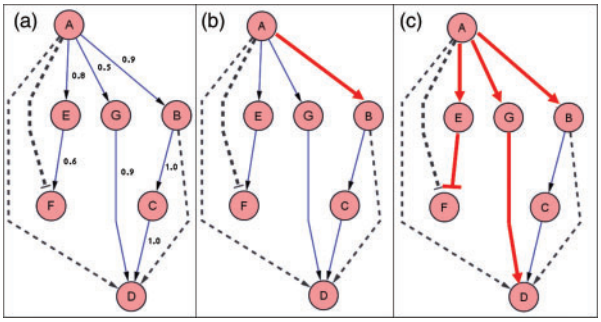

Fig. 3. A toy example. Nodes represent proteins and edges represent protein–DNA interactions. Knockout pairs are denoted by dashed lines, with the desired effect denoted by the arrow type: regular (positive) or cut (negative). Edges whose sign has been inferred by the algorithm are bold and colored red and those that were not inferred are non-bold and colored blue. (**a**) The input network, including edge reliabilities. (**b**) The inferred confident network in the edge variant. (**c**) The inferred confident network in the node variant.

The knockout pair indicators are:

$$K_{A,D} \equiv (M_{\pi_1} \vee M_{\pi_3})$$
$$K_{B,D} \equiv M_{\pi_2}$$
$$K_{A,F} \equiv M_{\pi_4}$$

Using Equation (3), we get the total expectation:

$$E(\sum_{(s,t)\in X} K_{s,t}) = \sum_{(s,t)\in X} p(K_{s,t} = 1) =$$
$$= p(K_{A,D} = 1) + p(K_{B,D} = 1) + p(K_{A,F} = 1) =$$
$$= p(\pi_1)M_{\pi_1} + p(\pi_3)M_{\pi_3} - p(\{\pi_1, \pi_3\})M_{\{\pi_1,\pi_3\}}$$
$$+ p(\pi_2)M_{\pi_2} + p(\pi_4)M_{\pi_4}$$

When looking for an assignment of signs to edges, there are many possible configurations that satisfy all those paths. In all those configurations $A \rightarrow B$ must have a positive sign, since both knockout pairs (A, D) and (B, D) have the same effect. All the other edges remain unsigned, since there is more than one possibility to set their signs. Hence, there is a single confident edge in this variant, and no consistent path is confidently inferred (Fig. 3b).

In contrast, in the node-based variant, many more variables are determined (Fig. 3c). Indeed, A will be signed positive for the same argument as above. Thus, E has to be signed negative, and G has to be signed positive. The only unsigned nodes in this example are B and C: signing both of them as either positive or negative will make π_2 a consistent path.

3 RESULTS

We tested SPINE both in a small-scale and a large-scale setting on a yeast (*Saccharomyces cerevisiae*) data set. First, we applied SPINE to part of a yeast subnetwork involved in mating in order to test the different variants of the algorithm and for purposes of comparison to a previous analysis made in Yeang *et al.* (2004). Next, we applied SPINE to genome-wide yeast data and evaluated its prediction performance.

3.1 Data acquisition

We constructed an integrated yeast network of 15 147 protein–protein and 5568 protein–DNA interactions, spanning a total of 5313 genes/proteins.

Table 1. Cross-validation results for the yeast mating subnetwork

Method	Left out	Number of Trials	Correct	Incorrect	Undecided	Accuracy
Yeang *et al.*	1	103	97.1%	2.9%	0%	97%
SPINE—edge variant	1	103	98%	1%	1%	99%
Yeang *et al.*	5	200	96.5%	3.5%	0%	96.5%
SPINE—edge variant	5	200	96.9%	0.4%	2.7%	99%

A hidden knockout pair is considered to be successfully predicted if all explanatory paths for it are consistent with its true effect. The accuracy represents the percentage of correct predictions among predictions made. In the leave-five-out cross-validation, the rates were computed as in Yeang *et al.* (2004) by dividing the number of correct, incorrect or undecided held-out pairs by the total number of held-out pairs in all trials.

PPIs were taken from DIP (Salwinski *et al.*, 2004, April 2005 download) and were assigned confidence scores using a previously described logistic regression method (Shlomi *et al.*, 2006). Each PPI was represented by two oppositely directed edges. Protein–DNA (directed) interactions were taken from (Macisaac *et al.*, 2006) and were assigned a confidence of 0.96 based on the false positive rate estimations in (Lee *et al.*, 2002).

Gene expression data were taken from (Hughes *et al.*, 2000) and included genome-wide gene expression measurements in yeast under 210 different single-gene knockouts. A previous analysis by Yeang *et al.* (2004) yielded a collection X of 24 457 pairs of a knocked out gene and an affected gene, all with P-value ≤ 0.02.

3.2 Yeast mating subnetwork

Yeang *et al.* (2004) have defined a yeast subnetwork involved in mating, which consists of 33 protein–DNA interactions, and 25 PPIs, covering a set of 149 knockout pairs. Applying our algorithm to this network with path length bound of 5 (as in Yeang *et al.*, 2004) resulted in inferring consistent explanatory paths for all 103 reachable knockout pairs.

To evaluate the performance of our algorithm, we tested it in a cross-validation setting on this network, and compared our results to those attained by Yeang *et al.* (2004). For purposes of comparison, we first used the edge variant of our algorithm with the same evaluation criterion used in Yeang *et al.* (2004). In detail, at each iteration one or five knockout pairs were hidden, and each was considered to be successfully predicted if all explanatory paths for it were consistent with its true effect. The results are shown in Table 1.

Evidently, the results are very similar to those of Yeang *et al.* (2004) with SPINE providing a slightly higher accuracy, both in the leave-one-out and leave-5-out tests. Notably, some of the knockout pairs remain undecided in the SPINE application since no explanatory path for them is predicted confidently. For SPINE, the only incorrect prediction was for knockout pair (FUS3, PRY2), which is the only knockout pair involving FUS3. The only explanatory path from FUS3 to PRY2 is FUS3 $\rightarrow$ STE12 $\rightarrow$ PRY2; both interactions were signed as positive, yielding a positive path, in contrast to the needed knockout effect. Interestingly, the interaction FUS3 $\rightarrow$ STE12 is known to be positive, as FUS3 is known to activate STE12.

Next, we wished to test the effect of maximizing the expected number of satisfied knockout pairs, rather than simply their number. In these tests, we used a more relaxed criterion for computing prediction accuracy: a knockout pair was considered to be predicted successfully if the expected number of explanatory paths that are consistent with its true effect exceeded the expected number of the non-consistent ones (if the numbers are equal, we consider the pair as undecided). Using this criterion seems more natural under our model and indeed improved the accuracy for the node variant, although the results for the edge variant remained the same (data not shown). We tested the maximization criteria in both the edge- and node-variant. As shown in Table 2, maximizing the mere number of satisfied pairs resulted in markedly smaller numbers of confident sign variables, in both variants, although both criteria attained similar accuracy levels.

By comparing the cross-validation results for SPINE's edge and node variants, it is evident that the latter attains similar accuracy levels although it has significantly less variables to be optimized and, hence, is less prone to overfitting.

Finally, we conducted a noisy cross-validation test. In each trial, the sign of one knockout pair was flipped, and the sign of a second knockout pair was hidden and inferred. The results show that the model is robust to these perturbations and the accuracy is maintained at very high levels (Table 2).

3.3 Genome-wide application

Next, we applied our method to the genome-wide data. Due to the size of the network, and following Yeang *et al.* (2004), we restricted the computation to paths of length at most 3. Out of 231 proteins participating in such paths, 183 protein signs were confidently inferred. Among 974 knockout pairs that had explanatory paths, 861 were confidently explained using the inferred signs. The contribution of each level in SPINE's execution is shown in Table 3. The amount of explained knockout pairs and inferred proteins increases in each level, and this demonstrates the importance of looking at paths rather than considering direct edges only. In comparison, when using only protein–DNA interactions the explanatory power of the model is markedly decreased: only 188 knockout pairs are confidently explained and only 30 protein signs are inferred.

In order to validate our predictions, we gathered information on activator and repressor proteins from four different sources: GO (Ashburner *et al.*, 2000, December 2006 download)—molecular functions 'transcriptional activator activity' and 'transcriptional repressor activity'; KEGG (Kanehisa and Goto, 2000), Guelzim *et al.* (2002) and Myers and Kornberg (2000). We excluded 19 genes that appeared as both activators and repressors in the different sources. The results are summarized in Table 4, showing a significant overlap between the model's prediction and the known signs. Examining the predictions made in each level of the algorithm we observe that most predictions, and also most errors, are made in level 3: 24/27 activators and 8/15 repressors are correctly predicted.

Table 2. Performance in cross-validation on the yeast mating subnetwork

Method variant	Maximization criterion	Left out	Correct	Incorrect	Undecided	Accuracy
Edge variant	Expectation	1	98%	1%	1%	99%
Edge variant	Number	1	47.6%	0%	52.4%	100%
Node variant	Expectation	1	89.3%	10.7%	0%	89.3%
Node variant	Number	1	60.2%	9.7%	30.1%	86.1%
Edge variant	Expectation	1 (noisy)	93.5%	3.9%	2.6%	96%
Node variant	Expectation	1 (noisy)	88.3%	11.7%	0%	88.3%

A hidden knockout pair is considered to be successfully predicted if the expected number of explanatory paths with consistent signs is higher than the expected number of explanatory pathways with non-consistent signs. The accuracy represents the percentage of correct predictions among all predictions made.

Table 3. Cumulative contributions of different path lengths (levels) to the inference process

Level	Number of Explained knockouts	Number of Inferred signs
1	107	14
2	183	30
3	861	183

Table 4. Results of the comparison to known regulators, for both Yeang *et al.* and SPINE

Type	Number of Known	Number of Predicted	Number of Correct	Significance
Yeang *et al.* - Activators	119	31	28	0.002
Yeang *et al.* - Repressors	57	22	8	0.4
SPINE - Activators	120	37	32	0.003
SPINE - Repressors	60	22	12	0.02

Shown are the number of known activators and repressors that appear in the network, the number of sign predictions on this known set, the number of correct predictions and a hypergeometric *P*-value. The latter was computed separately for the activator and repressor predictions.

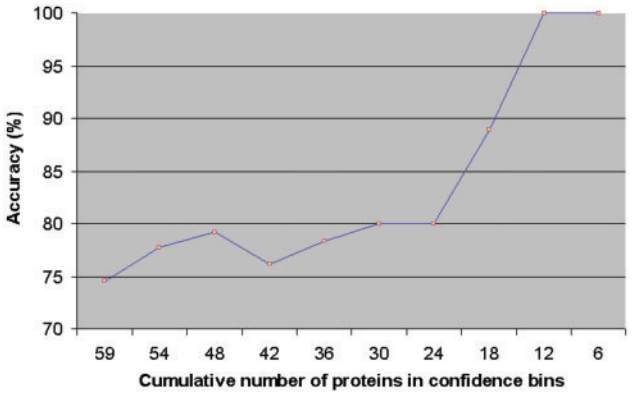

Fig. 4. The prediction accuracy as a function of the confidence threshold. The cumulative number of proteins in each bin appears on the *x*-axis, and the accuracy percentage appears on the *y*-axis. The confidence bins cover the threshold range between 0.01 and 34.56.

The confidence measure can vary from one inferred sign to another. Thus, we checked the accuracy of our predictions when using varying thresholds of the confidence measure for determining the set of predicted signs. The inferred signs were grouped into bins, such as each bin contains the inferred signs above a certain confidence threshold. The results clearly show that increasing the confidence threshold also increases the accuracy of the predictions (Fig. 4).

Next, we wished to compare our prediction performance to that of Yeang *et al.* (2005). As we could not readily apply the latter method to our network, nor to the intersection of our network and that in Yeang *et al.* (2005), we used the results of applying their method to the genome-wide network that appears in Yeang *et al.* (2005). We note that the two networks differ in ~20% of their constituent interactions, although their sizes are about the same; thus, the comparison presented below should be interpreted with caution as the difference in networks may bias the results.

Recall that the method of Yeang *et al.* infers edge signs. To infer node signs, we applied a majority rule to the set of edge signs incident to each node (nodes for which no majority existed remained undecided). The statistics of these predictions are shown in Table 4. It can be seen that the overall performance of the algorithms is similar with respect to activator predictions, but SPINE attains higher accuracy in repressor predictions while the results of Yeang *et al.* are not significant. To further evaluate the results of both algorithms, we computed their precision and recall. *Precision* is the percent of correct predictions out of all predictions (for both activators and repressors); *recall* is the percent of correct predictions out of all known signs. The precision and recall for SPINE were 75% and 24%, respectively. The method of Yeang *et al.* yielded lower rates in both measures with 68% precision and 20% recall.

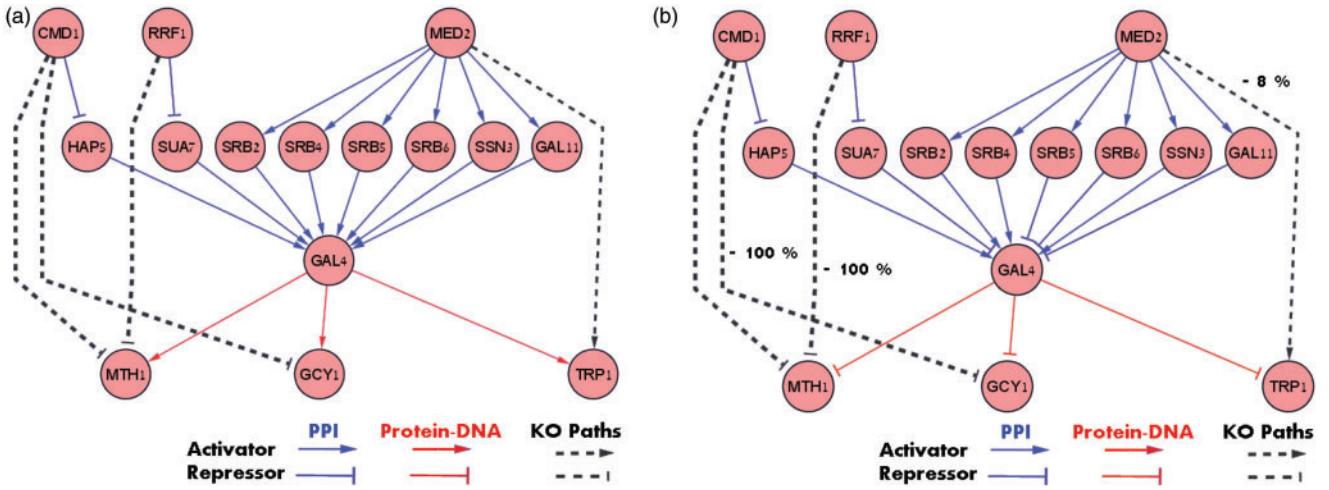

Fig. 5. A GAL4-centered subnetwork. (**a**) GAL4 is correctly identified as an activator and all the depicted explanatory pathways are consistent. (**b**) A sign assignment in which GAL4 is forced to be a repressor, which is as close as possible to that chosen in (a). It yields several non-consistent pathways, implying a decrease in expectation. The decrease in the expectation percentage for each changed knockout pair appears on its edge.

In contrast to Yeang *et al*. SPINE does not constrain the directions of the PPIs. Interestingly, most (312 out of 340) PPIs appeared in one direction only in the inferred consistent pathways.

Figure 5 shows an example subnetwork inferred by the model, including both molecular and functional data. The explanatory pathways that appear in the subnetwork go through GAL4. GAL4 is a known activator. It is a DNA-binding transcription factor required for the activation of the GAL genes in response to galactose. Other components in this subnetwork are MED2, GAL11, SRB4, SRB5 and SRB6, which are mediators that function as a bridge between the regulatory proteins and the basal polymerase II transcription machinery (Myers and Kornberg, 2000).

SPINE found a confident optimal solution, S_{optimal}, in which GAL4 is a confident activator, i.e. signing GAL4 as repressor reduces the expected number of explained knockout pairs. We checked the effect of signing GAL4 as a repressor, by running SPINE again, while fixing GAL4 as a repressor. Once reached an optimum, we searched for a solution with the same optimal value, but with minimum changes compared to solution S_{optimal}. As shown in Figure 5a, all the explanatory pathways that go through GAL4 are consistent when it is an activator, whereas when GAL4 is a repressor some of those pathways become non-consistent. Some genes, such as, e.g. HAP5, remain with the same sign even when GAL4 is a repressor, due to their importance in the different pathways, including pathways that does not appear in the network. However, some genes do change their sign, such as, e.g. SUA7. Figure 5b shows the decrease in the expectation of knockout pairs connected in the network.

4 CONCLUSIONS

This article addresses the problem of inferring signaling-regulatory pathways explaining cause-effect experiments. We have translated the reconstruction problem into that of assigning an activation/repression attribute with each protein so as to explain (in expectation) a maximum number of the knockout effects observed. We provided an integer programming formulation for the latter problem and solved it using a commercial solver. The resulting framework, SPINE, was tested on both small- and large-scale networks, and provided highly accurate results, comparing favorably to a previous approach by Yeang *et al*. (2004).

There are several important contributions of the current work: (i) a basic optimization criterion and a biologically motivated scheme to solve it in practice via multi-level runs; (ii) a confidence measure for the signs inferred by the model and (iii) a general approach for assigning both edge and node signs, providing a balance between the flexibility of the assignment and its confidence.

While our algorithm provides a valuable framework for analyzing gene expression data in the context of a physical network, several of its limitations should be acknowledged. First, the algorithm is computationally intensive due to the resulting large integer programs. A practical approach to analyze larger networks could be to limit the accuracy of the solver. Second, it could be beneficial to integrate into the framework information on protein complexes (Gavin *et al*., 2006; Hollunder *et al*., 2005), which could be perhaps viewed as unified single nodes in the context of the current analysis. Finally, our algorithm does not consider the strength of the knockout effect (and the associated *P*-value). A refined optimization criterion that combines this information could improve the accuracy of the predictions.

We have tested our method on cause-effect data gathered from gene knockout experiments. Although gene knockouts are routine in model organisms such as yeast, they are more technically involved in mammalian species such as mouse. For these higher eukaryotes, other ways of genetically perturbing the cell are gaining in practice, such as RNA interference and

eQTL analysis. In addition, although DNA microarrays are currently the most widespread technology for measuring global changes in cellular state, a variety of other measurement systems, such as mass spectrometry for monitoring protein abundance, protein post-translational modification or small molecule abundance, are gaining in popularity. The general formulation of our model allows its application to those types of cause-effect data as well.

ACKNOWLEDGEMENTS

R.S. was supported by an Alon Fellowship. T.S. was supported by an Eshkol Fellowship from the Israeli Ministry of Science. This research was supported by the Israel Science Foundation (grant no. 385/06).

Conflict of Interest: none declared.

REFERENCES

Andrec,M. *et al*. (2005) Inference of signaling and gene regulatory networks by steady-state perturbation experiments: structure and accuracy. *J. Theor. Biol.*, **232**, 427–441.

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

Fields,S. and Song,O. (1989) A novel genetic system to detect protein-protein interactions. *Nature*, **340**, 245–246.

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

Gavin,A.C. *et al*. (2006) Proteome survey reveals modularity of the yeast cell machinery. *Nature*, **440**, 631–636.

Guelzim,N. *et al*. (2002) Topological and causal structure of the yeast transcriptional regulatory network. *Nat. Genet.*, **31**, 60–63.

Hollunder,J. *et al*. (2005) Identification and characterization of protein subcomplexes in yeast. *Proteomics*, **8**, 2082–2089.

Hughes,T.R. *et al*. (2000) Functional discovery via a compendium of expression profiles. *Cell*, **102**, 109–126.

Kanehisa,M. and Goto,S. (2000) Kegg: kyoto encyclopedia of genes and genomes. *Nucleic Acids Res.*, **28**, 27–30.

Kholodenko,B.N. *et al*. (2002) Untangling the wires: a strategy to trace functional interactions in signaling and gene networks. *PNAS*, **99**, 12841–12846.

Lee,T.I. *et al*. (2002) Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, **298**, 799–804.

Macisaac,K.D. *et al*. (2006) An improved map of conserved regulatory sites for saccharomyces cerevisiae. *BMC Bioinformatics*, **7**, 113.

Myers,L.C. and Kornberg,R.D. (2000) Mediator of transcriptional regulation. *Ann. Rev. Biochem.*, **69**, 729–749.

Salwinski,L. *et al*. (2004) The database of interacting proteins: 2004 update. *Nucleic Acids Res.*, **32**, D449.

Shlomi,T. *et al*. (2006) Qpath: a method for querying pathways in a protein-protein interaction network. *BMC Bioinformatics*, **7**.

Workman,C.T. *et al*. (2006) A systems approach to mapping DNA damage response pathways. *Science*, **312**, 1054–1059.

Yalamanchili,N. *et al*. (2006) Quantifying gene network connectivity in silico: scalability and accuracy of a modular approach. *Syst. Biol. (Stevenage)*, **153**, 236–246.

Yeang,C.H. and Vingron,M. (2006) A joint model of regulatory and metabolic networks. *BMC Bioinformatics*, **7**, 332.

Yeang,C.H. *et al*. (2004) Physical network models. *J. Comput. Biol.*, **11**, 243–262.

Yeang,C.H. *et al*. (2005) Validation and refinement of gene-regulatory pathways on a network of physical interactions. *Genome Biol.*, **6**.