Molecular BioSystems

RSC Publishing

PAPER

View Article Online
View Journal

Cite this: DOI: 10.1039/c3mb25432a

iPoint: an integer programming based algorithm for inferring protein subnetworks

Nir Atias and Roded Sharan*

Received 11th October 2012,
Accepted 9th January 2013

DOI: 10.1039/c3mb25432a

www.rsc.org/molecularbiosystems

Large scale screening experiments have become the workhorse of molecular biology, producing data at an ever increasing scale. The interpretation of such data, particularly in the context of a protein interaction network, has the potential to shed light on the molecular pathways underlying the phenotype or the process in question. A host of approaches have been developed in recent years to tackle this reconstruction challenge. These approaches aim to infer a compact subnetwork that connects the genes revealed by the screen while optimizing local (individual path lengths) or global (likelihood) aspects of the subnetwork. Yosef *et al.* [*Mol. Syst. Biol.*, 2009, **5**, 248] were the first to provide a joint optimization of both criteria, albeit approximate in nature. Here we devise an integer linear programming formulation for the joint optimization problem, allowing us to solve it to optimality in minutes on current networks. We apply our algorithm, iPoint, to various data sets in yeast and human and evaluate its performance against state-of-the-art algorithms. We show that iPoint attains very compact and accurate solutions that outperform previous network inference algorithms with respect to their local and global attributes, their consistency across multiple experiments targeting the same pathway, and their agreement with current biological knowledge.

1 Introduction

The molecular mechanisms that guide cells in their response to a changing environment are far from understood. High-throughput techniques are constantly being developed with the goal of charting the pertaining molecular landscape in greater detail. However, the sheer mass of data being produced is hard to decipher. Challenges involve dealing with noisy and incomplete measurements, observing only snapshots at specific time points and conditions and, most notably, integrating different types of data whose inter-relations are not completely understood.

A fundamental problem in this domain is the inference of a subnetwork underlying a process of interest. Commonly, a process is characterized by a set of relevant genes that are either collected from the literature or obtained via a high-throughput expression or phenotypic screen. These *terminal* genes can be projected onto a protein network in an effort to gain pathway level understanding of the process.[1–3] Additionally, in many cases the process or response in question can be further characterized by a small set of *anchor* genes that mediate it. It is then appealing to assume that the sought subnetwork should ''economically'' connect the anchor to the terminals, as discussed in detail in Yosef *et al.*[4] For example, in a knockout screen, a gene is knocked out and as a result other genes are observed to change their expression levels. Here the knocked out gene serves as a natural anchor and the differentially expressed genes serve as the terminals for the reconstruction problem. In the following we focus on such ''anchored'' reconstruction problems.

Surprisingly, there are relatively few previous methods for anchored reconstruction. Yosef *et al.*[4] modeled the reconstruction task as a maximization problem where the likelihood of individual paths leading from the anchor node to the terminal nodes (local criterion), and that of the entire network (global criterion) were jointly maximized. They showed that the resulting formulation is NP hard and gave an approximation algorithm to solve it. Importantly, they showed that under many scenarios, the algorithm outperformed local-based or global-based solutions, demonstrating the added value of the joint optimization.

Additional state-of-the-art algorithms emphasized the global criterion. Yeger-Lotem *et al.*[5] used a flow-based approach to integrate data on the genetic interactors of a given gene with information on genes that change their expression level following its knockout. Briefly, flow originating at a root node, connected to the genetic hits, is drained at a sink node which is

Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: roded@post.tau.ac.il; Tel: +972-3-6407139

Published on 10 January 2013 on http://pubs.rsc.org | doi:10.1039/C3MB25432A
Downloaded by Tel Aviv University on 06 February 2013

connected to the set of differentially expressed genes. The goal is to maximize the flow between the root and the sink (under some capacity constraints) and at the same time minimize its cost which is derived from the confidence values associated with the edges. A second tool of the Fraenkel lab, called SteinerNet,[3] is based on reformulating the reconstruction problem as a prize-collecting Steiner tree problem. This variant maximizes the likelihood of the entire network but may produce a subnetwork spanning only a subset of the input genes when the cost of connecting them to the network is too high.

Here we provide an integer linear programming based algorithm, iPoint (integer PrOgramming based INference of Trees), to solve the optimization problem defined by Yosef *et al.* to optimality. This allows us, for the first time, to accurately compare the optimization criteria of Yosef *et al.*, Yeger-Lotem *et al.* and Huang *et al.* and, further, to evaluate the utility of an exact solution over an approximate one. To this end we use diverse data sets from yeast and human, running all four algorithms across a range of parameters. We evaluate a solution by its objective value (local and global attributes), by its agreement with biological knowledge and by its consistency across different experiments targeting the same pathway.

The results demonstrate across many different data sets that an optimal solution for the joint local–global optimization problem yields solutions that substantially outperform their approximate counterparts. They also show that iPoint consistently leads to biologically significant results across different data sets and species, outperforming previous approaches to the reconstruction problem.

2 Preliminaries

We assume we are given an initial connected network $G = (V,E)$ of physical protein–protein and possibly protein–DNA interactions covering the process in question. For convenience, we assume the network to be directed, where undirected edges are designated by two oppositely-directed edges. Every edge in the network has some confidence level associated with it. We define the *length* of an edge $(u,v) \in E$ as $-$log its confidence and denoted it by c_{uv}.

The input to a network inference problem includes: (i) the source of the response, conveniently represented by a single *anchor* node r, from which the signaling-regulatory pathways related to the response are assumed to emanate; and (ii) the targets of the response – a subset $T \subset V\backslash\{r\}$ of *terminal* nodes. The goal is to find the most probable subnetwork (under a scoring scheme to be defined), $H \subseteq G$, connecting the anchor node with the terminal nodes. To deal with multiple anchors, the network is augmented with an auxiliary node, connected to these anchors through edges with arbitrarily-small length, and the computation proceeds with the auxiliary node serving as a single anchor.

There are two natural criteria to score a subnetwork. The first, *local* criterion scores H according to the length (likelihood) of its constituent anchor-terminal paths.[2] Precisely,

$$L_l(H) = \sum_{t \in T} \sum_{(u,v) \in P_t} c_{uv} \tag{1}$$

where P_t is a path from r to terminal t in H. Clearly, to minimize L_l each P_t can be minimized independently by a shortest path computation. The second, *global* criterion is the overall likelihood of:[1,3,6]

$$L_g(H) = \sum_{(u,v) \in H} c_{uv} \tag{2}$$

As mentioned above, Yosef *et al.* gave an approximation algorithm that jointly minimizes these measures. Formally, they formed a combined objective

$$L(H) = cL_g(H) + L_l(H) \tag{3}$$

where the balancing factor c is set to $\mathrm{OPT}_l/\mathrm{OPT}_g \geq 1$, where OPT_l, OPT_g are the optimal values for L_l and L_g, respectively.

3 An exact algorithm

In this section we formulate the joint optimization problem defined in eqn (3) as an integer linear program (ILP) that may be optimally solved. The main challenges in formulating an efficient ILP are: (i) ensuring the connectivity of the chosen subnetwork. We overcome this challenge using a characterization of optimal solutions and a careful branch and cut search. (ii) Formulating the local measure. We address this problem using flow computations.

3.1 Expressing the connectivity constraints

We start by observing that the optimal solution $H^* = \arg\min_H \{L(H)\}$ must be a tree (more accurately, its underlying undirected graph is a tree). To see this, note that H^* cannot contain any edge that does not lie on a shortest path (in H^*) from r to some terminal, since its removal would improve the solution (L_g would decrease and L_l would remain constant). It follows that H^* does not contain directed cycles. By its optimality w.r.t. the global criterion, its underlying undirected graph cannot contain cycles. Since G is connected, H^* must be a tree.

Our ILP has the following obvious sets of variables: per-edge binary variables x_{uv}, indicating whether $(u,v) \in E$ should be included in the solution H^*; and per-node variables y_v, indiciating whether $v \in V$ should be included in H^*. Additionally, we use per-edge variables, σ_{uv}, indicating a flow going through (u,v). We use these flow variables to evaluate the local objective (eqn (1)) as explained below.

To ensure the resulting subnetwork is a tree we use two sets of constraints: the first ensures that every node $v \neq r$ in H^* has exactly one predecessor and the second ensures that H^* is connected or, equivalently, that any partition of the nodes of G, separating the anchor from some terminal, includes at least one edge from H^* crossing its subsets. Formally, given any subset of nodes $S \subset V$ containing some terminal t, the set of edges entering S is given by $\bar{\delta}(S) = \{(u,v)|u \in V\backslash S, v \in S\}$; the corresponding connectivity constraint is given by:

$$\sum_{(u,v) \in \bar{\delta}(S)} x_{uv} \geq 1 \tag{4}$$

We avoid the explicit specification of these exponentially many connectivity constraints by introducing them into the formulation only when they are violated. Specifically, while solving the linear program, the connected components of a candidate solution reveal such violations.

Finally, we use two additional sets of constraints to strengthen the formulation and improve the running time of the algorithm. First, we require the out-degree of an inner node v in H^* to be at least one. Second, we only allow one of two oppositely-directed edges to be included in H^* (to avoid cycles). The full set of constraints is given in Code Listing 1

$$\begin{aligned}
&\text{minimize} \quad c \cdot \textstyle\sum_{(u,v)\in E} c_{uv} \cdot x_{uv} + \sum_{(u,v)\in E} c_{uv} \cdot \sigma_{uv} \\
&\text{s.t.} && (5)
\end{aligned}$$

$$\sum_{(u,v)\in E} x_{uv} = y_v \qquad \forall v \in V \backslash \{r\} \qquad (6)$$

$$\sum_{(u,v)\in \bar{\delta}(S)} x_{uv} \geq 1 \qquad S \subset V \qquad (7)$$

$$y_u - \sum_{(u,v)\in E} x_{uv} \leq 0 \qquad \forall u \in V \backslash T \qquad (8)$$

$$x_{uv} + x_{vu} \leq y_u \qquad \forall (u,v),\ (v,u) \in E \qquad (9)$$

$$\sum_{(v,u)\in E} \sigma_{vu} - \sum_{(u,v)\in E} \sigma_{uv} = \begin{cases} |T| & \text{if } v = r \\ -1 & \text{if } v \in T \\ 0 & \text{o/w} \end{cases} \qquad \forall v \in V \qquad (10)$$

$$\sum_{(v,r)\in E} \sigma_{vr} = 0 \qquad (11)$$

$$\sigma_{uv} - x_{uv} \geq 0 \qquad \forall (u,v) \in E \qquad (12)$$

Code Listing 1 Complete ILP formulation employed by iPoint. To ensure the solution forms a tree, eqn (6) requires all solution nodes to have exactly one predecessor and eqn (7) ensures connectivity. Eqn (8) ensures that the out-degree of inner nodes is greater than one. Eqn (9) avoids trivial cycles by allowing at most one of two oppositely-directed edges in the reconstructed network. Eqn (10) and (11) maintain flow preservation for inner nodes, the generation of flow at the anchor and its drainage in the terminals. Eqn (12) ensures that flow is only admitted through edges that participate in the solution.

When the input set of terminals represents differentially expressed genes, we may wish to further constrain the solution subnetwork so that all anchor-terminal paths end with a protein–DNA interaction (when such edges are available), to reflect the nature of the response (see, *e.g.*, Yeang *et al.*[7]). Let E_{reg} be the (possibly empty) set of protein–DNA interactions. We then add the following constraints to the model:

$$\sum_{(u,t)\in E_{\text{reg}}} x_{ut} \geq y_t \quad \forall t \in T \qquad (13)$$

We call the resulting variant iPoint$_{\text{reg}}$.

3.2 Expressing the objective function

The objective function of our ILP has two terms: a global term (eqn (2)), trivially given by $\sum_{(u,v)\in E} c_{uv}x_{uv}$, and a local one.

To express the local term, we change the summation order and sum over all paths going through a given edge:

$$L_1(H) = \sum_{(u,v)\in H} t_{uv}c_{uv} \qquad (14)$$

where t_{uv} is the number of terminals connected to the anchor through (u,v) in H. We calculate t_{uv} using a flow formulation, where $|T|$ units of flow leave the anchor and each terminal drains only a single unit of flow. All other nodes preserve the amount of flow going through them. This formulation is expressed using the following linear constraints:

$$\sum_{(v,u)\in E} \sigma_{vu} - \sum_{(u,v)\in E} \sigma_{uv} = \begin{cases} |T| & \text{if } v = r \\ -1 & \text{if } v \in T \\ 0 & \text{o/w} \end{cases} \quad \forall v \in V \qquad (15)$$

$$\sigma_{uv} \geq x_{uv} \quad \forall (u,v) \in E \qquad (16)$$

$$\sum_{(v,r)\in E} \sigma_{vr} = 0 \qquad (17)$$

We argue that σ_{uv}, the flow going through (u,v), is equal to t_{uv}. To show this, we first prove by induction over the level of v in the tree H^* that $\sigma_{uv} \geq t_{uv}$. The base case is for edges leading to a leaf v (which must be a terminal). By eqn (15) and (16), $\sigma_{uv} \geq 1$ as claimed. Consider now the case where v is an internal node in H^*. Let $(v,w_1)\cdots(v,w_k)$ be the edges leaving v. By the induction hypothesis, for all i, $\sigma_{vw_i} \geq t_{vw_i}$. Furthermore, all the terminals in the subtree rooted at w_i admit a path from the root going through (v,w_i). There are now two cases to consider. If v is a terminal then by eqn (15) we have $\sigma_{uv} = \sum_i \sigma_{vw_i} + 1 \geq \sum_i t_{vw_i} + 1 = t_{uv}$. Otherwise, $\sigma_{uv} = \sum_i \sigma_{vw_i} \geq \sum_i t_{vw_i} = t_{uv}$. It remains to show that equality holds. By eqn (15) and (17) this must be the case at the root r. It is easy to see that equality must hold in the rest of the tree by a top-down induction at the node level in the tree. This completes the proof.

Interestingly, a similar flow formulation can be used to express the connectivity of the solution using only a polynomial number of constraints (see also *e.g.*, Bruckner *et al.*[8]). However, despite the theoretical succinctness of this formulation, adding violated constraints was much faster in practice.

4 Experimental setup and performance evaluation

We applied iPoint and iPoint$_{\text{reg}}$ in two scenarios: (i) reconstructing the response to a pheromone subnetwork in yeast; and (ii) reconstructing a subnetwork underlying Huntington's disease (HD) in human. The performance of our approach was compared with those of Yosef *et al.*, ResponseNet and SteinerNet. To execute the algorithm of Yosef *et al.* we used the ANAT tool.[9] To run ResponseNet[10] and SteinerNet[11] we used their respective websites.

4.1 Data sources

The analysis of the response to the pheromone pathway was based on a network downloaded from the ResponseNet website. These networks contained 54 545 protein–protein interactions (PPIs) and 14 023 protein–DNA interactions, spanning 6634 yeast proteins. Edge confidence was computed using a Bayesian approach taking into account the type of evidence available for the given interaction as described in Yeger-Lotem *et al.*[5] Genetic interactions were downloaded from the Saccharomyces Genome Database.[12] Differentially expressed genes, defined as those having at least two-fold change with $p < 0.05$, were extracted from gene signatures published by Hughes *et al.*[13]

For the HD pathway analysis, we used the human network available as part of ANAT. This network contained 44 737 PPIs and 4299 protein–DNA interactions spanning 10 380 proteins. Confidence values were calculated according to a logistic regression model,[14] taking into account the experimental techniques by which each of the interactions was discovered and the number of times it was reported. Differentially expressed genes, defined as those displaying more than two-fold change, were taken from the Caduate Nucleus data of Hodges *et al.*[15]

4.2 Execution parameters

To ensure a fair comparison we varied the execution parameters of the different approaches across a wide range, as recommended by their respective authors.

For the method of Yosef *et al.*, we used $\alpha \in \{0,0.25,0.5\}$, covering the local–global spectrum (with 0 preferring local solutions, 0.25 balancing the two criteria and 0.5 preferring global solutions). We additionally varied the *margin* parameter to integrate solutions that deviate by at most 2% from optimum.

For ResponseNet, we used *capping* = 0.9 which places an upper bound on the edges' confidence to reduce bias toward extensively studied interactions. We varied the Γ parameter which controls the surplus of flow in the flow-maximization problem in the range of 5–20, thereby affecting the size of the resulting network.

For SteinerNet, we used the ''rooted'' version and varied the β parameter, which controls the penalty on exclusion of terminals from the network and thereby the resulting solution size, in the range of 1–6. The nodes were connected *via* protein–DNA edges and their prizes were defined as absolute value of their log fold change divided by the largest log fold change in the data, as previously described. Some of the reconstructed networks did not include the anchor nodes. In those cases we used the ''unrooted'' version and specified the anchor as an additional terminal, associated with an arbitrarily (and connected *via* a physical interaction).

The ILP formulation of iPoint variants was optimally solved using CPLEX,[16] typically within minutes (see Section 4.3 for detailed running times). We additionally merged with the optimal solution up to 35 near-optimal solutions deviating from it by no more than 5%. These perform almost equally well under the objective and are therefore deemed as biologically relevant. Note that the merged structure need not be a tree. Following Yosef *et al.*, we added a penalizing factor, equal to the length of an edge at the 25th percentile, to the confidence of edges, thereby favoring shorter paths from the anchor to terminals.

4.3 Running time

To solve our formulation, two complex problems should be solved: (i) optimally solving the global criterion, also known as the Steiner tree problem, to obtain the balancing factor c; and (ii) optimally solving the joint optimization criterion. In addition, our implementation may optionally merge multiple, equally good solutions.

Surprisingly, solving the joint optimization criterion required less than two minutes in both yeast and human datasets. The comptuation of the balancing factor was more costly, taking six minutes on average on the yeast data set and half an hour on the human dataset. Interestingly, this cost may be avoided as we noted the balancing factor was highly correlated (Pearson 0.8, $p < 0.0003$) with the number of terminals and could possibly be estimated using a linear regression model. The approximation algorithm of Yosef *et al.*[4] is much faster requiring only a few seconds to complete.

Enumeration of multiple solutions was a time consuming task. For the yeast data sets, it took on average two hours to enumerate all 35 solutions and on the human data set 100 minutes were needed.

4.4 Performance evaluation

We evaluated a solution based on the enrichment of its genes with respect to functionally related gene sets. To this end, we used two scores: the first computing the hypergeometric enrichment of all inferred (non-input) genes of the reconstructed subnetwork; and the second capturing the topology of the subnetwork by computing the number of enriched anchor-to-terminal paths of the subnetwork ($p < 0.05$, FDR corrected). For each terminal, we arbitrarily chose a shortest path connecting it to the anchor and focused on paths with at least three nodes. For each algorithm we picked a solution (across the different parameters values) maximizing the sum of these two criteria by normalizing each score relative to the maximum achieved by the algorithm. For ease of comparison we report scores normalized across all algorithmic variants in a similar manner.

For the response to pheromone pathway data sets we computed the enrichment with genes annotated as related to pheromone response (GO:0019236). For these data sets we further assessed the consistency of the solutions. The *consistency score* of an algorithm (consisting of several experiments targeting the same pathway) was defined as the number of nodes (resp., edges) appearing in at least two subnetworks (reconstructed for different experiments) divided by the total number of nodes (resp., edges) across all subnetworks produced by the algorithm.

For the Huntington's disease data set we computed the enrichment of the solutions's genes with genes belonging to one of: KEGG HD pathway (hsa:05016), HD-associated genes in the Genetic Association Database (GAD),[17] HD-associated genes in the Comparative Toxicogenomics Database (CTD),[18] and with apoptosis-related genes according to KEGG (Apoptosis Pathway; hsa:04210) and the gene ontology (Apoptotic Process, GO:006915).

5 Results

We applied iPoint and iPoint$_{\text{reg}}$ to several data sets in yeast and human and compared them to state-of-the-art algorithms of Yosef *et al.*, Yeger-Lotem *et al.* and Huang and Fraenkel. We evaluated the performance of the algorithms according to three criteria: (i) the joint objective function; (ii) enrichment of solutions with relevant gene sets such as related GO terms; and (iii) the consistency of solutions across different experiments targeting the same pathway.

As an initial assessment of iPoint, we examined the contribution of iPoint's optimal solution (with respect to the joint objective) to the objective function and to the size of the resulting subnetwork. Applying our algorithm and that of Yosef *et al.* on the various data sets, we found that on average the approximation algorithm deviated by roughly 10% from optimum in both measures (see Fig. 1). However, for one of the data sets the objective deviated by more than 30% from optimum and in general the deviation could be much larger and is a function of the number of terminals.[4]

5.1 Pheromone response pathway

Our second set of experiments concerned the reconstruction of the well-studied pheromone response pathway in yeast. Following Yeger-Lotem *et al.*,[5] we used multiple data sets in which different genes in this pathway were perturbed (STE5/7/11/12/18) and data were collected on their genetic (synthetic lethal) interactors and differentially expressed genes. The different algorithms were used to connect the genetic interactors to the differentially expressed genes using an integrated network of protein–protein and protein–DNA interactions.

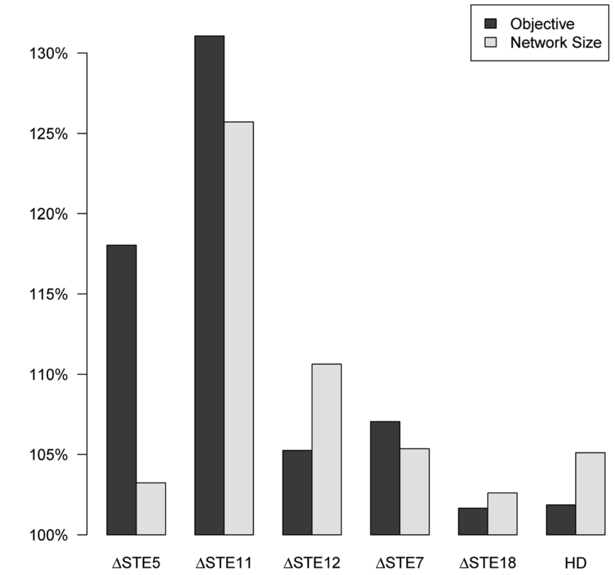

Fig. 1 Optimal *vs.* approximate solution. The dark bars denote the ratio between the objective value of the approximation algorithm to the optimum (attained by iPoint). The light bars denote the ratio between the sizes of the networks inferred by the approximation algorithm and iPoint. Data sets are ordered according to the number of terminals.

We scored the obtained solutions by their enrichment and that of their constituent pathways with the response to the pheromone biological process (GO:0019236), as suggested by Huang and Fraenkel.[3] The results are summarized in Fig. 2 and Table 1. In all but one data set (ΔSTE5) iPoint variants attained the highest combined score. Ranking the algorithms according to their average normalized scores across the different datasets (Table 1), consistently places iPoint variants as the best performing algorithms, followed by ResponseNet, Yosef *et al.* and SteinerNet. The ranking also shows that the iPoint$_{\text{reg}}$ variant is preferable in this setting.

An interesting feature of this data set is that all perturbations target the same pathway and, thus, allow estimating the consistency or stability of the produced solutions. Consistent algorithms allow integrating solutions from multiple perturbations in a confident manner into a consensus solution. We measured the consistency of any given algorithm by its ability to recover similar sets of nodes and edges across data sets. As shown in Fig. 3, the SteinerNet algorithm achieved the highest consistency scores on these data sets, followed by the iPoint variants.

5.2 Huntington's disease

Huntington's disease is a neurodegenerative disease caused by a mutation in the Huntingtin (HTT) gene. The exact mechanisms underlying the pathogenesis of the disease are

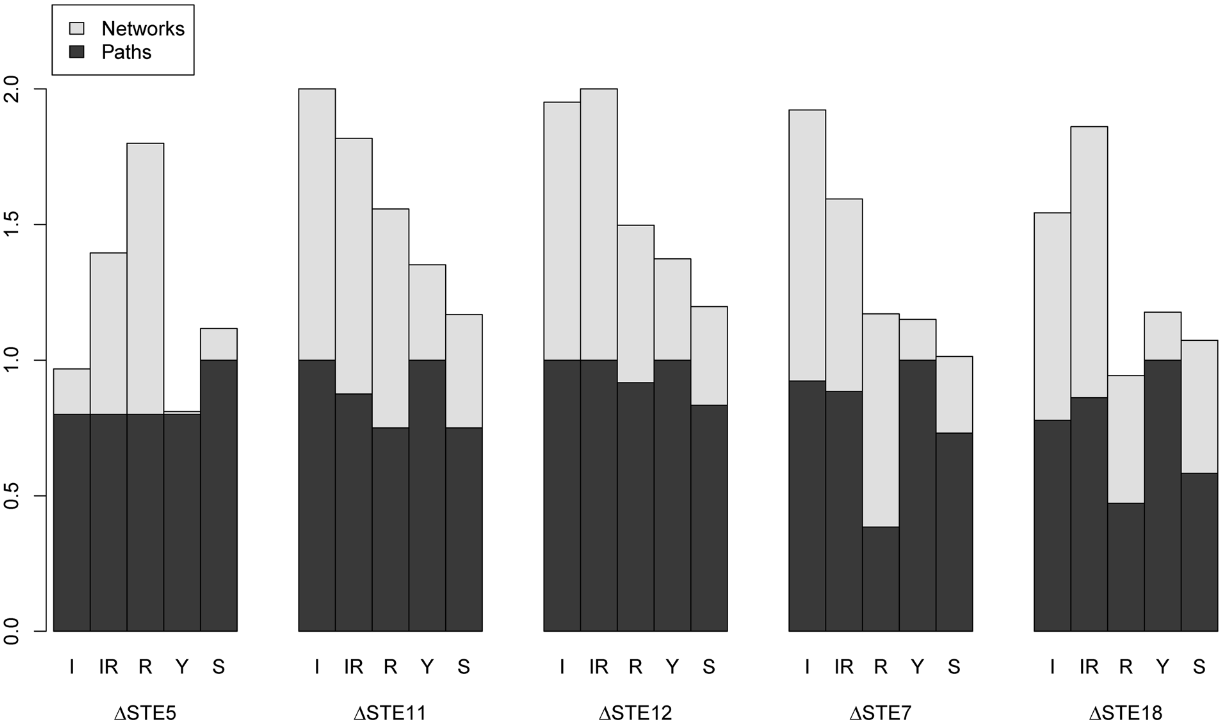

Fig. 2 Functional enrichment of inferred subnetworks on the response to pheromone data. Evaluation of enrichment of the entire network solution (light gray bars) and paths connecting anchors with terminals (dark gray bars) of different approaches for network reconstruction across multiple data sets related to the response to the pheromone pathway. Method abbreviations are as follows: I – iPoint, IR – iPoint$_{reg}$, R – ResponseNet, S – SteinerNet and Y – Yosef *et al.*

Table 1 Performance evaluation on the response to pheromone data sets. Values denote average normalized scores across the different data sets. *Network* details the enrichment score for the entire network and *Paths* details the enrichment score for individual paths connecting anchors with terminals

Algorithm	Network	Paths	Total
iPoint	0.78	0.90	1.68
iPoint$_{reg}$	*0.85*	0.88	*1.73*
ResponseNet	0.73	0.66	1.39
Yosef *et al.*	0.21	*0.96*	1.17
SteinerNet	0.33	0.78	1.11

not fully understood. In this case, we applied the four algorithms to infer subnetworks connecting HTT to genes that are differentially expressed in the disease state. Notably, across a wide range of algorithmic parameters ResponseNet did not produce any solution for this data set. Therefore, we compared iPoint's performance to that of Yosef *et al.* and SteinerNet only.

We calculated the enrichment of the solutions produced with respect to three gene sets related to HD: (i) KEGG HD pathway; (ii) HD genes in CTD; and (iii) HD genes in GAD. Additionally, as HD is strongly associated with apoptotic processes,[19,20] we scored the enrichment of the reconstructed networks with genes annotated for apoptosis in KEGG and GO.

In all but one data set (GAD genes), iPoint variants outperformed the other algorithms (see Fig. 4). Of note is the poor performance of SteinerNet with respect to path enrichments in these data, even when forcing its solutions to include the anchor (HTT) as described above.

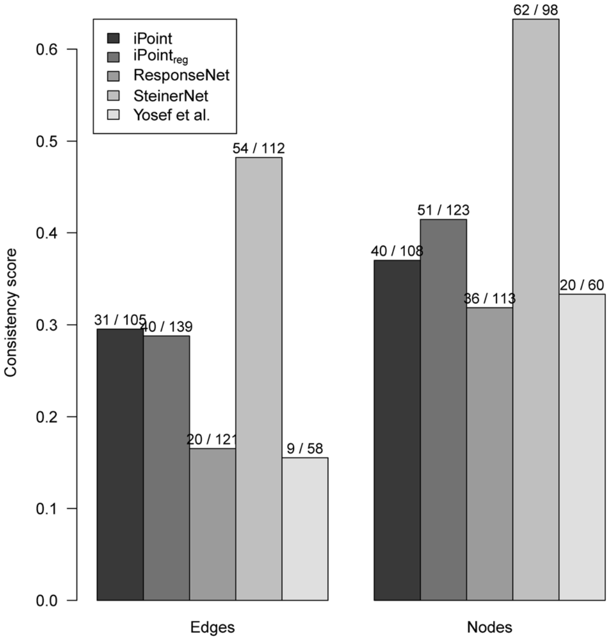

Fig. 3 Consistency evaluation. Shown are percents of consistent edges and nodes across solutions to the different pheromone response data sets. The numbers above the bars denote the number of consistent nodes or edges *versus* their total number across the inferred networks.

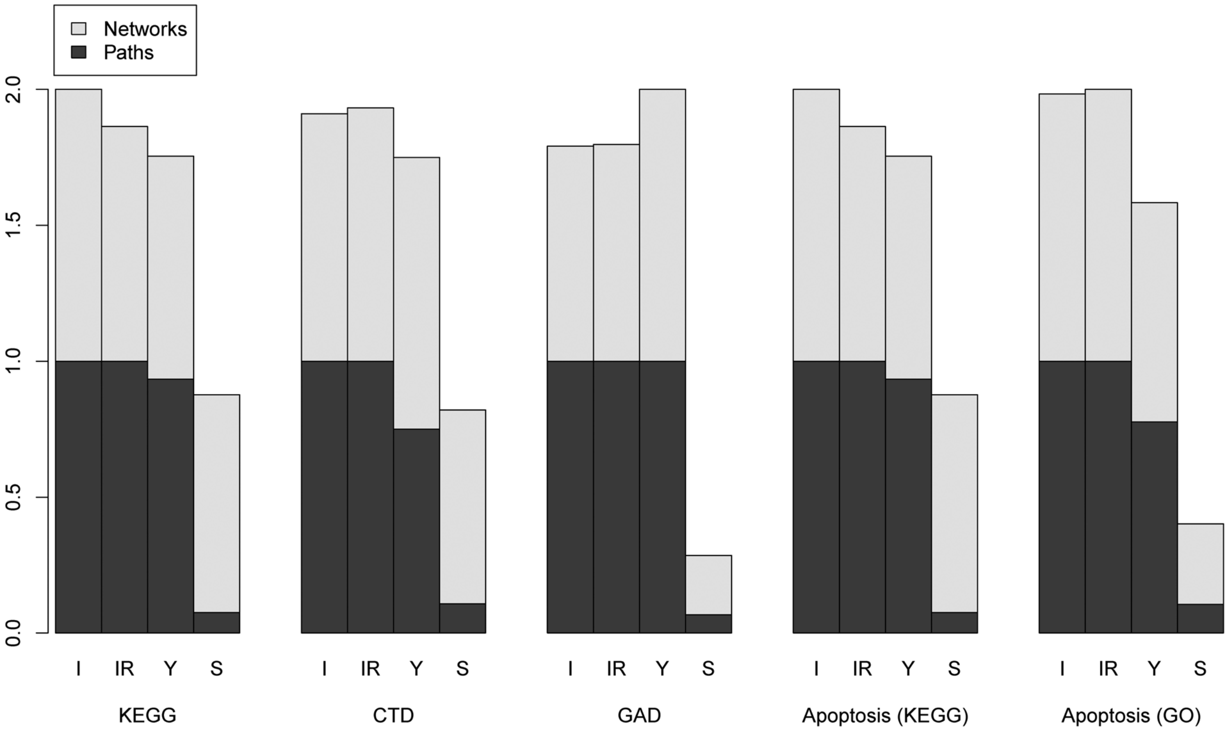

Fig. 4 Performance analysis on Huntington's disease data set. Evaluation of the functional enrichment for the inferred networks (light gray bars) as well as paths (dark gray bars) of different approaches for network reconstruction across multiple validation sets. Method abbreviations are as in the legend of Fig. 2.

6 Conclusions

In this paper we presented iPoint, an exact algorithm for subnetwork inference. The algorithm uses an ILP formulation to optimize global and local aspects of the reconstructed network. Our contribution is threefold: (i) we formulate the theoretical model of Yosef *et al.* as an integer linear program and optimally solve it in an efficient manner; (ii) we give an empirical assessment for the approximation performance of the algorithm with respect to network size and objective; and (iii) we demonstrate the advantages of our framework over state-of-the-art inference algorithms on various data sets in yeast and human, providing high quality and consistent networks.

Several extensions of our work are possible. First, following the approach of Huang and Fraenkel,[3] the algorithm could be easily adapted to account for confidence values of the input terminals, thereby including in the inferred subnetwork high scoring terminals and, possibly, discarding low scoring terminals. Second, the ILP could serve as a convenient platform for integrating additional data sets by placing explicit constraints to ensure a reconstruction that is highly consistent across the data sets.

Acknowledgements

We thank Nir Yosef for his help and fruitful discussions regarding this work. NA was supported in part by a fellowship from the Edmond J. Safra Center for Bioinformatics at Tel Aviv University. RS was supported by a research grant from the Israel Science Foundation (grant no. 241/11).

References

1 M. S. Scott, T. Perkins, S. Bunnell, F. Pepin, D. Y. Thomas and M. Hallett, *Mol. Cell. Proteomics*, 2005, **4**, 683–692.
2 R. Shachar, L. Ungar, M. Kupiec, E. Ruppin and R. Sharan, *Mol. Syst. Biol.*, 2008, **4**, 172.
3 S.-S. C. Huang and E. Fraenkel, *Sci. Signaling*, 2009, **2**, ra40.
4 N. Yosef, L. Ungar, E. Zalckvar, A. Kimchi, M. Kupiec, E. Ruppin and R. Sharan, *Mol. Syst. Biol.*, 2009, **5**, 248.
5 E. Yeger-Lotem, L. Riva, L. J. Su, A. D. Gitler, A. G. Cashikar, O. D. King, P. K. Auluck, M. L. Geddie, J. S. Valastyan, D. R. Karger, S. Lindquist and E. Fraenkel, *Nat. Genet.*, 2009, **41**, 316–323.
6 M. Bailly-Bechet, C. Borgs, A. Braunstein, J. Chayes, A. Dagkessamanskaia, J.-M. François and R. Zecchina, *Proc. Natl. Acad. Sci. U. S. A.*, 2011, **108**, 882–887.
7 C.-H. Yeang, T. Ideker and T. Jaakkola, *J. Comput. Biol.*, 2004, **11**, 243–262.
8 S. Bruckner, F. Hüffner, R. M. Karp, R. Shamir and R. Sharan, *J. Comput. Biol.*, 2010, **17**, 237–252.
9 N. Yosef, E. Zalckvar, A. D. Rubinstein, M. Homilius, N. Atias, L. Vardi, I. Berman, H. Zur, A. Kimchi, E. Ruppin and R. Sharan, *Sci. Signaling*, 2011, **4**, pl1.
10 A. Lan, I. Y. Smoly, G. Rapaport, S. Lindquist, E. Fraenkel and E. Yeger-Lotem, *Nucleic Acids Res.*, 2011, **39**, W424–W429.

11 N. Tuncbag, S. McCallum, S.-S. C. Huang and E. Fraenkel, *Nucleic Acids Res.*, 2012, **40**, W505–W509.

12 J. M. Cherry, C. Ball, S. Weng, G. Juvik, R. Schmidt, C. Adler, B. Dunn, S. Dwight, L. Riles, R. K. Mortimer and D. Botstein, *Nature*, 1997, **387**, 67–73.

13 T. R. Hughes, M. J. Marton, A. R. Jones, C. J. Roberts, R. Stoughton, C. D. Armour, H. A. Bennett, E. Coffey, H. Dai, Y. D. He, M. J. Kidd, A. M. King, M. R. Meyer, D. Slade, P. Y. Lum, S. B. Stepaniants, D. D. Shoemaker, D. Gachotte, K. Chakraburtty, J. Simon, M. Bard and S. H. Friend, *Cell*, 2000, **102**, 109–126.

14 N. Yosef, M. Kupiec, E. Ruppin and R. Sharan, *Nucleic Acids Res.*, 2009, **37**, e88.

15 A. Hodges, A. D. Strand, A. K. Aragaki, A. Kuhn, T. Sengstag, G. Hughes, L. A. Elliston, C. Hartog, D. R. Goldstein, D. Thu, Z. R. Hollingsworth, F. Collin, B. Synek, P. A. Holmans, A. B. Young, N. S. Wexler, M. Delorenzi, C. Kooperberg, S. J. Augood, R. L. M. Faull, J. M. Olson, L. Jones and R. Luthi-Carter, *Hum. Mol. Genet.*, 2006, **15**, 965–977.

16 IBM, *IBM ILOG CPLEX V12.1 User's Manual for CPLEX*, 2009.

17 K. G. Becker, K. C. Barnes, T. J. Bright and S. A. Wang, *Nat. Genet.*, 2004, **36**, 431–432.

18 A. P. Davis, B. L. King, S. Mockus, C. G. Murphy, C. Saraceni-Richards, M. Rosenstein, T. Wiegers and C. J. Mattingly, *Nucleic Acids Res.*, 2011, **39**, D1067–D1072.

19 M. A. Hickey and M. F. Chesselet, *Prog. Neuro-Psychopharmacol. Biol. Psychiatry*, 2003, **27**, 255–265.

20 D. Yang, C.-E. Wang, B. Zhao, W. Li, Z. Ouyang, Z. Liu, H. Yang, P. Fan, A. O'Neill, W. Gu, H. Yi, S. Li, L. Lai and X.-J. Li, *Hum. Mol. Genet.*, 2010, **19**, 3983–3994.