A Novel Method for Signal Transduction Network Inference from Indirect Experimental Evidence*

Réka Albert[1] **, Bhaskar DasGupta[2] ***, Riccardo Dondi[3], Sema Kachalo[4] †, Eduardo Sontag[5] ‡, Alexander Zelikovsky[6], and Kelly Westbrooks[6]

[1] Department of Physics, Pennsylvania State University, University Park, PA 16802. ralbert@phys.psu.edu

[2] Department of Computer Science, University of Illinois at Chicago, Chicago, IL 60607. dasgupta@cs.uic.edu

[3] Dipartimento di Scienze dei Linguaggi, della Comunicazione e degli Studi Culturali, Università degli Studi di Bergamo, Bergamo, Italy, 24129. riccardo.dondi@unibg.it

[4] Department of Bioengineering, University of Illinois at Chicago, Chicago, IL 60607. sema@uic.edu

[5] Department of Mathematics, Rutgers University, New Brunswick, NJ 08903. sontag@math.rutgers.edu

[6] Department of Computer Science, Georgia State University, Atlanta, GA 30303. {alexz,kelly}@cs.gsu.edu

Abstract. In this paper we introduce a new method of combined synthesis and inference of biological signal transduction networks. A main idea of our method lies in representing observed causal relationships as network paths and using techniques from combinatorial optimization to find the sparsest graph consistent with all experimental observations. Our contributions are twofold: on the theoretical and algorithmic side, we formalize our approach, study its computational complexity and prove new results for exact and approximate solutions of the computationally hard transitive reduction substep of the approach. On the application side, we validate the biological usability of our approach by successfully applying it to a previously published signal transduction network by Li et al. [20] and show that our algorithm for the transitive reduction substep performs well on graphs with a structure similar to those observed in transcriptional regulatory and signal transduction networks.

1 Introduction

Most biological characteristics of a cell arise from the complex interactions between its numerous constituents such as DNA, RNA, proteins and small

* A full version of this paper will appear in Journal of Computational Biology.

** Partly supported by a Sloan Research Fellowship, NSF grants DMI-0537992, MCB-0618402 and USDA grant 2006-35100-17254.

*** **Corresponding author**. Partly supported by NSF grants IIS-0346973, IIS-0612044 and DBI-0543365.

† Supported by NSF grant IIS-0346973.

‡ Partly supported by NSF grant DMS-0614371.

molecules [3]. Cells use signaling pathways and regulatory mechanisms to coordinate multiple functions, allowing them to respond to and acclimate to an ever-changing environment. Genome-wide experimental methods now identify interactions among thousands of proteins [11, 12, 18, 19]; however these experiments are rarely conducted in the specific cell type of interest and are not able to probe the directionality of the interactions (*i.e.*, to distinguish between the regulatory source and target). Identification of every reaction and regulatory interaction participating even in a relatively simple function of a single-celled organism requires a concerted and decades-long effort. Consequently, the state of the art understanding of many signaling processes is limited to the knowledge of key mediators and of their positive or negative effects on the whole process.

Experimental information about the involvement of a specific component in a given signal transduction network can be partitioned into three categories. First, biochemical evidence that provides information on enzymatic activity or protein-protein interactions. This first category is a *direct interaction*, *e.g.*, binding of two proteins or a transcription factor activating the transcription of a gene or a chemical reaction with a single reactant and single product. Second, pharmacological evidence, in which a chemical is used either to mimic the elimination of a particular component, or to exogenously provide a certain component, leads to observed relationships that are not direct interactions but indirect causal effects most probably resulting from a chain of interactions and reactions. For example, binding of a chemical to a receptor protein starts a cascade of protein-protein interactions and chemical reactions that ultimately results in the transcription of a gene. Observing gene transcription after exogeneous application of the chemical allows inferring a causal relationship between the chemical and the gene that however is not a direct interaction. Third, genetic evidence of differential responses to a stimulus in wild-type organisms versus a mutant organism implicates the product of the mutated gene in the signal transduction process. This category is a three-component inference that in a minority of cases could correspond to a single reaction (namely, when the stimulus is the reactant of the reaction, the mutated gene encodes the enzyme catalysing the reaction and the studied output is the product of the reaction), but more often it is indirect. As stated above, the last two types of inference do not give direct interactions but indirect causal relationships that correspond to reachability relationships in the unknown interaction network. Here we describe a method for synthesizing indirect (path-level) information into a consistent network by constructing the sparsest graph that maintains all reachability relationships.

This method's novelty over other network inference approaches is that it does not require expression information (as all reverse engineering approaches do, for a review see [5]). Moreover, our method significantly expands the capability for incorporating indirect (pathway-level) information. Previous methods of synthesizing signal transduction networks [21] only include direct biochemical interactions, and are therefore restricted by the incompleteness of the experimental knowledge on pairwise interactions. Our method is able to incorporate

indirect causal effects as network paths with known starting and end vertices and (yet) unknown intermediary vertices.

The first step of our method is to distill experimental conclusions into qualitative regulatory relations between cellular components. Following [8, 20], we distinguish between positive and negative regulation, usually denoted by the verbs "promote" and "inhibit" and represented graphically as $\rightarrow$ and $\dashv$. Biochemical and pharmacological evidence is represented as component-to-component relationships, such as "A promotes B", and is incorporated as a directed arc from A to B. Arcs corresponding to direct interactions are marked as such. Genetic evidence leads to double causal inferences of the type "C promotes the process through which A promotes B". The only way this statement can correspond to a direct interaction is if C is an enzyme catalyzing a reaction in which A is transformed into B. We represent supported enzyme-catalized reactions as both A (the substrate) and C (the enzyme) activating B (the product). If the interaction between A and B is direct and C is not a catalyst of the A-B interaction, we assume that C activates A. In all other cases we assume that the three-node indirect inference corresponds to an intersection of two paths ($A \Rightarrow B$ and $C \Rightarrow B$) in the interaction network; in other words, we assume that C activates an unknown intermediary (pseudo)-vertex of the AB path. The main idea of our method is finding the minimal graph, both in terms of pseudo vertex numbers and non-critical edge numbers, that is consistent with all reachability relationships between real vertices. The algorithms involved are of two kinds: (i) transitive reduction of the resulting graph subject to the constraints that no edges flagged as direct are eliminated and (ii) pseudo-vertex collapse subject to the constraints that real vertices are not eliminated.

Note that we are not claiming that real signal transduction networks are the sparsest possible; our goal is to minimize false positive (spurious) inferences, even if risking false negatives. Thus we want to be as close as possible to a "tree topology" while supporting all experimental observations. The implicit assumption of chain-like or tree-like topologies permeates the traditional molecular biology literature: signal transduction and metabolic pathways were assumed to be close to linear chains, genes were assumed to be regulated by one or two transcription factors [3]. According to current observations the reality is not far: the average in/out degree of transcriptional regulatory networks[18, 23] and the mammalian signal transduction network [21] is close to 1.

2 A Formal Description of the Network Synthesis

The goal of this section is to introduce a formal framework of the network synthesis procedure that is sufficiently general in nature, and amenable to algorithmic analysis and consequent automation. First, we need to describe a graph-theoretic problem which we refer to as the *binary transitive reduction* (BTR) problem. We are given a directed graph $G = (V, E)$ with an edge labeling function $w : E \mapsto \{0, 1\}$. Biologically, edge labels 0 and 1 in edges $u \overset{0}{\rightarrow} v$ and $u \overset{1}{\rightarrow} v$ correspond to the "u promotes v" and "u inhibits v", respectively.

The following definitions and notations are used throughout the paper. All paths are (possibly self-intersecting) directed paths unless otherwise stated. A non-self-intersecting path or cycle is called a *simple* path or cycle. If edge labels are removed or not mentioned, they are assumed to be 0 for the purpose of any problem that needs them. The *parity* of a path P from vertex u to vertex v is $\sum_{e \in P} w(e) \pmod 2$. A path of parity 0 (resp., 1) is called a path of *even* (resp, *odd*) parity. The same notions carries over to cycles in an obvious manner. The notation $u \stackrel{x}{\Rightarrow} v$ denotes a path from u to v of parity $x \in \{0,1\}$. If we do not care about the parity, we simply denote the path as $u \Rightarrow v$. An edge will simply be denoted by $u \stackrel{x}{\rightarrow} v$ or $u \rightarrow v$. For a subset of edges $E' \subseteq E$, reachable(E') is the set of all ordered triples (u, v, x) such that $u \stackrel{x}{\Rightarrow} v$ is a path of the restricted subgraph (V, E').

The BTR problem is defined as follows. An input instance is a directed graph $G = (V, E)$ with an edge labeling function $w : E \mapsto \{0, 1\}$ and a set of critical edges $E_{\text{critical}} \subseteq E$. A valid solution is a subgraph $G' = (V, E')$ where $E_{\text{critical}} \subseteq E' \subseteq E$ and reachable(E') =reachable(E). The objective is to find a valid solution that *minimizes* $|E'|$. Intuitively, the BTR problem is useful for determining a sparsest graph consistent with a set of experimental observations. The set of "critical edges" represent edges which are known to be direct interactions with concrete evidence. By maximizing sparseness we do not simply mean to minimize the number of edges per se, but seek to minimize the number of spurious feed-forward loops (*i.e.*, a node regulating another both directly and indirectly). Thus we want to be as close as possible to a "tree topology" while supporting the experimental observations.

We also need to define one more problem that will be used in the formal framework of the network synthesis approach. **The pseudo-vertex collapse (PVC) problem is defined as follows**. An input instance is a directed graph $G = (V, E)$ with an edge labeling function $w : E \mapsto \{0, 1\}$ and a subset $V' \subset V$ of vertices called pseudo-vertices. The vertices in $V \setminus V'$ are called "real" vertices. For any vertex v, let $\text{in}(v) = \{(u, x) \mid u \stackrel{x}{\Rightarrow} v, x \in \{0, 1\}\} \setminus \{v\}$ and let $\text{out}(v) = \{(u, x) \mid v \stackrel{x}{\Rightarrow} u, x \in \{0, 1\}\} \setminus \{v\}$. Collapsing two vertices u and v is permissible provided both are not "real" vertices and in(u) =in(v) and out(u) =out(v). If permissible, the collapse of two vertices u and v creates a new vertex w, makes every incoming (resp. outgoing) edges to (resp. from) either u or v an incoming (resp. outgoing) edge from w, removes any parallel edge that may result from the collapse operation and also removes both vertices u and v. A valid solution of the problem is then any graph $G'' = (V'', E'')$ that can be obtained from G by a sequence of permissible collapse operations and the objective is to find a valid solution that *minimizes* $|V''|$. Intuitively, the PVC problem is useful for reducing the pseudo-vertex set to the the minimal set that maintains the graph consistent with all indirect experimental observations. As in the case of the BTR problem, our goal is to minimize false positive (spurious) inferences of additional components in the network.

A formal framework for the network synthesis procedure is presented in Figure 1. As described in Section 1, in the first step we incorporate biochemical interaction or causal evidence as labeled edges, noting the critical edges corresponding to direct interactions. Then we perform a binary transitive reduction to eliminate spurious inferred edges (*i.e.*, edges that can be explained by paths of the same label). In step two we incorporate double causal relationships $A \xrightarrow{x} (B \xrightarrow{y} C)$ by (i) adding a new edge $A \xrightarrow{x} B$ if $B \xrightarrow{y} C$ is a critical edge, (ii) doing nothing if existing paths in the network already explain the relationship, or (iii) adding a new pseudo-vertex and three new edges. To correctly incorporate the parity of the $A \xrightarrow{x+y \pmod 2} C$ relationship, $B \xrightarrow{y} C$ paths, with $y \pmod 2 = 0$, will be broken into two edges with 0 parity, while paths of odd parity will be broken into an edge of $a = 0$ parity and an edge of $b = 1$ parity, summarized in a concise way by the equation $b = a + b = y \pmod 2$. The unnecessary redundancy of the resulting graph is reduced by performing pseudo-vertex collapse, then a second round of binary transitive reduction. Intuitively, the approach in Figure 1 first expands the network by the addition of the pseudo-vertices at the intersection of the two paths corresponding to three-node inferences, then uses the additional information available in the network to collapse these pseudo-vertices, *i.e.*, to identify them with real vertices or with each other. The PVC is the heart of the algorithm, the final BTR is akin to a final cleanup step; thus it is important to perform PVC before BTR in Step 2.2 of Figure 1.

1 [**encoding single causal inferences**]

- **1.1** Build a network for each causal inference of the type $A \xrightarrow{0} B$ or $A \xrightarrow{1} B$ noting each critical edge.
- **1.2** Solve the BTR problem for this network.

2 [**encoding double causal inferences**]

- **2.1** Consider each indirect causal relationship $A \xrightarrow{x} (B \xrightarrow{y} C)$ where $x, y \in \{0, 1\}$. We add new nodes and/or edges in the network based on the following cases:
  - If $B \xrightarrow{y} C \in E_{\text{critical}}$, then we add $A \xrightarrow{x} (B \xrightarrow{y} C)$.
  - If there is no subgraph of the form $\begin{array}{ccc} & A & \\ & \Downarrow x & \\ B & \overset{a}{\Rightarrow} \; D \; \overset{b}{\Rightarrow} & C \end{array}$ for some node D where $b = a + b = y \pmod 2$ then add the subgraph $\begin{array}{ccc} & A & \\ & \downarrow x & \\ B & \xrightarrow{a} \; \mathrm{P} \; \xrightarrow{b} & C \end{array}$ to the network where a new "pseudo-node" P is added and $b = a + b = y \pmod 2$.
- **2.2** Solve the PVC problem for the resulting graph.

3 [**final reduction**] Solve the BTR problem for the network.

Fig. 1. The overall network synthesis approach.

Proposition 1 *All the steps in the network synthesis procedure except the steps that involve BTR can be solved in polynomial time.*

3 Summary of Pertinent Previous Works

The idea of transitive reduction, though in a more simplistic setting and/or integrated in an approach different from what appears in this paper, has been used by a few researchers before. For example, in [25] Wagner's goal is to find the network from the *reachability information*. He constructs uniformly random graphs and scale-free networks in a range of connectivities (average degrees), and matches their reachability information to the range of gene reachability information found from yeast perturbation studies. He concludes that the expected number of direct regulatory interactions per gene is around 1 (if the underlying graph is uniformly random) or less than 0.5 (if the underlying graph is scale free with a degree exponent of 2). Chen et al. in [6] use time-dependent gene expression information to determine candidate activators and inhibitors of each gene, then prune the edges by assuming that no single gene functions both as activator and inhibitor. This assumption is too restrictive given that transcription factors can have both activation and inhibition domains, and the same protein-level interactions (for example phosphorylation by a kinase) can have positive or negative functional character depending on the target. Li et al. in [20] manually synthesize a plant signal transduction network from indirect (single and double) inferences introducing a first version of pseudo-vertex collapse. They assume that if $A \xrightarrow{0} B$, $A \xrightarrow{0} C$ and $C \xrightarrow{0} (A \xrightarrow{0} B)$, the most parsimonious explanation is that $A \xrightarrow{0} C \xrightarrow{0} B$. The reader is referred to the excellent surveys in [9, 15] for further general information on biological network inference and modelling.

Special cases of the BTR problem have been looked at by the theoretical computer science community in a different context of designing reliable communication networks. Obviously, BTR is NP-complete since the special case with all-zero edge labels includes the problem of finding a directed Hamiltonian cycle in a graph. If $E_{\text{critical}} = \emptyset$, BTR with all-zero edge labels is known as the *minimum equivalent digraph* (MED) problem. MED is known to be MAX-SNP-hard, admits a polynomial time algorithm with an approximation ratio of $1.617 + \varepsilon$ for *any* constant $\varepsilon > 0$ [16] and can be solved in polynomial time for directed acyclic graphs [1]. More recently, Vetta [24] has claimed a $\frac{3}{2}$-approximation for the MED problem. A weighted version of the MED problem admits a 2-approximation [10]; this implies a 2-approximation for the BTR problem when all-zero edge labels.

In a previous publication [4], we considered the BTR problem, generalized it to a so-called *p-ary transitive reduction* problem and provided an approximation algorithm for this generalization. In particular, we designed a $2 + o(1)$-approximation for the generalized problem, observed that the general problem can be solved in polynomial time if the input graph is a DAG and provided a 1.78-approximation for the BTR problem when all edge labels are zero but critical edges are allowed. The results in [4] are purely theoretical in nature with no

experimental or implementation results, moreover the network synthesis process described in Figure 1 does *not* appear in [4]. All the theoretical results reported in this paper are *disjoint* from the results reported in [4].

4 New Algorithmic Results for BTR

Theorem 1 *BTR can be solved in polynomial time if the graph has no cycles of length more than* 3.

Theorem 2 *The GREEDY procedure, namely the following approach:*

> **Definition** *an edge* $u \xrightarrow{x} v$ *is redundant if there is an alternate path* $u \overset{x}{\Rightarrow} v$
> *GREEDY*
> **while** *(there exists a redundant edge) delete the redundant edge*

is a 3*-approximation for the BTR problem. Moreover, there are input instances of BTR for which GREEDY has an approximation ratio of at least* 2.

5 Our Implemention for the BTR Problem

Given an instance graph $G = (V, E)$ of the BTR problem, one can design a straightforward dynamic programming approach to determine, for every $u, v \in V$ and every $x \in \{0, 1\}$, if $u \overset{x}{\Rightarrow} v$ exists in G. The worst-case running time of the algorithm is $O(|V|^3)$. To solve the BTR problem within a acceptable time complexity while ensuring a good accuracy, we have implemented the following two major approaches. In *Approach 1* (applicable for smaller graphs), if the number of nodes in the graph is at most a threshold N, we implemented the GREEDY heuristic of Theorem 2 on the *entire graph*. The heuristic is implemented by iteratively selecting a new non-critical edge $e = u \rightarrow v$ for removal, tentatively removing it from G and checking if the resulting graph has a path $u \overset{x}{\Rightarrow} v$. If so, we remove the edge; otherwise, we keep it and mark it so that we never select it again. We stop when we have no more edges to select for deletion. In *Approach 2* (applicable for larger graphs), if the number of nodes in the graph is above the threshold N, we first use Approach 1 for every strongly connected component of G. Then we use two procedures $T_{\text{cycle-to-gadget}}$ and $T_{\text{gadget-to-cycle}}$, described in the proof of Theorem 2 in the full version of this paper, to identify the remaining edges that can be deleted. To speed up our implementations and to improve accuracy, we also use some algorithmic engineering approaches. For example, we stop the Floyd-Warshall iteration in Approach 1 as soon as an alternate path $u \overset{x}{\Rightarrow} v$ is discovered, we randomize the selection of the next edge for removal and, in Approach 2, if the strongly connected component has very few vertices, calculate an exact solution of BTR on this component exhaustively. Both Approach 1 and Approach 2 are guaranteed to be a 3-approximate solution by Theorem 2. However, in Approach 1 there is no bias towards a particular candidate edge for removal among all candidate edges; in contrast, in Approach 2 a bias is introduced via removal of duplicate edges in the gadget replacement procedure. Thus,

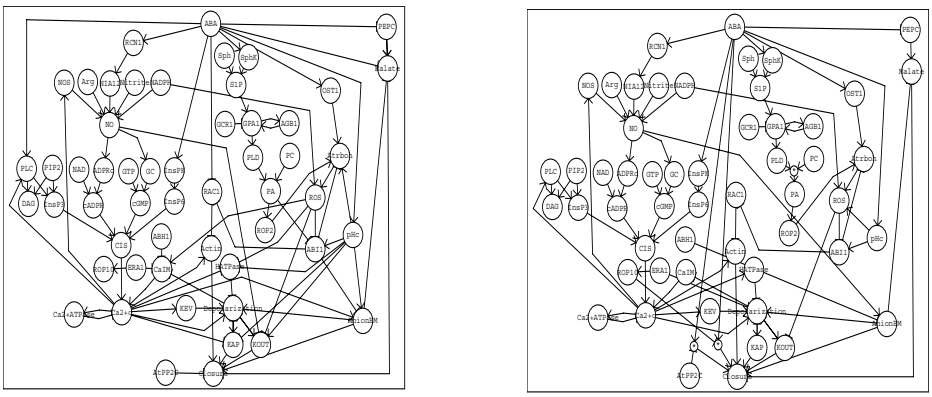
the two approaches may return slightly different solutions in practice. Choosing N to be 150, **our implementation takes mostly negligible time** to run on networks with up to thousands of nodes, taking time of the order of seconds for the manually curated network that is described in Section 6 to about a minute for the 1000 node random biological networks described in Section 7 on which we tested the performance of our implementations. Theoretical worst-case estimates of the running times of the two approaches are as follows. Approach 1 runs in $O(d \cdot |V|^3)$ time where d is the number of non-critical edges. By using a linear-time solution of the BTR problem on a DAG, Approach 2 runs in $O(m^2 + |E| + \sum_{i=1}^{m} d_i \cdot n_i^3)$ time where the given graph has m strongly connected components and d_i (n_i) are the number of non-critical edges (vertices) in the i^{th} strongly connected component.

6 Synthesizing ABA-induced Stomatal Closure Network

Network inference algorithms applied to gene expression (microarray) data based on several types of analysis lead to indirect causal relationships among genes. Large-scale repositories for microarray data for several organisms such as Many Microbe Microarrays, NASCArrays and Gene Expression Omnibus contain expression information for thousands of genes under tens to hundreds of experimental conditions. Our methods are applicable for filtering redundant information by binary transitive reduction of indirect pairwise data and for incorporating differential gene expression under experimental perturbations by pseudo-vertex collapse. Signal transduction pathway repositories such as TRANSPATH and protein interaction databases contain up to thousands of interactions, a large number of which are not supported by direct binding evidence. Our methods can be used to selectively filter redundant information while keeping all direct interactions.

In this section we discuss our computational results on synthesizing experimental results into a consistent guard cell signal transduction network for ABA-induced stomatal closure using our detailed procedure described in Section 2 and compare it with the manually curated network obtained in [20]. Our starting point is the list of experimentally observed causal relationships in ABA-induced closure collected by Li et al. and published as Table S1 in [20]. This table contains around 140 interactions and causal inferences, both of type "A promotes B" and "C promotes process(A promotes B)". We augment this list with critical edges drawn from biophysical/biochemical knowledge on enzymatic reactions and ion flows and with simplifying hypotheses made by Li et al., both described in Text S1 of [20].

The synthesis of the network is carried out using the formal method described in Section 2. We also formalize an additional rule specific to the context of this network (and implicitly assumed by [20]) regarding enzyme-catalyzed reactions. We follow Li et al. in representing each of these reactions by two directed critical edges, one from the reaction substrate to the product and one from the enzyme to the product. As the reactants (substrates) of the reactions in [20] are abundant, the only way to regulate the product is by regulating the enzyme. The



(a) **(b)**

Fig. 2. (a) The network manually synthesized by Li et al. [20]. **(b)** The network synthesized in this paper. A pseudo-vertex is displayed as ⊛.

enzyme, being a catalyst, is always promoting the product's synthesis, thus positive indirect regulation of a product will be interpreted as positive regulation of the enzyme, and negative indirect regulation of the product will be interpreted as negative regulation of the enzyme. In graph-theoretic terms, this leads to the following rule. We have a subset $E_{\text{enzymatic}} \subseteq E_{\text{critical}}$ of edges that are all labeled 0. Suppose that we have a path $A \xrightarrow{a} x \xrightarrow{b} B$, an edge $C \xrightarrow{0} B \in E_{\text{enzymatic}}$. Then, we identify the node C with x by collapsing them together and set the parities of the edges $A \to (x = C)$ and $(x = C) \to B$ based on the following two cases: if $a + b = 0 \pmod 2$ then both $A \to (x = C)$ and $(x = C) \to B$ have zero parities, otherwise if $a + b = 1 \pmod 2$ then $A \to (x = C)$ has parity 1 and $(x = C) \to B$ has parity 0. The manually synthesized network of Li et al. includes a pseudo-vertex for each non-critical edge, indicating the existence of unknown biological mediators. For the ease of comparison we omit these degree two pseudo-vertices. The two networks are shown in Figures 2 (a)–(b). Here is a brief summary of an overall comparison of the two networks:

- [20] has 54 vertices and 92 edges; our network has 57 vertices (3 extra pseudo-vertices) but only 84 edges. The two networks have 71 common edges.
- Both [20] and our network has identical strongly connected component (SCC) of vertices. There is one SCC of size 18 (KOUT Depolarization KAP CaIM Ca2+c Ca2+ATPase HATPase KEV PLC InsP3 NOS NO GC cGMP ADPRc cADPR CIS AnionEM), one SCC of size 3 (Atrboh ROS ABI1), one SCC of size 2 (GPA1 AGB1) and the rest of the SCCs are of size 1 each.
- All the paths present in the [20] reconstruction are present in our network as well. Our network has the extra path ROP10$\overset{1}{\Rightarrow}$Closure that Li et al. cited in their Table S1 but did not include in their network due to weak supporting evidence.

Thus the two networks are highly similar but diverge on a number of edges. Li et al. keep a few graph-theoretically redundant edges such as ABA$\xrightarrow{0}$PLC, PA$\xrightarrow{1}$ABI1 and ROS$\xrightarrow{0}$CaIM that would be explainable by feedback processes. Some of our edges such as NO$\xrightarrow{0}$AnionEM correspond to paths in Li et al.'s reconstruction. Our graph contains the full pseudo-vertex-using representation of the process AtPP2C$\xrightarrow{1}$(ABA$\xrightarrow{0}$Closure) that Li et al. simplifies to AtPP2C$\xrightarrow{1}$ABA. We have pHc$\xrightarrow{0}$ROS and ROS$\overset{0}{\Rightarrow}$Atrboh where [20] has pH$\xrightarrow{0}$Atrboh and a positive feedback loop on Atrboh. All these discrepancies are due not to algorithmic deficiencies but to human decisions. Finally, the entire network synthesis process was done within a few seconds by our implemented algorithm.

7 BTR Algorithm's Performance on Simulated Networks

A variety of cellular interaction and regulatory networks have been mapped and graph theoretically characterized. One of the most frequently reported graph measures is the distribution of node degrees, *i.e.*, the distribution of the number of incoming or outgoing edges per node. A variety of networks, including many cellular interaction networks, are heterogeneous (diverse) in terms of node degrees and exhibit a degree distribution that is close to a power-law or a mixture of a power law and an exponential distribution[2, 11, 14, 19, 21]. Transcriptional regulatory networks exhibit a power-law out-degree distribution, while the in-degree distribution is more restricted [18, 23]. To test our algorithm on a network similar to the observed features, we generate random networks with a prescribed degree distribution using the methods in [22]. We base the degree distributions on the yeast transcriptional regulatory network that has a maximum out-degree ~ 150 and maximum in-degree ~ 15 [18]. In our generated network the distribution of in-degree of the network is *exponential*, *i.e.*, Pr[in-degree$= x$]$= L\mathbf{e}^{-Lx}$ with L between $1/2$ and $1/3$ and the maximum in-degree is 12. The distribution of out-degree of the network is governed by a power-law, *i.e.*, for $x \geq 1$ Pr[out-degree$= x$]$= cx^{-c}$ and for $x = 0$ Pr[out-degree$= 0$]$\geq c$ with c between 2 and 3 and the maximum

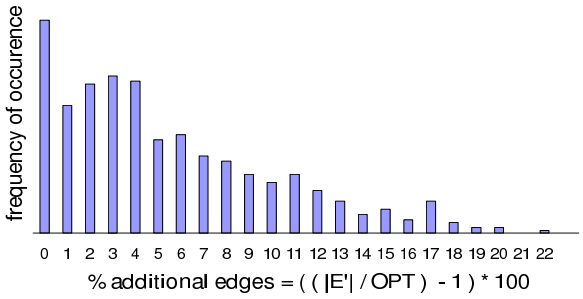


Fig. 3. A plot of the empirical performance of our BTR algorithm on the 561 simulated interaction networks. E' is our solution, OPT is the loose lower bound on the minimum number of edges and $100 \times \left(\frac{|E'|}{\text{OPT}} - 1\right)$ is the percentage of additional edges that our algorithm keeps. *On an average, we use no more than* 5.5% *more edges than the optimum (with about* 4.8% *as the standard deviation*).

out-degree is 200. We varied the ratio of excitory to inhibitory edges between 2 and 4. Since there are no known biological estimates of critical edges, we tried a few small and large values, such as 1%, 2% and 50%, for the percentage of edges that are critical to catch qualitatively all regions of dynamics of the network that are of interest. [7] To empirically test the performance of our algorithm, we used the (rather loose) lower bound OPT for the optimal solution $\mathbf{OPT} \geq \max\{\mathbf{n} + \mathbf{s} - \mathbf{c}, \mathbf{t}, \mathcal{L}\}$ where n is the number of vertices, s is the number of strongly connected components, c is the number of connected components of the underlying undirected graph, t is the number of those edges $u \xrightarrow{x} v$ such that either $u \xrightarrow{x} v \in E_{\text{critical}}$ or there is no alternate path $u \overset{x}{\Rightarrow} v$ in the graph and $\mathcal{L}$ is a lower bound described in the full version of the paper.

We tested the performance of our BTR algorithm on 561 randomly generated networks varying the number of vertices between roughly 100 and 900. A summary of the performance is shown in Figure 3 indicating that our transitive reduction procedure returns solutions close to optimal in many cases even with such a simple lower bound of OPT. The running time of BTR on an individual network is negligible (from about one second for a 100 node networks to about no more than a minute for a 1000 node network). A summary of the various statistics of these 561 networks is given in Figure 4. To verify the performance of our BTR algorithm we perturb most of these networks with increasing amounts of additional random edges chosen such they do not change the optimal solution of the original graph. In most cases, our algorithm returns a solution that is either optimal or very close to the original network on which additional edges are added.

number of nodes	average number of edges			
(range)	total	excitory	inhibitory	critical
98–100	206	147	59	31
250–282	690	552	138	33
882-907	2489	1991	498	118

Fig. 4. Basic statistics of the simulated networks used in Figure 3.

Software. See `http://www.cs.uic.edu/~dasgupta/network-synthesis/`.

References

1. A. Aho, M. R. Garey and J. D. Ullman. *The transitive reduction of a directed graph*, SIAM Journal of Computing, 1 (2), 131-137, 1972.

[7] By "estimates of critical edges", we mean an accurate estimate of the percentage of total edges that are critical on an average in a biological network. Depending on the experimental or inference methods, different network reconstructions have *widely varying* expected fractions of critical edges. For example, the curated network of Ma'ayan et al. [21] is expected to have close to 100% critical edges as they specifically focused on collecting direct interactions only. Protein interaction networks are expected to be mostly critical [11, 12, 19]. The so-called genetic interactions (*e.g.*, synthetic lethal interactions) represent compensatory relationships, and only a minority of them are direct interactions. Network inference (reverse engineering) approaches lead to networks whose interactions are close to 0% critical.

2. R. Albert and A.-L. Barabási. *Statistical mechanics of complex networks*, Reviews of Modern Physics, 74 (1), 47-97, 2002.
3. B. Alberts. Molecular biology of the cell, New York: Garland Pub., 1994.
4. R. Albert, B. DasGupta, R. Dondi and E. Sontag. *Inferring (Biological) Signal Transduction Networks via Transitive Reductions of Directed Graphs*, Algorithmica, to appear.
5. G. W. Carter. *Inferring network interactions within a cell*, Briefings in Bioinformatics, 6 (4), 380-389, 2005.
6. T. Chen, V. Filkov and S. Skiena. *Identifying Gene Regulatory Networks from Experimental Data*, Proc. of third RECOMB, 94-103, 1999.
7. T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein. *Introduction to Algorithms*, The MIT Press, 2001.
8. B. DasGupta, G. A. Enciso, E. D. Sontag and Y. Zhang. *Algorithmic and Complexity Results for Decompositions of Biological Networks into Monotone Subsystems*, to appear in Biosystems (also in WEA-2006, LNCS 4007, 253-264, 2006).
9. V. Filkov. *Identifying Gene Regulatory Networks from Gene Expression Data*, in Handbook of Computational Molecular Biology (S. Aluru editor), Chapman & Hall/CRC Press, 2005.
10. G. N. Frederickson and J. JàJà. *Approximation algorithms for several graph augmentation problems*, SIAM Journal of Computing, 10 (2), 270-283, 1981.
11. L. Giot, J. S. Bader et al. *A protein interaction map of Drosophila melanogaster*, Science, 302, 1727-1736, 2003.
12. J. D. Han, N. Bertin et al. *Evidence for dynamically organized modularity in the yeast protein-protein interaction network*, Nature 430, 88-93, 2004.
13. R. Heinrich and S. Schuster. The regulation of cellular systems, New York: Chapman & Hall, 1996.
14. H. Jeong, B. Tombor, R. Albert, Z. N. Oltvai and A.-L. Barabási. *The large-scale organization of metabolic networks* Nature, 407, 651-654, 2000.
15. H. D. Jong. *Modelling and Simulation of Genetic Regulatory Systems: A Literature Review*, Journal of Computational Biology, 9 (1), 67-103, 2002.
16. S. Khuller, B. Raghavachari and N. Young. *Approximating the minimum equivalent digraph*, SIAM Journal of Computing, 24 (4), 859-872, 1995.
17. S. Khuller, B. Raghavachari and N. Young. *On strongly connected digraphs with bounded cycle length*, Discrete Applied Mathematics, 69 (3), 281-289, 1996.
18. T. I. Lee, N. J. Rinaldi et al. *Transcriptional regulatory networks in Saccharomyces cerevisiae*, Science 298, 799-804, 2002.
19. S. Li, C. M. Armstrong et al. *A map of the interactome network of the metazoan C. elegans*, Science 303, 540-543, 2004.
20. S. Li, S. M. Assmann and R. Albert. *Predicting Essential Components of Signal Transduction Networks: A Dynamic Model of Guard Cell Abscisic Acid Signaling*, PLoS Biology, 4 (10), October 2006.
21. A. Ma'ayan et al. *Formation of Regulatory Patterns During Signal Propagation in a Mammalian Cellular Network*, Science, 309 (5737), 1078-1083, 2005.
22. M. E. J. Newman, S. H. Strogatz and D. J. Watts. *Random graphs with arbitrary degree distributions and their applications*, Phys. Rev. E, 64 (2), 26118-26134, 2001.
23. S. S. Shen-Orr, R. Milo, S. Mangan and U. Alon. *Network motifs in the transcriptional regulation network of Escherichia coli*, Nature Genetics 31, 64-68, 2002.
24. A. Vetta. *Approximating the minimum strongly connected subgraph via a matching lower bound*, 12th ACM-SIAM Symposium on Discrete Algorithms, 417-426, 2001.
25. A. Wagner. *Estimating Coarse Gene Network Structure from Large-Scale Gene Perturbation Data*, Genome Research, 12, 309-315, 2002.