Chapter 16

Inference of Signal Transduction Networks from Double Causal Evidence

Réka Albert, Bhaskar DasGupta, and Eduardo Sontag

Abstract

Here, we present a novel computational method, and related software, to synthesize signal transduction networks from single and double causal evidences. This is a significant and topical problem because there are currently no high-throughput experimental methods for constructing signal transduction networks, and because the understanding of many signaling processes is limited to the knowledge of the signal(s) and of key mediators' positive or negative effects on the whole process. Our software NET-SYNTHESIS is freely downloadable from http://www.cs.uic.edu/~dasgupta/network-synthesis/.

Our methodology serves as an important first step in formalizing the logical substrate of a signal transduction network, allowing biologists to simultaneously synthesize their knowledge and formalize their hypotheses regarding a signal transduction network. Therefore, we expect that our work will appeal to a broad audience of biologists. The novelty of our algorithmic methodology based on nontrivial combinatorial optimization techniques makes it appealing to computational biologists as well.

Key words: Computational biology, Network inference, Signal transduction, Systems biology, Double causal evidence

1. Introduction

Most biological characteristics of a cell involve complex interactions between its numerous constituents such as DNA, RNA, proteins, and small molecules (1). Cells use signaling pathways and regulatory mechanisms to coordinate multiple functions, allowing them to respond to and acclimate to an ever-changing environment. In a signal transduction network (pathway), there is typically an input, perceived by a receptor, followed by a series of elements through which the signal percolates to the output node, which represents the final outcome of the signal transduction process. For a cellular signal transduction pathway not involving alterations in

David Fenyö (ed.), *Computational Biology*, Methods in Molecular Biology, vol. 673,
DOI 10.1007/978-1-60761-842-3_16, © Springer Science+Business Media, LLC 2010

gene expression, elements often consist of proteinaceous receptors, intermediary signaling proteins, metabolites, effector proteins, and a final output that represents the ultimate combined effect of the effector proteins. If the signal transduction process includes regulation of the transcript level of a particular gene, the intermediate signaling elements will also include the gene itself and the transcription factors that regulate it, as well any small RNAs that regulate the transcript's abundance, with the final output being presence or absence of transcript. Genome-wide experimental methods now identify interactions among thousands of proteins (2–5). However, 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.

The experimental evidence about the involvement of specific components in a given signal transduction network frequently belongs to one of these three categories:

(a) *Biochemical evidence*. This type of evidence provides information on enzymatic activity or protein–protein interactions. These are "direct," physical interactions. Examples include:

- Binding of two proteins,
- A transcription factor activating the transcription of a gene, or
- A simple chemical reaction with a single reactant and single product.

(b) *Pharmacological evidence*. This type of experimental evidence is generated by processes in which a chemical is used either to mimic the elimination of a particular component or to exogenously provide a certain component, leading to observed relationships that are not direct interactions but indirect causal relationships most probably resulting from a chain of direct interactions and/or reactions.

(c) *Genetic evidence of differential responses to a stimulus*. Such evidence in a wild-type organism versus a mutant organism implicates the product of the mutated gene in the signal transduction process. This category is a three-component inference as it involves the stimulus, the mutated gene product, and the response. We will call this category as a *double causal inference*.

In this chapter, we describe a method for synthesizing single and double causal information into a consistent network. Our method significantly expands the capability for incorporating indirect (pathway-level) information. Previous methods of synthesizing signal transduction networks only include direct biochemical interactions, and are therefore restricted by the incompleteness of the experimental knowledge on pair-wise interactions. Figure 1 shows a schematic diagram of our overall goal. Mathematical and more technical details about our method are available in our publications (6–9).

A starting point in applying our method involves distilling experimental conclusions into qualitative regulatory relations between cellular components. We differentiate between positive and negative regulation by using the verbs "promote" and "inhibit" and representing them graphically as → and ⊣, respectively (see Fig. 2). Biochemical and pharmacological evidence is represented as a component-to-component relationship, such as "A promotes B," and is incorporated as a directed edge (also called link) from vertex (also called node) A to B (see Fig. 2). Edges corresponding to "known" (documented) direct interactions are marked as "critical." Genetic evidence leads to double

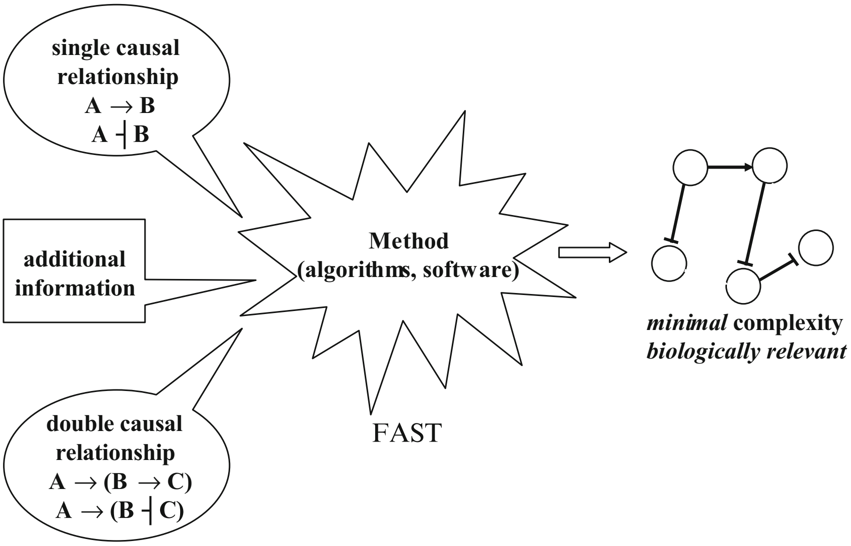


Fig. 1. A schematic diagram of the overall goal of our method.

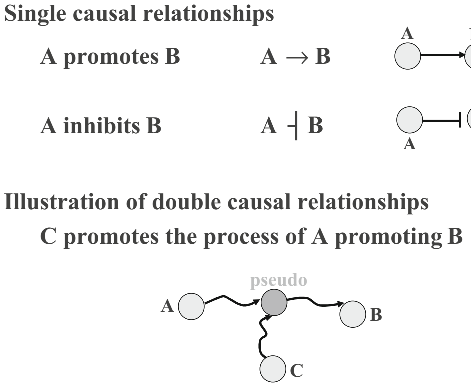

Single causal relationships
A promotes B
A → B
A
B
A inhibits B
A ⊣ B
A
B
Illustration of double causal relationships
C promotes the process of A promoting B
pseudo
A
B
C

Fig. 2. *Direct and double causal interactions*. Illustration of graph–theoretic interpretations of various types of interactions.

causal inferences of the type "C promotes the process through which A promotes B." We assume that a three-node double causal inference corresponds to an intersection of two paths (one path from A to B and another path from C to B) in the interaction network; in other words, we assume that C activates an unknown intermediary (*pseudo*) vertex of the AB path; see Fig. 2 for a pictorial illustration.

The main idea of our method is to find a minimal graph, both in terms of pseudo-vertex numbers and noncritical edge numbers, that is consistent with all reachability relationships between nonpseudo ("real") vertices. A schematic diagram of an overall high-level view of our method is shown in Fig. 3 and a detailed diagram appears in Fig. 4. Two main computational steps involved are: (1) *binary transitive reduction* (BTR) of a resulting graph subject to the constraints that *no* edges flagged as direct are eliminated and (2) *pseudo-vertex collapse* (PVC) subject to the constraints that real vertices are *not* eliminated. In the next two subsections, we discuss these two computational substeps in more detail.

1.1. Pseudo-vertex Collapse

Intuitively, the PVC problem is useful for reducing the pseudo-vertex set to the minimal set that maintains the graph consistent with all double causal experimental observations. Computationally, an exact solution of this problem can be obtained in polynomial time.

The PVC operation is shown schematically in Fig. 5. Mathematically, the PVC computational problem can be defined as follows. Our input is a signal transduction network $G=(V, E)$ with vertex set V and edge set E in which a subset of vertices are pseudo-vertices. For any vertex v, the vertex sets are defined as follows:

$$\text{in }(v)=\{(u,x)\mid u \text{ has a path to } v \text{ of type } x \text{ with } x\in\{\rightarrow$$

$$\text{out }(v)=\{(u,x)\mid u \text{ has a path to } v \text{ of type } x \text{ with } x\in\{\rightarrow$$

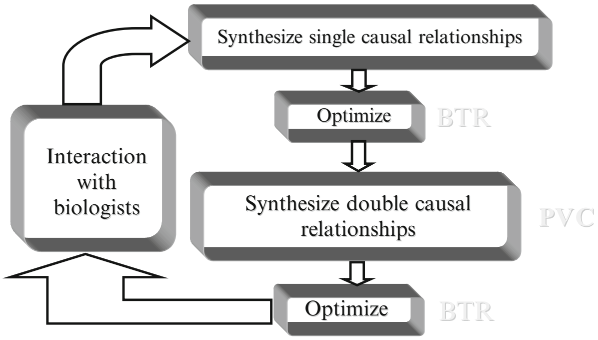

Synthesize single causal relationships
Optimize
BTR
Interaction with biologists
Synthesize double causal relationships
PVC
Optimize
BTR

Fig. 3. *High-level description of the network synthesis process.* PVC and BTR refer to the pseudo-vertex collapse and the binary transitive reduction computational steps, respectively.

1. [**encode single causal relationships**]

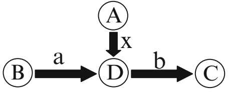
   1.1 Build networks for connections like A → B and A ⊣ B noting each critical edge.

   1.2 Apply BTR

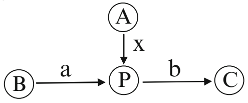
2. [**encode double causal reltionships**]

   2.1 For each double causal relationship of the form A $\overset{x}{\rightarrow}$ (B $\overset{y}{\rightarrow}$ C) with x,y ∈ {0,1}, add new nodes and/or edges as follows:

   - if B $\overset{y}{\rightarrow}$ C ∈ E_{critical} then add A $\overset{x}{\rightarrow}$ (B $\overset{y}{\rightarrow}$ C)
   - if no subgraph of the form (for some node D with b = a+b = y (mod 2))

     then add the subgraph (where P is a new pseudo-node and b = a+b = y (mod 2))

   2.2 Apply PVC

3. [**final reduction**] Apply BTR

Fig. 4. *Algorithmic details of the basic network synthesis procedure* (8). In this diagram, a *right arrow* → labeled by 0 denotes a "promotes" relation and a *right arrow* → labeled by 1 denotes an "inhibits" relation. Similarly, a *right double arrow* ⇒ labeled by 0 denotes a "promotes" *path* and a *right double arrow* ⇒ labeled by 1 denotes an "inhibits" *path*. E_{critical} denotes the set of critical edges. The mathematical notation like $a + b = c$ (mod 2) indicates that $a + b$ has the same remainder as c when divided by 2.

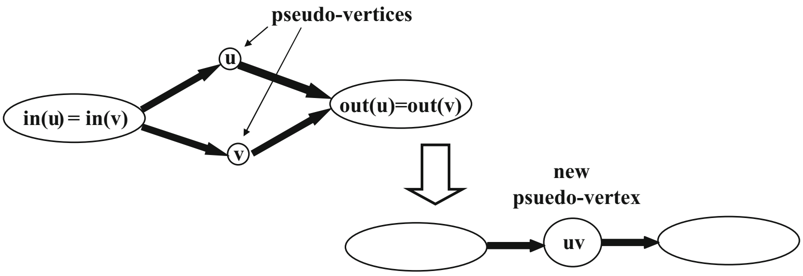


Fig. 5. *Pictorial illustration of a PVC operation*. Repeatedly performing this operation results in a graph with fewer nodes and edges

Collapsing two vertices u and v is *permissible* provided *both* are *not* real vertices, in(u) = in(v) and out(u) = out(v). A PVC operation is as follows: if permissible, *collapse* two vertices u and v to create a new vertex w, make every incoming (respectively, outgoing) edge to (respectively, from) either u or v an incoming (respectively, outgoing) edge from w, remove any parallel edges that may result from the collapse operation and also remove both vertices u and v. A *valid solution* consists of a network $G' = (V', E')$ obtained from G by a sequence of permissible collapse operations; the goal is to *minimize* the number of edges in E'.

1.2. Binary Transitive Reduction

Intuitively, the BTR problem is useful for determining a sparsest graph consistent with a set of experimental observations. Computationally, in contrast to the PVC problem, an exact solution of this problem is *hard*.

The BTR operation is shown schematically in Fig. 6. Mathematically, the BTR computational problem can be defined as follows. Our input is a signal transduction network $G=(V, E)$ with a subset $E_c \subseteq E$ of edges marked as *critical*. A valid solution is a subset of edges E', with $E_c \subseteq E' \subseteq E$, that maintains the same "reachability": u has a path to v in G of nature x ($x \in \{\rightarrow, \dashv\}$) if and only if u has a path to v in $G' = (V, E')$ of the *same nature*. The goal is to *minimize* the size of E'.

The BTR problem is known to be NP-hard as a consequence of the results in (10). A few results were obtained for certain versions of BTR (11, 12) before our work in (6–9), but they were either special cases or biologically more restrictive versions. A special case of the BTR problem, called the *minimum-equivalent-digraph* problem, has been of special interest to computer scientists for a long time with regard to optimizing computer networks with connectivity requirements (13–17) and has also found applications in the context of visualization of social networks (18). Our theoretical results (6) resulted in efficient 2-approximation algorithms for BTR, which has been recently improved further to a 1.5-approximation (19).

The final product of our method led to a custom software package NET-SYNTHESIS (available at http://www.cs.uic.edu/~dasgupta/network-synthesis/) that can be simply downloaded and run in almost any machine with Microsoft Windows as the operating system (for LINUX users, source C/C++ codes for a nongraphic version of the software can be provided on request). The input to NET-SYNTHESIS is a list of relationships among biological components (single causal and double causal)

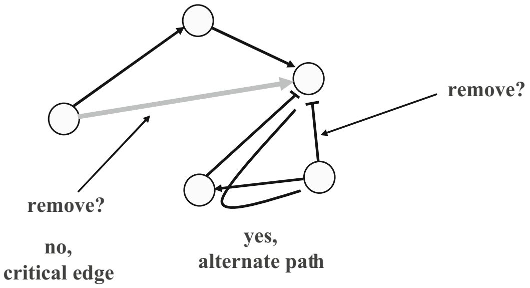


Fig. 6. *Pictorial illustration of a BTR operation.* The *lighter edge* is a critical edge and thus cannot ever be removed. The indicated inhibitory edge can be removed because there is an alternate inhibitory path from the beginning node of the edge to the end node of the edge.

and its output is a network diagram and a text file with the edges of the signal transduction network.

Below is a summary of the standard steps necessary for carrying out the network synthesis and simplification task using NET-SYNTHESIS:

1. Gather the direct interactions, single causal inferences, and double causal inferences regarding your signal transduction network.
2. Read the single inferences into NET-SYNTHESIS to form a graph. Perform BTR on the graph.
3. Integrate the double causal inferences into the graph.
4. Perform PVC.
5. Perform a follow-up round of BTR and vertex collapse until the graph cannot be reduced further.
6. If warranted, simplify the graph further by designating known vertices as pseudo-vertices and performing PVCs.

2. Materials

2.1. Information and Data Sources

Large-scale repositories such as Many Microbe Microarrays (http://m3d.bu.edu/cgi-bin/web/array/index.pl?read=aboutM3D), NASCArrays (http://affymetrix.arabidopsis.info/narrays/experimentbrowse.pl), and Gene Expression Omnibus (http://www.ncbi.nlm.nih.gov/geo/) contain expression information for thousands of genes under tens to hundreds of experimental conditions. In addition, information about differentially expressed genes responding to a combination of two experimental perturbations, e.g., the presence of a signal in normal versus mutant organisms, can be expressed as double causal inferences. Signal transduction pathway repositories such as TRANSPATH (http://www.gene-regulation.com/pub/databases.html#transpath) and protein interaction databases such as the Search Tool for the Retrieval of Interacting Proteins (http://string.embl.de/) contain up to thousands of interactions, a large number of which are not supported by direct physical evidence and thus are best treated as single causal inferences.

2.2. Software

The input to the NET-SYNTHESIS software package is a list of relationships among biological components (single causal and double causal) and its output is a network diagram and a text file with the edges of the signal transduction network. We note that “nodes” and “vertices” are used interchangeably in the software and in this chapter. In the following, we explain a few menu options for NET-SYNTHESIS; a user manual is available at the software's webpage, http://www.cs.uic.edu/~dasgupta/network-synthesis/help.html.

2.2.1. File Menu

- *Read*. Reads an input file from your local directory. After reading, it builds a network for single causal inferences (i.e., edges) only.
- *Write*. Writes the current result to a text file in your local directory.

2.2.2. Action Menu

- *Redundant edges*. Finds out and removes if there are duplicate edges in your file or in the current graph.
- *Add pseudonodes*. Adds the double causal (i.e., three-vertex) inferences in the input file to the network via introducing pseudo-nodes if necessary.
- *Collapse pseudonodes*. Collapses pseudo-nodes using the PVC algorithm.
- *Reduction* (*slower*). Performs BTR on the current network. Recommended for networks of no more than 150 nodes.
- *Reduction* (*faster*). Performs BTR on the current network. Recommended for networks of more than 150 nodes.
- *Collapse degree-2 pseudonodes*. Collapses pseudo-nodes that have a single incoming edge and a single outgoing edge.
- *Randomize before reduction*. The transitive reduction algorithm has steps where ties are broken arbitrarily. If you turn on this action, then such tie-breaking steps will be randomized, thus potentially giving different solutions at different runs of the transitive reduction. This option may be useful if you wanted to check out more than one solution for the transitive reduction step.

2.2.3. View Menu

- *Info*. Shows basic information about the current graph such as the number of vertices and edges.
- *Edge handle*. Displays the edges more visibly (and, hopefully more nicely).
- *Show critical*. Shows critical edges with a different color.

2.2.4. Other Functions

- You can right click on a vertex on the canvas to change the name of that node. This may be especially useful in changing a real node to a pseudo-node or vice versa because the program assumes that nodes whose names start with an asterisk (*) are pseudo-nodes.
- You can right click on the edge handle to change the nature of an edge (e.g., from excitatory to inhibitory or vice versa).

3. Methods

3.1. Gather the Direct Interactions, Single Causal Inferences, and Double Causal Inferences Regarding Your Signal Transduction Network

First, thoroughly read the relevant literature concerning the signal transduction pathway of interest. After reading all available literature on the topic, assess whether sufficient information is on hand such that network synthesis is necessary. For example, if all that is known about a system is that component/process X activates component Y which in turn activates component Z, one can draw a simple linear network and deduce that knockout of Y will eliminate signaling, but a formal analysis is hardly required.

In assessing the literature, the modeler should especially focus on experiments that provide information of the type relevant to network construction. Experiments that identify nodes belonging to a signaling pathway and the relationships between them include: (1) in vivo or in vitro experiments which show that the properties (e.g., activity or subcellular localization) of a protein change upon application of the input signal or upon modulation of components already definitively known to be associated with the input signal; (2) experiments that directly assay a small molecule or metabolite (e.g., imaging of cytosolic Ca^{2+} concentrations) and show that the concentration of that metabolite changes upon application of the input signal or modulation of its associated elements; (3) experiments that demonstrate physical interaction between two nodes, such as protein–protein interaction observed from yeast two-hybrid assays or in vitro or in vivo coimmunoprecipitation; (4) pharmacological experiments which demonstrate that the output of the pathway of interest is altered in the presence of an inhibitory agent that blocks signaling from the candidate intermediary node (e.g., a pharmacological inhibitor of an enzyme or strong buffering of an ionic species); (5) experiments which show that artificial addition of the candidate intermediary node (e.g., exogenous provision of a metabolite) alters the output of the signaling pathway; (6) experiments in which genetic knockout or overexpression of a candidate node is shown to affect the output of the signaling pathway. The first three types of experiments correspond to single causal inferences that will become edges of the network; the third also corresponds to direct interactions that will become critical edges of the network. The fourth to sixth types of experiments correspond to double causal inferences.

The experimental conclusions need to be distilled into two kinds of regulation: positive (usually described by the verbs "promotes," "activates," and "enhances") and negative (usually described by the verbs "inhibits," "reduces," and "deactivates"), and represented graphically as → and ⊣ (see Fig. 2). As the input to NET-SYNTHESIS is simple text files, the graphical symbols are replaced by "→" and "⊣." Component-to-component relationships are represented such as “A→B.” Double causal inferences are of the type “C promotes the process through which A activates 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-catalyzed 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 represent the double causal inference such as “C→(A→B).”

Note that some choices may have to be made in distilling the relationships, especially in the case where there are two conflicting reports in the literature. For example, imagine that in one report it is stated that proteins X and Y do not physically interact based on yeast two-hybrid analysis, while in a second report, it is described that proteins X and Y do interact, based on coimmunoprecipitation from the native tissue. The modeler will need to decide which information is more reliable, and proceed accordingly. Such aspects dictate that human intervention will inevitably be an important component of the literature curation process, even as automated text search engines such as GENIES (20–22) grow in sophistication.

We will illustrate the five analysis steps following the data-gathering phase on a sample collection of single and double causal inferences. This sample is a small subset of the evidence gathered for the signal transduction network responsible for abscicic acid-induced closure of plant stomata (23). The vertices correspond to the signal, denoted “ABA,” the output, denoted “Closure,” and seven mediators of ABA-induced closure, the heterotrimeric G protein α subunit (GPA1), the small molecules NO and phosphatidic acid (PA), the enzymes Phospholipase C (PLC) and Phospholipase D (PLD), K^+ efflux through slowly activating outwardly rectifying K^+ channels at the plasma membrane (KOUT). The compilation includes nine single causal inferences, two of which correspond to direct interactions and two double causal inferences.

The input to NET-SYNTHESIS is given as follows:

ABA ⊣ NO
ABA → PLD
ABA → GPA1
ABA → PLC
GPA1 → PLD Y
PLD → PA
NO ⊣ KOUT
KOUT → Closure Y
PA → Closure
PLC → (ABA → KOUT)
PLD → (ABA → Closure)

The single inferences need to precede the double inferences. The direct interactions are marked by the letter "Y" following the component-to-component relationship.

3.2. Read the Single Inferences into NET-SYNTHESIS to Form a Graph. Perform BTR on the Graph

To use NET-SYNTHESIS on this example, it needs to be saved into a text file, e.g., "example.txt." After starting NET-SYNTHESIS, select the command "Read" from the File menu, and open the input file "example.txt." This will display the vertices and edges corresponding to the single inferences. You can move the nodes by clicking and holding your left mouse button on them. Try to arrange the nodes so the edges do not cross each other. Note that the small circles correspond to edge handles (if you have the option of edge handles chosen in the View menu) which can also be moved to make the graph clearer. Clicking on Info in the View menu indicates that currently the network contains eight vertices and nine edges. To perform BTR, select "Reduction (slower)" from the Action menu. This reduction method is the better choice for networks smaller than 150 vertices. A pop-up window will indicate that one edge was removed. Indeed, the edge from ABA to PLD was superfluous as it did not indicate a direct interaction and had no effect on the reachability of any node in the network.

3.3. Integrate the Double Causal Inferences into the Graph

To read in the double causal inferences, select "Add pseudonodes" from the Action menu. The pop-up window will indicate that two pseudo-vertices and six edges were added to account for the two double causal inferences. Rearrange the nodes to see what is new. Indeed, the PLD→(ABA→Closure) inference created a new pseudo-vertex, indicated by a circle with a star in it, and three new edges, one from PLD to the pseudo-vertex, one from ABA to the pseudo-node, and one from the pseudo-node to Closure. The second inference was incorporated in a similar manner. The newly added edges created new redundancies in the network. For example, the newly introduced pseudo-node connecting ABA and PLD to Closure has the same in and out reachability as the node PA, i.e., it can be reached from ABA, GPA1, and PLD and it can reach Closure. Therefore, the pseudo-vertex is a candidate for PVC.

3.4. Perform PVC

To perform PVC, select "Collapse pseudonodes" from the Action menu. The pop-up window will indicate that one pseudo-node was removed. An inspection of the network will tell you that indeed the pseudo-vertex indicated above was collapsed with the real node PA. This decreased the number of vertices by one and the number of edges by two. As an effect of the collapse, ABA is now directly connected to PA in addition to being connected by the chain GPA1–PLD. The ABA→PA edge is redundant with the path, thus it is a candidate for BTR. In addition, an edge among the three that connect ABA, PLC, and the remaining pseudo-vertex is also redundant. Thus, we should try to simplify the network further.

3.5. Perform a Follow-up Round of BTR and Vertex Collapse Until the Graph Cannot be Reduced Further

Select “Reduction (slower)” again and you will see that indeed the two edges have been removed. The remaining pseudo-vertex is now simply a mediator between PLC and KOUT. But because its existence does not add any further information, it should be removed. You can do that by selecting “Collapse degree-2 pseudonodes” from the action menu. Now the network has eight vertices and nine edges. Select “Reduction (slower)” to make sure no more reduction is possible.

3.6. If Warranted, Simplify the Graph Further by Designating Known Vertices as Pseudo-vertices and Performing PVC

In the example above, we succeeded in integrating single and double causal inferences into a signal transduction network whose nodes are all known (i.e., they are not pseudo-nodes). For a real situation, as opposed to an illustrative example, the resulting network can be quite large and complex. In cases when some of the nodes are clearly more documented, more important, or more interesting than others, it may be beneficial to focus on the reachability among these more important nodes and disregard the others without explicitly removing them. One can do this by designating the less important nodes as pseudo-nodes and then simplifying the network by using PVC and BTR.

Let us designate the node NO as a pseudo-node. We can do this by right-clicking on the node, prepending a `*` to the node name that appears in a pop-up window, and press Enter. The node will now become a pseudo-node, indicated by the fact that the symbol corresponding to the node becomes a small circle with a star in the middle. Selecting “Collapse degree-2 pseudonodes” will remove the pseudo-node and connect ABA and KOUT with a positive edge. This is because a path with an even number of negative edges is positive. The new edge is redundant with the path going through PLC and “Reduction (slower)” will delete it.

4. Conclusion

We have previously successfully illustrated the usefulness of our software by applying it to synthesize an improved version of a previously published signal transduction network (7, 23) and by using it to simplify a novel network corresponding to activation-induced cell death of T cells in large granular lymphocyte leukemia (7, 24). It is our hope that this method, in assistance with interactive human intervention as discussed before, will be useful in the future in synthesizing and analyzing networks in a broader context.

References

1. B. Alberts. Molecular Biology of the Cell. Garland Publishing: New York, 1994.
2. T. I. Lee, N. J. Rinaldi et al. Transcriptional regulatory networks in *Saccharomyces cerevisiae*, Science, 298, 799–804, 2002.
3. L. Giot, J. S. Bader et al. A protein interaction map of *Drosophila melanogaster*, Science, 302, 1727–1736, 2003.
4. J. D. Han, N. Bertin et al. Evidence for dynamically organized modularity in the yeast protein–protein interaction network, Nature, 430, 88–93, 2004.
5. S. Li, C. M. Armstrong et al. A map of the interactome network of the metazoan *C. elegans*, Science, 303, 540–543, 2004.
6. R. Albert, B. DasGupta et al. Inferring (biological) signal transduction networks via transitive reductions of directed graphs, Algorithmica, 51 (2), 129–159, 2008.
7. S. Kachalo, R. Zhang et al. NET-SYNTHESIS: A software for synthesis, inference and simplification of signal transduction networks, Bioinformatics, 24 (2), 293–295, 2008.
8. R. Albert, B. DasGupta et al. A novel method for signal transduction network inference from indirect experimental evidence, Journal of Computational Biology, 14 (7), 927–949, 2007.
9. R. Albert, B. DasGupta et al. A novel method for signal transduction network inference from indirect experimental evidence, in 7th Workshop on Algorithms in Bioinformatics, R. Giancarlo and S. Hannenhalli (Eds.), LNBI 4645, Springer: Berlin/Heidelberg, 407–419, 2007.
10. 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.
11. A. Wagner. Estimating coarse gene network structure from large-scale gene perturbation data, Genome Research, 12, 309–315, 2002.
12. T. Chen, V. Filkov and S. Skiena, Identifying gene regulatory networks from experimental data, in 3rd Annual International Conference on Computational Molecular Biology, 94–103, 1999.
13. S. Khuller, B. Raghavachari and N. Young. Approximating the minimum equivalent digraph, SIAM Journal of Computing, 24 (4), 859–872, 1995.
14. S. Khuller, B. Raghavachari and N. Young. On strongly connected digraphs with bounded cycle length, Discrete Applied Mathematics, 69 (3), 281–289, 1996.
15. S. Khuller, B. Raghavachari and A. Zhu. A uniform framework for approximating weighted connectivity problems, in 19th Annual ACM-SIAM Symposium on Discrete Algorithms, 937–938, 1999.
16. G. N. Frederickson and J. JàJà. Approximation algorithms for several graph augmentation problems, SIAM Journal of Computing, 10 (2), 270–283, 1981.
17. A. Vetta. Approximating the minimum strongly connected subgraph via a matching lower bound, in 12th ACM-SIAM Symposium on Discrete Algorithms, 417–426, 2001.
18. V. Dubois and C. Bothorel. Transitive reduction for social network analysis and visualization, in IEEE/WIC/ACM International Conference on Web Intelligence, 128–131, 2008.
19. P. Berman, B. DasGupta and M. Karpinski. Approximating Transitivity in Directed Networks, arXiv:0809.0188v1 (available online at http://arxiv.org/abs/0809.0188v1).
20. C. Friedman, P. Kra, H. Yu, M. Krauthammer and A. Rzhetsky. GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles, Bioinformatics, 17 (Suppl 1), S74–S82, 2001.
21. E. M. Marcotte, I. Xenarios and D. Eisenberg. Mining literature for protein-protein interactions. Bioinformatics, 17 (4), 359–363, 2001.
22. L. J. Jensen, J. Saric and P. Bork. Literature mining for the biologist: from information retrieval to biological discovery, Nature Reviews Genetics, 7 (2), 119–129, 2006.
23. 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), e312, 2006.
24. R. Zhang, M. V. Shah, J. Yang et al. Network model of survival signaling in large granular lymphocyte leukemia. Proceedings of the National Academy of Sciences of the United States of America, 105 (42), 16308–16313, 2008.