JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 15, Number 7, 2008
© Mary Ann Liebert, Inc.
Pp. 829–844
DOI: 10.1089/cmb.2007.0139

Network Legos: Building Blocks of Cellular Wiring Diagrams

T. M. MURALI and CORBAN G. RIVERA

ABSTRACT

Publicly available datasets provide detailed and large-scale information on multiple types of molecular interaction networks in a number of model organisms. The wiring diagrams composed of these interaction networks capture a static view of cellular state. An important challenge in systems biology is obtaining a dynamic perspective on these networks by integrating them with gene expression measurements taken under multiple conditions. We present a top-down computational approach to identify building blocks of molecular interaction networks by: (i) integrating gene expression measurements for a particular disease state (e.g., leukemia) or experimental condition (e.g., treatment with growth serum) with molecular interactions to reveal an *active network*, which is the network of interactions active in the cell in that disease state or condition; and (ii) systematically combining active networks computed for different experimental conditions using set-theoretic formulae to reveal *network legos*, which are modules of coherently interacting genes and gene products in the wiring diagram. We propose efficient methods to compute active networks, systematically mine candidate legos, assess the statistical significance of these candidates, arrange them in a directed acyclic graph (DAG), and exploit the structure of the DAG to identify true network legos. We describe methods to assess the stability of our computations to changes in the input and to recover active networks by composing network legos. We analyze two human datasets using our method. A comparison of three leukemias demonstrates how a biologist can use our system to identify specific differences between these diseases. A larger-scale analysis of 13 distinct stresses illustrates our ability to compute the building blocks of the interaction networks activated in response to these stresses. Source code implementing our algorithms is available under version 2 of the GNU General Public License at *http://bioinformatics.cs.vt.edu/∼murali/software/network-lego*.

Key words: molecular interaction networks, response networks, gene modules, bioclustering.

1. INTRODUCTION

THE FUNCTIONING OF A LIVING CELL is governed by an intricate network of interactions among different types of molecules. These interactions transduce external signals, control gene expression

Department of Computer Science, Virginia Polytechnic Institute and State University, Blacksburg, VA.

and protein localization, modify protein activities, and drive biochemical reactions. Recent experimental advances and literature curation have provided us with large-scale publicly available datasets of molecular interactions, especially for model organisms such as *S. cerevisiae*, *C. elegans*, and *D. melanogaster*; for pathogens such as *P. falciparum*; and for *H. sapiens* itself. Taken together, the known molecular interactions for an organism constitute its *wiring diagram*, a graph where each node is a molecule and each edge is an interaction between two molecules. Existing wiring diagrams are tremendous resources for systems biology, since they integrate information on multiple types of molecular interactions obtained from a variety of different experimental sources. However, their potential impact is diluted since they typically represent the *universe* of interactions that take place across diverse contexts in the cell. Consequently, a fundamental challenge in molecular systems biology is automatically identifying building blocks of cellular wiring diagrams, where each building block is a network of interactions that is associated with a set of experimental conditions in which the building block is activated.

In this paper, we present a top-down computational approach that identifies building blocks of molecular interaction networks by:

(i) Integrating gene expression measurements for a particular disease state (e.g., leukemia) or experimental condition (e.g., treatment with growth serum) with molecular interactions to reveal an *active network*, which is the network of interactions active in the cell in that disease state or condition and
(ii) Systematically combining active networks computed for different experimental conditions using set-theoretic formulae to reveal *network legos*, which are functional modules of coherently interacting genes and gene products in the wiring diagram. These network legos are potential building blocks of the wiring diagram, since we can express each active network as a composition of network legos.

Given a wiring diagram and the transcriptional measurements for a particular condition, we use the gene expression data to induce edge weights in the wiring diagram. We find dense subgraphs (Charikar, 2000) in this weighted graph to compute the active network for that condition. Given the active networks for a number of different conditions, we first represent the active networks in an appropriately defined binary matrix and compute closed biclusters (Agrawal and Srikant, 1994; Zaki and Hsiao, 2002) in the matrix. Each bicluster simultaneously represents a set-theoretic combination of particular active networks and a subgraph of the wiring diagram; we call such a subgraph a "network block." We exploit the subset structure between blocks to arrange them in a directed acyclic graph (DAG). When the number of active networks is large, we may compute a very large number of highly similar blocks. Not all these blocks are likely to be network legos. We assess the statistical significance of each block by simulation and identify those that are maximally significant (i.e., more significant than any descendant or an ancestor in the DAG). We deem these blocks to be network legos.

We develop two measures to assess the quality of the network legos we compute. *Stability* measures to what degree we can recompute the same legos when we remove each active network in turn from the input. *Recoverability* measures to what extent we recoup the original active networks when we combine network legos. These two notions test two different aspects of network lego computation. Considering active networks to be the inputs and network legos to be the outputs, stability measures how much the outputs change when we perturb the inputs by removing one of the inputs at a time. In contrast, recoverability asks whether we can reclaim the inputs by combining the outputs; thus recoverability is a measure of how well the network legos serve as building blocks. To assess the biological content of network legos, we measure the functional enrichment of the genes and interactions that belong to a network lego. For each function, we track its degree of enrichment in the DAG to highlight differences among the network legos. For each network lego, we also ask if any functions are enriched only in that network lego and correlate such functions with the expression patterns of the genes in that network lego.

We demonstrate two ways in which a biologist can use our system. In the first, our system allows the systematic comparison of responses to a small number of different conditions, diseases, or perturbations tested in the same lab. The comparison of three leukemias (ALL, AML, and MLL) (Armstrong et al., 2002) we discuss in Section 4.1 is such an application. Using our method, we show that the activation of the Kit receptor pathway is a hallmark of AML but not of ALL and MLL; thus, the activation of this pathway distinguishes AML from the other two leukemias. In the second, a biologist can analyze a specific condition of interest in the context of a large compendium of other conditions, compute the building blocks of the networks activated in these conditions, and ask how the building blocks compose the active network for the specific condition of interest to the biologist. In Section 4.2, we apply our approach to a collection of 178 arrays measuring the gene expression responses of HeLa cells and primary human lung fibroblasts to 13 distinct stresses, including cell cycle arrest, heat shock, endoplasmic reticulum stress, oxidative stress, and crowding (Murray et al., 2004). Our method computes 143 network legos. We carefully examine the compositions of these network legos to demonstrate that they are true building blocks of the active networks for these 13 stresses. We use leave-one-out validation to prove that our algorithm to construct network legos is stable: when we remove each active network and recompute network legos, we are able to recompute most network legos with at least 95% fidelity. We also demonstrate that we can recover active networks with almost perfect accuracy by composing network legos. Further analysis of the network legos reveals that the active networks corresponding to cell cycle arrest contain interactions that are quite distinct from the network of interactions activated by the other stresses. When we remove the two cell cycle arrest datasets, we compute only 15 network legos. Of the 11 remaining active networks, we recover five with complete accuracy and one with 99.9% accuracy. We recover the other five active networks with accuracies of 71–92%. Functional enrichment analysis of these network legos shows that the only lego enriched in genes controlling and participating in the cell cycle is one that distinguishes the reaction of fibroblasts to endoplasmic reticulum stress from the other stresses. Taken together, these statistics indicate that the network legos we detect are indeed building blocks of the networks activated in response to the stresses studied by Murray et al. (2004) and that the network legos yield biologically-useful insights into the similarities and differences between the two cell types.

The success of our approach stems from a number of factors. First, unlike other approaches discussed in Section 2 that simultaneously integrate multiple gene expression datasets in the context of the network scaffold, we compute individual active networks for each dataset and associate the active network with the corresponding disease or perturbation. This approach allows us to explicitly compare and contrast different conditions. Second, we treat interactions (rather than genes or proteins) as the elementary objects of our analysis. Therefore, different network legos may share genes, allowing for the situation when a gene participates in multiple biological processes and is activated differently in these processes. Finally, we develop a simple but effective method to assess the statistical significance of a network lego and to recursively weed out sub-networks that masquerade as building blocks but contain true network legos. Taken together, network legos and the accompanying set-theoretic formulae provide a dynamic and multi-dimensional view of cell circuitry obtained by integrating molecular interaction networks, gene expression data, and descriptions of experimental conditions.

2. RELATED RESEARCH

A number of approaches, recently surveyed by Joyce and Palsson (2006) and by Sharan and Ideker (2006) have been developed to integrate diverse types of biological data and "mine" these datasets to find groups of molecules (usually genes and/or proteins) that act in concert to perform a specific biological task. We briefly discuss these approaches and place our work in context.

2.1. *Active networks*

A number of techniques overlay gene expression data for a condition on the wiring diagram to compute the active network for that condition (Haugen et al., 2004; Ideker et al., 2002; Reiss et al., 2005; Segal et al., 2003). Ulitsky and Shamir (2007) use the wiring diagram as a constraint network and compute dense subgraphs in the gene co-expression network as long as the genes in the dense subgraphs induce a connected subgraph of the wiring diagram. These methods typically focus on a single condition of interest.

2.2. *Biclustering*

Biclustering has emerged as a powerful algorithmic tool, especially for analyzing gene expression data. A bicluster in a gene expression dataset is a subset of genes and a subset of conditions with the property that the selected genes are co-expressed in the selected conditions; these genes may not have any coherent patterns of expression in the other conditions in the dataset. Since a bicluster includes only a subset of genes and samples, it models condition-specific patterns of co-expression. Biclustering algorithms allow a gene or a sample to participate in multiple biclusters, each of which may correspond to a different pathway. A number of different methods have emerged for computing biclusters in gene expression data; two papers provide excellent surveys (Madeira and Oliveira, 2004; Tanay et al., 2006). Two methods have used biclustering to integrate analysis of heterogeneous genome-wide data (Bonneau et al., 2006; Tanay et al., 2004).

2.3. *Itemset and graph mining*

The algorithm we propose for computing network legos in Section 3.3 uses well-studied approaches in data mining for computing itemsets (Agrawal and Srikant, 1994; Ganter and Wille, 1997; Zaki and Hsiao, 2002). Recent work of frequent graph mining (Hu et al., 2005; Koyuturk et al., 2006; Yan and Han, 2003) takes a set $\mathcal{G}$ of labeled graphs as input and finds all connected and/or dense graphs that occur frequently as subgraphs of graphs in $\mathcal{G}$. Our computation of network legos bears some resemblance to these methods, except that our goal is to find frequently-occurring subgraphs that we can use to reconstruct each graph in $\mathcal{G}$ as well as possible.

2.4. *Graph clustering and decomposition*

Graph clustering, or automatic decomposition of a network into modules or communities, has a rich history, with many problem formulations (Bansal et al., 2004; Chung, 1997; Ng et al., 2002; Radicchi et al., 2004), techniques (Dhillon et al., 2007; Drineas et al., 2004; Kannan et al., 2004; Newman, 2006; White and Smyth, 2005), and applications in diverse domains (Dunn et al., 2005; Koyuturk et al., 2006; Enright et al., 2002; Sharan and Shamir, 2000; Shi and Malik, 2000). The top-down data-driven construction of network legos complements the bottom-up assembly of network motifs (Grochow and Kellis, 2007; Milo et al., 2002; Yeger-Lotem et al., 2004), motif super-families (Milo et al., 2004), and thematic maps (Zhang et al., 2005). These approaches typically operate by computing the number of subgraphs in the wiring diagram isomorphic to a query graph Q, and assembling frequently occurring subgraphs into modules.

2.5. *Knowledge-based approaches*

A few methods have used pre-defined gene sets (e.g., based on functional annotations or literature curation) to discriminate among or classify diseases and tissues (Barry et al., 2005; Edelman et al., 2006; Guo et al., 2005; Huang et al., 2006; Levine et al., 2006; Subramanian et al., 2005). Unlike these techniques, our methods specifically take into account the structure of the interactions between the genes in the wiring diagram.

2.6. *Compendium-based approaches*

Many methods have computed gene modules by integrating gene expression data across multiple cellular conditions; they analyze large compendia of such data to reveal similarities and differences between organisms (Bergmann et al., 2003; Stuart et al., 2003), predict functional links and annotations (Hu et al., 2005; Huttenhower et al., 2006; Lee et al., 2004), reconstruct regulatory networks (Bar-Joseph et al., 2003; Zhou et al., 2005) and networks activated in diseases (Basso et al., 2005), zero in on biomarkers for diseases (Rhodes et al., 2004), compute pathway-specific networks that include query genes input by a biologist (Myers et al., 2005), and identify the gene products and associated pathways that a drug compound targets (di Bernardo et al., 2005). A feature common to most of these approaches is that they find patterns woven by genes that share similar expression across the *entire* compendium.

2.7. *Context-specific approaches*

A few recently published methods integrate the wiring diagram with gene expression compendia to detect similarities and differences between conditions, for example, to dissect under what conditions hubs bind their partners (Han et al., 2004) and to obtain insights into changes in topological properties of the wiring diagram across different conditions (Luscombe et al., 2004). Recently, two powerful methods have emerged for comparing gene expression measurements for multiple conditions. Segal et al. (2004) analyze expression profiles in different tumors to compute modules, sets of genes that act in concert to carry out a specific function. They characterize gene-expression profiles in specific (sets of) tumors as a combination of activated and deactivated modules. Tanay et al. (2005) integrate a diverse collection of datasets into a bipartite graph representing connections between genes and gene properties. Their modules are statistically-significant biclusters (Tanay et al., 2002) in this graph. They represent a target gene expression dataset as a bipartite graph and determine which already-computed modules respond in the target dataset. Our approach differs from theirs in two respects. First, we represent differences and similarities between multiple conditions explicitly as a set theoretic formula involving the interaction network activated in each condition. Second, when we analyze a compendium of gene expression datasets, we exploit the subset structure between these formulae to detect network legos, statistically significant building blocks of these active networks.

3. ALGORITHMS

We describe the main computational ingredients of our approach in stages. We first define useful terminology. Next, we present our method to integrate a cellular wiring diagram with the gene expression data for a single condition to compute the active network for that condition. Third, we describe how we combine active networks for different conditions to form blocks. Fourth, we discuss how we compute the statistical significance of blocks, arrange them in a DAG, and exploit the DAG to identify network legos, which are the most statistically-significant blocks in the DAG. Finally, we present our methods to measure the stability of network legos and assess how well we can recover active networks from the network legos.

3.1. Definitions

We denote the wiring diagram of molecular interactions for an organism by W; each node of W is a gene (or gene product) and each edge represents an interaction. Let G be the set of genes in W. The gene expression dataset for a condition c consists of a set of samples S_c, each with an expression value for each gene in G; we denote by g_c the vector of values for gene g in the condition c. Our method takes as input the wiring diagram W for an organism and a compendium of gene expression datasets, each for a different condition.

Active networks. Given a gene expression dataset for a condition c, we say that a gene *responds in c* if the expression values of the gene vary by more than an input threshold. Let g and h be two genes that respond in c and let $e = (g, h)$ be an interaction in W. We say that e is *active in c* if g_c and h_c are correlated to a statistically-significant extent. Let the weight of interaction e be this degree of correlation. We define the *active network A_c in c* to be the subgraph of W with maximum density, where we define the density of a graph as the total weight of its edges divided by the number of nodes in it. We describe the details of how we detect responding genes, active interactions, and active networks in Section 3.2.

Network blocks. Let $\mathcal{A}$ denote a set of active networks, one for each of the conditions in the input compendium. We define a *block* to be a triple $(G, \mathcal{P}, \mathcal{N})$, where G is a subgraph of W, $\mathcal{P}$ and $\mathcal{N}$ are disjoint subsets of $\mathcal{A}$, and $\mathcal{P} \neq \emptyset$ such that

$$G = \Bigg(\bigcap_{P\in\mathcal{P}} P\Bigg) \bigcap \Bigg(\bigcap_{N\in\mathcal{N}} (W - N)\Bigg) = \Bigg(\bigcap_{P\in\mathcal{P}} P\Bigg) - \Bigg(\bigcup_{N\in\mathcal{N}} N\Bigg),$$

where "$\cap$," "$-$," and "$\cup$" respectively denote the intersection, difference, and union of the edge sets of two graphs, and

1. $\mathcal{P}$ is maximal, i.e., there is no active network $P \in \mathcal{A} - \mathcal{P}$ such that $G \subseteq P$, and
2. $\mathcal{N}$ is maximal, i.e., there is no active network $N \in \mathcal{A} - \mathcal{N}$ such that $G \cap N = \emptyset$.

In other words, we can form G by taking the intersection of all the active networks in $\mathcal{P}$ and removing any edge that appears in any of the active networks in $\mathcal{N}$. We require that $\mathcal{P}$ contain at least one active network so that G is not formed solely by the intersection of the networks in $\mathcal{N}$; such a block is unlikely to be biologically interesting. We also require that $\mathcal{P}$ and $\mathcal{N}$ be disjoint so that G is not the empty graph. Requiring $\mathcal{P}$ and $\mathcal{N}$ to be maximal ensures that we include all the relevant active networks in the construction of G. These criteria imply that it is enough to specify $\mathcal{P}$ and $\mathcal{N}$ to compute G uniquely;

we include G in the notation for a block for convenience and drop $\mathcal{P}$ and $\mathcal{N}$ when they are understood from the context. We refer to $\left(\bigcap_{P\in\mathcal{P}} P\right)\bigcap\left(\bigcap_{N\in\mathcal{N}}(W - N)\right)$ as the *formula* for the block. Figure 1a displays three toy active networks and Figure 1b shows examples of three blocks formed from these active networks.

Network legos. We now describe how we identify network legos in a set $\mathcal{B}$ of blocks. We note that n active networks can compose at most $3^n - 2^n$ blocks. Given two distinct blocks $(G_1, \mathcal{P}_1, \mathcal{N}_1)$ and $(G_2, \mathcal{P}_2, \mathcal{N}_2)$ in $\mathcal{B}$, we say that $G_1 \prec G_2$ if

(i) $\mathcal{P}_1 \subseteq \mathcal{P}_2$ and $\mathcal{N}_1 \subseteq \mathcal{N}_2$ or
(ii) $\mathcal{P}_1 \subseteq \mathcal{N}_2$ and $\mathcal{N}_1 \subseteq \mathcal{P}_2$.

Further, we say that $G_1 < G_2$ if there is no block $G_3 \in \mathcal{B}$ such that $G_1 \prec G_3 \prec G_2$. The operators $<$ and $\prec$ represent partial orders between blocks, with $\prec$ being the transitive closure of $<$. Given a set $\mathcal{B}$ of blocks, let $\mathcal{D}_\mathcal{B}$ denote the directed acyclic graph representing the partial order $<$: each node in $\mathcal{D}_\mathcal{B}$ is a block in $\mathcal{B}$ and an edge connects two blocks related by $<$. For a block G, let $\sigma_G \in [0, 1]$ denote the statistical significance of G. We describe a method to compute this value in Section 3.4. We define a *network lego* to be a block $(G, \mathcal{P}, \mathcal{N}) \in \mathcal{B}$ such that $\sigma_G < \sigma_H$, for every $H \in \mathcal{B}$ where $G \prec H$ or $H \prec G$. In other words, $(G, \mathcal{P}, \mathcal{N})$ is a network lego if it is more statistically significant that blocks formed by combining any subset of $\mathcal{P}$ and $\mathcal{N}$ or by combining any superset of $\mathcal{P}$ and $\mathcal{N}$. In this sense, we claim that G is a building block of the active networks in $\mathcal{A}$.

3.2. Computing the active network for a single condition

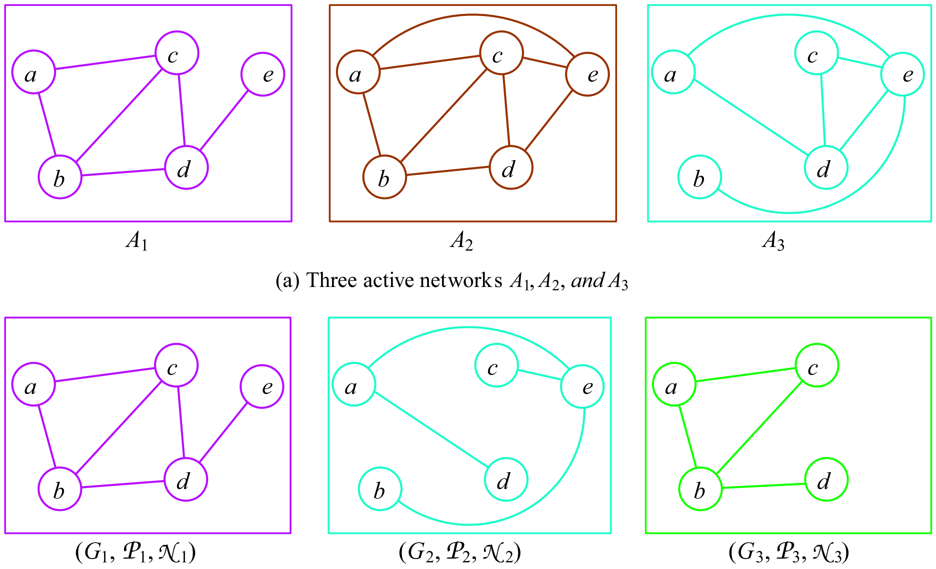
Given a gene expression dataset for a condition c, we compute its active network A_c using the following steps:

1. We use a variational filter to remove all genes whose expression profiles have a small dynamic range from W. Specifically, we log-transform and zero-centre each gene's expression values. We discard a gene and all its interactions in the wiring diagram W if all the transformed expression values of the gene lie between -1 and 1 (Segal et al., 2004). We deem the remaining genes to have responded in the condition.
2. To each interaction $e = (g, h)$ remaining in W, we assign a weight equal to the absolute value of the Pearson's correlation coefficient of g_c and h_c, reasoning that this weight indicates how "active" the interaction is in the experimental condition. We discard edges whose weights are not statistically significant by using the following procedure: (i) We construct 50 random versions of the gene expression dataset by permuting each gene's expression values independently. (ii) For each random dataset, we compute a histogram of the absolute value of the Pearson's correlation coefficient of the expression profiles of all pairs of genes. (iii) We average these 50 histograms and keep only those interactions in W whose edge weights are significant at the 0.01 level. Let W_c be the resulting weighted interaction network.
3. We compute A_c using a greedy algorithm (Charikar, 2000). We define the *weight* of a vertex $v \in W_c$ to be the total weight of the edges incident on v. We repeatedly delete the node of smallest weight in W_c. After each deletion, we update the weights of the neighbours (before deletion) of this node and record the density of the remaining network. We set A_c to be most dense of all the networks so generated.

Remarks. It is possible to find the subgraph of largest density using linear programming or parametric network flows (Charikar, 2000). The greedy algorithm described above finds a subgraph that is at least half as dense as the most dense subgraph. In practice, we embed the greedy algorithm in the following heuristic: we repeatedly apply this approximation algorithm, remove the edges of the subgraph it computes, and re-invoke the algorithm on the remaining graph until the density of the remaining graph is less than the density of W_c. We deem the union of the computed dense subgraphs to be the active network A_c.

3.3. Computing blocks in a set of active networks

Given a set $\mathcal{A}$ of active networks, we reduce the problem of computing blocks defined by the active networks in $\mathcal{A}$ to the problem of computing closed biclusters in a binary matrix (Agrawal and Srikant, 1994; Zaki and Hsiao, 2002). Consider a binary matrix M where each column corresponds to an interaction


a
b
c
d
e
A_1
A_2
A_3
$(G_1, \mathcal{P}_1, \mathcal{N}_1)$
$(G_2, \mathcal{P}_2, \mathcal{N}_2)$
$(G_3, \mathcal{P}_3, \mathcal{N}_3)$

(a) Three active networks $A_1, A_2, and A_3$

(b) Examples of three blocks formed by combining the active networks A_1, A_2, and A_3

FIG. 1. Examples of active networks and blocks. In the block $(G_1, \mathcal{P}_1, \mathcal{N}_1)$, $\mathcal{P}_1 = \{A_1, A_2\}$, $\mathcal{N}_1 = \emptyset$, and $G_1 = A_1 \cap A_2$; in the block $(G_2, \mathcal{P}_2, \mathcal{N}_2)$, $\mathcal{P}_2 = \{A_3\}$, $\mathcal{N}_2 = \{A_2\}$, and $G_2 = A_3 - A_1$; and in the block $(G_3, \mathcal{P}_3, \mathcal{N}_3)$, $\mathcal{P} = \{A_1, A_2\}$, $\mathcal{N} = \{A_3\}$, and $G_3 = A_1 \cap A_2 - A_3$. Therefore, we have the following relations: $G_1 < G_3$ and $G_2 < G_3$. (See this paper online for Fig. 1 in color.)

in W. The matrix M contains two rows for each active network $A \in \mathcal{A}$: the *positive* row corresponds to the interactions in A and the *negative* row to the interactions in $W - A$. In the positive row corresponding to A, we set a cell to be one if and only if the corresponding interaction belongs to A; this cell is zero in the negative row for A. Thus, M is a qualitative representation of which interactions are present in each active network and which are present in its complement.

In a binary matrix such as M, a *bicluster* (R, C) is a subset R of rows and a subset C of columns such that the sub-matrix spanned by these rows and columns only contains ones. A *closed* bicluster is a bicluster with the property that each row (respectively, column) not in the bicluster contains a zero in at least one column (respectively, row) in the bicluster. Therefore, it is not possible to add a row or a column to a closed bicluster without introducing a zero into the corresponding sub-matrix. We can partition R into two subsets R_P and R_N where R_P (respectively, R_N) consists of all the positive (respectively, negative) rows in R. There is a natural one-to-one mapping from a closed bicluster (R, C) to a block $(G, \mathcal{P}, \mathcal{N})$:

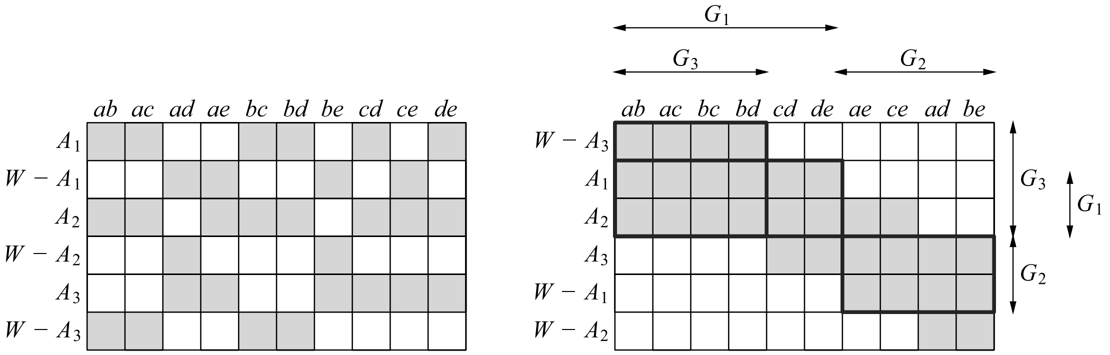
1. G is the subgraph of W induced by the interactions corresponding to the columns in C;
2. $\mathcal{P}$ is the set of active networks corresponding to the rows in R_P; and
3. $\mathcal{N}$ is the set of active networks corresponding to the rows in R_N.

Requiring a bicluster to be closed is equivalent to ensuring that $\mathcal{P}$ and $\mathcal{N}$ are maximal and that C contains all the interactions in G. Figure 2a displays the matrix corresponding to the three active networks in the example in Figure 1a, while Figure 2b displays a layout of all the biclusters in this matrix and highlights the three biclusters corresponding to the three blocks in Figure 1b.

Before describing our algorithm, we define one more concept. Given a set R of rows in M, we define the *closure* of R to be the closed bicluster (R^*, C^*), where C^* is the set of columns that contain ones in all the rows in R and $R^* \supseteq R$ is the set of rows that contain ones in all the columns in C. Given R, we can compute its closure by two scans over M.

To construct closed biclusters and the resulting set $\mathcal{B}$ of blocks, we use a variation of the well-known *Apriori* level-wise algorithm for computing itemsets (Agrawal and Srikant, 1994). In the algorithm described below, we do not distinguish between a closed bicluster and the corresponding block.

1. Compute the closure of each positive row r in M. Let $\mathcal{C}$ be the set of biclusters so computed.
2. $\mathcal{B} \leftarrow \mathcal{C}$.



(a) The binary matrix M. Each shaded square corresponds to a entry whose value is one.

(b) A layout of all closed biclusters in M. The three biclusters with solid boundaries correspond to the blocks G_1, G_2, and G_3 in Figure 1. We use arrows to dileneate the rows and columns that span these three biclusters.

FIG. 2. The binary matrix and biclusters corresponding to the example in Figure 1. Here, W is the complete graph on the five nodes.

3. Repeat the following steps until $\mathcal{C}$ is empty:
   (a) $\mathcal{C}' \leftarrow \emptyset$.
   (b) For each bicluster (R, C) in $\mathcal{C}$ and for each row $r \notin R$, compute the closure of $R \cup \{r\}$. If the closure contains at least one column, add it to $\mathcal{C}'$.
   (c) $\mathcal{C} \leftarrow \mathcal{C}'$.
   (d) $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{C}$.
4. Construct the DAG $\mathcal{D}_{\mathcal{B}}$ connecting the blocks in $\mathcal{B}$ as per the partial order $<$.

Remarks. Since we consider only the positive rows in M in the first step, every closed bicluster we compute contains at least one positive row. In practice, we hash the row sets of the biclusters to avoid reporting a bicluster more than once. The worst-case running time of the algorithm is exponential in the number of rows in M.

3.4. Assessing the statistical significance of a block

To measure the statistical significance of a block, we construct an empirical distribution of block sizes. We repeatedly select a subset of rows uniformly at random from the binary matrix M, compute the closure of these rows, and convert the resulting bicluster into a block. We ensure that the random subset of rows does not contain an active network and its complement, since such a subset will trivially result in an bicluster with zero columns. Given a block $(G, \mathcal{P}, \mathcal{N})$ computed in the real dataset, let m be the number of interactions in G. To estimate the statistical significance σ_G of $(G, \mathcal{P}, \mathcal{N})$, we only consider the distribution formed by random blocks $(H, \mathcal{P}', \mathcal{N}')$ where $|\mathcal{P}| = |\mathcal{P}'|$ and $|\mathcal{N}| = |\mathcal{N}'|$. We set σ_G to be the fraction of such blocks that have at least m interactions. Since the number of interactions in a block will decrease with an increase in $|\mathcal{P}|$ or in $|\mathcal{N}|$, these constraints ensure that we compare G with appropriate random blocks in order to estimate σ_G. We only retain blocks that are significant at the 0.01 level. We compute the DAG defined by these blocks. We perform two topological traversals of this DAG, one from the roots to the leaves and the other from the leaves to the roots, to identify the maximally-significant blocks. The resulting set of blocks are the network legos we desire to compute. Let $\mathcal{L}$ denote the set of network legos.

3.5. Stability and recoverability analysis

Stability. It is clear that the set $\mathcal{L}$ of network legos we compute depend on the active networks in $\mathcal{A}$. To assess this dependence, we modify a method for suggested by Segal et al. (2004). We remove each network $N \in \mathcal{A}$ in turn and recompute network legos from the set $\mathcal{A} - \{N\}$. Let $\mathcal{L}_N$ denote the resulting set of network legos. For each network lego L in $\mathcal{L}$, we compute the most similar network lego L' in $\mathcal{L}_N$ using the set-similarity measure $(|L \cap L'|/|L \cup L'|)$ and store this measure as $s_{L,N}$. Given a similarity threshold t, for each network lego L in $\mathcal{L}$, we compute the fraction of networks in $\mathcal{A}$ such that $s_{L,N} \geq t$. The higher this fraction is, the more resilient L is to perturbations in the input.

Recoverability. If the network legos in $\mathcal{L}$ are true building blocks of the active networks in $\mathcal{A}$ that they spring from, it should be possible to recover each active network in $\mathcal{A}$ from the network legos in $\mathcal{L}$. For each active network A, we define

$$\mathcal{L}_A = \{(G, \mathcal{P}, \mathcal{N}) \in \mathcal{L} | A \in \mathcal{P}\},$$

to be the set of network legos in $\mathcal{L}$ where A does not appear negated in the network lego. We compute the union of the network legos in $\mathcal{L}_A$ and compute what fraction of A's edge set appears in the union. The larger this fraction is, the more "recoverable" A is from the computed network legos.

4. RESULTS

We applied the algorithm described in the previous section to human data sets. We obtained a network of 31,108 molecular interactions between 9243 human gene products by integrating the interactions in the IDSERVE database (Ramani et al., 2005), the results of large scale yeast two-hybrid experiments (Rual et al., 2005; Stelzl et al., 2005), and 20 immune and cancer signalling pathways in the Netpath database (*www.netpath.org*). The IDSERVE database includes human curated interactions from BIND (Bader et al., 2003), HPRD (Peri et al., 2003), and Reactome (Joshi-Tope et al., 2005), interactions predicted based on co-citations in article abstracts, and interactions that transferred from lower eukaryotes based on sequence similarity (Lehner and Fraser, 2004). We derived functional annotations for the genes in our network from the Gene Ontology (GO) (Ashburner et al., 2000) and from MSigDB (Subramanian et al., 2005). In addition, we annotated each Netpath interaction in our network with the name of the pathway it belonged to. We used these annotations to compute the functional enrichment of the nodes and edges in the network legos using the hypergeometric distribution. We controlled the false discovery rate using the method proposed by Benjamini and Hochberg (1995).

We present two analyses below. In the first, we compare and contrast three types of leukemias. In the second, we compute the network legos for a set of environmental stresses imparted to two human cell types.

4.1. ALL, AML, and MLL

Armstrong et al. (2002) demonstrated that lymphoblastic leukemias involving translocations in the *MLL* gene constitute a disease different from conventional acute lymphoblastic (ALL) and acute myelogenous leukemia (AML). The authors based their analysis on the comparison of gene expression profiles from individuals diagnosed with ALL, AML, and MLL. We reasoned that the networks of molecular interactions activated in these diseases may also show distinct differences. First, we computed active networks for each leukemia, as described in Section 3.2. Next, we computed all 19 ($3^3 - 2^3$) blocks induced by these three active networks, using the method presented in Section 3.3. Since the number of blocks is small, we did not compute their statistical significance. Instead, we treated every block as a network lego. We connected the network legos in the directed acyclic graph (DAG) displayed in Figure 3. In this DAG, each node represents a single network lego, e.g., the leftmost node on the top row represents the MLL active network while the leftmost node in the middle row represents the interactions activated in AML but not in MLL (the formula AML – MLL). A solid blue edge directed from a child to a parent indicates that the formula for the child (e.g., MLL) appears as a part of the formula for the parent (e.g., MLL – AML), while a dashed green edge indicates that the child's formula (e.g., MLL) appears negated in the parent's formula (e.g., AML – MLL).

To assess the biological content of the results and to illustrate one type of analysis our approach facilitates, we computed Netpath pathways enriched in the interactions in the networks corresponding to the 19 formulae. Figure 3 demonstrates that the interactions in the KIT pathway are differentially enriched in the 19 networks. The darker the colour of a node, the more statistically significant is the enrichment of this pathway in the corresponding network. We first note that the only formulae enriched in this pathway are the ones that involve AML (and not the complement of AML). The statistical significance is the highest (FDR-corrected p-value 3.5×10^{-7}) for the formula AML – ALL – MLL. We interpret these statistics to imply that this pathway is activated only in AML and not in ALL or in MLL. Evidence in the literature supports this conclusion. The c-KIT receptor is activated in almost all subtypes of AML (Reuss-Borst et al.,

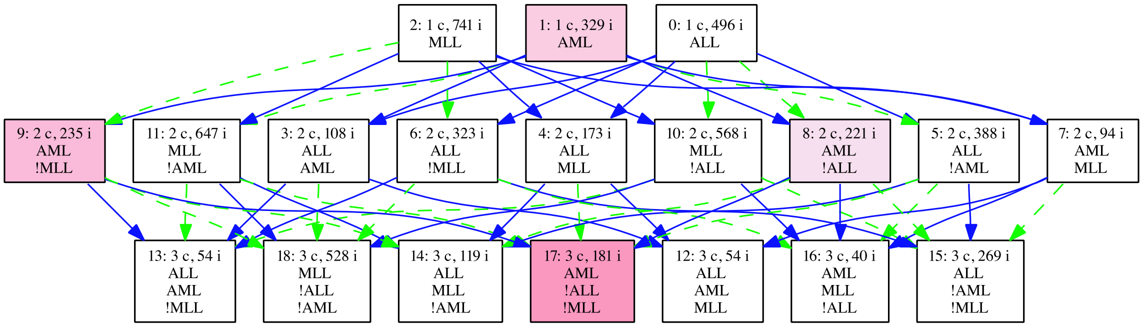


FIG. 3. The DAG connecting combinations of ALL, AML, and MLL active networks. Each node contains an index, the number of "c"onditions, the number of "i"nteractions and the active networks participating in the formula. We use "!" to indicate set difference. Colors indicate differential enrichment of the interactions in the KIT pathway in the computed combinations. Darker colors denote more significant enrichment values. (See this paper online for Fig. 3 in color.)

1994; Schwartz et al., 1999). Similarly, Schnittger et al. (2006) report that "mutations in codon D816 of the KIT gene represent a recurrent genetic alteration in AML." By studying 1937 patients diagnosed with acute leukemia, Bene et al. (1998) found that c-kit was expressed in 67% of AML cases but only in 4% of ALL cases, and that most of these ALL cases exhibited myeloid markers. We note that gain-of-function mutations in c-Kit have been observed in many human cancers (Cozma and Thomas-Tikhonenko, 2006). Our analysis only suggests that in the context of ALL, AML, and MLL, the KIT pathway may be activated only in AML.

4.2. *Human stresses*

We computed network legos by applying our methods to the human interaction network and the gene expression responses of HeLa cells and primary human lung fibroblasts to heat shock, endoplasmic reticulum stress, oxidative stress, and crowding (Murray et al., 2004). The dataset we analyzed includes transcriptional measurements obtained by Whitfield et al. (2002) for studying cell cycle arrest by using a double thymidine block or with a thymidine-nocodazole block. Overall, the dataset contains 13 distinct stresses over the two cell types. The authors note that each type of stress resulted in a distinct response and that there was no general stress response unlike in the case of *S. cerevisiae* (Gasch et al., 2000). Therefore, this dataset poses a challenge to our system. Can we find network legos that combine active networks for multiple stresses?

Structural analysis of network legos. The number of genes in the 13 active networks we computed ranged from 165 (for crowding of WI38 cells) to 1148 (for the thymidine-nocodazole block) with an average of 684 genes per active network. The number of interactions ranged from 257 to 3667 with an average of 1874 interactions per active network. Theoretically, we can compute 1,586,131 ($3^{13}-2^{13}$) blocks involving 13 distinct active networks. Our method computed 444,201 blocks, indicating that the remaining combinations of active networks are not closed or yield blocks without any interactions. We computed a null distribution of block sizes using a million random samples. Of the 444,201 blocks, 12,386 blocks were statistically significant at the 0.01 level. We identified 143 network legos in the DAG induced by the relation $<$ on these blocks. We observed that all but one of the 143 network legos involved at least six distinct active networks, indicating that these network legos are not the result of combining a small number of active networks. The following table displays the distribution of the number of legos involving k conditions, where $5 \leq k \leq 12$. Interestingly, no network lego involved all 13 active networks.

#conditions	5	6	7	8	9	10	11	12
#legos	1	6	10	36	34	20	28	8

In light of the statement by Murray et al. (2004) that each type of stress resulted in a distinct response, it is important to ask whether most of our network legos primarily involve complemented active networks. Over all network legos $(G, \mathcal{P}, \mathcal{N})$, we counted the total size of the "$\mathcal{P}$ sets" and the "$\mathcal{N}$ sets." The ratio of these numbers was 2:3, indicating that a large fraction of the network legos represented features common to multiple stresses. The active networks that appeared most often in the positive form were the two treatments that resulted in cell cycle arrest. Each participated in as many as 119 network legos. In most of these network legos, almost all the other active networks appeared in complemented form. The complements of the cell cycle arrest active networks did not participate in any network legos. This observation indicates that the interactions activated by cell cycle arrest are quite distinct from the network of interactions activated by the other stresses.

We obtained very good stability and recovery results. Upon the removal of each active network, we were able to recompute each network lego with at least 95% fidelity. We were also able to recover 11 active networks with 100% accuracy by composing network legos. The two active networks we could not recover completely were the double thymidine network (97% recovery) and the thymidine-nocodazole network (86% recovery). When we tested the recoverability of active networks using the blocks at the roots of the DAG connecting statistically-significant blocks, the recovery for these two active networks dropped to 85% and 75% respectively. This result underscores the fact that identifying network legos as those that are maximally statistically-significant in the DAG of blocks is a useful concept.

Since the cell-cycle treatments resulted in active networks that were quite distinct from those for the other stresses, we repeated the analysis after removing the double thymidine and thymidine-nocodazole active networks. The 11 remaining active networks yielded only 77,117 blocks (out of the 175,099 possible). Of these, 1629 blocks were statistically significant. These blocks yielded 15 network legos. This much smaller set of network legos suggests that a number of the 143 network legos in the complete analysis were needed to capture unique aspects of the cell cycle active networks. Each network lego involved at least seven active networks. No network lego involved all 11 stresses. The ratio of total size of the "$\mathcal{P}$ sets" and the "$\mathcal{N}$ sets." over the 15 network legos was 1:2. Of the 11 active networks, we recovered five with complete accuracy and one with 99.9% accuracy. We recovered the remaining with accuracies ranging from 71% to 92%. Taken together, these statistics indicate that the network legos we detect are indeed building blocks of the networks activated in response to the stresses studied by Murray et al. (2004).

Biological analysis of network legos. We focus on one of the 15 network legos we computed in our analysis without the cell cycle arrest treatments. This *ER stress* network lego corresponds to the formula

$$\text{(Fibroblast DTT} \cap \text{Fibroblast Menadione)} -$$

$$\text{(HeLa Crowding} \cup \text{HeLa Heat} \cup \text{HeLa Menadione} \cup \text{Fibroblast Crowding} \cup \text{Fibroblast Heat)}$$

The only two stresses that appear in positive form in this formula are the treatment of fibroblasts with DTT and menadione. These chemicals induce endoplasmic reticulum (ER) stress. This network lego is the only one significantly enriched in functions related to the cell cycle (e.g., p-value 3×10^{-30} for the KEGG (Kanehisa et al., 2006) "Cell cycle" pathway and 2.3×10^{-24} for the REACTOME (Joshi-Tope et al., 2005) pathway describing the transition from G1 to S) and in targets of the E2F1 transcription factor (Subramanian et al., 2005) (p-value 8×10^{-13}), which is a known regulator of cell cycle progression. E2F1 arrests cells in the G1 phase by forming a transcriptional repressor complex with the Retinoblastoma protein (Zhang et al., 1999). Figure 4 displays a layout of the *ER stress* network lego, specifically highlighting the genes annotated with the KEGG "Cell cycle" pathway and as targets of E2F1. Figure 5 shows a heat map of the expression profiles of the genes annotated with these two functions in the seven conditions in the network lego. Examination of the gene expression patterns in Figure 5 reveals that, at about 4–6 hours after treatment with DTT or menadione, fibroblasts shut down the cell cycle far more aggressively than fibroblasts or HeLa cells do in response to other treatments. Thus, this network lego lego automatically identifies a unique characteristic of fibroblast response to ER stress in the context of the other stresses in the compendium.

5. DISCUSSION

We have presented a novel approach for combining gene expression datasets with a multi-modal wiring diagram to compute network legos, which are context-sensitive building blocks of the wiring diagram. This combination provides a dynamic view of the interactions that are activated in the wiring diagram

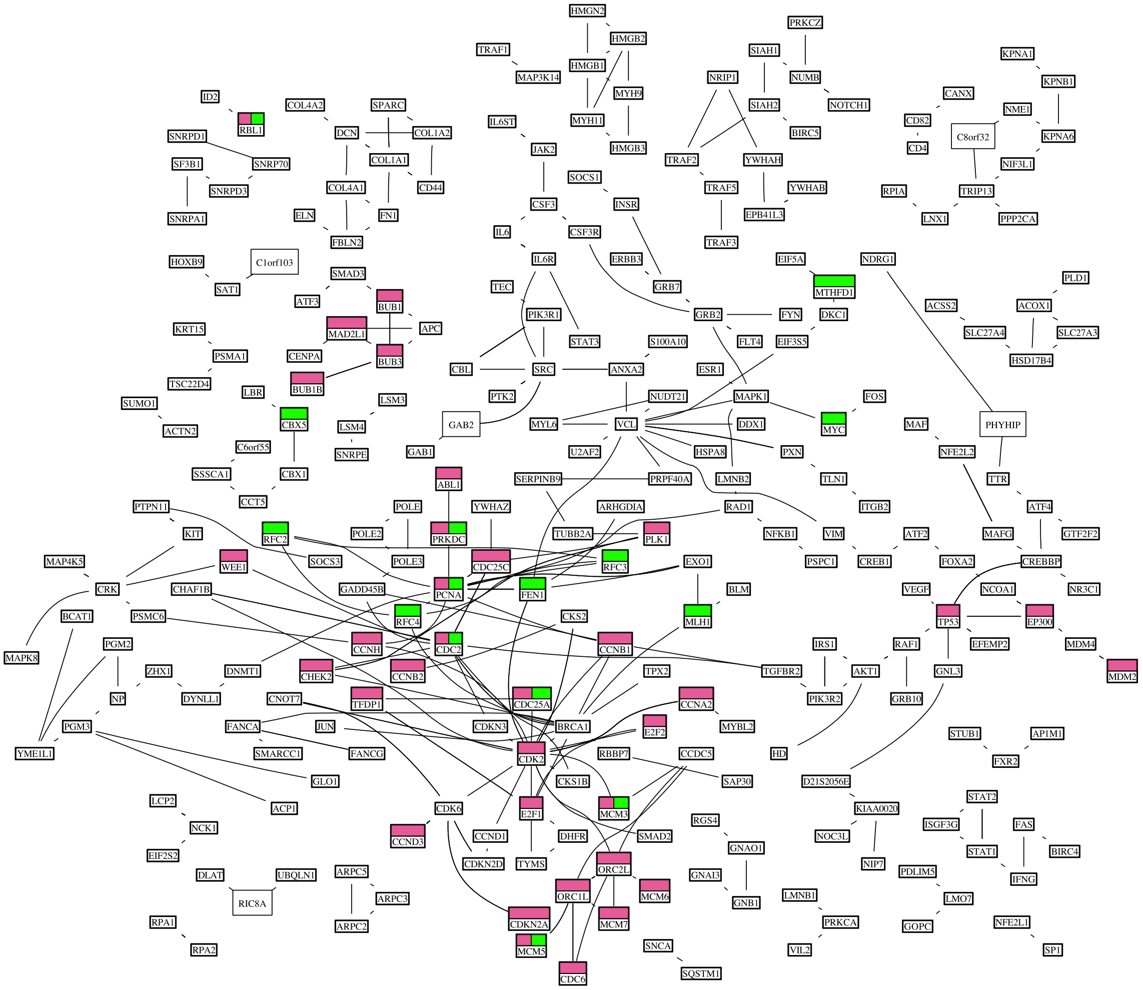


FIG. 4. A layout of the interactions in the *ER stress* network lego. Red nodes are annotated with "Cell cycle" and green nodes are targets of the E2F1 transcription factor. (See this paper online for Fig. 4 in color.)

under different conditions. We represent similarities and differences between the network of interactions activated in response to different cell states both as a set theoretic formula involving cell states and as a network lego, a functional module of co-expressed molecular interactions. A novel contribution of our work is the DAG that relates all cell states (and the active networks corresponding to the cell states). This DAG provides a high-level abstract view of the similarities and differences between cell states.

The literature on network motifs (Milo et al., 2002, 2004; Shen-Orr et al., 2002; Yeger-Lotem et al., 2004) provides an alternative perspective on finding the building blocks of cellular circuits. Zhang et al. (2005) constructed an integrated *S. cerevisiae* interaction network, identified three- and four-node network motifs, and organized these motifs into network themes and further into thematic maps. It would be interesting to study whether the top-down approach presented here to construct network legos yields network modules that are similar in structure and organization to those computed by the bottom-up approach used by Zhang et al. (2005).

Since we explicitly compute all closed biclusters in $\mathcal{B}$, the worst-case running time of our algorithm may be exponential in the number of active networks. An interesting avenue of future research is to develop a method that avoids this exorbitant running time, perhaps by computing network legos that directly optimize for stability and/or recoverability. Another important open question is that of developing an incremental algorithm that can efficiently recompute the network legos upon the addition or deletion of an active network.

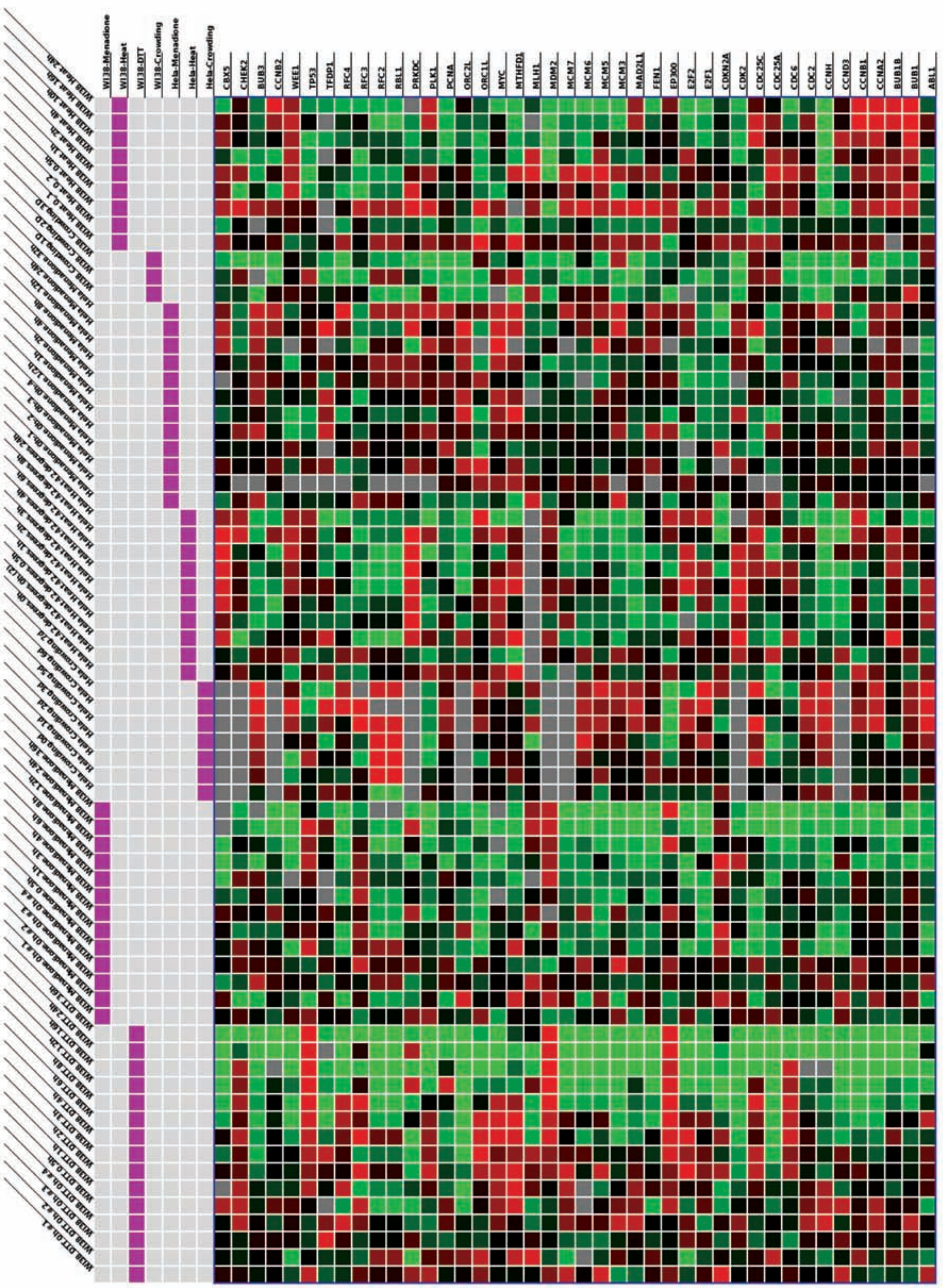


FIG. 5. A heat map of the gene expression measurements in the seven conditions participating in the *ER stress* network lego. Columns correspond to genes annotated with at least one of the functions mentioned in the text. Rows correspond to samples. Each vertical pink line delineates a set of samples belonging to one of the stresses in the network lego. The two lowermost pink lines correspond to fibroblast response to treatment with menadione and with DTT, which are the two stresses that appear in positive form in the *ER stress* network lego. (See this paper online for Fig. 5 in color.)

ACKNOWLEDGMENTS

We thank Naren Ramakrishnan for useful suggestions on the algorithm for computing blocks. Grants from the ASPIRES programme and the Institute for Critical Technology and Applied Science at the Virginia Polytechnic Institute and State University supported this research.

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Agrawal, R., and Srikant, R. 1994. Fast algorithms for mining association rules in large databases. *Proc. 20th Int. Conf. Very Large Databases* 487–499.

Armstrong, S., Staunton, J., Silverman, L., et al. 2002. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nat. Genet.* 30, 41–47.

Ashburner, M., Ball, C.A., Blake, J.A., et al. 2000. Gene Ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* 25, 25–29.

Bader, G.D., Betel, D., and Hogue, C.W.V. 2003. BIND: the Biomolecular Interaction Network Database. *Nucleic Acids Res.* 31, 248–250.

Bansal, N., Blum, A., and Chawla, S. 2004. Correlation clustering. *Mach. Learn.* 56, 89–113.

Bar-Joseph, Z., Gerber, G.K., Lee, T.I., et al. 2003. Computational discovery of gene modules and regulatory networks. *Nat. Biotechnol.* 21, 1337–1342.

Barry, W.T., Nobel, A.B., and Wright, F.A. 2005. Significance analysis of functional categories in gene expression studies: a structured permutation approach. *Bioinformatics* 21, 1943–1949.

Basso, K., Margolin, A., Stolovitzky, G., et al. 2005. Reverse engineering of regulatory networks in human B cells. *Nat. Genet.* 37, 382–390.

Bene, M., Bernier, M., Casasnovas, R., et al. The reliability and specificity of c-kit for the diagnosis of acute myeloid leukemias and undifferentiated leukemias. *Blood* 92, 596.

Benjamini, Y., and Hochberg, Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc.* 57, 289–300.

Bergmann, S., Ihmels, J., and Barkai, N. 2003. Similarities and differences in genome-wide expression data of six organisms. *PLoS Biol.* 2, E9.

Bonneau, R., Reiss, D.J., Shannon, P., et al. 2006. The Inferelator: an algorithm for learning parsimonious regulatory networks from systems-biology datasets *de novo*. *Genome Biol.* 7, R36.

Charikar, M. 2000. Greedy approximation algorithms for finding dense components in a graph. *Proc. 3rd Int. Workshop Approx. Algorithms Combin. Optim.* 84–95.

Chung, F.R.K. 1997. *Spectral Graph Theory*. AMS.

Cozma, D., and Thomas-Tikhonenko, A. 2006. Kit-activating mutations in AML: lessons from PU.1-induced murine erythroleukemia. *Cancer. Biol. Ther.* 5, 579–581.

Dhillon, I., Guan, Y., and Kulis, B. 2007. Weighted graph cuts without eigenvectors: a multilevel approach. *IEEE Trans. Pattern Analysis Mach. Intellig.* 29, 1944–1957.

di Bernardo, D., Thompson, M.J., Gardner, T.S., et al. 2005. Chemogenomic profiling on a genome-wide scale using reverse-engineered gene networks. *Nat. Biotechnol.*, 23, 377–383.

Drineas, P., Frieze, A., Kannan, R., et al. Clustering large graphs via the singular value decomposition. *Mach. Learn.* 56, 9–33.

Dunn, R., Dudbridge, F., and Sanderson, C.M. 2005. The use of edge-betweenness clustering to investigate biological function in protein interaction networks. *BMC Bioinform.* 6.

Edelman, E., Porrello, A., Guinney, J., et al. 2006. Analysis of sample set enrichment scores: assaying the enrichment of sets of genes for individual samples in genome-wide expression profiles. *Bioinformatics* 22, e108–e116.

Enright, A.J., Van Dongen, S., and Ouzounis, C.A. 2002. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* 30, 1575–1584.

Ganter, B., and Wille, R. 1997. *Formal Concept Analysis: Mathematical Foundations*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Gasch, A.P., Spellman, P.T., Kao, C.M., et al. 2000. Genomic expression programs in the response of yeast cells to environmental changes. *Mol. Biol. Cell.*, 11, 4241–4257.

Grochow, J., and Kellis, M. 2007. Network motif discovery using subgraph enumeration and symmetry-breaking. *Lect. Notes Comput. Sci.* 92–106.

Guo, Z., Zhang, T., Li, X., et al. 2005. Towards precise classification of cancers based on robust gene functional expression profiles. *BMC Bioinform.* 6, 58.

Han, J., Bertin, N., Hao, T., et al. 2004. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature* 430, 88–93.

Haugen, A., Kelley, R., Collins, J., et al. 2004. Integrating phenotypic and expression profiles to map arsenic-response networks. *Genome Biol.* 5, R95.

Hu, H., Yan, X., Huang, Y., et al. 2005. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics* 21, Suppl 1, i213–i221.

Huang, R., Wallqvist, A., and Covell, D.G. 2006. Targeting changes in cancer: assessing pathway stability by comparing pathway gene expression coherence levels in tumor and normal tissues. *Mol. Cancer Ther.* 5, 2417–2427.

Huttenhower, C., Hibbs, M., Myers, C., et al. 2006. A scalable method for integration and functional analysis of multiple microarray datasets. *Bioinformatics* 22, 2890–2897.

Ideker, T., Ozier, O., Schwikowski, B., et al. 2002. Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, 18, Suppl 1, S233–S240.

Joshi-Tope, G., Gillespie, M., Vastrik, I., et al. 2005. Reactome: a knowledgebase of biological pathways. *Nucleic Acids Res.* 33, D428–D432.

Joyce, A.R., and Palsson, B.O. 2006. The model organism as a system: integrating "omics" datasets. *Nat. Rev. Mol. Cell Biol.* 7, 198–210.

Kanehisa, M., Goto, S., Hattori, M., et al. 2006. From genomics to chemical genomics: new developments in KEGG. *Nucleic Acids Res.* 34, D354–D357.

Kannan, R., Vempala, S., and Vetta, A. 2004. On clusterings-good, bad and spectral. In *Proc. FOCS '00. JACM* 51, 497–515.

Koyuturk, M., Kim, Y., Subramaniam, S., et al. 2006. Detecting conserved interaction patterns in biological networks. *J. Comput. Biol.*, 13, 1299–1322.

Lee, H., Hsu, A., Sajdak, J., et al. 2004. Coexpression analysis of human genes across many microarray data sets. *Genome Res.* 14, 1085–1094.

Lehner, B., and Fraser, A.B. 2004. A first-draft human protein-interaction map. *Genome Biol.* 5, R63.

Levine, D.M., Haynor, D.R., Castle, J.C., 2006. Pathway and gene-set activation measurement from mRNA expression data: the tissue distribution of human pathways. *Genome Biol.* 7, R93.

Luscombe, N., Babu, M., Yu, H., et al. 2004. Genomic analysis of regulatory network dynamics reveals large topological changes. *Nature* 431, 308–312.

Madeira, S.C., and Oliveira, A.L. 2004. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Trans. Comput. Biol. Bioinform.*

Milo, R., Shen-Orr, S., Itzkovitz, S., et al. 2002. Network motifs: simple building blocks of complex networks. *Science* 298, 824–827.

Milo, R., Itzkovitz, S., Kashtan, N., et al. 2004. Superfamilies of evolved and designed networks. *Science* 303, 1538–1542.

Murray, J.I., Whitfield, M.L., Trinklein, N.D., et al. 2004. Diverse and specific gene expression responses to stresses in cultured human cells. *Mol. Biol. Cell.* 15, 2361–2374.

Myers, C.L., Robson, D., Wible, A., et al. 2005. Discovery of biological networks from diverse functional genomic data. *Genome Biol.* 6, R114.

Newman, M. 2006. Modularity and community structure in networks. *Proc. Natl. Acad. Sci.* 103, 8577–8582.

Ng, A.Y., Jordan, M.I., and Weiss, Y. 2002. On spectral clustering: analysis and an algorithm. *Adv. NIPS 14.* 2002.

Peri, S., Navarro, J., Amanchy, R., et al. 2003. Development of human protein reference database as an initial platform for approaching systems biology in humans. *Genome Res.* 13, 2363–2371.

Radicchi, F., Castellano, C., Cecconi, F., et al. 2004. Defining and identifying communities in networks. *Proc. Natl. Acad. Sci.* 101, 2658–2663.

Ramani, A.K., Bunescu, R.C., Mooney, R.J., et al. 2005. Consolidating the set of known human protein-protein interactions in preparation for large-scale mapping of the human interactome. *Genome Biol.* 6, R40.

Reiss, D., Avila-Campillo, I., Thorsson, V., et al. 2005. Tools enabling the elucidation of molecular pathways active in human disease: application to hepatitis C virus infection. *BMC Bioinform.* 6, 154.

Reuss-Borst, M., Buhring, H., Schmidt, H., et al. 1994. AML: immunophenotypic heterogeneity and prognostic significance of c-kit expression. *Leukemia* 8, 258–263.

Rhodes, D., Yu, J., Shanker, K., et al. 2004. Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression. *Proc. Natl. Acad. Sci. USA* 101, 9309–9314.

Rual, J., Venkatesan, K., Hao, T., et al. 2005. Towards a proteome-scale map of the human protein-protein interaction network. *Nature* 437, 1173–1178.

Schnittger, S., Kohl, T.M., Haferlach, T., et al. 2006. KIT-D816 mutations in AML1-ETO-positive AML are associated with impaired event-free and overall survival. *Blood* 107, 1791–1799.

Schwartz, S., Heinecke, A., Zimmermann, M., et al. 1999. Expression of the C-kit receptor (CD117) is a feature of almost all subtypes of de novo acute myeloblastic leukemia (AML), including cytogenetically good-risk AML, and lacks prognostic significance. *Leuk. Lymphoma* 34, 85–94.

Segal, E., Wang, H., and Koller, D. 2003. Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics* 19, I264–I272.

Segal, E., Friedman, N., Koller, D., et al. 2004. A module map showing conditional activity of expression modules in cancer. *Nat. Genet.* 36, 1090–1098.

Sharan, R., and Ideker, T. 2006. Modeling cellular machinery through biological network comparison. *Nat. Biotechnol.* 24, 427–433.

Sharan, R., and Shamir, R. 2000. Click: a clustering algorithm with applications to gene expression analysis. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 8, 307–316.

Shen-Orr, S.S., Milo, R., Mangan, S., et al. 2002. Network motifs in the transcriptional regulation network of *Escherichia coli*. *Nat. Genet.* 31, 64–68.

Shi, J., and Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Analysis Mach. Intellig.* 22, 888–905.

Stelzl, U., Worm, U., Lalowski, M., 2005. A human protein-protein interaction network: a resource for annotating the proteome. *Cell* 122, 957–968.

Stuart, J.M., Segal, E., Koller, D., et al. 2003. A gene-coexpression network for global discovery of conserved genetic modules. *Science* 302, 249–255.

Subramanian, A., Tamayo, P., Mootha, V., et al. 2005. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci. USA*.

Tanay, A., Sharan, R., and Shamir, R. 2002. Discovering statistically significant biclusters in gene expression data. *Proc. ISMB 2002* S136–S144.

Tanay, A., Sharan, R., Kupiec, M., et al. 2004. Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data. *Proc. Natl. Acad. Sci. USA* 101, 2981–2986.

Tanay, A., Steinfeld, I., Kupiec, M., et al. 2005. Integrative analysis of genome-wide experiments in the context of a large high-throughput data compendium. *Mol. Syst. Biol.* 1, msb4100005-E1–msb4100005-E10.

Tanay, A., Sharan, R., and Shamir, R. 2006. *Handbook of Computational Molecular Biology*. CRC Press.

Ulitsky I., and Shamir, R. 2007. Identification of functional modules using network topology and high-throughput data. *BMC Syst. Biol.* 1, 8.

White, S., and Smyth, P. 2005. A spectral clustering approach to finding communities in graph. *Proc. Fourth SIAM Int. Conf. Data Mining*.

Whitfield, M.L., Sherlock, G., Saldanha, A.J., et al. 2002. Identification of genes periodically expressed in the human cell cycle and their expression in tumors. *Mol. Biol. Cell.* 13, 1977–2000.

Yan, X., and Han, J. 2003. Closegraph: mining closed frequent graph patterns. *Proc. KDD '03* 286–295.

Yeger-Lotem, E., Sattath, S., Kashtan, N., et al. 2004. Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *Proc. Natl. Acad. Sci. USA* 101, 5934–5939.

Zaki, M.J., and Hsiao, C.-J. 2002. CHARM: an efficient algorithm for closed itemset mining. In *SIAM Int. Conf. Data Mining* 457–473.

Zhang, H.S., Postigo, A.A., and Dean, D.C. 1999. Active transcriptional repression by the Rb-E2F complex mediates G1 arrest triggered by p16INK4a, TGFbeta, and contact inhibition. *Cell* 97, 53–61.

Zhang, L.V., King, O.D., Wong, S.L., et al. 2005. Motifs, themes and thematic maps of an integrated *Saccharomyces cerevisiae* interaction network. *J. Biol.* 4, 6.

Zhou, X.J., Kao, M.C., Huang, H., et al. 2005. Functional annotation and network reconstruction through cross-platform integration of microarray data. *Nat. Biotechnol.* 23, 238–243.

Address reprint requests to:
Dr. T. M. Murali
2050 Torgerson Hall
Department of Computer Science
Virginia Polytechnic Institute and State University
Blacksburg, VA 24061

E-mail: murali@cs.vt.edu