JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 16, Number 8, 2009
© Mary Ann Liebert, Inc.
Pp. 1023–1034
DOI: 10.1089/cmb.2009.0037

Research Article

An Integrative Network Approach to Map the Transcriptome to the Phenome

MICHAEL R. MEHAN, JUAN NUNEZ-IGLESIAS, MRINAL KALAKRISHNAN,
MICHAEL S. WATERMAN, and XIANGHONG JASMINE ZHOU

ABSTRACT

Although many studies have been successful in the discovery of cooperating groups of genes, mapping these groups to phenotypes has proved a much more challenging task. In this article, we present the first genome-wide mapping of gene coexpression modules onto the phenome. We annotated coexpression networks from 136 microarray datasets with phenotypes from the Unified Medical Language System (UMLS). We then designed an efficient graph-based simulated annealing approach to identify coexpression modules frequently and specifically occurring in datasets related to individual phenotypes. By requiring phenotype-specific recurrence, we ensure the robustness of our findings. We discovered 118,772 modules specific to 42 phenotypes, and developed validation tests combining Gene Ontology, GeneRIF and UMLS. Our method is generally applicable to any kind of abundant network data with defined phenotype association, and thus paves the way for genome-wide, gene network-phenotype maps.

Key words: algorithms, computational molecular biology, dynamic programming.

1. INTRODUCTION

The fundamental aim of genetics is to link phenotype to genotype, and traditional genetic studies have sought to associate single genes to a particular phenotypic trait. However, it has become clear that complex diseases, such as cancer, autoimmune disease, or heart disease, are effected by the interaction of many different genes. For this problem, genetic association studies lack power. Locus heterogeneity, epistasis, low penetrance, and pleiotropy all contribute to mask or reduce the detectable signal (Lander and Schork, 1994; Risch, 2000).

In recent years, high-throughput approaches have been used to study the interaction of groups of genes. In a gene network, nodes represent genes (or gene products), and links between nodes represent functional relationship between the nodes. Examples include protein-protein interaction networks, genetic interaction networks, and gene coexpression networks. Borrowing or expanding tools from the fields of network analysis and graph theory, researchers have devised numerous ways to use these networks to determine which genes work together (Zhou et al., 2002; Bader and Hogue, 2003; Spirin and Mirny, 2003; Kelley et al., 2003; Hu et al., 2005; Yip and Horvath, 2007). However, virtually all of this work fails to complete

Program in Computational Biology, Department of Biological Sciences, University of Southern California, Los Angeles, California.

the link between genotype and phenotype. Genes and gene products are grouped into modules and complexes, but these are not linked to phenotypes. We note two remarkable exceptions: Butte and Kohane (2006) used differential expression analysis to systematically associate genes with specific phenotypes and environments, using data from the Gene Expression Omnibus (Barrett et al., 2007); and Lage et al. (2007) used OMIM protein annotations to associate protein complexes with disease phenotypes. However, the former approach does not consider genes in a network context, while the latter approach only considers annotated nodes in a single static network. Neither approach, nor any other, has systematically mapped gene networks to the experimental phenotype conditions under which they are activated.

In this article, we introduce the first approach to explicitly bridge this gap. Like Butte and Kohane, we used the large amount of microarray gene expression data from the Gene Expression Omnibus. Here, instead of gene-phenotype associations, we used integrative network analysis to infer network module to phenotype associations. A series of microarray datasets can be modeled as a series of coexpression networks as follows: each node represents a gene, and a link is placed between two nodes if their expression profiles in that dataset are highly similar. The crucial advantage of this approach is that each generated network can be labeled with the phenotypic information of that dataset, such as the type of biological sample, the disease state, drug treatment, etc. The Unified Medical Language System (UMLS) (Bodenreider, 2004) provides an extensive catalog of medical concepts and their relationships, as well as language processing tools that enable the automated mapping of text onto UMLS concepts. This allowed us to automatically annotate each microarray dataset with UMLS phenotype classes by using the associated MEDLINE reference.

For each phenotype, we partitioned the datasets into a *phenotype class*, consisting of datasets annotated with that phenotype, and a *background class*, consisting of the rest of the datasets. We designed a graph-based simulated annealing (Kirkpatrick et al., 1983) approach to efficiently identify groups of genes which form dense subnetworks preferentially and repeatedly in the phenotype class. Note that a dense subnetwork in a coexpression graph represents a coexpression cluster. Although microarray data is noisy, we have shown in our previous work (Zhou et al., 2005; Yan et al., 2007) that coexpression clusters recurrent across multiple datasets represent true functional or transcriptional modules with high probability. Here, we further show that, if a frequent coexpression cluster additionally is specific to a phenotype class, it is likely to effect that phenotype.

We applied our approach to the analysis of 136 microarray datasets, covering 47 phenotype conditions. We discovered approximately 120,000 modules specific to 42 of these phenotypes, and developed a novel way to validate this specificity by integrating gene and dataset annotations from Gene Ontology (Gene Ontology Consortium, 2006), Gene Reference Into Function (GeneRIF) (Mitchell et al., 2003), and UMLS. Our method lays the foundation for a genome-wide, gene network-phenotype map, which will benefit our understanding of complex diseases and their treatment. Our present map of network patterns to phenotypes has many applications, such as predicting the phenotypic effects of multiple interacting genetic perturbations, *in silico* testing of genetically complex hypotheses, and prioritization of candidate genes for targeted intervention. Furthermore, the concept of our approach is general, and can be easily extended to incorporate any standardized phenomic procedures, as suggested, for example, by the Human Phenome Project (Freimer and Sabatti, 2003).

2. METHODS

2.1. Dataset preparation

2.1.1. Dataset selection. We selected every microarray dataset from NCBI's Gene Expression Omnibus that met the following criteria: all samples were of human origin; the dataset had at least eight samples (a minimum for accurate correlation estimation); and the platform was either GPL91 (corresponding to Affymetrix HG-U95A) or GPL96 (Affymetrix HG-U133A). Throughout this study, we only considered the genes shared by the two platforms (and therefore all datasets), of which there are 8,635. We averaged expression values for probe that map to the same gene within a dataset. The 136 datasets that met these criteria on 28 February 2007 were used for the analysis described herein.

2.1.2. Dataset annotation. We determined the phenotypic context of a microarray dataset by mapping the Medical Subject Headings (MeSH) of its corresponding PubMed record to UMLS concepts using the MetaMap Transfer tool provided by the UMLS. This is more refined than attempting to scan the abstract or full text of the paper, and in practice it results in much cleaner and more reliable annotations (Butte and Kohane, 2006; Butte and Chen, 2006). UMLS is the largest available compendium of biomedical vocabularies, spanning approximately one million interrelated concepts, including diseases, treatments, and phenotypic concepts at different levels of resolution (molecules, cells, tissues and whole organisms). In order to infer higher-order links between datasets, we annotated datasets with the matched UMLS concept and, in addition, all its ancestor concepts. This resulted in a total of 467 annotations, of which 80 mapped to more than five datasets, or 60 after merging annotations that mapped onto identical sets of datasets.

2.1.3. Correlation estimation and graph generation. For each dataset, we used the Jackknife Pearson correlation as a measure of similarity between two genes (the minimum of the leave-one-out Pearson correlations). To determine the coexpression network, we selected a cutoff corresponding to the top-ranking 150,000 correlations of the total $\binom{8635}{2} \approx 3.73 \times 10^7$ gene pairs (0.4%). The cutoff was generated by exploring frequency of coexpressed pairs of genes that share a functional annotation for a range of correlation values. We found that this cutoff was consistent across many datasets and therefore provides coexpression graphs with similar densities, regardless of the number of experimental samples.

Once a cutoff was determined, we defined that dataset's coexpression network as the graph $G_i = (V, E_i)$, where V corresponds to the set of genes being investigated, and $(g_a, g_b) \in E_i$ if the correlation between g_a and g_b is higher than the cutoff.

2.1.4. Differential coexpression graphs. To dramatically increase the probability of finding optimal modules across the many massive networks, we wished to narrow down the search space. We therefore constructed a weighted *differential coexpression graph* for each phenotype, which summarizes the differences between the gene coexpression networks in the phenotype class and those in the background class. This graph was used by the simulated annealing algorithm to create neighboring states (see Section 2.2). We describe it formally as follows.

To begin, we define $\mathcal{G}$ as the set of all graphs constructed from the microarray datasets. For each phenotype $\mathcal{P}$, we partition $\mathcal{G}$ into the *phenotype* graphs $\mathcal{G}_{\mathcal{P}}$, corresponding to datasets annotated with $\mathcal{P}$, and the *background* graphs $\mathcal{G}^c_{\mathcal{P}} = \mathcal{G} \setminus \mathcal{G}_{\mathcal{P}}$, corresponding to the rest of the datasets.

We then construct a weighted differential coexpression graph $G_\Delta = (V, E_\Delta)$ to reflect edges (coexpression relationships) that are present frequently in $\mathcal{G}_{\mathcal{P}}$ but not in $\mathcal{G}^c_{\mathcal{P}}$. This specificity can be measured by the significance p of a hypergeometric test, assessing the abundance of an edge in $\mathcal{G}_{\mathcal{P}}$ relative to its overall abundance in $\mathcal{G}$. In G_Δ, the vertex set V is the same as in every graph in $\mathcal{G}$, and the weight associated with (g_a, g_b) is then $w_\Delta(ga, gb) = -\log(p)$. Edges of weight 0 are not in E_Δ. In this way, heavier edges in this graph represent pairs of genes that exhibit elevated coexpression highly specific to $\mathcal{G}_{\mathcal{P}}$.

2.2. Simulated annealing design

2.2.1. Goal and rationale. Our aim was to find sets of genes that satisfy three criteria: first, the genes must be tighly coexpressed in multiple datasets; second, the annotations of these datasets must be enriched for some specific phenotype; and third, the gene set must be sufficiently large while meeting the first two criteria.

As explained in Section 2.1, from each annotated dataset we derived a coexpression graph. For a set of vertices $V' \subset V$ having m edges between them, the density is $\delta(V') = m / \binom{|V'|}{2} = 2m/(|V'|(|V'| - 1))$. This is exactly the proportion of gene pairs from V' that are coexpressed, taken over all possible pairs $\{(u, v) : u \in V', v \in V'\}$. We say that a vertex set is *dense* if δ is large (typically greater than 0.66). Then, for each phenotype, we wanted to find a set of vertices that is dense in a large proportion of datasets annotated with that phenotype, and that is *not* dense in datasets not annotated with it.

As we demonstrated in our earlier work (Yan et al., 2007), the problem of identifying frequent dense vertex sets is NP-complete. Much work has been done on identifying dense vertex sets in single graphs (Ding and Peng, 2005; Asahiro et al., 2002; Feige et al., 2001; Srivastav and Wolf, 1998), and it is easy to show that the additional requirement of phenotype specificity does not decrease the complexity of the problem. Hence, we decided to use simulated annealing, a well-established stochastic algorithm with successful application in other NP-complete problems (Suman and Kumar, 2006). Our design for the simulated annealing (SA) algorithm follows.

2.2.2. Search space. A state in our SA design is defined as a set of vertices, and the search space is the set of all sets of vertices, although for simplicity and for computational considerations we limited ourselves to sets smaller than 30 vertices. We believe this to be an ample margin for phenotypically relevant gene sets. Formally, we define the search space as $S = \{x : x \subset V, |x| \leq 30, |x| \geq 3\}$.

2.2.3. Objective functions. Recall that we needed to optimize three different objectives: size, density, and specificity. We created one objective function for each of these goals, and then supplemented them with a fourth objective, called *density differential.*

Much work has been done to generalize the simulated annealing process to multiple objectives, collectively known as MOSA (Multiple Objective Simulated Annealing). The general strategy is to create an energy function f_i for each objective i, and then combine them into a single energy function by using a weighted sum $f(x) = \sum_{i=1}^{k} w_i f_i(x)$. The key difficulty with this approach is determining an appropriate set of weights. In previous studies, this has been accomplished empirically (Collette and Siarry, 2004), and this is the approach that we take for the following reasons: we were interested in a single optimal combination of objective functions, rather than exploring the extremes of each; our design for individual functions was such that overall effectiveness of the algorithm was consistent throughout a range of weights; and the parameters we chose based on performance on simulated data behaved well on the real data. The weights we chose for size, density, specificity, and density differential were 0.05, 0.05, 5, and 50, respectively.

The individual energy functions that we designed take the following forms:

$$f_{size}(x) = \exp\left\{ -\alpha\left(\frac{|x|}{\gamma} - o_s\right)\right\} \tag{1}$$

$$f_{dens}(x) = \exp\left\{ -\alpha\left(\min_{i \in \mathcal{G}_{\mathcal{A}}} (\delta_i(x)) - o_\delta\right)\right\} \tag{2}$$

$$f_{spec}(x) = \log\left(\mathbb{P}(Y \geq |\mathcal{G}_{\mathcal{A}} \cap \mathcal{G}_{\mathcal{P}}|)\right) \tag{3}$$

$$f_{diff}(x) = \left(\frac{1}{|\mathcal{G}_{\mathcal{P}}^c|} \sum_{i \in \mathcal{G}_{\mathcal{P}}^c} \delta_i(x) - \frac{1}{|\mathcal{G}_{\mathcal{P}}|} \sum_{i \in \mathcal{G}_{\mathcal{P}}} \delta_i(x) \right) \tag{4}$$

where $\mathcal{G}_{\mathcal{P}}$ is the set of datasets annotated with the current phenotype; $\mathcal{G}_{\mathcal{A}}$ is the set of datasets in which the gene cluster is dense; and $Y \sim \textit{hypergeometric}\ (|\mathcal{G}_{\mathcal{A}}|, |\mathcal{G}_{\mathcal{P}}|, |\mathcal{G}_{\mathcal{P}}^c|)$.

From previous studies, we have determined criteria for favorable coexpression clusters: size of 7 or more and density greater than 0.66 (Yan et al., 2007). For simulated annealing, however, we cannot simply enforce these thresholds as we need to accept intermediate states that may be unfavorable. We therefore designed the energy functions for size (1) and density (2) to enforce soft thresholds, by using an exponential increase in energy for unfavorable values. Since we combine the functions using a linear weighted sum, extreme solutions (such as a single triangle that is very dense, but very small) will be rejected by the exponential energy increase of this soft threshold, but states slightly below our threshold will still have a probability of being accepted.

The specificity function (3) is the log of a hypergeometric p-value for enrichment, so that more significant enrichment for the phenotype datasets will be rewarded with lower energies. Unlike the first two objectives, the specificity function has no threshold component. It does however continue to reward more significant enrichment with large decreases in energy, whereas the soft thresholds have very small decreases in energy once the threshold is achieved. Therefore the first three objective functions attempt to maximize the phenotype specificity while enforcing soft thresholds on both size and density. To ensure that our module is not only phenotype-specific but also a frequently occurring cluster, we enforce a minimum of five active datasets when evaluating both the density and specificity functions. This prevents the algorithm from settling on a module that is very dense in a single dataset related to the phenotype, rather than identifying a recurrent module present in many phenotype-related datasets.

Finally, equation (4) shows the density differential objective function, which consists of the difference between the average density of the cluster in the background datasets and that in the phenotype datasets. The density differential objective function is designed to complement the density and specificity objective functions. Since the specificity function takes a state's active datasets as its argument, only neighboring states with a new set of active datasets will have a different specificity energy value than the previous state. However, many neighboring states can have subtle changes in the density distribution among the active and inactive datasets that is not captured by the density and specificity functions alone. The density differential function is therefore designed to reward these subtle density changes, and thus direct the simulated annealing process towards more phenotype-specific clusters. We found that using the density differential objective function in combination with specificity and density allowed the algorithm to converge faster and to better clusters than either function alone.

We selected the parameters $\alpha = 20$, $\gamma = 30$, $o_\delta = 0.85$, and $o_s = 0.2$ based on our simulation results with biologically validated clusters compared with clusters arising from random chance.

2.2.4. Initial state. A SA approach aims to find a global optimum during each run. Therefore, if we were to use random initial states and run the algorithm for a long enough time, we will always find approximately the same set of vertices, representing the largest set having the most evidence for coexpression and phenotype specificity. We were, however, interested in a large number of vertex sets showing evidence for coexpression and phenotype specificity. To this end, we designed a systematic way of generating initial states, or seeds, and we restricted the SA search space to clusters containing these seeds.

We define a *triangle* as a set of three vertices that is fully connected in at least one dataset. The hypothesis underlying our strategy is that if a set of genes is coexpressed specifically in datasets annotated with the phenotype of interest, then at least one recurrent triangle will appear in the phenotype datasets and it is unlikely to appear in many of the background datasets.

Therefore, for each phenotype, we tested every possible gene triplet for enrichment (using the hypergeometric test) of triangles in the phenotype datasets with respect to the background datasets. For each seed having a hypergeometric p-value less than 0.01, we ran the SA algorithm, with the constraint that states in that run must be supersets of the initial triplet. Of the 60 non-redundant phenotypes, 47 had at least one signficant seed at a false discovery rate (FDR) of 0.01.

2.2.5. Selection of neighboring states. We defined a neighbor as a state that contains either one more or one less vertex than the current state. We created neighboring states by first determining whether to add to or remove a vertex, then choosing the vertex based on the appropriate probability distribution.

If a cluster has size 3, it consists only of the initial seed, and so a vertex must be added. Conversely, if a cluster has size 30, it has reached the maximum size, and a vertex must be removed. For intermediate values, we proceeded as follows.

Let x be the current cluster. We narrowed the search space of vertices to be added by considering only vertices that have at least one edge to a vertex in x in at least one of the phenotype datasets. This is easily justified because vertices not meeting this criterion could not possibly contribute to x as a dense, phenotype-specific cluster, even as an intermediate step. It can be shown that this set corresponds exactly to $\mathcal{N}_x = \left\{ g : g \notin x, \sum_{h \in x} w_\Delta(g, h) > 0 \right\}$ (see Section 2.1).

The probability of removing a vertex is then given by $p_{rem} = s_0 / |\mathcal{N}_x|$, where s_0 is an estimate of how many vertices will improve the cluster. This is to allow the SA process ample time to consider many neighbors before attempting to remove a vertex, since the number of neighboring vertices vastly outnumbers the number of vertices in a cluster. We heuristically used $s_0 = 20$ as an appropriate average number. In the future, an iterative estimation of s_0 as the average size of the returned clusters might improve the performance of the algorithm.

In the event that a gene is to be removed, it is chosen uniformly from the cluster. When adding a gene, however, we made the probability that a vertex $g \in \mathcal{N}_x$ is added proportional to the sum of the weights of edges from g to the members of x in the differential coexpression graph. Formally, we have: $P(g_a \text{ is added}) = \sum_{a \in x} w_\Delta(g_a, a) / \sum_{b \in \mathcal{N}_x} \sum_{a \in x} w_\Delta(a, b)$.

2.2.6. Annealing schedule. We used the schedule $T_k = T_{max} / \log(k + 1)$, where k is the iteration number and T_k is the temperature at that iteration (Geman and Geman, 1984). The initial temperature for our study was 4 degrees. This schedule form guarantees optimality for long running times. Although it can be argued that the exponential running times required makes this schedule impractical, we found that for an identical number of SA iterations, it resulted in lower-energy clusters than the often-used exponential schedule, $T_{k+1} = \alpha T_k = \alpha^k T_{max}$. We ran the algorithm for a maximum of 1,000,000 iterations per run or until the simulated annealing converged. If the maximum number of iterations was reached, we forced convergence to the best local minimum by a near-greedy exploration of the neighborhood of the best state, achieved by decreasing the temperature to near-zero.

2.2.7. Post-filtering. Recall that we enforced the inclusion of the initial seed triangle in the final result. Clearly, some seeds will result from noise alone, and therefore the final output will not be biologically significant. To remove these clusters, we filtered the SA output clusters by discarding any vertex set not meeting the following criteria: size greater than 6; density greater than 0.66; FDR-corrected phenotype-specificity p-value less than 0.01; and dense in at least 3 datasets related to the target phenotype. After filtering, we merged redundant clusters, defined as pairs clusters for which intersection/union was higher than 0.8.

3. RESULTS

3.1. The modules returned by our algorithm are functionally and conceptually homogeneous

We applied our simulated annealing approach to the 136 microarray datasets, covering 42 phenotype classes that contained initial seeds that were statistically significant after FDR correction. These included a range of diseases (e.g., leukemia, myopathy, and nervous system disorders) and tissues (e.g., brain, lung, and muscle). Starting from the recurrent triangle seeds for each of the 42 phenotypes, we identified 118,772 clusters that satisfied our criteria for a concept-specific coexpression cluster. The number of clusters we found for a given phenotype increased with the number of datasets annotated with it: most of the phenotypes with only a few associated datasets yielded few clusters. The most represented phenotype we studied was "nervous system disorders," which had 15 associated datasets and a total of 22,388 clusters.

We used two different methods to evaluate cluster quality. First, we assessed cluster functional homogeneity by testing for enrichment for specific Gene Ontology (Gene Ontology Consortium, 2006) biological process terms. If a cluster is enriched in a GO term with a hypergeometric p-value less than 0.01, we declare the cluster functionally homogeneous. Of the 118,772 clusters derived from all phenotypes, 78.98% were functionally homogenous by this measure. An advantage of our approach is demonstrated by this validation: since we identified clusters specific to only subsets of all our datasets, we were less likely than previous studies to detect constitutively expressed clusters, such as those consisting of ribosomal genes or genes involved in protein synthesis.

While the GO approach provides information about gene function, it fails to describe its phenotypic implications. To map individual genes to phenotypes, we used GeneRIF (Mitchell et al., 2003). The GeneRIF database contains short statements derived directly from publications describing functions, processes, and diseases in which a gene is implicated. We annotated genes with phenotypes by mapping the GeneRIF notes to UMLS metathesaurus terms as we did with the dataset MeSH headings (see Section 2.1). Similar to GO annotations, we assessed the *conceptual homogeneity* of gene clusters in specific UMLS keywords with the hypergeometric test, enforcing a minimum p-value of 0.01. The proportion of modules that were conceptually homogeneous was 46.83%. Clusters usually show less conceptual homogeneity than functional homogeneity, which is likely due to the sparsity of GeneRIF annotations. There are cases, however, in which GeneRIF performs very well. For example, many of the cancer related phenotypes, such as "Neoplasm Metastasis" and "Neoplastic Processes," show higher GeneRIF homogeneity, which could be attributed to the abundance of related literature. The functional and conceptual homogeneity of clusters derived from different phenotype classes is summarized in Figure 1.

3.2. The modules returned by our algorithm are specific to particular phenotypes

In addition to testing for functional and conceptual homogeneity, we assessed whether the returned clusters were involved in the phenotype condition in which they were found. Again, we used both GO and GeneRIF independently for this.

Recall that each functionally homogeneous module is associated with one or more GO biological functions, and that it is also associated with the phenotype in which it was found. We summarized these GO functions by mapping them to "informative nodes," which we introduced in our earlier work (Zhou et al., 2002), and then tested them for overrepresentation in that phenotype class. This provided, for each of 33

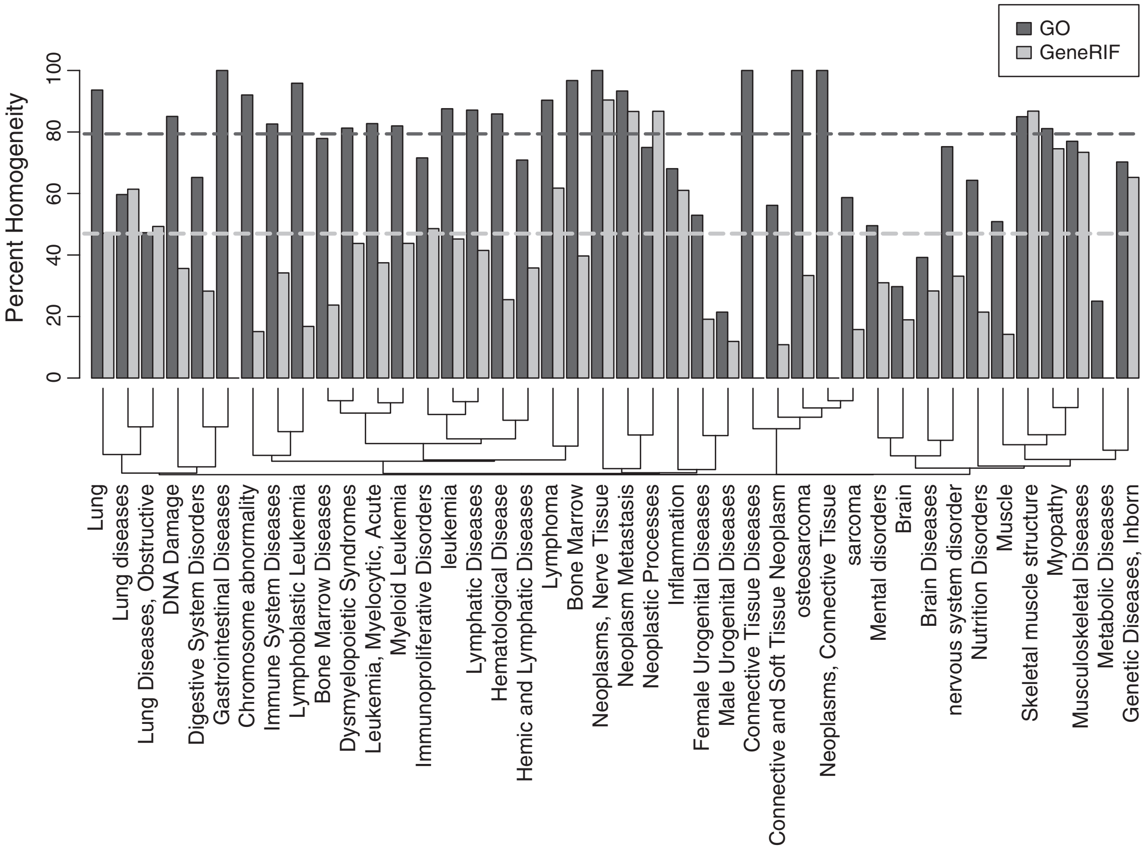

FIG. 1. Cluster homogeneity by phenotype. For each phenotype, the proportion of clusters that are significantly enriched (p-value < 0.01) for a GO biological process (black) or a GeneRIF UMLS concept (gray). The dotted lines show the overall homogeneity for all clusters. The dendrogram shows the distance between phenotypes in terms of dataset overlap.

phenotypes (out of 42 phenotypes having at least one module), a list of gene module functions that are active in that phenotype more often than expected by chance. Many of these GO functions are clearly related to the phenotype in which they were found. For example, the phenotype "Mental disorders" has three GO biological processes related to brain function: "synaptic transmission" (2.3e-62), "neuron differentiation" (5.4e-42), and "central nervous system development" (7.9e-25). Furthermore, our approach identified biological processes related not only to disease phenotypes, but also to tissue phenotypes. For example, the "Skeletal muscle structure" phenotype is significantly enriched with modules homogeneous for the biological functions "muscle system process" (4.0e-221), "actin filament-based process" (1.23e-150), and "skeletal development" (1.53e-03). The functional association between a module's GO function and the phenotype in which it is active suggests that our clusters are indeed linked to the phenotype conditions under which they were identified. In addition to GO informative nodes, we also tested each phenotype for over-representation of UMLS concepts from GeneRIF. This over-representation shows which diseases, tissues, and biological concepts are significantly enriched in each phenotype. In Table 1, we highlight some of these over-represented functions and concepts. The full table can be found on the Zhou Lab group webpage (see Section 4).

The preceding analysis relies on our subjective evaluation of matches between UMLS and GO terms. We reasoned that we could make a more objective analysis with GeneRIF, as it can be mapped directly to the same UMLS terms as the dataset phenotypes. We thus counted the modules that were conceptually homogeneous for the same UMLS annotation as the datasets in which they were identified. Of the 42 phenotypes represented in our study, 26 had one or more matching modules. The proportion of matching modules among total modules in these 26 phenotypes ranged from 0.04% to 33.6%. Although these numbers may not sound immediately impressive, we showed that these proportions are larger than expected

Table 1. Selected UMLS Concepts and Their Principal Annotations

Concept	*Total*	*Over-represented GO annotations*	*Over-represented GeneRIF annotations*
Lymphoma	890	Cell cycle phase (9.2e-276)	Lymphoreticular tumor (2.6e-93)
		Cell cycle checkpoint (1.2e-14)	Abnormal hematopoietic and lymphoid cell (2.6e-22)
		Regulation of cell cycle process (3.2e-08)	Low grade B-cell lymphoma morphology (3.5e-19)
		Antigen processing and presentation (7.7e-03)	
Mental disorders	866	Synaptic transmission (2.3e-62)	Schizophrenia (4.3e-12)
		Neuron differentiation (5.4e-42)	Neurons (1.2e-11)
		Central nervous system development (7.9e-25)	Alzheimer's Disease (3.4e-04)
Muscle	584	Muscle system process (7.9e-52)	Heart (1.2e-20)
			Intrathoracic cardiovascular structure (3.1e-19)
			Muscle, striated (8.2e-15)
Myopathy	6328	Actin filament-based process (7.2e-21)	Coronary heart disease (<le-324)
		Muscle system process (4.6e-06)	Disorder of skeletal muscle (<le-324)
Neoplastic Processes	1486	Keratinocyte differentiation (<1e-324)	Lung neoplasms (2.6e-207)
		Cell cycle checkpoint (1.0e-124)	Triploidy and polyploidy (2.8e-179)
		Regulation of mitotic cell cycle (7.4e-122)	Tumor of dermis (8.2e-123)
		Cell division (1.7e-107)	Glioma (6.4e-121)
Skeletal muscle structure	6719	Muscle system process (4.0e-221)	Musculoskeletal structure of limb (4.3e-46)
		Actin filament-based process (1.2e-150)	Heart (7.6e-46)
		Skeletal development (1.5e-03)	

We annotated clusters using Gene Ontology and GeneRIF, as detailed in the text. We then identified the annotations that were preferentially found in one concept relative to the others, as assessed by the hypergeometric test (Bonferroni-corrected *p*-values shown in parentheses).

by chance. We used a permutation test to assess the statistical significance of our analysis. For each of 1,000,000 permutations, we randomly assigned existing clusters to the 42 phenotypes that had at least one cluster, maintaining the number of clusters assigned to each phenotype constant. The thirteen phenotypes that were statistically significant after FDR correction are shown in Table 2. The high significance for many of the phenotypes indicates that the low percentages are probably due to a dearth of GeneRIF annotations, and as GeneRIF becomes more comprehensive we would expect the performance to improve in both the percentage of matching clusters as well as the number of phenotypes that are significant. We also found that the UMLS text mining of the GeneRIF database and the MeSH headers is not perfect, so further improvements and refinements in those areas should also improve our validation results.

3.3. *Example modules identified by our algorithm*

Below we illustrate two examples of identified phenotype-specific modules, one from a disease phenotype and another from a tissue phenotype.

The first example is an 8-gene module (CD14, CFP, FCER1G, IFI30, ITGB2, MPO, S100A9, TYROBP) which is specific to the phenotype "leukemia" (Fig. 2a). The module has density higher than 0.67 in five out of the total 136 datasets (GDS1059, GDS1064, GDS1067, GDS1388, GDS1454), all of which are annotated with the phenotype "leukemia," which gives a specificity *p*-value of 2.2e-6.

Strikingly, seven genes are annotated as being involved in "defense response" in the GO biological process database, and the eighth gene, IFI30, is an interferon gene that is known to have immune system function although it is not annotated as such in GO. Additionally, three of the genes are associated with the UMLS concept "Acute Leukemia," including CD14 and ITGB2, which are known to have a direct protein-protein interaction *in vitro*. The dataset specificity, interactions between module members, and near complete immune system functional homogeneity all suggest a role for this module in leukemia.

Table 2. Phenotypes for Which the Annotated Clusters Are Consistent with the Phenotype Class in which They Were Derived

Phenotype	*Total clusters in phenotype class*	*Matching clusters in phenotype class (%)*	*Matching clusters in background class (%)*	q-*value*
Mental disorders	791	3.12	0.17	<4.7e-06
Lymphoma	409	20.11	0.97	<4.7e-06
Myopathy	645	15.46	3.65	<4.7e-06
Musculoskeletal diseases	1,619	2.26	1.33	<4.7e-06
Genetic diseases, inborn	1,470	7.86	1.82	<4.7e-06
Neoplasms, nerve tissue	765	33.60	2.02	<4.7e-06
Neoplastic processes	794	9.08	4.19	<4.7e-06
Nervous system disorder	2,214	4.44	2.69	<4.7e-06
Skeletal muscle structure	154	0.94	0.18	<4.7e-06
Hemic and lymphatic diseases	1,129	1.17	0.65	1.3e-05
Bone marrow diseases	523	1.31	0.52	5.3e-03
Leukemia	460	0.55	0.36	2.9e-02
Muscle	483	1.03	0.31	3.5e-02

The first column indicates a UMLS phenotype. The second column displays the total number of clusters active in that phenotype class. The third and fourth columns show the percentage of clusters annotated with that phenotype in that phenotype class and in the background class, respectively, and the fifth column shows the FDR-corrected *p*-value for the difference between the classes. Statistical significance was calculated by permuting the clusters across the dataset phenotypes 1,000,000 times. Concepts with a *q*-value less than 4.7e-6 were never outperformed by the permutations.

Knowledge of modules such as this could guide further experiments in the study of these diseases, as well as to elucidate the module's regulators.

The second example module consists of 12 genes, and is specific to the ''Skeletal muscle structure'' phenotype (Fig. 2b). Four of the five active datasets study expression in muscle tissue (GDS268, GDS198, GDS563, GDS2055). This cluster is highly functionally enriched, containing seven genes that are annotated with a GO biological process related to muscle contraction. Specifically, in muscle fibers, troponin genes (TNNC2, TNNI2, TNNT3) along with tropomyosin associate with actin (ACTA1) to regulate muscle contraction via binding to the myosin complex (MYH2, MYLPF). The module also contains SLN, which regulates the ATP-dependent transport (ATP2A1) of Ca^{2+} in muscle cells. ALDOA and ENO3 are known

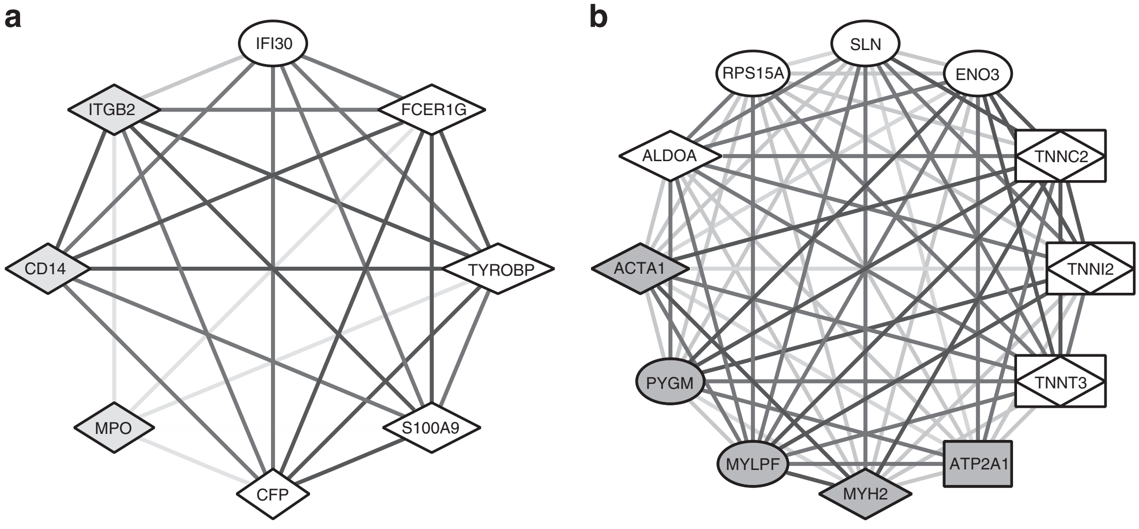

FIG. 2. Two examples of phenotype-specific modules. The opacity of an edge is proportional to the recurrence of the edge in the active datasets. (**a**) A module specific to ''leukemia'' datasets. Genes represented as diamonds are annotated with the GO term ''defense response.'' Shaded nodes represent genes known to be implicated in ''leukemia'' via GeneRIF text mining. (**b**) A module specific to the ''Skeletal muscle structure'' datasets. Genes represented as diamonds and rectangles are annotated with GO terms ''muscle contraction'' and ''regulation of muscle contraction'' respectively. The shaded genes are annotated as ''Muscle'' tissue genes via GeneRIF text mining.

to be expressed specifically in skeletal muscle. The final cluster gene, PYGM, is known to be involved in glycogen metabolism in muscle. In total, 11 of the 12 genes in the cluster are known to be related to muscle, providing strong evidence for the cluster's phenotypic specificity.

4. DISCUSSION

The importance of considering the phenotypic context of gene modules cannot be overstated. Ultimately, molecular understanding is most useful when its macroscopic effects are well understood. In this article, we described a graph-based approach integrating many microarray datasets to derive a genome-wide mapping of coexpression modules to phenotypes.

The provable computational complexity of this problem drew us to stochastic algorithms, and as a result we developed a number of useful graph-mining optimizations to the simulated annealing method. Firstly, we devised a strategy to divide the search space effectively by defining fully connected triplet (triangle) seeds. Secondly, we designed highly robust energy functions that could be linearly combined over a range of weights. And thirdly, we designed a method to prioritize neighbor searching. Overall, we have demonstrated that simulated annealing is a highly effective and adaptable strategy for pattern-mining in graphs.

We associated gene modules with human diseases on a genome-wide scale. The resulting map emphasizes that multiple genes must act together to effect phenotype, and, more specifically, that a gene in different contexts may participate in the manifestation different phenotypes. It has not escaped our notice that our map may represent the largest collection of examples of genetic pleiotropy to date. We reserve the results of this analysis for a future work (Jeffery, 2003b,a; Zhang, 2004).

In this study, we applied our method to microarray data, which is so far the most abundant data measuring the genome-wide molecular activity under different phenotype conditions. We are well aware that microarray data has limitations, and that not all module activities can be assessed with expression profiling. We emphasize, however, that our method is generally applicable to any kind of abundant network data having clearly defined phenotype annotations. One possibility is a dynamically-annotated protein-protein interaction network consisting of conditional interactions. Given the current unrelenting pace of technological innovation in the biological sciences, we envision that a vast amount of genome-wide, phenome-annotated profiling data will soon complement our current view of the genome-phenome association, for not only mRNA but also other molecules, such as protein and miRNA.

5. SUPPLEMENTARY MATERIAL

Availability

The complete catalog of phenotype-specific gene clusters can be found as online Supplementary Material at *www.liebertonline.com.*

Phenotype-specific modules

Table of phenotype-specific gene modules. The file contains Module ID, Module Gene IDs, Module Gene Symbols, Min Density, Enrichment p-value, Module Active Datasets, Module Phenotype Datasets (subset), and Dataset Phenotype. The first column contains a unique identifier for the module in the format c[i]_m[j] which corresponds to module number j from phenotype i. The phenotypes were identified by performing UMLS text mining on the dataset MeSH headings. The full table of the dataset phenotypes can be found below. The second and third columns list the genes found in the module. Negative gene ids are listed for Affymetrix probe ids that did not map to an NCBI gene id. The active datasets are the datasets in which the module had a density greater than 0.66. The module phenotype datasets are the subset of active that were annotated with the phenotype.

Dataset phenotypes

Table containing the UMLS phenotypes associated with at least 5 datasets. The table contains the Phenotype ID that we assigned for the purposes of the study, the UMLS phenotype(s), and the associated datasets.

Dataset description

This file contains the description of the 136 GEO datasets. The file contains dataset ID, number of experimental columns, dataset title, and a description for each dataset.

Go enrichment data

The Gene Ontology enrichment data for each of the phenotypes. Each file begins with a prefix concepts3_n5_##, where ## is the phenotype id. This id can be found in the Dataset Phenotypes file.

ACKNOWLEDGMENTS

This work was supported by the NIH (grants R01GM074163, P50HG002790, and U54CA112952) and the NSF (grants 0515936 and 0747475).

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Asahiro, Y., Hassin, R., and Iwama, K. 2002. Complexity of finding dense subgraphs. *Discrete Appl. Math.* 121, 15–26.

Bader, G.D., and Hogue, C.W.V. 2003. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinform.* 4, 2.

Barrett, T., Troup, D.B., Wilhite, S.E., et al. 2007. NCBI GEO: mining tens of millions of expression profiles–database and tools update. *Nucleic Acids Res.* 35, D760–D765.

Bodenreider, O. 2004. The Unified Medical Language System (UMLS): integrating biomedical terminology. *Nucleic Acids Res.* 32, D267–D270.

Butte, A.J., and Chen, R. 2006. Finding disease-related genomic experiments within an international repository: first steps in translational bioinformatics. *AMIA Annu. Symp. Proc.* 106–110.

Butte, A.J., and Kohane, I.S. 2006. Creation and implications of a phenome-genome network. *Nat. Biotechnol.* 24, 55–62.

Collette, Y., and Siarry, P. 2004. *Multiobjective Optimization: Principles and Case Studies.* 2nd ed. Springer, New York.

Ding, C., and Peng, H. 2005. Minimum redundancy feature selection from microarray gene expression data. *J. Bioinform. Comput. Biol.* 3, 185–205.

Feige, U., Peleg, D., and Kortsarz, G. 2001. The dense k-subgraph problem. *Algorithmica* 29, 410–421.

Freimer, N., and Sabatti, C. 2003. The human phenome project. *Nat. Genet.* 34, 15–21.

Geman, S., and Geman, D. 1984. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE-PAMI* 6, 721–741.

Gene Ontology Consortium. 2006. The Gene Ontology (GO) project in 2006. *Nucleic Acids Res.* 34, D322–D326.

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.

Jeffery, C.J. 2003a. Moonlighting proteins: old proteins learning new tricks. *Trends Genet.* 19, 415–417.

Jeffery, C.J. 2003b. Multifunctional proteins: examples of gene sharing. *Ann. Med.* 35, 28–35.

Kelley, B.P., Sharan, R., Karp, R.M., et al. 2003. Conserved pathways within bacteria and yeast as revealed by global protein network alignment. *Proc. Natl. Acad. Sci. U.S.A.* 100, 11394–11399.

Kirkpatrick, S., Gelatt, C., and Vecchi, M. 1983. Optimization by simulated annealing. *Science* 220, 671–680.

Lage, K., Karlberg, E.O., Størling, Z.M., et al. 2007. A human phenome-interactome network of protein complexes implicated in genetic disorders. *Nat. Biotechnol.* 25, 309–316.

Lander, E.S. and Schork, N.J. 1994. Genetic dissection of complex traits. *Science* 265, 2037–2048.

Mitchell, J.A., Aronson, A.R., Mork, J.G., et al. 2003. Gene indexing: characterization and analysis of NLM's geneRIFs. *AMIA Annu. Symp. Proc.* 460–404.

Risch, N.J. 2000. Searching for genetic determinants in the new millennium. *Nature* 405, 847–856.

Spirin, V., and Mirny, L.A. 2003. Protein complexes and functional modules in molecular networks. *Proc. Natl. Acad. Sci. U.S.A.* 100, 12123–12128.

Srivastav, A., and Wolf, K. 1998. Finding dense subgraphs with semidefinite programming. *Lect. Notes Comput. Sci.* 181–192.

Suman, B., and Kumar, P. 2006. A survey of simulated annealing as a tool for single and multiobjective optimization. *J. Operat. Res. Soc.* 57, 1143–1160.

Yan, X., Mehan, M.R., Huang, Y., et al. 2007. A graph-based approach to systematically reconstruct human transcriptional regulatory modules. *Bioinformatics* 23, i577–i586.

Yip, A.M., and Horvath, S. 2007. Gene network interconnectedness and the generalized topological overlap measure. *BMC Bioinform.* 8, 22.

Zhang, M. 2004. Multiple functions of maspin in tumor progression and mouse development. *Front. Biosci.* 9, 2218–2226.

Zhou, X., Kao, M.-C.J., and Wong, W.H. 2002. Transitive functional annotation by shortest-path analysis of gene expression data. *Proc. Natl. Acad. Sci. U.S.A.* 99, 12783–12788.

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

Address correspondence to:
Dr. Xianghong Jasmine Zhou
Program in Computational Biology
Department of Biological Sciences
University of Southern California
Los Angeles, CA 90089

E-mail: xjzhou@usc.edu