BIOINFORMATICS ORIGINAL PAPER

Vol. 23 no. 7 2007, pages 850–858
doi:10.1093/bioinformatics/btm019

Gene expression

Gene expression network analysis and applications to immunology

Şerban Nacu[1,3,*], Rebecca Critchley-Thorne[2], Peter Lee[2] and Susan Holmes[1]

[1]Department of Statistics, Stanford University, Stanford CA 94305, [2]Stanford School of Medicine, Stanford CA 94305, and [3]Ecole Normale Superieure, Paris, France

Received on August 5, 2006; revised on January 17, 2007; accepted on January 18, 2007

Advance Access publication January 31, 2007

Associate Editor: Martin Bishop

ABSTRACT

We address the problem of using expression data and prior biological knowledge to identify differentially expressed pathways or groups of genes. Following an idea of Ideker *et al*. (2002), we construct a gene interaction network and search for high-scoring subnetworks. We make several improvements in terms of scoring functions and algorithms, resulting in higher speed and accuracy and easier biological interpretation. We also assign significance levels to our results, adjusted for multiple testing. Our methods are succesfully applied to three human microarray data sets, related to cancer and the immune system, retrieving several known and potential pathways. The method, denoted by the acronym GXNA (Gene eXpression Network Analysis) is implemented in software that is publicly available and can be used on virtually any microarray data set.

Contact: serban@stat.stanford.edu

Supplementary information: The source code and executable for the software, as well as certain supplemental materials, can be downloaded from http://stat.stanford.edu/~serban/gxna.

1 INTRODUCTION

A central problem in biology is the identification of genes or pathways involved in diseases and other biological processes. The development of microarray technology (Schena *et al*., 1995) and other high-throughput techniques has enabled massively parallel approaches to this problem. In a typical experiment, two or more phenotypes are compared, with several replicates used for each phenotype. Each replicate measures expression data for a large number of genes.

Standard analysis starts with filtering and normalizing the data, followed by the computation of test statistics for each gene, comparing expression levels in different phenotypes. Various techniques (Dudoit *et al*., 2002) can be used to account for the large number of genes being tested. Finally, genes are sorted in increasing order of their adjusted *p*-values, and the most significant genes are used to generate biological hypotheses and/or subjected to experimental validation. For a survey, see, for example, Slonim (2002).

*To whom correspondence should be addressed.

The power of this strategy is limited by the fact that it analyzes genes one-at-a-time. Real genes function in concert rather than alone, their products interact with each other and with DNA and there has been a lot of interest in methods that analyze groups of genes. One of the earliest approaches has been hierarchical clustering. This method produces useful visualizations, but it lacks a sound statistical basis. It is also entirely driven by experimental data, not using any prior biological knowledge about the genes of interest.

Several other methods were developed, from visualization tools such as GenMAPP (Dahlquist *et al*., 2002) to algorithms such as GO (Gene Ontology) analysis; see Curtis *et al*. (2005) for a survey. A useful way to classify them is according to the way they represent prior knowledge:

(1) The Gene Ontology (The Gene Ontology Consortium, 2000) is a database of biological terms structured as a directed acyclic graph. A typical analysis seeks GO terms that contain a large number of differentially expressed genes. Pros include speed, simplicity and the ability to assign *p*-values. However, this analysis ignores a lot of information (all the top genes are assigned equal weight, regardless of their scores) and often outputs very general GO terms that are not very useful.

(2) Another method starts with a predefined list of groups of genes, such as known pathways. This is used in the GSEA algorithm (Subramanian *et al*., 2005) and also in Tian *et al*. (2005). Every such group is assigned a score that is essentially the average of the test statistics of its member genes; groups with high scores are more likely to be differentially expressed. *p*-values can be obtained by permutation methods.

(3) Yet another method uses a simple but powerful idea of Ideker *et al*. (2002); it has also been used in Rajagopalan and Agarwal (2005), Sohler *et al*. (2004) and Cabusora *et al*. (2004). Rather than using a list of known pathways, prior knowledge is represented as an interaction network. The nodes of the graph correspond to genes; there is an edge between two nodes if their genes interact. Various types of interactions may be considered, such as protein-to-protein, protein-to-DNA or co-expression. A group of related genes corresponds to a connected subgraph of the

interaction graph. Each subgraph is assigned a score (typically the sum of the scores of its component genes), and a search algorithm is used to find subgraphs with high scores.

The interaction network is a more precise way to represent information than lists of genes or pathways, as it describes which genes are closely connected *within* a given pathway. Hence, it has the potential to detect more subtle signals, such as local disturbances within known pathways, as well as within pathways that have not yet been described. This comes at the price of increased complexity, making it more difficult to design fast algorithms and compute significance levels.

We essentially follow the approach (3), but also integrating some elements of (2). We make several improvements in terms of design, scoring and algorithm that address the problems mentioned above. These are described in detail in the Methods section; we briefly emphasize the most important ones here:

- We focus on finding small networks, which are easier to interpret and validate.
- We use a fast algorithm that allows us to compute *p*-values, adjusted for multiple testing using the FWER (familywise error rate) method.
- While most previous work on interaction networks was done on simple organisms such as yeast and bacteria, we succesfully apply our method on human data.

2 METHODS

2.1 Data

2.1.1 Expression data We test the algorithm on three data sets: two are new, while one was previously published.

Lymphocyte data. This data set was generated as part of a study on the role of the immune system in cancer Critchley-Thorne *et al.*, (2006*b*). Blood samples were collected from melanoma and healthy patients; there were 26 healthy and 30 melanoma phenotypes. Lymphocytes were sorted according to their type into B, CD4 T, CD8 T and NK (natural killer) cells. In order to increase power, the various types were pooled together (see the discussion in Section 2.5.3). Gene expression data was obtained using 56 Agilent Human 1A version 2 microarrays. After removing saturated genes, there were 20 901 genes left.

Regulatory T-cell data. Another experiment Critchley-Thorne *et al.*, (2006*a*) in the same study compared the expression profiles of regulatory CD4 T-cells in healthy controls and melanoma patients. The same microarray platform and protocols were used. Due to the difficulty in isolating large amounts of regulatory T-cells, there were only four healthy and four melanoma phenotypes.

Serum data. The third data set was generated by Chang *et al.* (2004) and is publicly available in the SMD database. The authors studied similarities between the biology of tumor growth and wound recovery. To characterize wound response, they compared the expression profiles of 50 fibroblast cultures in the presence and absence of serum. We selected this data set because of its relevance to the study of the immune system in cancer.

All three data sets were processed using the R packages `bioconductor` (Gentleman *et al.*, 2004) and `limma` (Smyth and Speed, 2003). M-values were computed by a vsn-transform (Huber *et al.*, 2002) and normalization between arrays, and *t*-statistics were computed based on the normalized M-values. We use these as inputs to our analysis.

2.1.2 Gene interaction data Interaction data was downloaded from two public databases: EntrezGene (December 2005) and 33 human pathways in KEGG (March 2006). The data was represented as an undirected graph where each node is a gene and two nodes are connected by an edge if their genes interact. Loops (nodes connected to themselves) were eliminated. This results in a graph with 7180 nodes and 27 082 edges (Fig. 1). The highest degree is 180 (gene TP53). The most common degree is 1 (genes interacting with only one other gene); there are 1869 such nodes.

2.2 Filtering

To compare the single-gene and network-based approaches, we only look at genes with at least one interaction. To reduce the number of false positives, we select for multiple testing only the genes that show enough variability across arrays (typically we use a threshold of 0.5 for the standard deviation of the M-value).

2.3 Scoring functions

Given a gene or a set of genes, we need to compute a score that measures to what extent it is differentially expressed. We discuss several possible choices and their pros and cons.

2.3.1 Scores for a single gene Consider first a single gene. The most popular scoring function is the *t* statistic:

$$T_i = (\mu_{i1} - \mu_{i0})/\sqrt{\sigma_{i1}^2/n_1 + \sigma_{i0}^2/n_0} \quad (1)$$

where the mean and standard deviation μ_{i1}, σ_{i1} are for gene i and the case phenotype, and μ_{i0}, σ_{i0} are for gene i and the control phenotype.

Several alternatives exist. For simplicity, we focus on the *t* statistic, though most of our methods remain valid if we replace it with any reasonable competitor. For example, *t* can be converted into a *z*-score using the Student and normal distributions. This is not required (the *p*-values we derive do not assume normality) but it makes it easier to compare single gene statistics across different experiments. For large samples, the Student distribution is close to normal, so the effect of the transform is likely to be small.

2.3.2 Scores based on averaging test statistics Now consider a set $S = \{g_1, \ldots, g_k\}$ of k genes. A natural way to assign it a score is to average the scores of its individual genes, leading to the scoring function used in Tian *et al.* (2005):

$$f_1(S) = \frac{1}{k}\sum_{i=1}^{k} T_{g_i} \quad (2)$$

We refer to this class of score functions as ΣT. Often, pathways contain both upregulated and downregulated genes; as pointed out in Ideker *et al.* (2002), this can be captured by taking absolute values of the test statistic, possibly at the cost of creating more false positives:

$$f_2(S) = \frac{1}{k}\sum_{i=1}^{k} |T_{g_i}| \quad (3)$$

Either way, the distribution of the score depends on the set S (e.g. on its size), so ideally it should be normalized before it is used to compare different sets. Tian *et al.* (2005) propose a nonparametric normalization method: permutations of the phenotypes are used to estimate the null distribution of the score, and the score is adjusted by essentially

Fig. 1. A subset of the gene interaction network.

replacing it with its quantile. The advantage of this method is that it is nonparametric. Its main drawback is that estimates of the null distribution are reliable only if enough permutations are used, which may not be possible if the number of phenotypes is small [see the discussion of the minP algorithm in Dudoit *et al.* (2002)]. It is also more computationally intensive. We implement a variant of this idea called maxTscaled, discussed below in Section 2.5.2.

The alternative is to make some kind of parametric assumption. For example, Ideker *et al.* (2002) normalize all sets of size k by comparing with a single reference distribution, computed by sampling from random sets of k genes. Ignoring the small effect due to sampling without replacement, this amounts to using

$$f_3(S) = \frac{1}{\sqrt{k}}\left(\sum_{i=1}^{k}|T_{g_i}| - k\mu\right) \tag{4}$$

where μ is the mean of $|T|$ over all genes. The implicit assumptions here are that (1) the normalization need only depend on the size of the set, and (2) individual gene scores are independent. The latter assumption in particular is not realistic; it would be better to normalize by sampling among connected sets of k genes, leading to

$$f_4(S) = \frac{1}{\sigma_k}\left(\sum_{i=1}^{k}|T_{g_i}| - \mu_k\right) \tag{5}$$

where μ_k and σ_k are the mean and standard deviation score for random connected sets of k genes. Here 'random' need not mean 'uniformly random'; ideally the sampling should be similar in spirit to the one used in the search algorithm. Also, μ_k need not equal $k\mu$ (some nodes may be sampled more than others), and σ_k need not be proportional to $\sqrt{k}$; one would expect it to scale like k^α for some exponent $1/2 < \alpha < 1$, where 1/2 would correspond to independence and 1 to full dependence. This leads to

$$f_5(S) = \frac{1}{k^\alpha}\left(\sum_{i=1}^{k}|T_{g_i}| - k\mu\right) \tag{6}$$

2.3.3 Scores based on averaging gene expression Scores based on single-gene test statistics ignore most of the correlation structure among genes. The alternative is to first sum the expression levels for the genes in the group S within each microarray, and then take the test statistic. We denote this class of score functions by $T\Sigma$.

Let X_{ij} be the expression level (e.g. normalized M value) for gene i on array j. First we compute the group expression:

$$S_j = \sum_{i=1}^{k} X_{g_i j} \tag{7}$$

and then compute the score of the group S as the t statistic of those values:

$$f_6(S) = (\mu_{i1} - \mu_{i0})/\sqrt{\sigma_{i1}^2/n_1 + \sigma_{i0}^2/n_0} \quad (8)$$

where the mean and standard deviation μ_{i1}, σ_{i1} are for the set $\{S_j\}$ where j is a case, and μ_{i0}, σ_{i0} are for $\{S_j\}$ where j is a control.

To allow for both up- and downregulated genes in the same pathway, we can include signs in the group expression formula:

$$S_j = \sum_{i=1}^{k} \epsilon_i X_{g_i j} \quad (9)$$

where ϵ_i is -1 if gene i is underexpressed in cases (its t statistic is negative) and $+1$ otherwise. As noted before, this yields a more sensitive scoring function, but may also produce more false positives.

Unlike ΣT, this method takes into account probe correlations across arrays. It is also less likely to require normalization: taking the t statistic at the last step adjusts for differences in group means and variances. On the downside, it is computationally slower. It also works best when the genes in the group have variances of the same order of magnitude, an assumption that does not always hold. This last problem can be avoided by rescaling gene scores to equalize their variances.

We discuss the performance of various score functions in the results section.

2.4 Group selection and search algorithms

Given a scoring function, we need to find groups of interacting genes with high scores. There are two possible approaches: go through a limited list of pre-defined groups and select the ones with high scores, or search for high-scoring sets among all possible sets subject to some structural constraints (e.g. being connected).

2.4.1 Using pre-defined groups One option is to extract pathways or groups of related genes from databases such as KEGG and score each of them. This approach is dependent on the quality of the database, but has been applied succesfully in Tian *et al.* (2005). However, many pathways extracted this way contain many genes, so it is unlikely to detect changes that only affect a small part of a pathway. It will also miss pathways that are not in the database.

To address these problems, we look at the local neighborhood of nodes in the gene interaction network. This is useful in situations where a gene appears to be differentially expressed, but its test statistic is not significant. Because of the multiple testing adjustments, this occurs quite frequently in microarray data (for example, a t-statistic of 3 is often not significant). However, if both a gene and the genes it interacts with have relatively high test statistics, this is a good indication that the effect is real.

To make this formal, given a node x and a positive integer r, we consider the ball $B(x, r)$ centered at x with radius r, which is the set of all nodes that are connected to x by a path with at most r edges. For example, $B(x, 0)$ is just x; $B(x, 1)$ consists of x and its immediate neighbors; $B(x, 2)$ consists of x, its immediate neighbors and their neighbors; and so on.

We set r to some small value (for example, $r = 1$), compute the score of $B(x, r)$ for all genes x in the network and sort them according to these scores. This can be seen as a generalization of standard microarray analysis (and in fact the two are the same for $r = 0$), using the interaction graph to smooth gene expression values. It is fast and simple to implement. Some overlap is to be expected between the top genes found this way and the ones found by traditional techniques, but it does have the potential to discover new interesting genes. However, it will not perform well for high degree nodes (genes with many interactions) when only a few of the neighbors have high scores. This problem is addressed by adaptive search algorithms.

2.4.2 Using the subgraph search algorithms

Selection of target groups. Given the gene interaction network, we want to find differentially expressed pathways or groups of related genes. We need to determine what kind of objects to search for, and the simplest approach is to look for sets of interacting genes, hence for connected subgraphs of the interaction graph. This is the approach we have primarily followed, and in this article the terms 'pathway', 'network' and 'subgraph' are mostly used interchangeably.

From the biological standpoint, a pathway is much more than just a set of interacting genes. Graph searches can easily yield connected subgraphs that do not form a pathway, so this simple definition may yield to overfitting. To control this problem, we compute objective, permutation-based significance levels (Section 2.5). These are important to have in any multiple testing context, but particularly so when searching a large sample space.

Ideally our target networks should model as closely as possible the structure of real pathways, and in future work we plan to include information such as pathway motifs and gene interaction type and direction. Among the methods discussed so far, using balls is most likely too coarse, while allowing any connected subgraph is likely too loose. We add a third option, where the adapted search algorithm only searches for chains (each gene being added must interact with the last gene that was added). This attempts to capture the structure of a sequence of genes within a pathway that successively activate one another. We discuss the performance of various methods in the results section.

Search algorithms. Since the problem of finding the maximal subgraph of a generic graph is NP-hard (Ideker *et al.* 2002), various approximate algorithms have been proposed. Ideker *et al.* (2002) use simulated annealing, however, this is slow and tends to produce large subgraphs that are difficult to interpret. Rajagopalan and Agarwal (2005) offer several improvements, but these are based on heuristics that may not be optimal and require estimation of additional parameters.

We are primarily interested in small networks, so we use a different approach, where we start with a seed vertex and gradually expand around it. After k steps, we will have constructed a connected subgraph G_k with k nodes. Let N_k be the set of all nodes that are outside G_k but have at least one neighbor on G_k. We update G_k by choosing a vertex in N_k and attaching it to G_k.

The choice can be done in various ways. One natural way is to use a greedy algorithm: pick a vertex such that the new graph has maximal score. Variants of this are used in Sohler *et al.* (2004) and Breitling *et al.* (2004). It is fast, but reduces the number of subgraphs searched and may get stuck in local maxima.

An alternative is to use a randomized algorithm: pick a random vertex, with higher probabilities assigned to vertices that yield high scores. This is similar in spirit to Metropolis/Markov Chain Monte Carlo algorithms and avoids some of the problems of the greedy search. However, it is slower, which puts it at a disadvantage when computing resampling-based p-values. We found that, in practice, greedy search works reasonably well, so we decided to use it for its speed and simplicity.

There are two other ingredients to the algorithm: starting and stopping rules. *A priori* any node can be used as a root (starting node). This may lead to overlapping networks, but the multiple testing adjustment methods (Section 2.5.1) can in principle adjust for that. If we desire to reduce overlapping, we can use filters (for example, require that all roots have a certain degree) or require that roots be relatively far from each other in the graph distance. Both options are implemented in our software.

We can stop the search either when reaching a certain size (fixed-size search; we tried sizes of 5, 10 and 20) or when adding any extra node decreases the score of the current subgraph (flexible-size search). The former is simpler, at the cost of an artificial constraint, and less sensitive to normalization (all graphs have the same size, so their scores are easier to compare). The latter is more natural, but depends on the scoring function being used.

2.5 Computing significance levels

We would like to assign *p*-values to the graphs identified by the search algorithms. Clearly, adjustment for multiple testing is required: searching a large network will yield some high scoring subgraphs by mere chance, even if they have no biological significance.

The two standard measures in multiple testing problems are FWER (the family-wise error rate) and FDR (false discovery rate). FWER is more conservative, thus selecting fewer hypotheses as statistically significant; indeed, the GSEA algorithm (Subramanian *et al.*, 2005) switched from FWER to FDR because of difficulties in getting any significant genes in some experiments. However, FWER provides much better protection against false positives. Our goal is to obtain results that were at least partially normative, rather than merely exploratory. Hence, we choose to control FWER.

2.5.1 Permutations Since our algorithm uses root nodes, the nonparametric techniques developed for standard microarray analysis (Dudoit *et al.*, 2002) can be applied. In the two-phenotype case (n_0 controls and n_1 cases), the indices are permuted, thus relabeling some controls as cases and viceversa. The analysis (scoring and graph searching starting from each node) is repeated for each permutation. If enough permutations are available, this gives a reasonable estimate for the null distribution of the subgraph scores, and allows us to compute adjusted *p*-values that control the FWER.

Rajagopalan and Agarwal (2005) propose using permutations of the genes instead of the phenotypes. However, as discussed in Tian *et al.* (2005), this tests a different null hypothesis: whether genes are different, not whether phenotypes are different. While easier to implement, the *p*-values it produces are not relevant to our main question (Are the subgraphs we find truly differentially expressed among phenotypes?).

We note that the assumption of subset pivotality (Dudoit *et al.*, 2002), which guarantees strong control of the FWER, is not completely correct in our setting, as different subnetworks may overlap. However, since our search method is essentially local and produces small graphs, deviations from subset pivotality are likely to be smaller than for methods [such as in Tian *et al.* (2005) or GSEA] that involve large sets of genes.

2.5.2 maxT vs. minP Dudoit *et al.* (2002) discuss two methods for computing FWER, both implemented in the R package `bioconductor`. The maxT algorithm assumes that the null distribution is the same for all objects (in our case, for the scores of graphs obtained from different root genes). The minP algorithm makes no such assumption, and essentially replaces the null test statistics with their quantiles. While in theory minP seems superior, research shows that it may need a large number of permutations (up to 1 000 000) to obtain good estimates of the null, and that maxT often performs better for fewer permutations.

Our goal was to have a fast algorithm, so we decided to use fewer permutations. We implement two algorithms: the standard maxT, and a version called maxTscaled, which adjusts each *t*-statistics by subtracting its null mean and dividing by its null standard deviation; this is essentially a parametric (and faster) version of minP. Since we normalize group scores to reduce dependency on group size, it is realistic to assume that the null distributions do not vary too much and hence maxT can have adequate performance.

Unless specified otherwise, the *p*-values in the results section are obtained using $N = 1000$ permutations.

2.5.3 Choice of permutations Depending on experiment design, uniform random permutations may not capture the null hypothesis. The lymphocyte data is one example: to gain power, we pool data for several kinds of cells (B, CD4, CD8, NK). Thus, in addition to the main phenotype (healthy or melanoma), there is a 'ghost' phenotype (cell type). The null hypothesis asserts that healthy and melanoma are similar; it does not require that B and CD4 be similar, and in fact we know they are not. Hence it is desirable to use permutations that preserve cell type; we call them 'invariant' permutations. We implement this option and use it to analyze the lymphocyte data; the significance levels are better than the ones obtained using uniform random permutations.

3 RESULTS

The value of a new method can be judged by asking two questions: Does it tell us anything new? Does it tell us anything useful? For each of our three data sets, we run our algorithm and compare our results with ones obtained using standard single-gene analysis. Our method is validated if we obtain (1) statistically significant genes and pathways that (2) are biologically relevant and (3) are not obtained by other methods.

The ultimate proof of relevance is of course biological confirmation, and in some cases it is possible to check that our method retrieves known pathways. When this cannot be done, then we require statistical significance (low adjusted *p*-values) and also check gene annotations for connections with the biological problem at hand. The latter is inherently biased (it is easier to make up stories after seeing the results) but still helpful.

The networks described in this section are obtained as follows. For each of the three data sets, we run the algorithm twice, once looking for balls centered at each gene (Section 2.4.1) and once performing an adapted graph search (Section 2.4.2). For each set of results, we report the adjusted *p*-values for the top-most-significant networks and select the ones with the most interesting annotations. We leave out several networks that are statistically significant, but whose annotation is missing.

We conclude the section with a comparison of various parameter values, scoring functions and search methods.

3.1 Lymphocyte data

In a single-gene analysis, we obtain seven significant genes (FWER, $p < 0.05$). Several top genes are downregulated and associated with the Jak/Stat signaling pathway, suggesting its disruption may be a cause or effect of tumor development. See Critchley-Thorne *et al.* (2006*b*) for a more detailed discussion. Network-based analysis retrieves this result, and more.

Using balls of radius 1 only yields four significant genes, three of which are new: TNFRSF1B, CXCL9 and CXCL11. These genes are also common in the top subgraphs in the adapted search algorithm. Adapted search (fixed search depth 10) yields in fact 99 subgraphs with $p < 0.05$, but many overlap.

One of the top-scoring networks is represented in Figure 2. It includes STAT1, but also the chemokine ligands CXCL9,

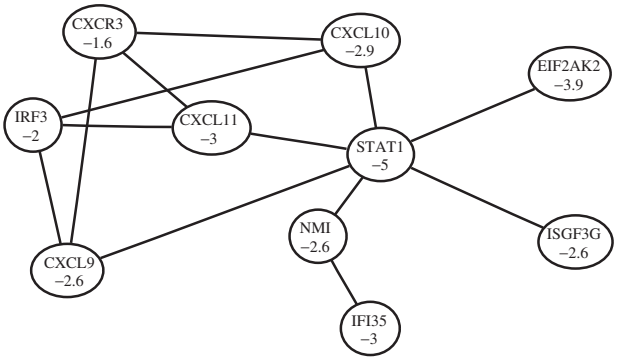

Fig. 2. A chemokine pathway is downregulated in melanoma. Each node contains the gene name and its t statistic.

CXCL10 and CXCL11, which bind to the chemokine receptor CXCR3. The ligands have high scores ($|t|>2.5$) but are not large enough to be significant as single genes after adjusting for multiple testing. As a group, however, they are are significant; they strongly suggest that a chemokine pathway is downregulated in melanoma patients.

Looking more closely at TNFRSF1B (tumor necrosis factor receptor superfamily, member 1B) also elucidates an interesting pathway that is altered in melanoma lymphocytes versus healthy. TNFRSF1B associates with TRAF1 ($t = 1.77$), which belongs to the TNF receptor-associated factor family that interact with and transduce signals for members of the TNF receptor superfamily. TRAF1 binds to TRAF2, which contains a C terminal homology domain that enables association with the cytoplasmic domain of TNF receptors. TRAF1 thereby indirectly interacts with TNF receptors. Of the eleven TNFRSFx genes that interact with TRAF1, nine are overexpressed in lymphocytes from melanoma patients, and four have $t>2$. TNF is a pleiotropic cytokine involved in biological processes such as cell proliferation, differentiation and apoptosis. Alterations in these processes in lymphocytes may be part of the mechanism of immune dysfunction in cancer.

This TNF receptor-signaling network is unusual because most of its genes are overexpressed in melanoma lymphocytes, while most genes identified by standard single-gene techniques are underexpressed in melanoma.

3.2 Regulatory T-cell data

The analysis of this data set is more challenging, due to the small number of arrays (four healthy, four melanoma). The number of available permutations is small, and significant p-values are difficult to obtain. In a standard single-gene analysis, the lowest value obtained is $p = 0.46$ (gene TCN2).

For this data set, a network-based analysis actually produces better p-values. Considering single genes together with their immediate neighbors, and averaging first within each array, the top balls are centered at TESK2 ($p = 0.08$) and GULP1 ($p = 0.11$). None appears in the top 100 genes in the single-gene analysis.

These all lead to potentially interesting findings. TESK2, testis-specific kinase 2 ($t = -3.7$) is involved in actin cytoskeletal reorganization and has only one neighbor, YWHAB ($t = 3.2$). YWHAB encodes a protein belonging to the 14-3-3 family, which bind to phosphoserine-containing proteins and may function in transducing mitotic signals to the cell cycle machinery. YWHAB is known to inhibit TESK2, consistent with the signs of their t-values.

The engulfment adaptor PTB domain GULP1 ($t = -1.8$) also has only one interactant, LRP1 ($t = -2.9$), a low-density-lipoprotein-receptor-related protein. GULP1 and LRP1 specifically interact during rapid clearance of apoptotic cells, e.g. in tissue turnover and inflammation. It is interesting that, despite their relatively low test statistics, they become significant when taken together.

In a graph search for high-scoring subgraphs, the top result has $p = 0.18$ (Fig. 3). It is rooted at the tyrosine-protein kinase LYN ($t = -1.8$). LYN is an important component of the KEGG B-cell receptor signaling pathway; it phosphorylates CD19 ($t = -6.4$), BTK ($t = -4.5$) and SYK ($t = -2.4$; not drawn). LYN also interacts with EPOR, the erythropoietin receptor ($t = 3.3$). Binding of erythropoietin to its receptor activates JAK2 tyrosine kinase, resulting in activation of STAT5.

EPOR also interacts with SOCS2, a STAT-induced STAT inhibitor that is upregulated in melanoma ($t = 3.9$), which in turn interacts with IL12B1, IL22RA2 and IL22RA1 ($t = -3.9, -2.6, -2.8$, respectively) cytokine receptors of the interleukin family. SOCS2 negatively regulates cytokine signaling by interacting with the cytoplasmic tails of cytokine receptors such as interleukin receptors. This network analysis suggests that these signaling pathways are disturbed in T regulatory cells from the melanoma patients and may provide insight into the hypothesized alterations in function of these cells in the cancer state.

3.3 Serum data

Chang *et al.* (2004) studied the gene expression response of fibroblasts to serum, which is a major initiator of wound healing responses. Since wound healing is a complex process, we predicted that many genes would be differentially expressed in serum-treated versus untreated fibroblasts. In a single gene analysis, out of 842 genes remaining after filtering, there are 22 differentially expressed genes (FWER, $p<0.05$).

Network analysis using balls of radius 1 produces similar p-values (21 balls with $p<0.05$) and yields several new genes. The top ball ($p = 0.001$) consists of the gene KLKB1 ($t = -1.2$) and its neighbors SERPINA5 ($t = 1.5$) and TFPI2 ($t = 2.7$). Individual t scores are fairly low, but taken together they yield the highest-scoring ball. KLKB1 (plasma kallikrein) is a serine protease that functions in blood coagulation, fibrinolysis and complement fixation, and so the altered expression of KLKB1 due to serum treatment is consistent with the known roles that fibroblast play in wound healing.

The adapted graph search (flexible size, maximum depth 20) yields 80 sets with $p<0.05$; however, there is significant overlap between the sets at the top. The set in Figure 4 is rooted at the gene SMAD3 and has $p = 0.001$. Unsurprisingly, it contains several cell-cycle-related genes like CDK2, CDC2, CDC25C and CDC27.

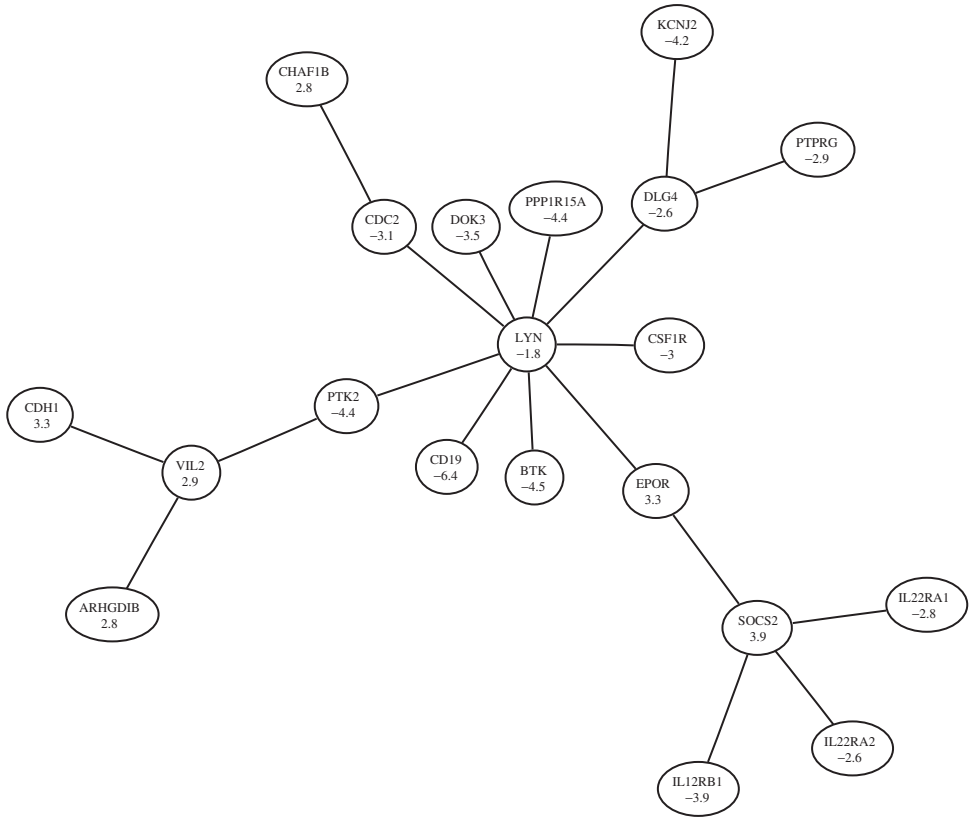

Fig. 3. The top-scoring network for regulatory T-cells.

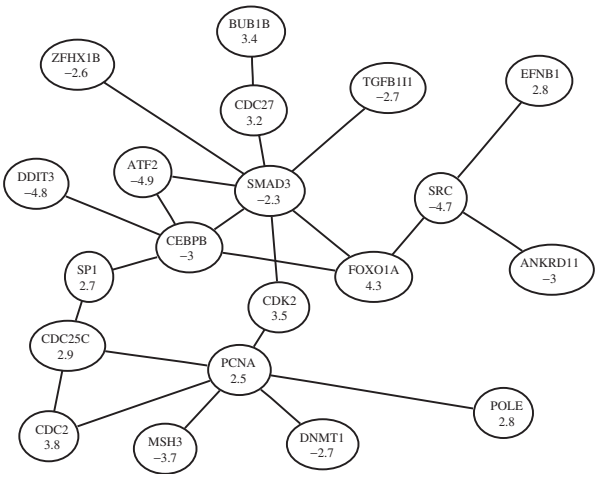

Fig. 4. A network involved in wound healing.

CDK2 ($t = 3.5$) is known to inhibit SMAD3 ($t = -2.3$) by phosphorylation and thereby regulate the anti-proliferative function of SMAD3. SMAD3 also interacts with TGFB1I1, the expression of which is induced by transforming growth factor beta 1. The action of TGF-beta is critical for resolution of inflammatory infiltrate and as such TGF-beta is a major mediator of normal wound healing. SMAD3 and SMAD7 are target genes of TGF-beta, further indicating the involvement of these SMAD genes in wound healing. This is reinforced by the statistics of two related genes not included in the subgraph: SMAD2 ($t = -1.7$) and TGFB2 ($t = -2.8$).

It is not straightforward to compare our results with the original results in Chang *et al*. (2004), as the authors' goal was to derive a global signature of wound healing, rather than identify individual genes; they also used different methods of normalization and analysis (the SAM algorithm). However, many of the genes we identify (KLKB1, SMAD3, the TGF genes) seem absent from the list of differentially expressed genes available as an online supplement to Chang *et al*. (2004). They are also not among the top genes in our standard, single-gene analysis. Our method identifies new networks of fibroblast serum-response genes and may further the understanding of the process of wound healing.

3.4 Comparing various methods

3.4.1 The scaling exponent for balls We consider balls of radius 1 centered at all genes having at least one interaction, and simulate the null distribution of their scores using $N = 10\,000$ permutations of the phenotypes. To compute the scaling exponent α for the ΣT scoring function, we perform a linear regression of the log of the standard deviation on the log of the number of nodes in each ball.

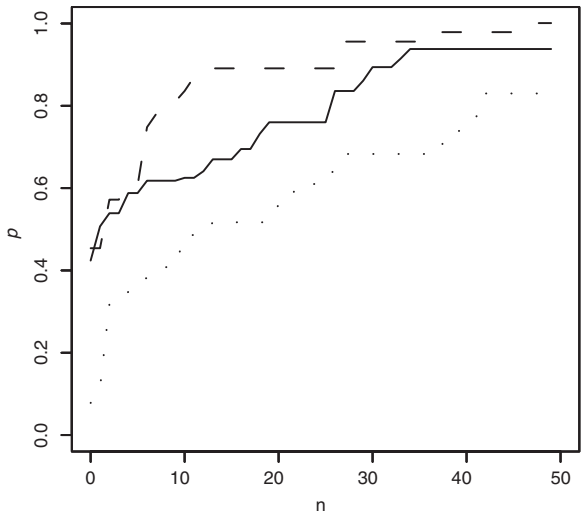

Fig. 5. Comparing three scoring functions for the regulatory T-cell data set. The x-axis has gene ranks, the y-axis has adjusted p-values. $T\Sigma$ (dotted curve) yields better p-values than ΣT (solid curve) and ΣT rescaled (dashed curve). Same qualitative behavior occurs for the other two data sets.

We obtain $\alpha = 0.60$ for the lymphocyte data set, $\alpha = 0.59$ for the regulatory T-cell data set and $\alpha = 0.49$ for the serum data set (the latter result is likely noisier, as the sample size is significantly smaller). This suggests that the optimal α is larger than 0.5, so neighboring genes tend to be positively correlated.

3.4.2 Scoring functions and permutation-based rescaling We compare the performance of the ΣT and $T\Sigma$ scoring functions for balls of fixed radius (the current implementation of adapted search only uses ΣT since it is faster). $T\Sigma$ yields better p-values (Fig. 5).

Permutation-based rescaling does not seems to improve ΣT; it also underperforms in the adapted search case. This does not change when increasing the number of permutations from 1000 to 10 000. Hence the behavior of maxTscaled appears similar to the behavior of minP described in Dudoit *et al.* (2002). We did not try larger numbers of permutations, as the algorithm slows down considerably and memory availability becomes an issue.

3.4.3 Adapted search parameters

Graphs versus chains. Restricting the search to chains instead of any connected subgraph does not seem very productive. The top sets in the two methods tend to overlap, but the p-values are higher for chains. This suggests that the chain model does not capture well the biology of pathways. The performance of chains may improve by using directed edges.

Graph size. We compare the algorithm for flexible graph size and fixed sizes of 5, 10 and 20. We do not find any strong size effect: different sizes provide better p-values for different data sets. In practice, we find that sizes of 10 and 20 both worked well. The results of flexible-depth search are influenced by the scaling exponent α; values close to 1 will make adding nodes more difficult and result in smaller networks. We compare four values of α: 0.5, 0.6, 0.7 and 0.9 for the three data sets. Again, we find no strong effects; $\alpha = 0.5$ or 0.6 worked well in most cases.

4 DISCUSSION

Our aim was to design an algorithm that uses interaction networks to obtain results not found by single-gene analysis. Remarkably, our method yields interesting findings in each of the three data sets we tried. This is one of the first succesful applications of this kind of analysis to human expression data.

Each data set conveniently illustrates a different use for the algorithm. For the lymphocyte data, simply looking at the top genes in single-gene analysis yields strong biological hypotheses; network analysis refines and reinforces them, and provides as well some new ones. For the regulatory T-cell data, traditional analysis yields no significant genes, but our algorithm identifies groups of genes with p-values very close to the conventional threshold of 5%. In the serum data, both methods find a large number of significant genes, but these do not completely overlap, and have different orderings. Network analysis highlights groups of high-scoring, interacting genes, and suggests several interesting pathways.

We offer several improvements over previous methods. Perhaps most importantly, we assign significance levels to the networks we find, so we can state with a high degree of confidence that they reflect underlying biological differences, rather than random chance. We focus on small networks, since they are easier to study and interpret. We also introduce the new scoring function $T\Sigma$, which outperforms the previously used ΣT. Our software was designed to run fast: a typical analysis (adapted search for 20 000 genes, 50 phenotypes, 1000 root nodes, 1000 permutations) takes at most a few minutes on a low-end 1GHz Pentium platform. The same analysis using balls of radius 1 takes only a few seconds.

We compared our results to Gene Ontology analysis using the top 50 most significant genes and found very little overlap; GO yields mostly fairly general terms of limited interest.

We also compared our method with the GSEA method for the lymphocyte data set (we attempted a similar analysis for the regulatory T-cell data set but the GSEA analysis did not terminate due to some unknown software issue). Overall, our method yields better p-values; the top four gene sets in GSEA have $p = 0.05, 0.14, 0.14$ and 0.43. The top GSEA gene sets are motif-based, with very little overlap with our top sets. These results suggest that the two methods are somewhat complementary. Heuristically, one would expect GSEA to be better at identifying large pathways with major disturbances, while our algorithm pinpoints a small number of genes within pathways that are most likely to be responsible. We are also likelier to detect local disturbances within a pathway.

The construction of an interaction network is a key step in our algorithm. The current implementation incorporates a broad definition of a network, including both protein–protein and DNA–protein interactions. All interactions are treated equally, regardless of type and direction. It is remarkable that even such a relatively coarse design captures enough information to produce interesting and statistically significant results. Refining network structure is a promising area of future

research. We are also working on increasing output quality by reducing the overlap among the top graphs.

ACKNOWLEDGEMENTS

This work was funded in part by NSF grant DMS-0241246. We also thank Justin Mungal for help in downloading and normalizing the serum data from the Stanford Microarray Database.

Conflict of Interest: none declared.

REFERENCES

Breitling,R. *et al*. (2004) Graph-based iterative group analysis enhances microarray interpretation. *BMC Bioinformatics*, **5**, 100.

Cabusora,L. *et al*. (2004) Differential network expression during drug and stress response. *Bioinformatics*, **21**, 2898–2905.

Chang,H. *et al*. (2004) Gene expression signature of fibroblast serum response predicts human cancer progression – similarities between tumors and wounds. *PLoS Bio*., **2**, 206–214.

Critchley-Thorne,R. *et al*. (2006*a*) Alterations in gene expression of regulatory T cells in melanoma patients. In preparation.

Critchley-Thorne,R. *et al*. (2006*b*) Inhibition of interferon signaling in lymphocytes in metastatic melanoma patients. To appear.

Curtis,R.K. *et al*. (2005) Pathways to the analysis of microarray data. *Trends Biotechnol*., **23**, 429–435.

Dahlquist,K.D. *et al*. (2002) GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. *Nat. Genet*., **31** 19–20.

Dudoit,S. *et al*. (2002) Statistical methods for identifying differentially expressed genes in replicated cDNA microarrray experiments. *Statistica Sinica*, **12**, 111–139.

Gentleman,R.C. *et al*. (2004) Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biol*., **5**, R80.

Huber,W. *et al*. (2002) Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, **18**, S96–104.

Ideker,T. *et al*. (2002) Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, **18**, S233–S240.

Rajagopalan,D. and Agarwal,P. (2005) Inferring pathways from gene lists using a literaturederived network of biological relationships. *Bioinformatics*, **21**, 788–793.

Schena,M. *et al*. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, **270**, 467–470.

Slonim,D. (2002) From patterns to pathways: gene expression data analysis comes of age. *Nat. Genet*., **32**, 502–508.

Smyth,G.K. and Speed,T.P. (2003) Normalization of cdna microarray data. *Methods*, **31**, 265–273.

Sohler,F. *et al*. (2004) New methods for joint analysis of biological networks and expression data. *Bioinformatics*, **20**, 1517–1521.

Subramanian,A. *et al*. (2005) Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *PNAS*, **102**, 15545–15550.