BIOINFORMATICS ORIGINAL PAPER

Vol. 21 no. 6 2005, pages 788–793
doi:10.1093/bioinformatics/bti069

Data and text mining

Inferring pathways from gene lists using a literature-derived network of biological relationships

Dilip Rajagopalan* and Pankaj Agarwal

Bioinformatics Sciences, GlaxoSmithKline Pharmaceuticals R&D, 709 Swedeland Road, UW2230, King of Prussia, PA 19406-0939, USA

Received on March 29, 2004; revised on September 12, 2004; accepted on September 13, 2004
Advance Access publication October 27, 2004

ABSTRACT

Motivation: A number of omic technologies such as transcriptional profiling, proteomics, literature searches, genetic association, etc. help in the identification of sets of important genes. A subset of these genes may act in a coordinated manner, possibly because they are part of the same biological pathway. Interpreting such gene lists and relating them to pathways is a challenging task. Databases of biological relationships between thousands of mammalian genes can help in deciphering omics data. The relationships between genes can be assembled into a biological network with each protein as a node and each relationship as an edge between two proteins (or nodes). This network may then be searched for subnetworks consisting largely of interesting genes from the omics experiment. The subset of genes in the subnetwork along with the web of relationships between them helps to decipher the underlying pathways. Finding such subnetworks that maximally include all proteins from the query set but few others is the focus for this paper.

Results: We present a heuristic algorithm and a scoring function that work well both on simulated data and on data from known pathways. The scoring function is an extension of a previous study for a single biological experiment. We use a simple set of heuristics that provide a more efficient solution than the simulated annealing method. We find that our method works on reasonably complex curated networks containing ∼9000 biological entities (genes and metabolites), and ∼30 000 biological relationships. We also show that our method can pick up a pathway signal from a query list including a moderate number of genes unrelated to the pathway. In addition, we quantify the sensitivity and specificity of the technique.

Contact: dilip_rajagopalan@gsk.com

1 INTRODUCTION

Increased use of high-throughput platform (omic) technologies has led to an important new problem in bioinformatics: biological interpretation of the lists of genes that are the typical output of such experiments. For example, transcriptome analysis of cell lines with and without drug treatment, results in a set of differentially expressed genes. It is important to understand whether some of these genes are functioning in a coordinated manner (a 'pathway'). Such an interpretation of this set of genes is useful in understanding the mechanism of action of the drug. As the number of genes in such lists can often be in the hundreds, computational tools are essential to assist in the interpretation of such gene lists.

One approach that has proven successful is based on quantifying the overlap of such a list of 'interesting' genes with a database of sets of genes associated with various biological processes (Tavazoie *et al*., 1999; Draghici *et al*., 2003; Hosack *et al*., 2003; Mootha *et al*., 2003). For example, if the gene list of interest overlaps significantly with the set of genes involved in glycolysis, one can conclude that the drug treatment experiment perturbed the glycolytic pathway. One disadvantage of such approaches is that genes must be placed in a limited number of static groups. For example, even the larger sources of pathways for signal transduction (such as BioCarta) are limited to about 300 pathways and phenomena such as cross talk are ignored.

In the pathway context, another useful approach is to map the query set of interesting genes onto a set of classical pathway maps such as KEGG, BioCarta, etc. Software such as GenMAPP (Dahlquist *et al*., 2002) and several transcriptome analysis packages provide such capability. A hit is represented by color coding the location of the gene on the pathway map. If many genes in the query set are mapped on to a single pathway, say fatty acid metabolism, one would conclude that the drug treatment plays a role in fatty acid metabolism. Although this approach is visually pleasing, it also suffers from the somewhat artificial grouping of genes into a limited number of small pathway maps. Furthermore, this visual approach by itself provides no guidance on the statistical significance of the result.

We present an alternative approach to the problem that is motivated by a systems biology perspective. We have assembled a large network of biological relationships between genes and metabolites derived from various databases created by manual curation of literature. These biological relationships span many types of cellular processes including signaling, transcriptional regulation and metabolism. Given such a network and a query set of interesting genes from an omics experiment, our goal is to search the network for subnetworks consisting mostly of query genes. The set of genes in such subnetworks and the web of literature-based relationships between them will provide some biological insight into the mechanism of action. The PubGene suite of tools developed by Jenssen *et al*. (2001) also helps to analyze gene expression data using a literature-based network. As we describe below, there are important distinctions between our method and PubGene.

In our work, we present a graph-based heuristic algorithm with an associated scoring function to dynamically construct subnetworks with a high score. Our approach is built on the work of Ideker *et al*. (2002) who developed a method to search Y2H-based

*To whom correspondence should be addressed.

protein interaction networks using a set of differentially expressed genes from a transcriptomics experiment. We believe network-based approaches will emerge as the preferred way to perform biological interpretation of gene lists derived from omics experiments.

2 DATA AND METHODS

2.1 Data used to build the network

We have constructed a network of biological relationships between genes and metabolites using three data sources:

(1) The Ingenuity Pathways Knowledge Base (www.ingenuity.com) includes over one million highly structured scientific findings manually curated from the literature relating genes, cells, diseases, drugs and other biological entities. These relationships are primarily from human, mouse and rat. We have extracted a subset of relationships from this knowledge base and constructed a network consisting of ∼25 000 relationships between ∼7300 human genes. The gene–gene relationships in this network include protein binding, protein phosphorylation, binding of transcription factors to DNA.

(2) The TransFac database (Matys *et al*., 2003) of transcriptional regulation (www.gene-regulation.com) contains ∼1000 relationships between ∼200 human transcription factors and 400 genes (Version 6.4).

(3) The HumanCyc database (May 2003 release) of human metabolism consists of a set of metabolic reactions and the genes whose products catalyze these reactions (humancyc.org). It includes data for ∼1400 genes and 900 metabolites.

We have integrated these three sources of data on biological relationships into a comprehensive network ($R1$). The total number of nodes in network ($R1$) is ∼9300, of which 900 are metabolites and the rest are genes. The network has ∼30 000 edges representing relationships between the genes and metabolites. The metabolic reactions were converted to a network by creating an edge between each enzyme and all its substrates and products. Common cofactors, such as ATP and molecules like water were excluded prior to building the network. Over 95% of the nodes in the integrated network form a single, large connected component. The degree (number of neighbors) of each node in this network ranges from 1 to ∼300. The topology of the network can be fit to a scale-free model (Barabasi and Oltvai, 2004) with a power-law coefficient of −1.9. The pathway results presented in the paper were all generated using network $R1$. However, for the purpose of testing and developing our scoring function and algorithm, we also included indirect relationships to build a more connected network ($R2$) containing ∼9500 nodes and 50 000 edges in which the maximum degree is ∼750.

2.2 Method to find subnetworks

We have developed a method to take a set of query genes that arise from an omics experiment and extract a subnetwork of genes and relationships between them from an interaction network. This method relies on a scoring function for subnetworks and an algorithm to find high-scoring subnetworks. The subnetworks found by this algorithm will predominantly consist of genes contained in the query set, but they can have some 'gaps'—genes not contained in the query set. The genes contained in the high-scoring subnetwork along with the relationships between them will provide useful insight into the mechanism of action underlying the omics experiment that gave rise to the query set of genes.

Our approach builds on the work of Ideker *et al*. (2002), which relies on a significance measure or p-value supplied for each gene in the query set. For example, in trying to determine pathways associated with a gene list arising from a transcriptomics experiment, the p-values supplied for the genes would typically be calculated from a statistical test of differential expression between a control and treated group. If significance measures are not available, our method can be applied to a query set of genes by assigning all genes in the query set a low, equal p-value. The remaining genes in the network are assigned a high (insignificant) p-value for the computation.

Our subnetwork scoring function is similar to the scoring function proposed by Ideker *et al*. (2002) for a single biological experiment or condition, but we introduced some important improvements. In addition to scoring functions for single and multiple conditions, Ideker *et al*. (2002) developed a simulated annealing algorithm to tackle the NP-hard problem of finding high-scoring subnetworks. However, our implementation of their simulated annealing algorithm proved too slow for the large network described earlier, and we introduced a novel, graph-based heuristic algorithm that produces high-scoring subnetworks with much shorter execution times. In the rest of this paper, we refer to our implementation of Ideker's simulated annealing method as algorithm $A1$, and we refer to our new heuristic method as algorithm $A2$.

2.3 Scoring function

We implemented the scoring function for a single experiment proposed by Ideker *et al*. (2002) along with a simple greedy search algorithm. This algorithm, which we denote $A3$, is derived from the simulated annealing method proposed by Ideker *et al*. (2002). In this algorithm, we rank the nodes by z-score and turn on the top 50% of nodes. We group the nodes turned on into connected components using a breadth-first search. Finally, we check whether turning on nodes adjacent to connected components improves their scores. This last step is done recursively, and it can result in the merging of separate connected components. For details of some of these steps [see Ideker *et al*. (2002)]. We tested this greedy algorithm ($A3$) using a random input gene set containing no biological pathway signal. As pointed out by Ideker *et al*. (2002), network nodes in this situation have uniformly distributed p-values between 0 and 1. For this kind of input, we do not expect the method to find large subnetworks. However, we found that the simple greedy search algorithm ($A3$) coupled with the above scoring function resulted in very large subnetworks consisting of up to 1000 nodes. As per the design of the greedy search algorithm, as the subnetwork grows in size, its score increases monotonically. Thus, it is a deficiency in the scoring function that is responsible for this phenomenon and not any inadequacy in the algorithm which is designed to find subnetworks with the highest score. Our goal was then to understand the root cause of this undesirable behavior and develop a modified scoring function that resulted in far smaller subnetworks for the case of random input.

For a subnetwork with M nodes, the scoring function for a single experiment proposed by Ideker *et al*. (2002, Equation 2) can be rewritten as

$$S = \frac{\sqrt{M}}{\sigma} \frac{\sum_i C_i}{M}, \tag{1}$$

where S is the subnetwork score (denoted as s_A by Ideker *et al*., 2002), σ is the SD of the distribution of z-scores for the entire network and C_i is a corrected score for each node. The z-scores for each node are derived from the p-values via the inverse normal distribution function. The corrected node score C_i in turn is given by $z_i - \mu$, where z_i is the z-score for the node and μ is the mean of the z-score distribution for the entire network.

Equation (1) is derived from the original scoring function of Ideker *et al*. (2002) using the approximate values of μ and $\sigma/\sqrt{M}$ for the mean and SD of a random sample of size M from the entire distribution of node z-scores. These approximations are fairly accurate as long as M is less than about one-fourth of the total number of nodes in the network.

The modified form of the scoring function [Equation (1)] immediately reveals the reason for producing large subnetworks from random inputs. About half the nodes in the network will have a positive value of $z_i - \mu$ (assuming the z-score distribution is not too skewed). With such a large number of nodes potentially able to contribute positively to a subnetwork score, there is a greater likelihood of generating large subnetworks from random input.

Our first modification to the scoring function is to introduce a different definition of corrected node score that is designed to produce far fewer nodes

with positive corrected score. We define the new corrected node score C_i as

$$C_i = z_i - \beta\mu, \tag{2}$$

where the empirical parameter β can be selected appropriately to reduce the number of nodes with positive C_i. We have chosen to use a value of β such that all nodes with p-value >0.01 have a negative value for C_i, but other choices could be equally appropriate.

In random input tests using network $R1$ and algorithm $A2$, this modification is sufficient to prevent the formation of very large subnetworks. For example, in 2000 random input tests using network $R1$, the largest subnetwork produced has 18 nodes, and the median and mean subnetwork size over these tests are 2 and 2.3, respectively. However, tests on the highly connected network $R2$ using random input continued to produce large subnetworks with hundreds of genes. We discovered that this behavior was due to promiscuous nodes in the network, and the scoring function had to be modified to properly account for such nodes.

The basic principle behind this modification is to recognize that for any node in the network, the likelihood of finding a neighbor with a good (low) p-value increases with the degree of the node. Considering a node with degree K in a network with N nodes, one would expect to find a neighboring node with roughly the N/K-th lowest p-value just by chance. A node adjacent to a promiscuous node should be included in a subnetwork only if its p-value is lower than what is expected by chance. As each node is typically surrounded by neighbors of varying degree, the hurdle on including a node depends on the path taken to include that node. In order to avoid creating a scoring function that depends on the order in which nodes are added to the subnetwork, the above principle is implemented in the following approximate way. An additional correction factor, which we term as the edge penalty, is calculated a priori for each node V_i in the network. We consider all nodes W_i that are neighbors of V_i. Given the degree D_i of each of these nodes, we compute the average $\bar{D}$ and extract the $(N/\bar{D})$-th lowest p-value from the list of N p-values for all the nodes in the network. This p-value is converted to a z-score z_i^{EP} using the inverse normal distribution function. The new definition of the corrected node score now becomes

$$C_i = z_i - \max(\beta\mu, z_i^{EP}). \tag{3}$$

These corrected node scores can be calculated up front and the subnetwork score for a groups of nodes is then given by Equation (1).

A final step is to calculate a normalized score s for a subnetwork of M nodes as

$$s = 100\frac{S}{S_{\max}}, \tag{4}$$

where $S_{\max}$ is the maximum possible value of S given the input set of p-values. It is calculated by ignoring network connectivity, starting with the node with lowest p-value, and adding nodes with sequentially higher p-values (irrespective of whether they are connected together in the network) until the S-value for this group of nodes no longer increases. This normalization step produces a score s guaranteed to lie between 0 and 100.

Using this form of the scoring function, the largest subnetwork produced in over 2000 runs on network $R2$ with random input and the algorithm described below contains 69 nodes, and the median and mean subnetwork size over these tests are 2 and 7.8, respectively.

Ideker *et al.* (2002) also proposed a scoring function for multiple biological experiments that is used to discover subnetworks active in many or all of these experiments. The improvements we have made to the single-experiment scoring function cannot be directly applied to the scoring of multiple experiments.

2.4 Heuristic algorithm

The steps in our heuristic algorithm are:

(1) Map the query genes and associated p-values to the corresponding nodes in the network. Assign all remaining nodes in the network a p-value of 1. Calculate corrected node score for every node in the network.

(2) Group nodes with positive corrected score into connected subnetworks using a breadth-first search from every positive scoring node not yet assigned to a subnetwork.

(3) Select a previously unselected subnetwork in decreasing order of score. If there is no subnetwork remaining with positive score go to Step 5.

(4) For each subnetwork selected (B), create a list of all non-positive nodes adjacent to nodes in B. Essentially, collapse the subnetwork into a single cluster node (V_B). Select the neighbors in decreasing order of degree. Perform a limited depth first search (DFS) for neighboring subnetworks that can be merged into subnetwork B. This DFS is limited in that it only extends over a maximum of d non-positive nodes. If an adjacent subnetwork A is found, merge A, B and the non-positive nodes between A and B into a new subnetwork B'. If the score of B' is greater or equal to the score of B, then accept the change and restart Step 3. Otherwise, reject the merge, and continue with other neighbors of V_B. Once the limited DFS from V_B has been exhausted, return to Step 3 and select the next subnetwork. Each time Step 3 is restarted, nodes previously used to initiate the DFS are not reconsidered for the DFS.

(5) The goal of the final pruning step is to try and remove nodes with small positive individual scores that might have been included in subnetworks in Step 2. The pruning step is performed for each subnetwork remaining after Step 4. The nodes in a subnetwork are considered in increasing order of score. If deleting a node would increase the score of the new subnetwork consisting of the rest of the nodes, and still keep the subnetwork connected, the node is deleted.

Execution time for this algorithm is dominated by Step 4, and the execution time for this step is largely a function of d. Both the size of the resulting subnetwork and the execution time of the algorithm increase with d. We used a value $d = 2$ for the results shown in this work, and we obtained run times on the order of 2 min on a Compaq Alpha (XP1000) workstation. In contrast, our implementation of the simulated annealing algorithm ran for several hours on the large networks described earlier. An important benefit of fast execution times is that it is possible to perform multiple permutation runs to assess the statistical significance of the subnetwork scores we obtain. We do this by randomly scrambling the association between nodes and p-values and repeating the search for high-scoring subnetworks. Subnetworks found using the original p-value assignment must have a high-score relative to the scores from the permutation runs.

The presence of hub nodes characteristic of scale-free networks is addressed by incorporating the edge penalty in our scoring function. However, our method does rely on a network of high-quality interactions and the presence of many spurious connections between genes or the absence of key connections between genes, will adversely impact the quality of results.

Our method differs substantially from the approach of Jenssen *et al.* (2001) implemented in PubGene. The main difference is, our approach tries to maximize a scoring function in the process of building subnetworks. In contrast, PubGene constructs a set of subnetworks guided by user inputs and graph properties, and the scoring is done as a post-processing step to generate a ranked list of subnetworks.

2.5 Simulated pathway data for validation

We validated our approach using simulated and known pathway data. We created 100 artificial 'pathways' to serve as a known answer by traversing the network of relationships. Each of these pathways comprised 57 genes of which 40 were randomly selected as input to our method. Each pathway was used to query the tool in turn, by setting these 40 nodes to have low p-values uniformly distributed between 0 and $p_{\max}$, where $p_{\max}$ was set to a low number like 10^{-2} or lower. The rest of the nodes had p-values uniformly distributed between $(0, 1)$. This assignment of p-values simulates the case of a real omics experiment where the list of important genes may contain a pathway signal along with some genes unrelated to any pathway. We explored

different proportions of pathway related and unrelated genes in the query gene list by varying p_{max}.

2.6 Known pathway data for validation

We used 267 pathways related primarily to signaling from BioCarta (www.biocarta.com) as additional tests of our method. We selected a subset of 219 pathways from the complete set for which we were able to assign identifiers automatically to at least six genes. The genes on each of these pathways were randomly assigned a low p-value uniformly distributed between $(0, p_{max})$, where p_{max} is a low number such as 10^{-3}. The remaining nodes in the network were assigned a p-value uniformly distributed between $(0, 1)$. It is important to note that we have not systematically extracted the relationships represented in the BioCarta maps and included them in our database of biological relationships. Hence, the tests we describe using the BioCarta pathways partly address the question of whether our database of relationships contains the information in these pathways. However, more importantly, these tests shed light on whether our method is able to pick out a pathway signal in a gene list containing varying numbers of genes unrelated to any pathway.

3 RESULTS AND DISCUSSION

We assessed the performance of our algorithm by examining how the best subnetwork found compares to what we expect (the artificial and BioCarta pathways). Our experiments also provide guidance as to the sensitivity and specificity of the technique. The quality of the omics data has to be reasonable for the tool to infer the correct pathway, i.e. the omic technology has to significantly highlight genes on the pathway.

Our first set of tests were for the case of random input with uniformly distributed (between 0 and 1) p-values for the nodes. The boxes labeled uniform in Figure 1 show a scatter plot of the size of the resulting network versus their score. In this set of 100 test cases, the score of the best subnetwork never exceeded 51.9 and the 95th percentile of the score distribution is 41.6.

We evaluated 100 artificially generated pathways of size 57 (as described in the Data and methods section), randomly selecting 40 nodes in each case to assign a uniformly distributed p-value between 0 and p_{max}. We explored three different values for p_{max}: 10^{-4}, 10^{-3} and 10^{-2}. The top subnetwork for each of the 100 artificial pathways for $p_{max} = 10^{-4}$ is represented by a cross in Figure 1. The subnetwork scores were between 70 and 100 and the size was between 34 and 50. These scores were clearly separated from the background distribution (squares in the figure). Similar (but declining) separation was obtained for $p_{max} = 10^{-3}$ and 10^{-2}. At $p_{max} = 10^{-2}$, there was some intermixing of the distributions, nevertheless, using a score threshold of 41.6, 80% of the tests yielded a subnetwork with a significant score. Given the ∼10 000 nodes in the network, ∼10 nodes from the background uniform distribution have p-values $<10^{-3}$, and ∼100 have p-values $<10^{-2}$. Thus, when the simulated pathway nodes are assigned p-values with $p_{max} = 10^{-3}$, ∼10 unrelated nodes have p-values in the same range as the simulated pathway nodes. At $p_{max} = 10^{-2}$, the number of unrelated nodes with p-values in the same range as simulated pathway nodes increases to ∼100. In summary, pathways of size 40 can be distinguished with varying difficulty dependent upon the number of unrelated genes in the query set.

While the algorithm successfully extracts a subnetwork with a low-probability of a false positive, we still need to show that this network is similar enough to the test pathway. This can be established via the number of missing nodes (in query list but not in subnetwork) and extra nodes (found by the algorithm but not in query list). For $p_{max} = 10^{-4}$, very few nodes are missing for any pathway, but up to 10 extra nodes are found. Even for pathways seeded with $p_{max} = 10^{-3}$, the top network found fits quite well with the initial gene list.

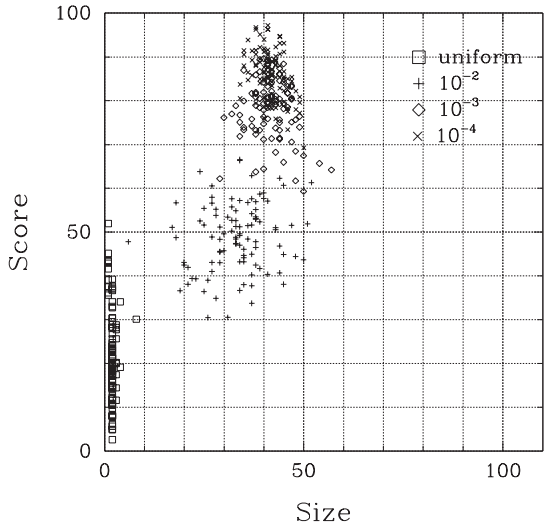

Fig. 1. The score (0–100) of the discovered subnetwork is plotted against its size measured in number of nodes. The legends show the p_{max} used for the pathway nodes. The boxes labeled uniform represent the background case where all the nodes had p-values drawn from the uniform distribution on [0,1]. All the input gene lists were of size 40.

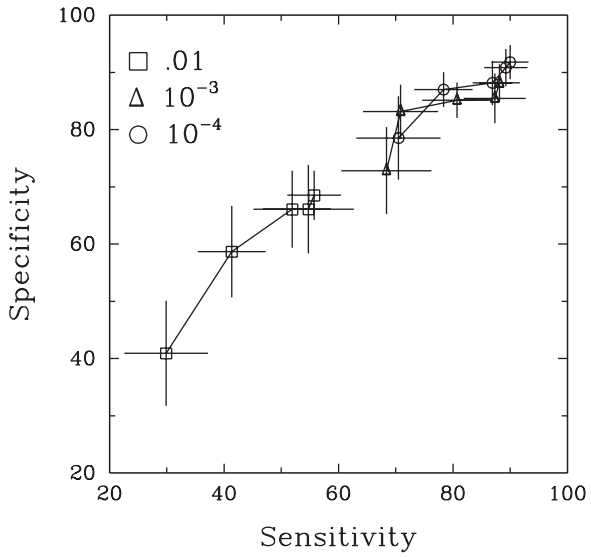

Fig. 2. The specificity of the technique versus sensitivity for 219 BioCarta pathways. The pathways were binned according to the number of identifiable proteins in the pathway. Each symbol represents the p_{max} value used to assign p-values to the genes in a set of BioCarta pathways. The successive points connected by lines represent different BioCarta pathway size ranges. The five size ranges used were 6–10, 11–15, 16–20, 21–25 and 26+. The number of pathways in each bin ranged from 25 to 61. The vertical and horizontal dashes represent the 95% confidence interval on the specificity and sensitivity estimates. The highest point for each p_{max} is the largest pathway bin.

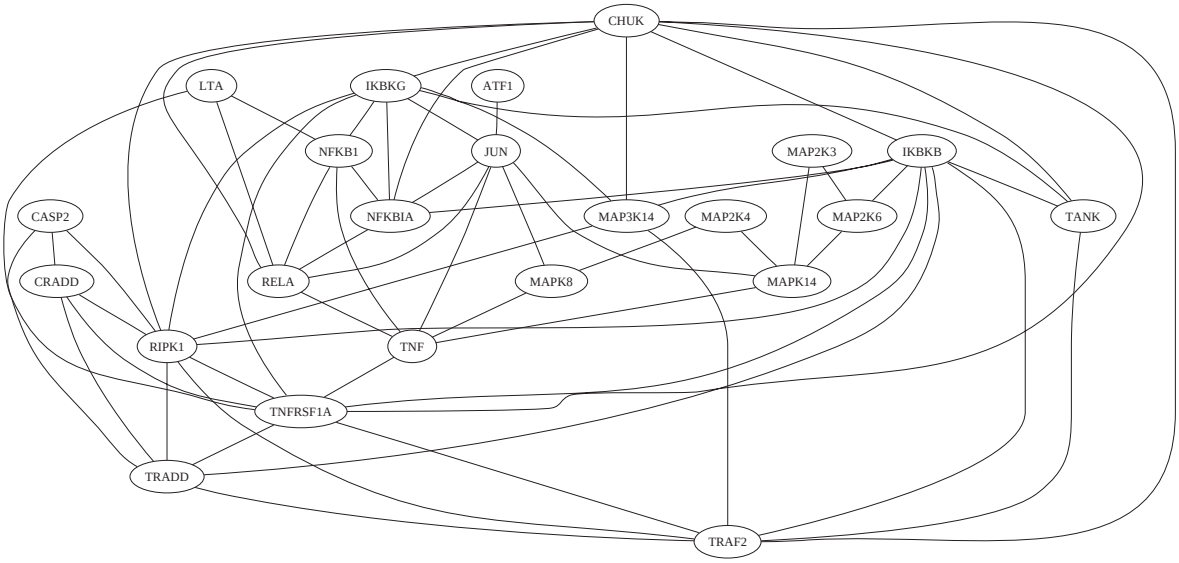

Fig. 3. BioCarta TNF/stress related signaling pathway rediscovered using $p_{\max} = 10^{-4}$. None of the pathway genes was missed and no extra nodes were added to the subnetwork found.

For $p = 10^{-2}$, the top network rarely misses >20 of the 40 nodes in the gene list and it includes up to ∼20 additional nodes. In such cases, it may not be trivial to extract the original pathway from the top subnetwork. Hopefully, the biology may still be inferred from the subnetwork.

The previous test cases demonstrated that the scoring function and heuristic algorithm are successful in extracting the relevant pathway when the query gene list contains genes mostly related to a coordinated process (pathway) with moderate numbers of unrelated genes.

To overcome any limitations in our testing based on artificially generated pathways, we also tested the tool using BioCarta pathway input as described above. These pathways ranged in size from 6 to 70 proteins. Figure 2 displays the quality of the retrieved subnetwork using the BioCarta-based gene lists as input. We used measures of Sensitivity (related to false negatives) and Specificity (related to false positives). Sensitivity is defined as the percentage of the input that is correctly identified in the resulting subnetwork. Specificity is defined as the percentage of the subnetwork predicted that is correct (i.e. was contained in the input).

Figure 2 shows the dependence of sensitivity and specificity on two important parameters: the number of unrelated genes in the query set with low p-values (related to $p_{\max}$), and the size of the BioCarta pathway. Data for different $p_{\max}$ values are plotted with different symbols and each point on the curve is not a different threshold as in a receiver operating characteristic (ROC), but represents the average size of the pathway. From the three curves, it is apparent that the seeding of the pathways at higher $p_{\max}$ decreases both the sensitivity and specificity of the technique. This is expected as at higher $p_{\max}$ the pathway signal is confounded by a substantial number of genes unrelated to the pathway. For example, at a $p_{\max}$ of 10^{-2}, there are ∼100 random, unrelated nodes with p-values in the same range as the BioCarta nodes. For pathways seeded at 10^{-3} or better, the sensitivity is over 70%, regardless of pathway size. This indicates that if the omics data clearly delineates most of the genes on the pathway, it can be recovered from the network even if the exact pathway is not previously known. This augurs well for recovering novel pathways that are not published previously. The specificity is also >70% for pathways seeded at $p_{\max} = 10^{-3}$. Even with the very large number of unrelated nodes for $p_{\max} = 10^{-2}$, sensitivity and specificity are, ∼60% for the larger pathways, dropping to ∼40% specificity and 30% sensitivity only for the smallest pathways. Specificity and sensitivity both decline with decreasing pathway size. For smaller pathway sizes there is more variation in sensitivity (i.e. larger 95% confidence intervals).

We also used the BioCarta inputs with $p_{\max} = 10^{-4}$ to test the original scoring function for a single experiment proposed by Ideker *et al.* (2002). Using our algorithm ($A2$) as well as the simple greedy algorithm ($A3$), we obtained subnetworks containing ∼3000 nodes. These subnetworks typically contained all the BioCarta nodes (100% sensitivity), but the specificity is ∼0. As the algorithm grew the subnetworks to this size, the score increases monotonically demonstrating the fundamental problem with their scoring function. We believe our improvements to the scoring function are necessary to obtain meaningful results on large networks.

The focus of our testing has been to determine whether our method can pick out a single set of interrelated genes from a query set containing varying numbers of unrelated genes. In these tests, the method usually returned a single high-scoring subnetwork. Pathway sets such as BioCarta typically have a lot of overlap between them. A test with a query set containing two BioCarta pathways would return a single subnetwork if even a single gene was common to the two pathways. On the other hand, a query set that contains two completely separate sets of interrelated genes would produce two distinct subnetworks. A detailed evaluation of inputs of this type is beyond the scope of the present work.

Figure 3 shows an example pathway recovered when the genes in the TNF/Stress related signaling pathway from BioCarta are used as input with $p_{\max} = 10^{-4}$. The layout is generated using the dot program within the Graphviz suite AT&T Labs (www.graphviz.org).

The input gene list contained 22 genes and the best subnetwork found contained all these genes and no extra nodes were added relative to the input set. Thus, both sensitivity and specificity are a 100% for this example.

4 CONCLUSIONS

A number of platform technologies including transcriptional profiling, proteomics, literature searches, genetic association, and high-throughput screening produce sets of genes or proteins that are perhaps linked by an underlying pathway or biological relationship. Interpreting gene lists from omic technologies continues to be a challenging task. Databases of biological relationships between thousands of proteins can help in deciphering such data. We constructed a large network of biological relationships between genes, and searched this network for subnetworks consisting largely of interesting genes from the omics experiment. The subset of genes in the subnetwork along with the web of relationships between them will provide insight into underlying pathways. Finding a subnetwork with maximal score (one that includes mostly low p-value nodes) is an NP-hard problem. Ideker *et al*. (2002) proposed a scoring function and simulated annealing algorithm to tackle this problem. However, we did not obtain very good results using this method on the large networks of interest to us. We present a more efficient heuristic algorithm and improvements to the scoring function that are necessary to obtain meaningful results on large networks. We demonstrate that our method works well on both simulated data and data from known pathways. Extrapolating the result, we feel that this algorithm may also shed light on unknown pathways. This is, of course, dependent on the known network including the biological pairwise relationships underlying the unknown pathway. Adopting a systematic view of cellular processes also enables study of cross talk between canonical pathways.

Additional improvements to our scoring function may be possible by exploiting network properties such as edge weights representing our confidence in a particular relationship. Interpreting these networks biologically in light of what would be called pathways is another challenging problem. Can they be laid out such that they are more recognizable as biological pathways? Regardless, we believe that network-based approaches will emerge as a preferred way to perform biological interpretation of gene lists derived from omics experiments.

ACKNOWLEDGEMENTS

We would like to thank Michael Lutz and David Searls for their encouragement, support and comments on the manuscript.

REFERENCES

Barabasi,A.-L. and Oltvai,Z.N. (2004) Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, **5**, 101–114.

Dahlquist,K.D., Salomonis,N., Vranizan,K., Lawlor,S.C. and Conklin,B.R. (2002) GenMAPP, a new tool for viewing and analyzing microarray data on biological pathways. *Nat. Genet.*, **31**, 19–20.

Draghici,S., Khatri,P., Martins,R.P., Ostermeier,G.C. and Krawetz,S.A. (2003) Global functional profiling of gene expression. *Genomics*, **81**, 98–104.

Hosack,D.A., Dennis,G.,Jr, Sherman,B.T., Lane,H.C. and Lempicki,R.A. (2003) Identifying biological themes within lists of genes with EASE. *Genome Biol.*, **4**, R70.

Ideker,T., Ozier,O., Schwikoswki,B. and Siegel,A.F. (2002) Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics*, **18**(Suppl. 1), S233–S240.

Jenssen,T.-K., Leagreid,A., Komorowski,J. and Hovig,E. (2001) A literature network of human genes for high-throughput analysis of gene expression. *Nat. Genet.*, **28**, 21–28.

Matys,V., Fricke,E., Geffers,R., Gossling,E., Haubrock,M., Hehl,R., Hornischer,K., Karas,D., Kel,A.E., Kel-Margoulis,O.V. *et al*. (2003) Transfac: transcriptional regulation, from patterns to profiles. *Nucleic Acids Res.*, **31**, 374–378.

Mootha,V., Lindgren,C., Eriksson,K., Subramanian,A., Sihag,S., Lehar,J., Puigserver,P., Carlsson,E., Ridderstrale,M., Laurila,E. *et al*. (2003) PGC-1alpha-responsive genes involved in oxidative phosphorylation are coordinately downregulated in human diabetes. *Nat. Genet.*, **34**, 267–273.

Tavazoie,S., Hughes,J.D., Campbell,M.J., Cho,R.J. and Church,G.M. (1999) Systematic determination of genetic network architechture. *Nat. Genet.*, **22**(3), 281–285.