Machine Learning, 52, 119–145, 2003
© 2003 Kluwer Academic Publishers. Manufactured in The Netherlands.

Inclusion of Textual Documentation in the Analysis of Multidimensional Data Sets: Application to Gene Expression Data

SOUMYA RAYCHAUDHURI tumpa@stanford.edu
HINRICH SCHÜTZE hinrich@stanfordalumni.org
RUSS B. ALTMAN russ.altman@stanford.edu
Department of Genetics, Stanford University, Stanford, CA 94305-5479, USA; Stanford Medical Informatics, Stanford University, Stanford, CA 94305-5479, USA

Editors: Paola Sebastiani, Isaac S. Kohane and Marco F. Ramoni

Abstract. Recently, biology has been confronted with large multidimensional gene expression data sets where the expression of thousands of genes is measured over dozens of conditions. The patterns in gene expression are frequently explained retrospectively by underlying biological principles. Here we present a method that uses text analysis to help find meaningful gene expression patterns that correlate with the underlying biology described in scientific literature. The main challenge is that the literature about an individual gene is not homogenous and may addresses many unrelated aspects of the gene. In the first part of the paper we present and evaluate the *neighbor divergence per gene* (*NDPG*) method that assigns a score to a given subgroup of genes indicating the likelihood that the genes share a biological property or function. To do this, it uses only a reference index that connects genes to documents, and a corpus including those documents. In the second part of the paper we present an approach, *optimizing separating projections* (*OSP*), to search for linear projections in gene expression data that separate functionally related groups of genes from the rest of the genes; the objective function in our search is the *NDPG* score of the positively projected genes. A successful search, therefore, should identify patterns in gene expression data that correlate with meaningful biology. We apply *OSP* to a published gene expression data set; it discovers many biologically relevant projections. Since the method requires only numerical measurements (in this case expression) about entities (genes) with textual documentation (literature), we conjecture that this method could be transferred easily to other domains. The method should be able to identify relevant patterns even if the documentation for each entity pertains to many disparate subjects that are unrelated to each other.

Keywords: gene annotation, natural language processing, text analysis, gene expression analysis, bioinformatics, functional genomics

1. Introduction

Manual or semi-automated analysis of large-scale biological data sets typically requires biological experts with vast knowledge of many genes to decipher the known biology accounting for genes with correlated experimental patterns. The goal is to identify the relevant “functions”, or the global cellular activities, at work in the experiment. For example, experts routinely scan gene expression clusters to see if any of the clusters are explained by a known biological function. Efficient interpretation of these data is challenging because the number and diversity of genes exceed the ability of any single investigator to track the complex relationships established by the data sets. However, much of the information relevant to the data is contained in the published literature about the individual genes; in fact it is the knowledge of this literature that experts rely on. Including the literature as a direct knowledge source for any algorithmic strategy to approach such data may greatly facilitate analysis.

Here we present a method to identify subgroups of genes with a distinct signature in gene expression data that also have established functional similarity in the published literature. By signature, we mean that these genes can be separated from the remainder by a linear projection in the data set. Gene expression data is typically available as a multidimensional data set. Gene expression signatures in the data set can be used to further understand all genes with that function, and also to annotate uncharacterized genes.

In this paper we present an algorithm that given an arbitrary multi-dimensional numerical data set of measurements and free text documentation about those measured entities, is able to identify a subgroup of entities that can be separated by a linear projection that share some property as assessed form the documentation. The application of such a method extends beyond biology to any other domain that has abundant multi-faceted textual documentation available. For example academic performance of college students could be analyzed by the text in their home pages and text in pages linked from their home page. Our method might indicate, for example, that students that share a personal interest in political issues perform better in social science classes. The documents about an individual student will be very diverse describing many different aspects about her that may not be related or even comparable; her personal, professional, and academic interests may all be described in different portions of text. Similarly the documents about an individual gene may pertain to different unrelated aspects about it. Critical to the effectiveness of our method is its ability to not be led astray by this diversity contained within the corpus.

1.1. Measuring gene expression

Gene expression arrays permit the rapid assaying of the relative expression of a gene within individual cells (Schena et al., 1995). They are used to determine relative induction of genes within an experimental condition where the cell is subjected to some sort of stimulus. These conditions may be different time points during a biological process, such as the yeast cell cycle (Cho et al., 1998; Spellman et al., 1998) and drosophila development (White et al., 1999); direct genetic manipulations on a population of cells such as gene deletions (Hughes et al., 2000); or they can be different tissue samples with some common phenotype (such as different cancer speciments) (Alizadeh et al., 2000; Ross et al., 2000). Besides gene expression, gene arrays have also been used to identify gene deletions (Behr et al., 1999), gene duplications (Pollack et al., 1999), transposon locations (Raychaudhuri et al., 2000), and single nucleotide polymorphisms (Halushka et al., 1999). A typical gene expression data set is a matrix, with each row representing a gene and each column representing condition. The value at each position in the matrix represents the relative expression of a gene under some condition.

1.2. Analysis of gene expression data

Large-scale gene-expression data sets include thousands of genes measured at dozens of conditions. The number and diversity of genes make manual analysis difficult and automatic analysis methods necessary. Initial efforts to analyze these data sets began with the application of unsupervised machine learning, or clustering, to group genes according to similarity in gene expression (Eisen et al., 1998; Toronen et al., 1999). Clustering provided a tool to reduce the size of the dataset to a palpable one that could more easily be manually examined. In typical studies, investigators examined the clusters to find those containing genes with common biological properties, such as the presence of common upstream promoter regions or involvement in the same biological processes. After commonalities were identified (often manually) it became possible to understand the global aspects of the biological phenomena being studied. As the community developed an interest in this area, additional novel clustering methods were introduced and evaluated for gene expression data (Ben-Dor, Shamir, & Yakhini, 1999; Heyer, Kruglyak, & Yooseph, 1999).

Supervised machine learning methods provided the earliest means of including external background information into the analysis. They were initially introduced to analyze gene expression data when investigators began looking for means to predict cancer types and their prognosis from gene expression data (Golub et al., 1999). Brown and colleagues used yeast expression data to classify genes into pre-selected gene functional categories using support vector machines, a supervised classification method (Brown et al., 2000). The categories they chose were known to have coherent expression signal from a clustering study done on the same data set (Eisen et al., 1998). They compared the performance of Support Vector Machines with other classification strategies.

The current challenge of gene expression analysis is in the inclusion of the vast amounts of external background information about the conditions and the genes available. While a clustering method reduces the dimensionality of the data to a size that a scientist can tackle, it does not identify the critical background information that helps the investigator understand the significance of each cluster. While supervised machine learning does permit the inclusion of outside labels to the data, the investigator exploring gene expression data for the first time rarely knows which sorts of labels will be applicable, and which will not. For example, it is not obvious from the beginning which genetic functions can be predicted from an expression data set. Only once these functions are known, can they be used as effective gene labels in a supervised machine learning method to make predictions on unknown genes. The issue is complex since most genes have many different functions and there are thousands of different functions that may be applicable to a specific data set; furthermore the definition of functions are often fuzzy, some being specific others being very general.

1.3. Text analysis in biology

Most of the relevant labels and pertinent background information is encoded in the biological literature. As of 30 September 2001, the PubMed database contains some 11,486,042 biomedical abstracts. Of these 1,465,797 are returned on a search for “genetics”; 49,338 abstracts are returned on a search for “saccharomyces cerevisae”; and 18,858 are returned on a search for “drosophila melanogaster”. Genome databases, such as Flybase and Sacharomyces Genome Database have identified articles that are relevant to specific genes within the respective organism (Ashburner et al., 1994; Cherry et al., 1998). The abstracts for many of these documents are available online at PubMed. In the future we anticipate that larger corpuses of full text will become available for analysis (through efforts such as PubMed central, for example Roberts et al., 2001).

Published literature is the largest and perhaps most valuable repository of biological information. Almost all biological discoveries of any significance are recorded in peer-reviewed publications. We hypothesize that the necessary information to analyze gene expression data or other large-scale biological data is in the literature.

Automatic analysis of text, or natural language processing (NLP), has great potential for its application to mining this biological literature. Many NLP techniques have already been used to annotate individual genes (Eisenhaber & Bork, 1999; Fleischmann et al., 1999; Raychaudhuri et al., 2002; Tamames et al., 1998), determine gene or protein interactions (Blaschke et al., 1999; Jenssen et al., 2001; Stephens et al., 2001; Thomas et al., 2000), and to assign keywords to genes or groups of genes (Andrade & Valencia, 1997; Masys et al., 2001; Shatkay et al., 2000).

1.4. Overview

We devise a method that find criterion that apply to an individual gene’s expression profile; the criterion are chosen so that all of the genes that are selected by them have some common biological or functional properties. The selected criterion defines the gene expression signature of the genes with that function. The algorithm presented here searches for the right criterion so that the selected genes represent a common biological concept or function.

The criteria we use are linear projections. A projection is a numerical vector of weights; there is one weight for each condition in the gene expression data. A gene is selected if the dot product of its expression profile with the projection vector is positive. To change the rules of selection, and therefore the selected genes, the projection vector can be reoriented so that different experiments are weighted differently.

There are many advantages to using linear projections. First, projections provide a very expressive set of potential rules; many effective machine learning methods, such as linear discriminant analysis, logistic regression, and the perceptron, are based on linear separation with projections (Ripley, 1996). Second, they are a reasonably constrained set of rules with a limited search space; there are only as many weight-parameters to define as there are conditions. Third, projections are a continuous rule set; so slight changes in the parameters change the selected genes only slightly.

First, in Section 2 we establish that subgroups of genes sharing a common function can be recognized by analysis of the gene-associated documents only. Identifying whether a group of genes has functional coherence is critical to the development of the algorithm presented here. The challenge is that genes have many diverse functions, many of which are described in the literature associated with it. The literature for an individual gene may contain information on the cloning of the gene, its sequence, its structure, its biochemical function, the cellular processes it is involved in, and diseases that it may have been implicated in. A group of genes may be coherent if they share only some of these properties. Also different genes have been studied to different extents, while some genes may have many papers written about them, some may have only a single one. We demonstrate that literature about genes can be used to create a scoring system that assigns significant scores to functional gene groups.

In Section 3 we describe our algorithm to search for projections in gene expression that separate functional groups of genes. The quality of a projection is assessed with the functional coherence score (described in Section 2) of the genes it selects. The algorithm searches for projections that score the highest. A search for an optimal projection is difficult and often confounded by local minima. We devise an algorithm that searches for an optimal projection that separates the most functionally coherent group of genes possible.

In Section 4, we demonstrate preliminary application of the method to a published *saccharomyces cerevisiae* (yeast) data set. We demonstrate that classes of genes with known expression fingerprints are identified. The method is easily applicable to other domains where multiple measurements are conducted on many entities for which much free text documentation is available.

2. Scoring subgroups of genes for functional coherence

In this section, we present a computational method, *neighbor divergence per gene* (*NDPG*), that rapidly assesses whether a subgroup of genes share a common biological function by automatic analysis of scientific text. The method utilizes statistical natural language processing techniques to interpret biological text. It requires only a corpus of articles relevant to the studied genes (e.g. all genes in an organism) and a reference index connecting the articles to appropriate genes. Such reference indices are often available online from genome databases, such as SWISS-PROT, Mouse Genome Database (mouse), Saccharomyces Genome Database (yeast), and FlyBase (drosophila) (Bairoch & Apweiler, 1999; Blake et al., 2002; Cherry et al., 1998; Gelbart et al., 1997). Alternatively they can be compiled automatically by scanning titles and abstracts of articles for gene names (Jenssen et al., 2001) or by transferring references to genes from homologous genes that have references assigned. Given a subgroup of genes, *NDPG* assigns a numerical score indicating how “functionally coherent” the gene group is from the perspective of the published literature.

Recognizing coherent gene groups from literature is a challenging problem, since there are disparities in the literature about genes. Some genes have been extensively studied while others have only been recently discovered. Furthermore most genes have multiple functions. A given gene may have many relevant documents or none, and the documents about it may cover a wide spectrum of functions. Consequently, the available text can skew performance of text analysis algorithms.

The intuition behind *NDPG* involves recognizing articles that are about the function represented in the group. If a group of genes shares some specific function, such as *DNA repair*, an article germane to that function will refer to at least one of the genes in the group. Furthermore, other articles that pertaining to the same function will tend to refer to the same gene or to other genes in the group.

NDPG assigns a functional coherence score to a group of genes based on literature. It uses document distance metrics to calculate semantic neighbors; two articles are semantic neighbors if there is similar word usage in each of them (Manning & Schutze, 1999). First, 199 semantic neighbors are pre-computed for each article in the corpus. Given a gene group, each article's relevance to the group is scored by counting the number of neighbors that have references to genes in the group. If the group represents a coherent biological function, the articles that discuss that function will have many referring neighbors and therefore score high (see figure 1). Other articles that address biological functions that are irrelevant to the group function will score low. If a few of the articles referring to a gene are high scoring articles, then the gene has a function that is relevant to that of the group. For each gene in the subgroup, *NDPG* scores its functional relevance to the subgroup by comparing its article scores to an expected random distribution of article scores; the difference between the two distributions

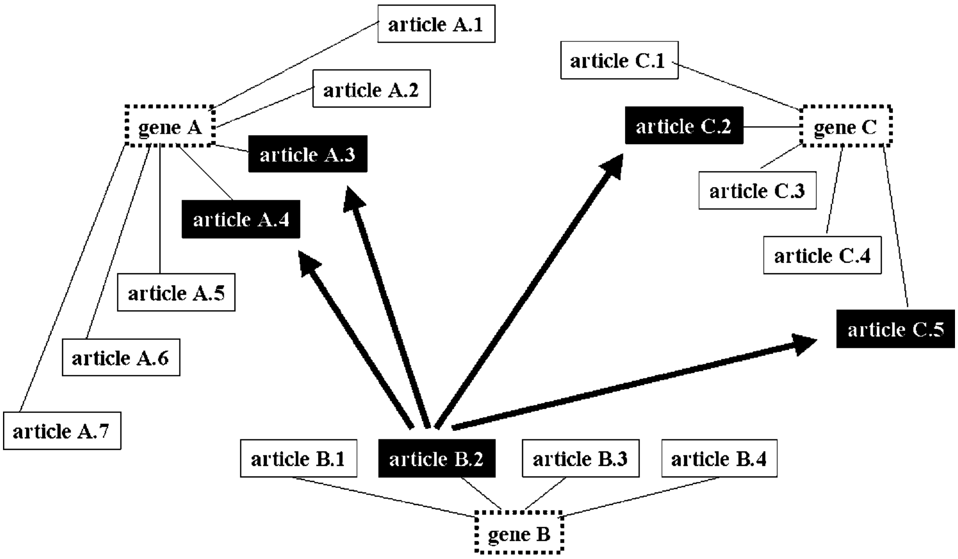

Figure 1. *Scoring articles relative to a gene groups.* Here we graphically depict a small gene group of three genes with the function *DNA repair* (boxes with dotted boundaries). The genes are connected to their respective article references (boxes with solid boundaries). Articles about the function *DNA repair* are dark boxes with white lettering. For all genes, only a few of the referenced articles are about *DNA repair*, the critical function that unites these genes in the group. The arrows are used to indicate the semantic neighbors of "article B.2", an *DNA repair* article. The significance of this article to the group's unifying function becomes apparent when we notice that many of its semantic neighbors, also *DNA repair* articles, are references for other genes in the same group.

is quantified with the KL divergence measure. The *NDPG* measure of functional coherence of a gene group is the mean divergence of all of the genes in the subgroup.

2.1. Neighbor divergence per gene algorithm

2.1.1. Data types: document corpus and reference index. *NDPG* calculation of a gene group requires a corpus of documents relevant to all genes in the organism, and a reference index indicating the articles that are germane to each gene. Here, the documents are PubMed abstracts. The title and abstract fields in the PubMed records are the only ones used. Unique tokens are obtained by tokenizing on white space, punctuation, and common non-alphanumeric characters such as hyphens and parentheses. Those tokens that were present in more than 4 abstracts and fewer than 10,000 abstracts were considered as vocabulary words. Abstracts are converted into vectors of word counts where each dimension represents a specific word.

Due to their availability, the current implementation of the method uses article abstracts only. A more complete version would use full text articles. Inclusion of full text articles in this method would be most effective if the text is broken into smaller, more specific documents such as individual paragraphs.

2.1.2. Identifying semantic neighbors for corpus articles. For each document, the 199 most similar documents (not including the article itself) are pre-computed. To quantify the similarity between two documents we used the cosine between the two weighted document word vectors. Word vectors are first converted into inverse document frequency weighted word vectors (Manning & Schutze, 1999):

$$W_{i,j} = \left\{ \begin{array}{ll} (1 + \log_2(tf_{i,j}))\log_2(N/df_i) & \text{if } tf_{i,j} > 0 \\ 0 & \text{if } tf_{i,j} = 0 \end{array} \right\}$$

where $W_{i,j}$ is the weighted count of word i in document j, $tf_{i,j}$ is the number of times word i is in document j, df_i is the number of documents that word i is present, and N is the total number of documents. Inverse document frequency weighting is used to reduce the impact of very common words. Document similarity is the cosine of the angle between these two weighted article vectors.

In the selection of the 199 similar documents for each document, we apply a simple filter. The neighbors of a seed document are selected from those documents that refer to at least one gene not in the subset of genes referred to in the seed document.

2.1.3. Scoring article relative to gene groups. Given a gene group, *NDPG* then assigns a score, S_i, to each document i. The score is the count of semantic neighbors that refer to group genes. Groups representing a genetic function will induce many documents to have high scores.

Practically, documents in the dataset may refer to multiple genes rather than a single one. Neighboring documents with some genes referring to gene groups are counted fractionally.

$$fr_{k,g} = n_{k,g}/n_k$$

where $n_{k,g}$ is the number of genes in the gene group g that the neighboring document k refers to, n_k is the number of genes that document k refers to, and $fr_{k,g}$ is the fractional reference for document k to group g.

To obtain the document score, the referring fractions of the 199 neighbors are summed and rounded to the nearest integer.

$$S_{i,g} = \text{round}\left(\sum_{j=1}^{199} fr_{sem_{i,j},g} \right)$$

where $S_{i,g}$ is the score for an document i for a group g calculated by rounding and summing the fractional reference of its 199 neighbor document whose indices are $sem_{i,j}$. $S_{i,g}$ is an integer that ranges between 0 and 199.

2.1.4. Calculating a theoretical distribution of scores. If the gene group has no coherent functional structure, the semantic neighbors of any given document should refer to group genes independently with a probability q. If each of these trials are independent, a Poisson distribution would estimate this distribution accurately for small values of q. In this case:

$$P(S = n) = \frac{\lambda^n}{n!} e^{-\lambda}$$

where $\lambda = 199 * q$. For a given gene group we estimate q, the fraction of documents referring to group genes, by summing all of the fractional references, fr, of all documents and dividing by the number of documents, N.

2.1.5. Quantifying the difference between the empirical score distribution and the theoretical one. An empirical distribution of the document scores for a gene is computed for each gene in the group. If the group contains no functional coherence, all of the distributions of scores should be similar to the Poisson distribution. The functional relevance of each gene to the subgroup is scored as the KL-divergence between its empirical distribution of article scores and the Poisson distribution (Manning & Schutze, 1999).

Given two distributions, a theoretical one, h, and an observed one, g, we calculate KL-divergence:

$$D(g \| h) = \sum_i g_i \log_2(g_i / h_i)$$

If two distributions are the same, the divergence is zero; the more disparate the two distributions the larger the divergence.

2.1.6. Functional coherence score of a group of genes. The functional coherence score assigned to a gene subgroup is the average KL-divergence for all genes in the subgroup. Each gene in the subgroup, should it be relevant to the dominant subgroup function, should have referring documents that score high. Therefore, the KL-divergence will be large. If many of the genes have relevant functions, the average KL-divergence of all genes in the subgroup will be high.

2.2. *Neighbor divergence: an alternative method to score functional coherence*

2.2.1. Neighbor divergence. The neighbor divergence scoring method was explored and evaluated in detail elsewhere (Raychaudhuri, Schutze, & Altman, 2002). Here all the documents are scored as described above; only 20 neighbors per article are utilized however. The KL divergence between the empirical distribution of all of the document scores and the theoretical Poisson distribution is used as a measure of functional coherence.

2.3. *Evaluation of NDPG*

To evaluate *NDPG* and to compare it with other approaches, we used 19 groups of yeast genes each representing a different function. We also devised 1900 decoy random yeast gene groups. We tested methods by scoring all groups. An appropriate method should assign high scores to functional groups and low scores to random groups. We calculate the precision and recall of a method at different score cutoff levels. The precision (or positive predictive value) is the number of functional groups scoring above the cutoff divided by the number of total groups scoring above the cutoff. The recall (or sensitivity) is the number of functional groups scoring above the cutoff divided by the total number of functional groups. A good method achieves 100% recall at 100% precision.

For comparison we have included the performance of *neighbor divergence*, a method similar to *NDPG*. Elsewhere, we have carefully compared *neighbor divergence* to other approaches to the same problem and explored its properties (Raychaudhuri, Schutze, & Altman, 2002). The reader is referred to that article for additional detail on that method.

2.3.1. Data types. All experiments described below are conducted in *Saccaharmyces Cerevisiae*. We used a reference index that contained PubMed abstract references to yeast genes from the *Saccharomyces Genome Database* (Cherry et al., 1998). The reference index included 20101 articles with 50860 references to 4205 genes; the article records were obtained from NCBI in Medline format. A total of 12,301 words were selected for the vocabulary. All documents were converted into 12,301 dimensional vectors of word counts.

2.3.2. Assembling gold standard functional gene groups. To test our method we assembled gold standard functional gene groups. Using manual GO annotations, we focused on "gene process" GO terms. We selected 19 diverse process GO terms relevant to yeast biology that had at least three genes. A functional group included genes assigned the listed term by the GO consortium or a more specific child of the listed term. The GO terms and properties of the groups they correspond to are described in Table 1(A). These groups varied in size and content; this diversity is representative of gene groups that experimental procedures may derive. Many genes were assigned to multiple gene groups (see Table 1(B)). This underscores the multiple-functionality that many genes have. We used the 2 Nov 2001 release of the GO process ontology and the 17 October 2001 GO gene associations for yeast.

Table 1A. Gold standard functional gene groups are created from GO codes.

Functional classification	Gene ontology code	Genes	Total article references
Signal_transduction	GO:0007165	94	3484
Cell_adhesion	GO:0007155	6	82
Autophagy	GO:0006914	16	110
Budding	GO:0007114	74	1692
Cell_cycle	GO:0007049	341	8399
Biogenesis	GO:0016043	459	6439
Shape_size_control	GO:0007148	54	1629
Cell_fusion	GO:0006947	89	2495
Ion_homeostasis	GO:0006873	43	667
Membrane_fusion	GO:0006944	6	212
Sporulation	GO:0007151	27	646
Stress_response	GO:0006950	94	2603
Transport	GO:0006810	313	4559
Amino_acid_metabolism	GO:0006519	78	1594
Carbohydrate_metabolism	GO:0005975	90	2719
Electron_transport	GO:0006118	8	205
Lipid_metabolism	GO:0006629	90	1035
Nitrogen_metabolism	GO:0006807	15	264
Nucleic_acid_metabolism	GO:0006139	676	12345

Table 1B. Many genes are in multiple groups.

Number of functional groups/gene	0	1	2	3	4	5	6
Number of genes	2412	1242	386	113	40	9	3
Total genes	4205						
Total functional group assignments	2576						

2.3.3. Assembling the decoy random gene groups. We assembled 1900 random gene groups as decoy gene groups. For each gold standard functional gene group, 100 random gene groups of the same size were assembled. The random gene groups constitute a poor negative set of gene groups since many experimentally derived groups are rarely completely random. However, it is sufficient for use in comparing different methods and also to establish a performance baseline for *NDPG*.

2.3.4. Performance. *NDPG* achieves 95% recall (18 out of 19 functional groups) at 100% precision; this is equivalent to 95% sensitivity at 100% specificity. In figure 2 we have plotted the precision and recall at different cutoff levels for *NDPG* and for *neighbor divergence*. As the cutoff score is selected to be more stringent, some functional groups are not obtained

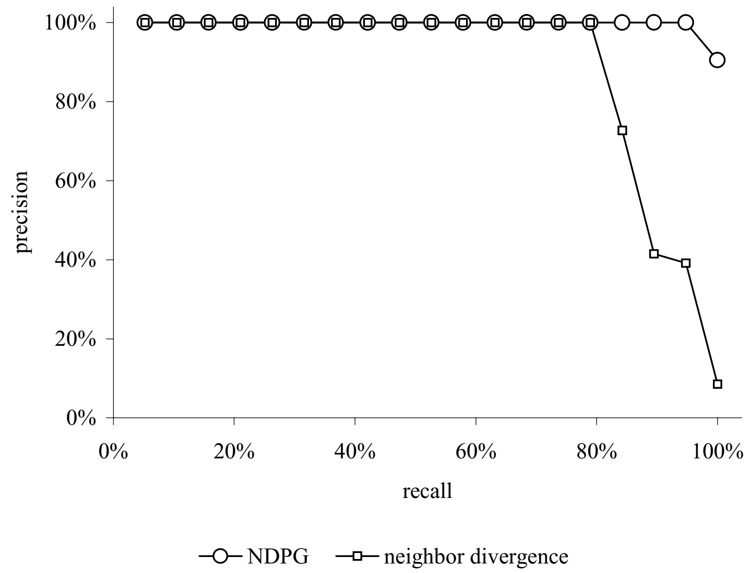

Figure 2. Precision-Recall plot for two functional coherence scoring methods. We used both methods to score the functional coherence of the 19 functional gene groups and the 1900 random gene groups. We calculated and plotted precision and recall at cutoff scores of different stringency. There is a tradeoff between precision and recall. More stringent cutoff values select fewer true functional groups and recall (or sensitivity) is compromised; however less stringent cutoff values cause many random groups to be selected inappropriately and precision is compromised. An ideal precision-recall plot achieves 100% precision for every value of recall. The *NDPG* method is closest to the optimal curve.

and therefore recall is lower. But, most random groups fail to make the cutoff and the precision is higher. For comparison, *Neighbor divergence* achieves 79% recall (15 out of 19 functional groups) at 100% precision; this is equivalent to 79% sensitivity at 100% specificity.

In figure 3 we have plotted the distribution of *NDPG* scores for the 1900 random gene groups and the 19 functional gene groups. While there is some overlap, most functional groups have scores that are about an order of magnitude higher than the highest score assigned to a random gene group.

The only adjustable parameter is the exact number of semantic neighbors that should be calculated for each article. The performance is robust to the number of neighbors; 95% recall at 100% precision is achieved with 19, 49, or 199 neighbors. However 199 neighbors achieves the highest precision at 100% recall. At 100% recall 199 neighbors achieves 90.5% precision, while 49 and 19 neighbors achieves 66% and 59% recall respectively.

2.4. Bias

We wanted to insure that the *NDPG* method was not biased towards specific functional group types. In the next section we use *NDPG* to search for projections that separate functional

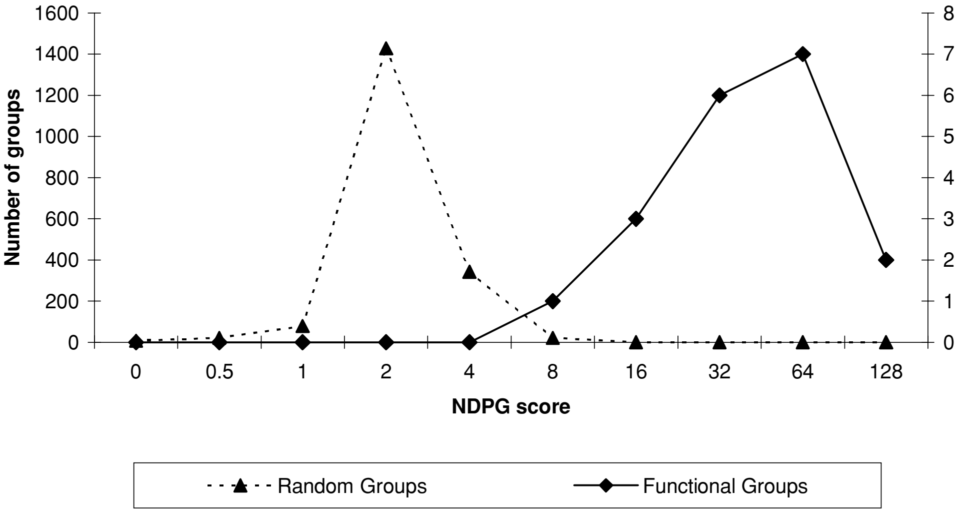

Figure 3. Histogram of NDPG scores. Each closed triangle represents (▲) the count of random gene group scores in the range indicated on the horizontal axis; each closed diamond (◆) represents the count of functional gene group scores in the range on the horizontal axis. There is little overlap between the two histograms. None of the random gene groups score above 12; most of the functional gene groups score well above 12.

gene groups. Here we want to insure that selected groups, obtained by search, are not subject to some sort of systematic error.

NDPG performance is robust to different size gene groups. Smaller functional groups usually contain fewer genes, fewer documents, and consequently fewer document references; they can be more difficult to discover. Figure 4 plots the *NDPG* and *neighbor divergence* scores of the functional groups as a function of the number of article references in the groups. *NDPG* scores do not appear to be biased toward or against larger functional groups with more references. This bias is apparent with *neighbor divergence*.

To insure that *NDPG* scores were not biased toward subgroups of genes with heterogeneous functions, for example gene subgroups that contain all genes with one of two very disparate functions, we conducted a simple experiment. We combined all pairs of the functional gene groups in Table 1(A) and scored them. Ideally, the combined group score should be less than that of the individual scores of the constituting groups. Of the 171 possible combined groups, only one had an *NDPG* score exceeding both parent groups (Table 2). In contrast *neighbor divergence* scores exceeded that of both parents 76 times.

Table 2. Functional coherence scores of two groups combined compared to individual parent scores.

	Combined group < both parent scores	Combined group between both parent scores	Combined group > both parent scores
Neighbor divergence	4	91	76
NDPG	72	98	1

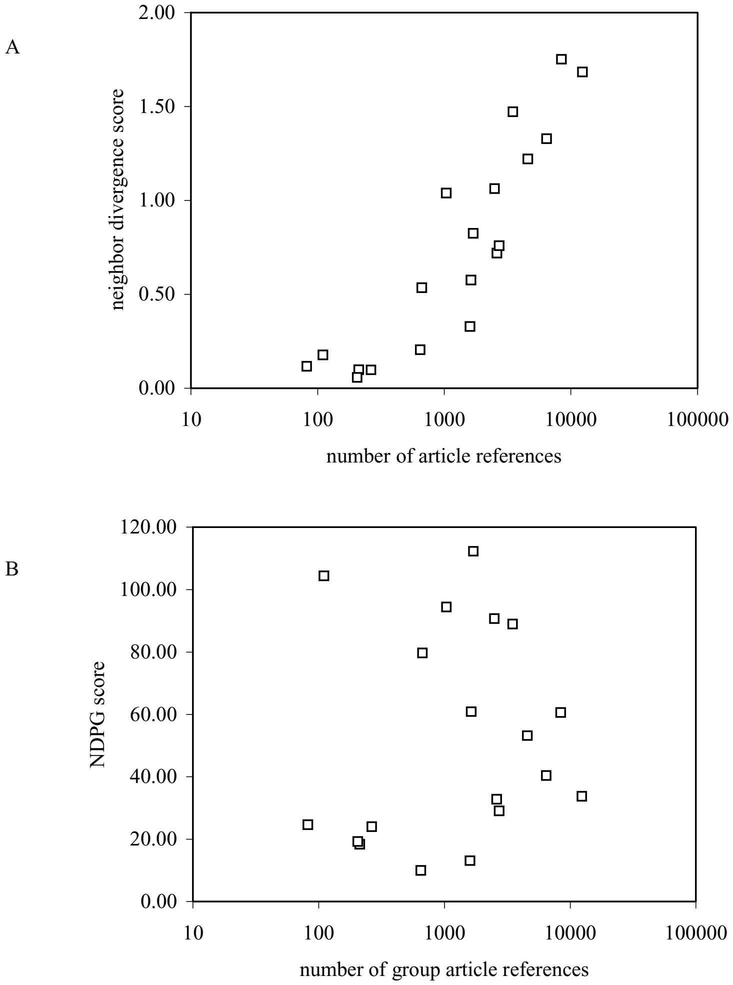

Figure 4. *NDPG is not biased by group size.* (A) *Neighbor divergence* score of the 19 functional groups as a function of number of article references. (B) Similar plot for *NDPG.*

3. Finding projections that separate coherent gene groups: *Optimal Scoring Projection* method

In this section we return the focus to multidimensional data sets. The general approach presented here is to choose criterion in gene expression data that separate genes with a common function from the remainder. We use linear projections as the criterion to separate genes. The optimization algorithm introduced here selects linear projections in the gene expression data for which positively projecting genes have a common biological function. Since the selected genes project positively along the same line in gene expression data, the genes share common features in gene expression.

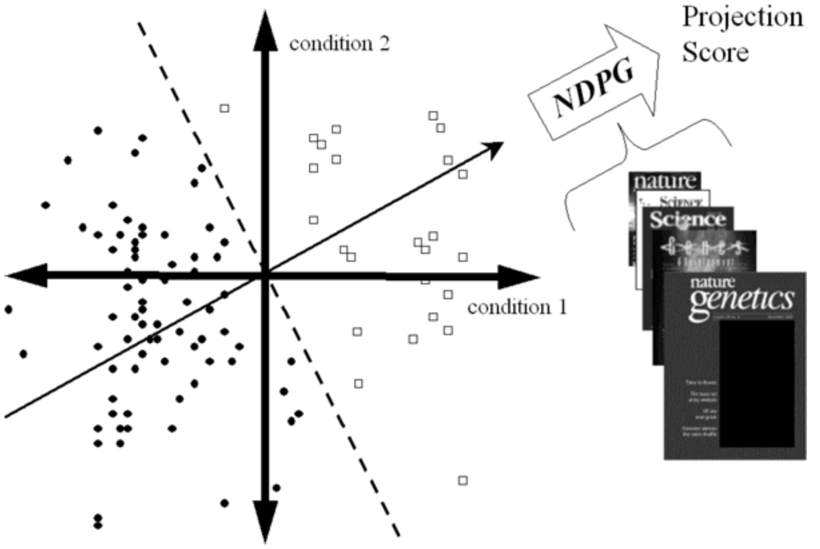

Figure 5. Scoring a linear projection in gene expression data for functional coherence. Here each axis represents gene expression condition in a two dimensional gene expression data set. Each point represents a gene; it is positioned to represent its expression value in both conditions. The tilted line with the arrow represents a projection in gene expression space used to separate the genes. The dotted line perpendicular to the vector represents the plane of separation; the genes that project positively on the vector are to the right of the dotted line. The genes that project positively are the ones selected by the projection (□). The remaining genes are not selected (●). The projections functional coherence score is the *NDPG* score of the genes it selects. The *NDPG* method leverages the scientific literature to assess functional coherence of a group of genes. The *OSP* selects projections with high a score.

To insure that there is a shared underlying function in the genes selected by the projection, the projection is selected so that the separated genes have a high *NDPG* score. In the previous section we demonstrate that given literature about genes, a group of genes can be scored for functional coherence with the *NDPG* approach. A random group of genes will score poor, while groups with genes sharing a common underlying biological function will score high.

We introduce the *optimizing separating projections* (*OSP*) method. The algorithm iteratively optimizes the projection so that the selected genes have improved functional coherence. Each linear projection is scored as the *NDPG* score of the positively projecting genes; the *NDPG* score acts as an objective function (figure 5). There are many conceivable means to find optimal high scoring linear projections in data. *OSP* starts with an initial training set of genes. Then, it uses linear discriminant analysis (LDA) line to define the projection that best separates those genes from the remainder. Then genes that project positively along the LDA projection are used to create several different candidate training sets. Some genes are removed from the candidate training sets to increase the *NDPG* score and define additional candidate training sets. The candidate training set whose LDA training line has the largest *NDPG* score is used in the next iteration. A brief overview of the algorithm is presented in Table 3. *OSP* converges to a local maximum.

Table 3. Basic *Optimal Scoring Projection* algorithm.

```
Given: initial training set of genes
⋆Repeat until the selected Set of genes does not change:
     ⋆Train LDA function with (positive set = training, negative set = other genes).
     ⋆Use LDA function to calculate log likelihood ration for all genes.
     ⋆Define candidate training sets j=1 to 7:
          ⋆Candidate training set 2j-1:
               All genes with likelihood ratio > 2^(-4+j)
          ⋆Candidate training set 2j:
               All genes after filtering candidate training set 2j-1
     ⋆Define Candidate training set 15
               Genes in original training set
     ⋆For each candidate training set:
          ⋆Train LDA with candidate training
          ⋆Calculate NDPG score of all genes with log likelihood ratio greater than 0
          ⋆If the candidate training set achieves highest NDPG score, replace
               training set with candidate training set
```

The data types employed in this algorithm are a corpus of articles, an index of references connecting documents to genes, and a gene expression data set containing log expression ratios for each gene across multiple experimental conditions. The first two data types are used to calculate *NDPG* score for groups of genes.

The approach is related to linear supervised machine learning methods (Ripley, 1996). In linear machine learning algorithms, the user specifies known examples of genes with a specific biological function. Then a machine learning algorithm identifies a linear projection that separate these genes from the remainder. Some functions are learnable in the context of gene expression data; that is, there is a projection that can separate them effectively from the gene expression data set. The method presented here does not require any previously known examples; it searches for possible training sets that can be separated by projections; these training sets also should have the property of a shared biological function.

The method is distinct from unsupervised machine learning which does not incorporate any previous knowledge about genes (Sherlock, 2000). Unsupervised machine learning groups genes also, but since it lacks previous knowledge, it cannot leverage flexible rules to select genes with known commonalities before hand. Unsupervised machine learning usually relies on an inflexible metric that defines similarity between two genes.

3.1. Linear discriminant analysis

Linear discriminant analysis (LDA) is a central aspect of the algorithm we implement, so we briefly review it here. Linear discriminant analysis (LDA) is a supervised machine learning method that finds a linear projection that separates two classes of genes in multidimensional data. Here we will refer to one class as the positive set, the other the negative set. Linear discriminant analysis uses the labeled examples from each of the two sets to estimate a probability distribution for the values of the features in that set. The densities are assumed to be normal distributions. Given an unclassified example, it uses the set densities to calculate log likelihood ratios (Ripley, 1996). Generally, positive log likelihood scores indicates that the unclassified example is in the positive set, while negative scores predict the example is in the negative set. Given an unclassified example with features X the log likelihood that it is in the positive set is estimated:

$$\log \frac{P(+ \mid X)}{P(- \mid X)} = \log \frac{\pi_+}{\pi_-} - \frac{1}{2}(\mu_+ + \mu_-)^T \sum^{-1}(\mu_+ - \mu_-) + X^T \sum^{-1}(\mu_+ - \mu_-)$$

where π is the prior probability of cases for the class, μ is the mean feature vector for the training examples in each class, and $\sum$ is the pooled covariance matrix of the two classes. Here we compute $\sum$:

$$\sum = \frac{1}{2}\left(\sum_{+} + \sum_{-}\right)$$

When we refer to the LDA line, we are referring to the log likelihood equation produced by LDA.

3.2. Filtering out genes

Critical to the method is a means to remove problematic genes whose function is not relevant to the group as a whole. To determine which genes do not belong to the set of genes given, the algorithm uses a filtering step. First the algorithm removes each gene individually and recalculates the *NDPG* functional coherence score without it. Any gene whose removal causes an increase in functional coherence is removed from the set.

3.3. Finding projections that discriminate functional subgroups of genes

An outline of the algorithm is presented in Table 3. The algorithm begins with a starting training set of genes. The starting training set may be a set of genes that have clustered together or a completely random set of genes. The training set is used as a positive training set while the remaining genes are used as a negative training set in LDA. The resulting LDA projection should separate the set form the remaining genes effectively if the genes are separable in the data set. The LDA linear function is used to score all genes.

Seven candidate training sets are devised. All genes whose LDA likelihood ratio of being in the positive set is greater than 8, 4, 2, 1, 1/2, 1/4, and 1/8 respectively are selected for each of the candidate training sets. We restrict the number of genes in these candidate sets to be no less than 8 and no more than 150. Application of the filtering procedure described in Section 3.2 to each of the seven candidate training sets creates an additional seven candidate training sets.

Each of the fourteen candidate training sets, and the original training set are used to create as positive sets to create new LDA lines. The *NDPG* score for all genes that project positively onto each LDA line (i.e. have a likelihood ratio of 1) is used to score it (figure 5). The highest scoring line and its corresponding training set are kept for the following iteration, the other candidate training sets are discarded. The procedure is repeated for ten iterations or until the same training set genes are selected.

4. Application to gene expression data set

To test the algorithm we applied it to a well known published yeast gene expression data set (Eisen et al., 1998). The data set measures gene expression for 2467 genes over 79 different conditions including cell cycle time series experiments, sporulation time series experiments, heat shock experiments, and metabolic time series experiments. Article abstracts relevant to these genes were obtained from a reference index connecting yeast genes to articles (Cherry et al., 1998). A total of 2394 genes had 40351 references to 17858 articles. The median number of gene references per article was 2; the median number of article references per gene was 8.

Initially, k-means algorithm was employed to produce 60 clusters. The resulting clusters were used as initial starting training sets for the algorithm described in Section 3. The resulting 60 projections were sorted according to the *NDPG* score of genes that projected positively onto them. Of the 60 projections obtained, 56 had significant *NDPG* scores greater than 16. To avoid projections that selected similar sets of genes, we removed projections if 50% of the genes it selected were contained in the selected genes of higher scoring projections. This resulting 21 projections and corresponding *NDPG* scores are listed in Table 4.

4.1. Evaluation

To evaluate these projections we examined the genes they selected. If the selected genes represent a known biological function, then the projection is informative and the method has succeeded. We use the Gene Ontology (GO) gene function assignments as a gold standard. We investigate whether the genes selected by the projections have high precision and recall for any Gene Ontology function code.

The GO Consortium has developed a controlled vocabulary for function (Ashburner et al., 2000). This vocabulary contains a set of codes associated with specific genetic functions. The GO is a hierarchically arranged set of codes organized into three broad components: molecular function, cellular location, and biological process. A term applies to a gene if it was directly associated to it by the GO consortium or if one of the children of the term was associated. We used the 23 January 2002 release of GO component ontology, the 24 January 2002 releases of the GO process and function ontologies, and the 24 January 2002 GO gene associations for yeast. It should be noted that since the commencement of this research many new GO yeast gene assignments and reassignments have been made and the vocabulary itself is in flux. As a gold standard, it is not yet perfect, but is today among the best yeast gene annotation resources available.

After removal of the redundant projections, we find the GO codes obtaining the highest precision and recall for each gene projection. Often several biologically related functions were obtained in the same projection. In Table 5 we list alongside each of the 21 projections

Table 4. Selected projections after application to gene expression data set.

Projection #	*NDPG* score of selected genes
1	107.18
2	91.60
4	76.90
5	75.88
12	69.95
16	57.52
18	42.54
19	41.55
20	40.55
21	39.93
23	35.55
25	34.84
28	34.52
32	32.19
35	29.50
45	24.30
47	23.44
48	22.69
51	20.79
57	15.20
60	11.56

the most relevant GO codes and the precision and recall of the selected genes for that GO code. If all selected genes have that function, 100% precision for that function is achieved. If all of the genes with that function are selected, 100% recall for that function is achieved.

Many of the projections correspond well to a specific GO code, or multiple related codes. For example, projection #4 selects 100% of the genes assigned the function *nucleosome* by GO. These genes are involved in the packaging of DNA into chromosomes. Of the 9 genes the projection selects, 8 of them are annotated with this function. Similarly, projection #12 selects 70.6% of the genes that were assigned the *heat shock* function by GO. These genes are expressed when the cell is exposed to sudden heat; many are involved in protein synthesis. Of the 18 genes selected, 12 have this function. Not surprisingly, projection #12 also selects many "protein folding" genes.

When the data was first published the investigators identified certain functional clusters. These clusters were assigned a function label that indicates the authors' impression of the function represented in the cluster. The clusters were derived from hierarchical clustering. Experts manually identified the clusters and their boundaries (which level of the tree to cut) so that they best represented some biological function. We list the precision and recall of the most relevant GO terms in Table 6.

Table 5. GO codes that correspond to discovered projections.

Projection #	*N*	Selected GO codes	Precision (%)	Recall (%)
1	39	Threonine endopeptidase	69.2	90.0
		26S proteasome	74.4	87.9
		20S core proteasome	33.3	92.9
		19S proteasome regulatory particle	41.0	88.9
2	12	Genetic exchange	75.0	20.9
		Developmental processes	75.0	12.7
		Mating (sensu Saccharomyces)	66.7	20.0
		Pheromone response (sensu Saccharomyces)	41.7	33.3
4	9	Nucleosome	88.9	100.0
5	18	Heat shock protein	66.7	70.6
		Protein folding	77.8	40.0
12	129	Cytosolic ribosome	89.1	92.7
		Protein biosynthesis	93.8	51.3
16	9	Protein disulfide oxidoreductase	22.2	66.7
		Protein metabolism and modification	55.6	0.9
		Endoplasmic reticulum	44.4	6.3
		SRP-dependent, co-translational membrane targeting	22.2	18.2
18	45	ATP dependent DNA helicase	11.1	71.4
		Pre-replicative complex	13.3	75.0
		DNA unwinding	15.6	77.8
		Replication fork	31.1	38.9
19	8	Fructose transporter	62.5	35.7
		Glucose transporter	62.5	29.4
20	28	Mitochondrial electron transport chain complex	17.9	100.0
		Electron transport	25.0	87.5
		Tricarboxylic acid cycle	28.6	53.3
		Proton-transporting ATP synthase complex, catalytic core F(1)	7.1	40.0
21	55	Mitochondrial electron transport chain complex	9.1	100.0
		Tricarboxylic acid cycle	21.8	80.0
		Electron transport	12.7	87.5
		Proton-transporting ATP synthase complex, catalytic core F(1)	7.3	80.0
		Proton-transporting ATP synthase complex, coupling factor F(0)	5.5	60.0

(*Continued on next page*.)

Table 5. (*Continued*).

Projection #	*N*	Selected GO codes	Precision (%)	Recall (%)
23	54	Nucleolus	48.1	37.7
		Transcription, from Pol I promoter	51.9	33.7
		rRNA processing	42.6	41.8
		Ribosome biogenesis	48.1	31.3
25	41	Inner plaque of spindle pole body	7.3	75.0
		G2/M-specific cyclin	7.3	75.0
		Anaphase-promoting complex	12.2	62.5
		Mitotic spindle assembly (sensu Saccharomyces)	4.9	66.7
		Microtubule cytoskeleton organization and biogenesis	34.1	36.8
		Mitotic spindle assembly	17.1	53.8
		Microtubule organizing center	24.4	45.5
		Microtubule-based process	34.1	34.1
		Degradation of cyclin	12.2	55.6
28	14	Developmental processes	64.3	12.7
		Conjugation (sensu Saccharomyces)	14.3	50.0
		Genetic exchange	42.9	14.0
		Sex determination	14.3	40.0
32	15	Steroid biosynthesis	40.0	30.0
		Steroid metabolism	40.0	27.3
35	65	Glycolysis	23.1	62.5
		Glucose catabolism	23.1	60.0
		Glyceraldehyde 3-phosphate dehydrogenase	4.6	100.0
		Glyoxylate cycle	4.6	75.0
45	23	Peroxisomal matrix	13.0	23.1
		Electrochemical potential-driven transporter	17.4	12.5
47	18	Glucan metabolism	11.1	50.0
		Polysaccharide metabolism	11.1	20.0
48	122	Mitochondrial ribosome	23.8	60.4
		Mitochondrial matrix	34.4	50.6
		50S ribosomal subunit	16.4	66.7
51	13	M phase	38.5	4.8
		G2/M-specific cyclin	15.4	50.0
		Transcriptional activator	15.4	33.3
		Regulation of CDK activity	15.4	22.2
57	6	DNA-directed RNA polymerase	50.0	10.7
		DNA-directed RNA polymerase II	33.3	18.2
60	12	Fatty-acid ligase	16.7	40.0
		Chitin synthase	8.3	33.3

Table 6. Published clusters from Eisen, and relevant GO codes.

Cluster label assigned by Eisen	*N*	Selected GO codes	Precision (%)	Recall (%)
ATP synthesis	14	Proton-transporting ATP synthase complex, coupling factor F(0)	35.7	100.0
		Proton-transporting ATP synthase complex	64.3	69.2
		Proton-transporting ATP synthase complex, catalytic core F(1)	28.6	80.0
Chromatin structure	8	Nucleosome	100.0	100.0
DNA replication	5	ATP dependent DNA helicase	80.0	57.1
		Pre-replicative complex	80.0	50.0
		DNA unwinding	80.0	44.4
		Replication fork	80.0	11.1
Glycolysis	17	Glycolysis	70.6	50.0
		Glucose catabolism	70.6	48.0
		Glyceraldehyde 3-phosphate dehydrogenase	17.6	100.0
		Carbohydrate catabolism	70.6	42.9
Mitochondrial ribosome	22	Mitochondrial ribosome	50.0	22.9
		Mitochondrial matrix	50.0	13.3
		50S ribosomal subunit	36.4	26.7
mRNA splicing	14	mRNA splicing	28.6	7.4
		RNA splicing	28.6	5.6
		Spliceosome	14.3	16.7
Proteasome	27	Threonine endopeptidase	92.6	83.3
		26S proteasome	96.3	78.8
		20S core proteasome	44.4	85.7
		19S proteasome regulatory particle	51.9	77.8
Ribsome and translation	125	Cytosolic ribosome	88.8	89.5
		Protein biosynthesis	96.0	50.8
Spindle pole body assembly and function	11	Septin ring (sensu Saccharomyces)	18.2	50.0
		Structural protein of cytoskeleton	45.5	18.5
		Spindle pole body	27.3	13.6
		Mitotic spindle elongation	18.2	22.2
Tricarboxylic acid cycle and respiration	16	Mitochondrial electron transport chain complex	31.3	100.0
		Electron transport	37.5	75.0
		Tricarboxylic acid cycle	31.3	33.3

Nine of the ten clusters correspond to one of the *OSP* projections; the genes selected by the projection and the genes in the published cluster have similar function.

The genes selected by projection #1, the highest *NDPG* scoring projection, have a similar function to the genes in the "Proteasome" cluster. These genes are involved in the natural breakdown and digestion of unnecessary proteins. For the relevant GO codes, the genes selected by *OSP* have slightly higher recall and slightly lower precision. For example 90% of the *threonine peptidase* are selected by the projection compared to 83.3% by the cluster. However, only 69.2% of the genes selected by projection #1 are *threonine peptidase* genes compared to 92.6% of the genes in the cluster.

Projection #4 selects *nucleosome* genes. Its precision and recall are comparable to that of the published "Chromatin Structure" cluster.

Projection #12 selects 92.7% of the *cytosolic ribosome* genes. The proteins encoded by these genes constitute a complex that synthesizes proteins. The genes selected by the projection obtain precision and recall that is comparable to that of the published cluster for the *cytosolic ribsome* function.

The 45 genes selected by projection #18 are similar in function to the 5 genes in the "DNA replication" cluster. These genes are involved in *DNA unwinding* and constitute the *replication fork*. These genes are involved in the initial stages of DNA replication. The projection achieves considerably higher recall, but trades off precision. For example the projection selects *replication fork* genes with 31.1% precision, but 38.9% recall; the cluster selects genes with the same function with 80% precision and 11.1% recall.

Projections #20 and #21 are similar in function. Both include genes that are involved in aerobic respiration and the synthesis of ATP. The functions represented are similar to the ones in "ATP Synthesis" and "Tricarboxylic acid cycle and respiration" clusters. Performance is comparable, except the projections combine the genes in the two separate clusters. Since the two related functions are combined by the projection, lower precision for each of the individual functions result, even though the recall is comparable.

Projection #25 selects 41 genes; many are involved in spindle pole body and assembly. Microtubules are heavily involved in this process, so many of the genes involved in microtubule reorganization are also selected by this projection. This projection is similar in function to the cluster "Spindle Pole Body Assembly and Function". But it selects for many genes with related function with higher recall.

Projection #35 selects genes involved in the glycolysis pathway; carbohydrates are broken down in this pathway into metabolically useful products. The selected genes have higher recall (62.5% vs. 50%) and lower precision (23.1% vs. 70.6%) than the published "Glycolysis" cluster for the function *glycolysis*. Other related functions are selected by the projections also, such as *glyoxylate cycle* genes.

Projection #48 selects mitochondrial ribosome genes; these genes are also involved in protein synthesis, but are located in a mitochondrial complex. It selects genes with the function *mitochondrial ribosome* with much higher recall (60.4% vs. 22.9%) but at the cost of lower precision (23.8% vs. 50%).

We did not find a projection that selected genes that had the same function as the "mRNA splicing" cluster. But that cluster did not correspond that well with any GO code. It contained only 7.4% of the genes assigned the *mRNA splicing* code by GO.

Many of the other projections selected genes that had a clear function. For example projection #2 selected genes involved in *mating* and *genetic exchange*. Also cluster #5 selected *heat shock* genes. Projection #19 selected for genes involved in *fructose transport*. Projection #23 selected *nucleolus* genes; since these genes constitute an organelle involved in ribosomal RNA (rRNA) synthesis this projection was relevant to *ribosome biogenesis*. Projection #32 selected genes involved in *steroid biosynthesis*.

Some of the projections do not seem to clearly represent any single GO code. Specifically, the common function of genes selected by projections #16, #28, 45, #47, #51, #57, #60 are not immediately obvious. Three of these were the lowest scoring projections that we evaluated. The most relevant GO codes suggest certain functions, but not with high enough precision or recall to be confident about. But a detailed review by a qualified expert may reveal the relationship between the genes selected by some of these clusters.

Generally, most of the projections derived by *OSP* select genes that have related function. The *OSP* method is able to find the functions published by Eisen. Eisen et al. discovered these functions after careful expert screening of clusters derived by an agglomerative hierarchical algorithm and subsequent careful cluster boundary determination. Our algorithm, on the other hand, required no human input. The genes selected by *OSP* projections, had, comparable precision and recall to GO function codes. In general *OSP* obtained higher recall, but lower precision for the same functions. *OSP* was also able to find other functions and their projections that had consistent gene expression patterns that were not published.

5. Conclusions

Gene expression data sets are large and contain diverse measurements over many genes. Successful analysis of these data sets, require investigators to find patterns of expression for genes with a common function. The difficulty is that not only are the number of genes large, but the number of possible relevant functions are enormous also. The paradigm of gene expression data today remains tedious. While clustering can reduce the large number of genes to palpable subsets, what remains is a large number of clusters that need to be independently manually examined. Investigators routinely spend weeks pouring over large cluster dendrograms of gene expression data in the hopes of identifying functionally relevant clusters.

Here we applied the *OSP* algorithm to gene expression data. Several hours of computation yielded a plethora of biologically relevant information. Twenty-one gene expression patterns that could potentially be explained by known documented biology were derived. The functions described included those functions recognized by Eisen among other ones not previously noted. We required no human input to analyze and distill; *OSP* did that automatically.

The advantage of our algorithm is that it uses literature directly as a knowledge source to interpret data. It searches for projections that separate groups of genes out that have common biological principles described within the literature. This is particularly valuable in instances where obvious labels for the data are undetermined or ambiguous, preventing the straightforward application of supervised machine learning approaches.

The approach presented here has another advantage in that it obtains a projection that separates functionally coherent genes from the remainder. Projections in gene expression can often be interpreted by examining the conditions that are heavily weighted (Raychaudhuri, Stuart, & Altman, 2000). Those conditions are the ones that the genes with that function are most actively repressed or induced. So the projections can help decipher the biology of the data set.

Furthermore, since genes frequently have multiple functions that they may be involved in, they may under some of the conditions exhibit the behavior of genes with one function and in other conditions exhibit the behavior of genes with a different function. Here a projection may be effective at selecting for a specific function. The same gene can be selected by multiple projections, each emphasizing the different experimental conditions most critical to the function it represents. Projections can focus on the critical features (in this case gene expression conditions).

We presented a simple and straightforward optimization strategy in *OSP*. Unfortunately, it does not find a global maximum. Other strategies may be more effective than the one presented here. A gradient ascent approach could be implemented, for example; we did not do so here because of the computational intensiveness of such an approach. Other search algorithms need to be explored.

We relied heavily on LDA, since it had been shown to be effective in gene expression analysis and can be trained rapidly (Brown et al., 2000). Other linear machine learning approaches could be substituted into *OSP* instead of LDA that may be more effective. However, it is critical that whatever machine learning approach is employed that it is not so flexible that it easy to over-train. Any method that can be over-trained easily will select the same training points over and over again, preventing proper opportunities to derive new and different candidate training sets.

An effective metric to measure the functional coherence of a group of genes is critical to the success of this measure. Here we used *Neighbor Divergence Per Gene* (*NDPG*). This method was 95% sensitive and 100% specific at identifying functional yeast gene groups. The method is not confounded by the multi-functionality of genes since it is an article-based approach. In biology, articles often address specific subjects, even though the genes they are about may have multiple unrelated functions. It is also robust to certain types of bias that another functional coherence measure that we employed was susceptible to. It does not favor larger functional groups, nor does it favor groups of genes that contain all genes relevant to multiple disparate functions.

An alternative approach is to cluster the data using k-means, self-organizing maps, or hierarchical clustering techniques first, and then score the individual clusters with *NDPG*. In the case of hierarchical clustering, *NDPG* can actually be used to determine which level of the tree to cut at also, thereby determining cluster boundaries. However, in clustering the definition of gene expression similarity is usually rigid, it is pre-defined according to a distance metric and cannot be adjusted to emphasize certain conditions more than others. In situations where only certain conditions are more critical for classification of the functional group, those groups may become difficult to find.

NDPG determines whether a group of genes has a coherent function. It does not tell us the function. The easiest way to determine the group's function is to examine the higher scoring articles for a gene group manually or automatically. These high scoring articles are the ones most relevant to the group's shared function. The high scoring articles could be collected and examined manually to determine group function. Alternatively, keywords for the group could be automatically determined automatically that describe the function of the group. Investigators have already developed algorithms to find keywords in collections of biological document that could be applied to these high scoring articles to determine functional keywords (Andrade & Valencia, 1997).

The methods described here rely on the content of the scientific literature. If the literature provides no indication whatsoever of some novel function that might be critical to the experiment, it will be difficult for our method to successfully identify that function. However, an organism's response to some novel stimuli usually includes activation of well-described pathways. Additionally, even if the function and the genes with that function are not explicitly elucidated in the literature, as long as there is some indications of genetic properties in the literature that they all share, our method should be able to assign a low, but significant score, to the group.

Currently *NDPG* does not exclude articles that have negative statements about a gene's involvement in a specific biological function or process. Since the vast majority of publications on biology are positive results, the method performs well empirically despite this. However, determining those articles with negative statements in advance and excluding them from the analysis could result in additional gains. This may be an avenue for future methodological improvement.

In the future we anticipate with additional availability of scientific text on-line that the applicability of *OSP* will improve. Full text versions of articles are becoming increasingly available on-line (Roberts et al., 2001). Also many of the genome databases are maintaining large reference indices connecting genes to appropriate articles references. Finally, genes not associated with any articles could be associated to the articles of homologous genes.

The analysis of multidimensional data assisted by textual documentation may have application in many domains. In biology, it is particularly potent, since the field is rich with published literature. Other large numerical biological data sets, such as proteomics data, could be examined in an identical manner. The methods presented here could also be applied to the analysis of any multidimensional data in other fields where there is an abundance of textual documentation. For example in medicine, free text documentation in the form of medical records is readily available, and investigators are frequently interpreting numerical data such as survival statistics and laboratory results. The *OSP* method may be an effective way to identify medical syndromes from lab data. Also, scientists can be attached to the free text documents that they have written over the course of their careers, and objective data such as grant monies received annually, number of citations, and number of students trained are obtainable. The *OSP* method can be used with this information to identify trends in science.

Acknowledgments

RBA is supported by NIH LM06244, GM61374, NSF DBI-9600637, and a grant from the Burroughs-Wellcome Foundation; SR is supported by NIH GM-07365. The authors thank Kara Dolinski of SGD for providing a reference index connecting yeast genes to articles, and Patrick D. Sutphin and Joshua M. Stuart for helpful discussions. SR thanks the members

of his thesis committee: Drs. Russ Altman, David Botstein, Serafim Batzoglou, Stuart Kim and Hinrich Schütze.

References

Alizadeh, A. A., Eisen, M.B., Davis, R. E., Ma, C., Lossos, I. S. et al. (2000). Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature, 403:6769*, 503–511.

Andrade, M. A., & Valencia, A. (1997). Automatic annotation for biological sequences by extraction of keywords from MEDLINE abstracts. Development of a prototype system. *Proc. Int. Conf. Intell. Syst. Mol. Biol., 5:1*, 25–32.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H. et al. (2000). Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet, 25:1*, 25–29.

Ashburner, M., & Drysdale, R. (1994). FlyBase—The Drosophila genetic database. *Development, 120:7*, 2077–2079.

Bairoch, A., & Apweiler, R. (1999). The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1999. *Nucleic Acids Res., 27:1*, 49–54.

Behr, M. A., Wilson, M. A., Gill, W. P., Salamon, H., Schoolnik, G. K. et al. (1999). Comparative genomics of BCG vaccines by whole-genome DNA microarray. *Science, 284:5419*, 1520–1523.

Ben-Dor, A., Shamir, R., & Yakhini, Z. (1999). Clustering gene expression patterns. *J. Comput. Biol., 6:3/4*, 281–297.

Blake, J. A., Richardson, J. E., Bult, C. J., Kadin, J. A., & Eppig, J. T. (2002). The mouse genome database (MGD): The model organism database for the laboratory mouse. *Nucleic Acids Res., 30:1*, 113–115.

Blaschke, C., Andrade, M. A., Ouzounis, C., & Valencia, A. (1999). Automatic extraction of biological information from scientific text: Protein-protein interactions. *Proc. Int. Conf. Intell. Syst. Mol. Biol., 2:1*, 60–67.

Brown, M. P., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W. et al. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA, 97:1*, 262–267.

Cherry, J. M., Adler, C., Ball, C., Chervitz, S. A., Dwight, S. S. et al. (1998). SGD: Saccharomyces genome database. *Nucleic Acids Res., 26:1*, 73–79.

Cho, R. J., Campbell, M. J., Winzeler, E. A., Steinmetz, L., Conway, A. et al. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell, 2:1*, 65–73.

Eisen, M. B., Spellman, P. T., Brown, P. O., & Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA, 95:25*, 14863–14868.

Eisenhaber, F., & Bork, P. (1999). Evaluation of human-readable annotation in biomolecular sequence databases with biological rule libraries. *Bioinformatics, 15:7/8*, 528–535.

Fleischmann, W., Moller, S., Gateau, A., & Apweiler, R. (1999). A novel method for automatic functional annotation of proteins. *Bioinformatics, 15:3*, 228–233.

Gelbart, W. M., Crosby, M., Matthews, B., Rindone, W. P., Chillemi, J. et al. (1997). FlyBase: A Drosophila database. The FlyBase consortium. *Nucleic Acids Res., 25:1*, 63–66.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M. et al. (1999). Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science, 286:5439*, 531–537.

Halushka, M. K., Fan, J. B., Bentley, K., Hsie, L., Shen, N. et al. (1999). Patterns of single-nucleotide polymorphisms in candidate genes for blood-pressure homeostasis. *Nat Genet, 22:3*, 239–247.

Heyer, L. J., Kruglyak, S., & Yooseph, S. (1999). Exploring expression data: Identification and analysis of coexpressed genes. *Genome Res., 9:11*, 1106–1115.

Hughes, T. R., Marton, M. J., Jones, A. R., Roberts, C. J., Stoughton, R. et al. (2000). Functional discovery via a compendium of expression profiles. *Cell, 102:1*, 109–126.

Jenssen, T. K., Laegreid, A., Komorowski, J., & Hovig, E. (2001). A literature network of human genes for high-throughput analysis of gene expression. *Nat Genet, 28:1*, 21–28.

Manning, C. M., & Schutze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge: The MIT Press.

Masys, D. R., Welsh, J. B., Lynn Fink, J., Gribskov, M., Klacansky, I. et al. (2001). Use of keyword hierarchies to interpret gene expression patterns. *Bioinformatics, 17:4*, 319–326.

Pollack, J. R., Perou, C. M., Alizadeh, A. A., Eisen, M. B., Pergamenschikov, A. et al. (1999). Genome-wide analysis of DNA copy-number changes using cDNA microarrays. *Nat Genet, 23:1*, 41–46.

Raychaudhuri, S., Chang, J. T., Sutphin, P. D., & Altman, R. B. (2002). Associating genes with gene ontology codes using a maximum entropy analysis of biomedical literature. *Genome Res., 12:1*, 203–214.

Raychaudhuri, S., Schutze, H., & Altman, R. B. (2002). Text analysis of scientific literature can automatically determine if a group of genes share a common biological function. *Genome Res., 12:10*, 1582–1590.

Raychaudhuri, S., Stuart, J. M., & Altman, R. B. (2000). Principal components analysis to summarize microarray experiments: Application to sporulation time series. *Pac. Symp. Biocomput*, 455–466.

Raychaudhuri, S., Stuart, J. M., Liu, X., Small, P. M., & Altman, R. B. (2000). Pattern recognition of genomic features with microarrays: Site typing of Mycobacterium tuberculosis strains. *Proc. Int. Conf. Intell. Syst. Mol. Biol., 8*, 286–295.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. New York: Cambridge University Press.

Roberts, R. J., Varmus, H. E., Ashburner, M., Brown, P. O., Eisen, M. B. et al. (2001). Information access. Building a "GenBank" of the published literature. *Science, 291:5512*, 2318–2319.

Ross, D. T., Scherf, U., Eisen, M. B., Perou, C. M., Rees, C. et al. (2000). Systematic variation in gene expression patterns in human cancer cell lines. *Nat Genet, 24:3*, 227–235.

Schena, M., Shalon, D., Davis, R. W., & Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science, 270:5235*, 467–470.

Shatkay, H., Edwards, S., Wilbur, W. J., & Boguski, M. (2000). Genes, themes and microarrays: Using information retrieval for large-scale gene analysis. *Proc. Int. Conf. Intell. Syst. Mol. Biol., 8:10*, 317–328.

Sherlock, G. (2000). Analysis of large-scale gene expression data. *Curr. Opin. Immunol., 12:2*, 201–205.

Spellman, P. T., Sherlock, G., Zhang, M. Q., Iyer, V. R., Anders, K. et al. (1998). Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell, 9:12*, 3273–3297.

Stephens, M., Palakal, M., Mukhopadhyay, S., Raje, R., & Mostafa, J. (2001). Detecting gene relations from Medline abstracts. *Pac. Symp. Biocomput, 52:3*, 483–495.

Tamames, J., Ouzounis, C., Casari, G., Sander, C., & Valencia, A. (1998). EUCLID: Automatic classification of proteins in functional classes by their database annotations. *Bioinformatics, 14:6*, 542–543.

Thomas, J., Milward, D., Ouzounis, C., Pulman, S., & Carroll, M. (2000). Automatic extraction of protein interactions from scientific abstracts. *Pac. Symp. Biocomput*, 541–552.

Toronen, P., Kolehmainen, M., Wong, G., & Castren, E. (1999). Analysis of gene expression data using self-organizing maps. *FEBS Lett., 451:2*, 142–146.

White, K. P., Rifkin, S. A., Hurban, P., & Hogness, D. S. (1999). Microarray analysis of Drosophila development during metamorphosis. *Science, 286:5447*, 2179–2184.

Received June 19, 2002
Revised September 16, 2002
Accepted November 8, 2002
Final manuscript November 12, 2002