Whole-Genome Functional Classification of Genes by Latent Semantic Analysis on Microarray Data

See-Kiong Ng∗ Zexuan Zhu† Yew-Soon Ong†

∗Knowledge Discovery Department, Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore 119613.
†School of Computer Engineering, Nanyang Technological University, Blk N4, #02a-32 Nanyang Avenue, Singapore 639798.

Email: skng012r.a-star.edu.sg, zhuexuan@pmail.ntu.edu.sg, ayzsong@ntu.edu.sg

Abstract

Quantitative simultaneous monitoring of the expression levels of thousands of genes under various experimental conditions using microarray experiments is very useful for elucidating the functional relationships among genes in the genomes. This paper introduces the application of latent semantic analysis (LSA) to microarray expression data for systematic, genome-wide functional classification of genes.

In the LSA approach considered here, singular value decomposition is first applied as a dimension-reducing step on the gene expression data, followed by an unsupervised clustering procedure based on vector similarities in the truncated space. Functional classification is then conducted through calling by majority on each of the resulting gene clusters. Using this semi-supervised LSA approach on microarray data, we have performed systematic functional classification on the genes in the partially-annotated yeast genome, annotating 1,753 genes under more than 200 distinct functional classes with promising results.

Keywords: whole-genome gene functional classification, microarray data analysis, latent semantic analysis, singular value decomposition.

1 Introduction

DNA microarray technology allows for the quantitative measurement of thousands of gene expression levels simultaneously. Through the use of this technology, it is possible for molecular biologists to study the differential gene expression across a set of related assays. As a high throughput technology, it allows whole genomes to be scanned, generating thousands of data points per microarray experiment. Analysis of these genomic data has enabled new ways of looking at the biology of living organisms.

To reveal the various functions of the genes in a genome, the gene expression profiles of a series of experimental assays (or conditions) can be analyzed to group genes into clusters based on the similarity in their patterns of expression. These co-expression clusters can then be interpreted as functional groupings for the genes, with each cluster containing genes that encode proteins required for a common function.

However, the learning of gene functional classes from whole-genome microarray expression data is not an easy one, even for sophisticated machine learning algorithms such as support vector machines (Brown, Grundy, Lin et al. 2000) and multi-layer perceptrons (Mateos, Dopazo, Jansen et al. 2002). Some inherent problems include:

1. For genome-wide functional analysis, the number of experimental conditions (the “features”) is easily out-numbered by the number of genes in the genome. Typically, there are several thousands of genes but only tens or hundreds of different experimental assays or conditions.

2. There is a large number of functional classes to be learned. For example, there are ∼100 functional classes cataloged in the MIPS database (Mewes, Prishman, Guldener et al. 2002). As a result, an inherent problem for whole-genome functional classification is the imbalance in the number of positive and negative examples with respect to each function class. The noise in the relatively large proportion of the negative training examples can easily outweigh the small number of positive examples in each class, making it difficult for machine learning. In fact, Mateos et al. (2002) found that only ∼10% of the gene functional classes are machine-learnable.

We propose here the application of latent semantic analysis (LSA) to the problem of systematic, genome-wide functional classification of genes from microarray expression data. In LSA, singular value decomposition (SVD) is first applied as a dimension-reducing step on the gene expression data, followed by an unsupervised clustering procedure to group genes with similar expression in the truncated gene expression space. Classification is then conducted with functional assignment by majority voting in each of the resulting gene clusters. The use of dimension reduction by SVD helps to de-noise the data as well as enable the clustering algorithm to focus only on significant components present in the expression data. The unsupervised pre-classification clustering considers the grouping of all classes globally, making it less susceptible to the imbalance of training examples by individual classes. Using this semi-supervised LSA approach on microarray data, we have performed systematic functional classification on the genes in the partially-annotated yeast genome and annotated 1,753 under 266 distinct functional classes from the Comprehensive Yeast Genome Database (CYGD) available from Munich Information Centre for Protein Sequences (MIPS) (Mewes et al. 2002).

The rest of this paper is organized as follows: in Section 2, we provide some background information on gene clustering and classification, and on latent semantic analysis. We describe the data used and our
method in details in Section 3. Finally, in Section 4, we present the performance evaluation of our classification approach on whole-genome yeast gene classification, and show the performance of annotation on unknown yeast genes.

2 Background

In this section, we provide the background information for the key concepts in this paper. First, we discuss about the differences between gene clustering and classification using supervised and unsupervised approaches. We then provide some background information on latent semantic analysis and singular value decomposition, and we describe some related work on using LSA or SVD for the analysis of gene expression data.

2.1 Gene Clustering and Classification

The living cell is a complex system comprising multiple cellular pathways that performs different biological functions. Through genome-wide measurements of the mRNA expression levels across different experimental conditions, we can construct a global map of the functional associations of the various genes in the genome based on their differential expression patterns under different conditions. This approach is called gene clustering—it involves the process of organizing genes into different functional groups using a similarity (or distance) measure on the gene expression data, but using no prior knowledge of the true functional classes of the genes. Gene clustering employs unsupervised machine learning techniques such as self-organizing maps (Tamayo, Slonim, Mesirov et al. 1999, Toronen, Kolehmainen, Wong & Castren 1999) and hierarchical clustering (Eisen, Spellman, Brown & Botstein 1998) to learn directly from the expression data.

When we have prior information about the functions of some of the genes, supervised approaches may be used. In fact, biologists often already knew a subset of genes involved in a biological pathway of interest. Such domain knowledge can be used—in the form of training sets—in gene classification for supervised machine learning techniques such as support vector machines (Brown et al. 2000) and neural networks (Mateos et al. 2002) which can then learn to assign function to unknown genes from annotated genes using the known training examples. However, for whole-genome functional classification, these supervised techniques can suffer from the inherent imbalance in the positive and negative training examples for each of the functional classes, as the total number of functional classes in a living cell is large. Researchers have attempted to combat this problem by refining the design of the machine learning algorithms—for example, Brown et al. (2000) modified the kernel values for their support vector machines to adjust for the misproportions in the positive and negative training examples.

In this paper, we adopt a combination of unsupervised clustering approach followed by a calling-by-majority semi-supervised approach to perform the multi-class functional annotation of genes in whole genomes. By not using prior classification information, the unsupervised clustering is unaffected by the positive-negative population imbalance in the predefined classes. However, in the absence of such prior information, an incompetent clustering algorithm would generate clusters of genes that do not correspond well to the true underlying functional classes. As such, we use singular value decomposition to mathematically pre-analyze the expression data, and we will show in Section 4 that this unsupervised clustering approach based on the latent semantic analysis approach (that have been previously proven successful in text mining applications) can indeed cluster the data with accuracy without supervision.

2.2 Latent Semantic Analysis

Latent semantic analysis, or LSA, is a popular analysis method in the text mining community. LSA uses singular value decomposition to map the high-dimensional word-document frequency count matrix to lower-dimensional latent “semantic” space wherein text terms and documents that are closely associated are placed near one another (Deerwester, Dumais, Furnas et al. 1990, Landauer & Dumais 1997). Dimension reduction can then be carried out as a preprocessing step, followed by clustering in the resulting truncated space. Here, we map this text mining approach into gene expression analysis by noting the similarity between the word-document count matrix and gene-sample expression data matrix—with “text terms” corresponding to genes, and “documents” corresponding to a sample (or an expression experiment). We can therefore apply LSA in a similar fashion for microarray data analysis as in text mining.

2.2.1 Singular Value Decomposition

LSA uses singular value decomposition (SVD) as a dimensionality reduction technique. In SVD, any $m \times n$ gene expression matrix $A$ (i.e., $m$ genes and $n$ experiment samples, where $m > n$ typically) can be decomposed into a product of three other component matrices in the relation $A = U \cdot W \cdot V^T$, where:

- The component $m \times n$ matrix $U$ describes the original row entities in $A$—i.e. the genes—as vectors of derived orthogonal column values (called the “eigensamples”), while the $n \times n$ matrix $V$ describes the experimental samples (the original column entities in $A$) in terms of the so-called “eigengenes”; and

- The third component matrix $W$ is an $n \times n$ diagonal matrix containing $n$ scaling values indicating the relative significance of the eigenvectors.

Using SVD, we can reduce the dimensionality of the problem space simply by ignoring the insignificant coefficients in the diagonal matrix $W$. The reconstructed matrix is a least-squares best fit that uses fewer than the number of components present in the original data.

2.2.2 LSA and Gene Expression Analysis

In text mining, LSA involves the application of SVD with dimension reduction in order to reveal the underlying semantic components. In gene expression analysis, Alter et al. (2000, 2001) have also shown that much of the expression information in the microarray data can be captured by several significant eigenvectors, indicating the suitability of dimensional reduction with SVD in gene expression data analysis. In fact, they have found that some of the significant eigenvectors indeed represented independent biological and experimental processes that contributed to the overall expression. Their observation suggests that SVD can indeed be used to reveal the implicit higher-order structure—such as the functional structures of the genes—in the gene expression data. Dimension reduction in LSA then constitutes an inductive step by which genes are represented by values on
a smaller set of abstract features (such as their functional classes), rather than their raw patterns of observed expression levels in the various samples, therefore allowing us to uncover the underlying genetic functional classification as expressed by the genes.

2.3 Related Work

There has been a recent increase in interest in the use of SVD and related approaches for analyzing microarray expression data. Most of the work has been focused on applying SVD to mathematically discover the underlying components in the microarray expression data that have corresponding biological significance. For example, using Principal Component Analysis (similar to SVD), Raychaudhuri et al. (2000) demonstrated that the mathematical components they discovered in the time series sporulation expression data corresponded to significant biological subprocesses. Holter et al. (2000) used SVD to uncover underlying patterns or so-called “characteristic modes” from gene expression data—they showed that the essential features of a given set of expression profiles are captured using just a small number of characteristic modes, suggesting the viability of dimension reduction process in LSA. In a similar work, Alter et al. (2000) analyzed yeast cell cycle data using SVD and showed that the components revealed by the mathematical analysis can also be assigned corresponding biological meanings. For example, they identified sinusoidal modes in the singular value decomposition of the expression data that corresponded to the various cell cycle modes.

In the application of SVD to the task of gene clustering, Wall et al. (2001) described a method for generating gene groups directly from the clustering cover the underlying components in the microarray expression data. Most of the work has been focused on applying SVD to mathematically discover the underlying components in the microarray expression data that have corresponding biological significance. For example, using Principal Component Analysis (similar to SVD), Raychaudhuri et al. (2000) demonstrated that the mathematical components they discovered in the time series sporulation expression data corresponded to significant biological subprocesses. Holter et al. (2000) used SVD to uncover underlying patterns or so-called “characteristic modes” from gene expression data—they showed that the essential features of a given set of expression profiles are captured using just a small number of characteristic modes, suggesting the viability of dimension reduction process in LSA. In a similar work, Alter et al. (2000) analyzed yeast cell cycle data using SVD and showed that the components revealed by the mathematical analysis can also be assigned corresponding biological meanings. For example, they identified sinusoidal modes in the singular value decomposition of the expression data that corresponded to the various cell cycle modes.

In the application of SVD to the task of gene clustering, Wall et al. (2001) described a method for generating gene groups directly from the eigenvalues. Their approach is different from the LSA approach we described in this paper—we do not perform clustering in the “eigenspace”. In a more recent work, Horn et al. (2003) presented an LSA approach that also uses SVD followed by dimensional reduction, but they have applied a quantum-clustering algorithm in the reduced SVD space for clustering of the genes. Here, we use LSA approach for semi-supervised whole-genome functional classification of genes. We apply singular value decomposition as a dimension-reducing step on gene expression data, followed by a similarity-based unsupervised clustering procedure in the truncated data matrix which we will describe in details in Section 3.2. Classification of genes is then conducted using majority voting on the known annotations in each of the resulting gene clusters. Using this semi-supervised LSA approach on microarray data, we were able to perform systematic functional classification on the partially-annotated yeast genome, classifying 1,753 genes into more than 200 distinct functional classes.

3 Materials and Methods

3.1 Data

For our study, we have applied our LSA classification method on the 6,221 genes in the Saccharomyces cerevisiae genome using the yeast gene expression data from Eisen’s Lab (Eisen et al. 1998) at http://rana.lbl.gov/EisenData.htm. For each gene, there were a total of 80 data points generated from spotted arrays using samples collected at various time points during sporulation (Chu, DeRisi, Eisen et al. 1998), diauxic shift (DeRisi, Iyer & Brown 1997), mitotic cell division cycle (Spellman, Sherlock, Zhang et al. 1998), and various other experimental conditions.

The microarray data are represented by a gene expression matrix $A$ of dimension $6221 \times 80$. Each value $a_{ij}$ in $A$ contains the relative expression of the $i$-th gene under the $j$-th assay (or condition). Each row in $A$ therefore represents the expression signature of a gene under the 80 experimental conditions (or assays), while the columns represent genome-wide expression profiles of a particular assay or condition.

For reference functional classification of the genes, we refer to the MIPS annotations given in the CYGD database (Mewes et al. 2002). This database has been compiled based on extensive knowledge in the literature, and it is available at http://mips.gsf.de/genre/proj/yeast.

3.2 Method

We describe our method as follows:

Step 1. Singular Value Decomposition.

1.1 Given a gene expression matrix $A$, perform singular value decomposition on it such that $A = U \cdot W \cdot V^T$.

1.2 If not already so, arrange the eigenvectors in the order of their relative significance as indicated by the diagonal scaling values in $W$.

Step 2. Dimension Reduction by Coverage.

2.1. Compute $r$, the number of eigenvectors to be retained based on the desired fraction $\theta$ of expression coverage by the eigenvectors. The expression coverage $C_i$ of the $i$-th eigenvector is defined as $C_i = w_i^2 / \sum_{k=1}^n w_k^2$, where $w_k$ is the $k$-th scaling value in $W$ (Alter, Brown & Botstein 2000). The number of eigenvectors to be retained is thus the smallest $r$ such that $\sum_{i=1}^r C_i \geq \theta$.

2.2 Create a new scaling matrix $W'$ by setting the $w_k = 0$ for $k = r + 1, \ldots, n$.

2.3 Reconstruct the reduced gene expression matrix $A'$ using $A' = U \cdot W' \cdot V^T$.

Step 3. Clustering by Vector Similarity.

3.1 Normalize each row in $A'$ such that $\sum_{j=1}^n \alpha_{ij}^2 = 1$.

3.2 For each normalized row $r_i$, generate its neighborhood set $F_i = \{k | r_i \cdot r_k \geq p_1\}$ for a pre-set value of $p_1$, $0 \leq p_1 \leq 1$.

3.3 Iteratively, merge any neighborhood sets with average inter-cluster similarity $\geq p_2$, where $p_2 \geq p_1$.

Step 4. Calling by Majority.

4.1 Each resulting $F_i$ is assigned a gene functional class label by majority voting on the annotated genes in the pre-set.

4.2 The function of the unannotated genes in each set is then predicted to be the corresponding functional class label.

In our current study, we used $\theta = 0.80, p_1 = 0.95, p_2 = 0.85$. Note that we have used the vector dot-product here as the measure of similarity rather than the proximity between vectors here—this is consistent to the standard application of LSA (Landauer & Dumais 1997). While other similarity measures such as the Pearson correlation—a common similarity measure for microarray data analysis—can also be used, our results showed that the LSA usage of vector dot-product as a similarity measure performed equally well (data not shown).
in the partially annotated Saccharomyces cerevisiae genome-wide functional annotation on the 6,221 genes. Using our LSA classification method, we perform functional classification on yeast gene expression data. Table 1: Classification of the top 30 major functional classes in MIPS using our LSA method for whole-genome functional annotation by using only up to GO level 2 as functional class labels.

4 Results

Using our LSA classification method, we perform genome-wide functional annotation on the 6,221 genes in the partially annotated Saccharomyces cerevisiae genome using the 80-experiment yeast gene expression data from Eisen’s Lab described in Section 3.1. As mentioned earlier, we refer to the CYGD database from MIPS the reference functional annotations. The CYGD database uses the standard Gene Ontology (GO) (Ashburner, Ball, Blake et al. 2000) for its functional annotation. Since GO is a hierarchical classification scheme, we normalize the genes’ functional annotation by using only up to GO level 2 as functional class labels.

Our reference set include MIPS-annotated yeast genes in non-trivial functional classes—i.e., functional classes with more than three genes. Unlike previous similar studies such as the study by Mateos et al., we have chosen in our analysis here to exclude genes with ambiguous functional assignments—namely, genes that belong to multiple functional classes—from our reference set. The inclusion of such genes in the classification process has been shown to corrode the results due to the so-called “Borges Effect” (Mateos et al. 2002). More on this will be discussed later in Section 5. Out of the 6,221 yeast genes studied in the microarray experiments, 4,095 contain MIPS functional annotation. After excluding genes that are assigned to multiple function classes, we have in our reference set 1,851 single-function genes covering 58 level-2 MIPS functional classes. In other words, we are using only 30% of the 6,221 yeast genes as a reference set to predict the functional classification of the other 4,370 unknown yeast genes.

In a related work by Mateos et al. (2002), they have defined functional classes with a true positive (or precision) rate ≥ 40% as “learnable”. Out of the 58 MIPS functional classes, we found that 40 of them—about 70% of the MIPS functional classes—are learnable using our LSA approach, with precision rates mostly in the range of 0.6 to 0.8, with an average of 0.7; see Figure 1 and Table 1, where we show the detailed classification results of the first 30 major MIPS functional classes. This represents a significant improvement in whole-genome gene functional classification. In the work by Mateos et al. (2002), they found that only ~10%—or 8 out of the 96 classes that they had looked at—of the MIPS functional classes are learnable with the supervised learning algorithm (multi-layer perceptrons) on the same set of yeast microarray data. They have attributed the cause of the poor learnability partly to the so-called “Borges Effect”, as they did not exclude genes that have multiple function class annotations from their training set as we have done for our reference set. We will show in Section 5 that the precision rates for our LSA whole-genome annotation approach is not affected by the inclusion of multi-function genes in our reference set.

In our LSA classification method, prediction of
Figure 1: Classification performance of 40 non-trivial learnable MIPS yeast gene functional class using our LSA approach.

Figure 2: Classification performance versus MIPS function class size.

5 Discussions

There are four main factors that influence the systematic learning of gene functional classes from DNA array expression data (Mateos et al., 2002): data noise, class size, class heterogeneity, and the internal structure of the catalog (or the so-called “Borges Effect”). In this section, we discuss how our LSA approach deals with each of these.

5.1 Data Noise

One of the major complications in analyzing high-throughput gene expression data is due to the noisy nature of the data. In our LSA approach, we use singular value decomposition as a de-noising mechanism. The dimension reduction step in our method allows us to normalize our data by filtering out the eigen components that correspond to additive or multiplicative experimental noise and background signals from irrelevant biological processes. These are usually represented as eigen components with low eigenvalues, and the decoupling of the eigen components in SVD ensures that they can be effectively filtered out from the data without eliminating any relevant information in the data.

To show that such data normalization by SVD can improve the further analysis of expression data, we applied our classification method without the SVD dimension reduction steps (namely, Steps 1 and 2 in Method) on the gene expression data using the same reference set. Instead of 38 learnable classes, there are only 16 learnable classes resulted, each with a very low recall rate—the mean recall rate is 0.14 instead of 0.40. This indicates that the SVD dimension reduction step clearly benefits the processing of expression data for functional classification by de-noising the expression data and enhancing the relevant signals for further analysis in our LSA approach.

5.2 Function Class Sizes

Another determinant of the learning rates of gene function classification, particularly by supervised machine learning algorithms such as neural networks, is the size of the function class. Mateos et al. (2002) showed that there is a clear trend for the true positive rate to increase with the class size—the more examples there are, the easier it is for a class to be learned. The larger class size also helps to offset the imbalance in the number of positive and negative examples with respect to each function class—a problem in multi-class whole-genome function classification.
5.3 Function Class Heterogeneity

Biologically, many of the function classes are expected to be heterogeneous, particularly for the larger classes. For example, genes in the class “assembly of protein complexes” are unlikely to be expressed in a coordinated fashion, as different complexes are clearly compiled under different conditions. The heterogeneity of the expression profiles of the different member genes can deteriorate the learning rates of classifiers, since there will be no clear-cut clusters in the expression profiles that correspond to these heterogeneous classes.

Just like other common classification methods, our LSA groupings can only capture genes that are homogeneous in expression within each group. However, our LSA approach handles class heterogeneity by allowing for multiple groups to be called under the same function label, as long as there is a majority of reference genes in each group that have the function in question. For example, the “assembly of protein complexes” class is made up of six eigen groups, each with a majority of such reference genes. In fact, the number of LSA subgroups for the various MIPS function classes can range from 2 to 85.

5.4 Borges Effect

While each function class can consist multiple subgroups, it is also possible for a nontrivial number of class members to intersect with other classes. Such cross-linking internal structure of the catalog of functional classes can confuse most machine learning procedures in distinguishing positive from negative examples.

From the biological point of view, most cellular processes are clearly not stand-alone as they are expected to interact with other processes. As such, it is not realistic to expect that the MIPS functional classes are equivalence classes. Mateos et al. (2002) have concluded that this inherent limitation of functional classification systems the “Borges effect”. They have shown that the Borges effect can be costly to the performance of supervised machine learning classifiers such as multi-layer perceptrons. This was the reason why we have chosen the 40 non-overlapping MIPS classes as our reference set.

To investigate the extent by which the Borges effect can affect our LSA approach in whole-genome functional annotation, we include genes that have multiple functional annotation in the reference set on the same 58 MIPS classes as before. The new reference set contains 8,674 functional annotation for 4,095 genes—our previous reference set contains only 1,851 single-annotation genes. For comparison, we show in Figure 4 the classification performance of our LSA approach for the same 40 MIPS classes using this larger reference set of non-equivalence classes. Comparing with the results shown in Figure 1, we can see that the recall rates of our LSA approach was affected by the Borges effect, with a mean decrease in performance of -0.25. However, the results also suggested that the precision rates of our LSA approach is fairly robust against the Borges effect with a mean decrease in performance of only -0.06. For future work, we will investigate ways to improve the recall rates against the Borges effect.

6 Conclusions

The recent advances in microarray technology has certainly revolutionized the way molecular biologists study the functional relationships among genes. While we are now able to monitor gene expression at the genomic scale using microarray technology, there is still some gap toward whole-genome functional annotation of genes using the gene expression data. Recent work by Brown et al. (2000) and Mateos et al. (2002) have shown that while it is possible to use machine learning algorithms to systematically learn the gene functional classes of some number of the genes in the genome, the number of genes that can be annotated this way is still not yet at the genomic scale. For example, Brown et al. focused on learning only 5 functional classes (using sophisticated support vector machines), while Mateos et al. concluded that only ∼10% of the functional classes are learnable by their neural networks.

In this paper, we have proposed to use an alternative semi-supervised approach based on latent semantic analysis (LSA) to the problem of whole-genome gene functional classification. Our approach is a 4-step procedure: singular value decomposition, dimension reduction by coverage, clustering by similarity, followed by assignment by majority, as outlined in Section 3.2. Our unsupervised pre-classification clustering is less susceptible to the many difficulties in whole-genome gene functional classification. It is able to handle the imbalance of training examples by considering the groupings of all classes globally without the a priori partitioning by the positive examples. It employs the use of dimension reduction by SVD to help de-noise the data and enable the clustering algorithm to focus only on significant components present in the expression data. For heterogeneous function classes, it allows multiple subgroups to be called as the same functional class. Our LSA approach is useful
for systematic whole-genome functional classification of genes, as shown by the promising results in the classification of more than 1,700 genes in the partially-annotated yeast genome into 40 distinct MIPS functional classes.

References


