Biologically-Interpretable Disease Classification Based on Gene Expression Data

Gregory Grothaus

Thesis submitted to the Faculty of the Virginia Polytechnic Institute and State University in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Science and Applications

T. M. Murali, Chair
Vicky Choi, Member
Alexey Onufriev, Member

May 13th, 2005
Blacksburg, Virginia

Keywords: Classification, Microarrays, Gene Expression, Biclustering

Biologically-Interpretable Disease Classification Based on Gene Expression Data

Gregory Grothaus

(ABSTRACT)

Classification of tissues and diseases based on gene expression data is a powerful application of DNA microarrays. Many popular classifiers like support vector machines, nearest-neighbour methods, and boosting have been applied successfully to this problem. However, it is difficult to determine from these classifiers which genes are responsible for the distinctions between the diseases. We propose a novel framework for classification of gene expression data based on notion of condition-specific clusters of co-expressed genes called xMotifs. Our xMotif-based classifier is biologically interpretable: we show how we can detect relationships between xMotifs and gene functional annotations. Our classifier achieves high-accuracy on leave-one-out cross-validation on both two-class and multi-class data. Our technique has the potential to be the method of choice for researchers interested in disease and tissue classification.

Acknowledgments

I would first like to acknowledge and thank my advisor, T. M. Murali, without whom this thesis would not be possible.
I would also like to thank Jonathan Myers for assisting me with some of the work on functional enrichment and writing the original version of libEnrichment which xMotif uses to assess biological significance.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Disease classification in the 20th century was based primarily on phenotypical symptoms of patients. Before the advent of the genomic era and the human genome project many of the more difficult diseases to treat, such as cancer, could only be differentiated on the basis of simple phenotypical symptoms, whereas the critical differences were at a subcellular level requiring different treatments. New approaches are now being considered which make use of the very same technologies that helped reveal the true nature of cancer in the first place: the full sequence of the human genome and proliferation of associated genetic measuring technologies such as DNA Microarrays, protein gels, and high-throughput gene knockout techniques like RNA interference. These new technologies lead to new understanding of disease that holds the promise of individualized medicine, customized to a patient's specific malady.

1.1 DNA Microarrays

The central dogma of molecular biology states that DNA contains the complete information used by an organism to define the organization and function of that organism. The genetic information contained by DNA is copied into molecules of mRNA through a process called transcription. mRNA molecules are translated into proteins within individual cells. Proteins in turn are directly responsible for cell organization and function.

DNA sequences are identical in any two cells within the same individual[1] and there are only minute differences between the DNA sequences of different individuals of the same

[1]This is true assumming no cell mutations have occurred as is the case with cancer.

organism. Despite this, the behavior of the two cells or individuals can be significantly different. The genes expressed in a cell, the levels at which they are expressed, the proteins formed, and the activation of the proteins can change several times a second within a single cell in response to changing conditions outside the cell. Changes in gene expression level are believed to account for a large portion of this difference in behavior between genetically similar or identical cells. Gene expression level is the degree to which a gene is being transcribed into mRNA and subsequently translated into protein.

Measuring the expression level of various genes within a cell is an important step in understanding the function of a cell. DNA microarrays, hereafter simply referred to as microarrays, are a relatively recent technology that can be used to measure gene expression level by measuring the amount of mRNA from different genes in a cell [SSDB95, LDB+96]. This technology offers the capability to measure the gene expression of the entire genome on a single chip. The ability to measure gene expression in such a high-throughput manner allows a researcher to get a view of "whole picture" genome interaction at a relatively low cost.

A microarray is a small surface, generally glass but sometimes nylon, which is spotted in an organized order with a large number of probe DNA sequences. Microarrays test for the amount of transcribed copies of DNA sequences (mRNA) in a solution. It is believed that the relative abundance of mRNA corresponds approximately to the level of protein produced in the cell and can thus be used as a rough surrogate for measuring protein activity. The relative abundance of each mRNA molecule is assessed against a control by first reverse transcribing the two mRNA samples into cDNA while labelling each sample using a different fluorescent dye. Next, the cDNA samples are both applied to the microarray and allowed to hybridize with the arrayed probe DNA sequences. Probabilistically, hybridization should occur in roughly the same proportion as the proportion of mRNA between the two mRNA samples. Finally, the microarray is photographed with a scanner that can measure the combined fluorescence from each of the two dyes. The resulting photograph is processed by software which makes calls about the relative amount of each dye for every probe DNA sequence. Thousands of probe DNA sequences can be put onto a single microarray, allowing for high throughput estimation of gene expression.

There are many sources for error and assumptions made in the DNA microarray pipeline as described. First, it is assumed that mRNA abundance correlates directly with levels of protein abundance. Next, since existing technology cannot measure the expression

levels of the genes in a single cell, the mRNA levels measured are actually sampled from a large group of cells and the assumption is made that the variance of expression levels in the individual cells is small. Next, the hybridization step is not without experimental error since hybridization is inherently a probabilistic process. If a gene is expressed at a low level, small variations in the expression level can lead to significant changes in the measured levels. The scanning technology is not error-free either. For example, certain regions of the scanner may have slight differences in light sensitivity. Finally, the image processing is also an imperfect approximation to the real fluorescence ratios. As a result of all of these errors, any analysis of microarray measurements needs to account for the presence of significant amounts of noise in the data.

1.2 Disease Classification

In 2000, Druker and Lydon [DL00] discovered that the chemical compound imatiniv mesylate inhibited the protein kinase Bcr-Abl. This kinase is the protein product of a gene created in Chronic Myeloid Leukemia. Clinical trials of imatiniv mesylate, renamed Gleevec after FDA Approval in 2001, revealed that the compound was relatively non-toxic to normal cells, while simultaneously causing apoptosis in Myeloid Leukemia cells. While relatively high success rates were reported for Myeloid Leukemia, Lymphoblastic Leukemia was not affected by imatiniv mesylate, emphasizing the need to differentiate between these clinically similar diseases. The success of imatiniv mesylate also indicated the need to understand not only the phenotypical distinctions between disease, but also the underlying molecular processes that could be used to identify potential drug targets.

One of the primary applications of microarrays is using measured gene expression profiles to distinguish between different types of diseases at a subcellular level. After training on a corpus of tissue samples with known disease classifications, we desire to predict the disease classification of new samples. Many machine learning algorithms such as k-nearest neighbors (k-NN) [PTG+02], weighted voting [GST+99], decision trees [ZYS03, ZYSX01], boosting [BDBF+00], and support vector machines (SVMs) [BDBF+00,PTG+02,FCD+00] have been applied to this problem with success.

Traditional machine learning algorithms have two drawbacks when applied to microarray data. First, these algorithms operate best when there are more objects than features. In microarray experiments, there are generally on the order of 10,000 genes measured on

one microarray for each sample. A microarray experiment is very expensive, thus most studies involving microarrays only use at most a few hundred microarray samples. The second problem is that it is difficult to interpret the results of traditional machine learning classifiers, especially in terms of determining the genes that cause the primary distinctions between the classes in the data. The k-NN classifier simply computes distances between samples in the space spanned by all the genes in the data. SVMs, boosting, and weighted voting compute a separating function between the between the samples belonging to the two classes. As Ramaswamy et al. report [RTR$^+$01], the SVM that has the best cross-validation performance on their multi-cancer data set assigns small non-zero coefficients to all the genes in the data set. As a result, it is difficult to ascertain from these coefficients which genes are most responsible for causing the distinctions between the samples in the data.

1.3 Formal Classification Problem

We are given a set of samples S; each sample $s \in S$ is associated with a class label C_s. We use C to denote the set $\{C_s, s \in S\}$ of class labels of all the samples in S. Each sample $s \in S$ is a point in $\mathbf{R}^d$; the coordinate of these points are the expression values of the d genes. A classifier is a function h:$\mathbf{R}^d \rightarrow C$ that when given the training set S, the class labels $\{C_s, s \in S\}$, and a new sample query vector q in $\mathbf{R}^d$ computes a predicted class for q, $h(q)$. A good classifier h is one where the predicted class $h(q)$ is frequently the same as the correct class for q.

In order to validate a classifier, a method called leave-one-out cross-validation (LOOCV) is used. LOOCV is frequently used when the number of available data points is small. LOOCV works by removing a single sample s from the dataset S of all samples and training the classifier on the remaining dataset. The trained classifier is then used to predict the class of s. This process is repeated for all values of $s \in S$ and the accuracy is simply the percentage of samples whose class was correctly identified. This allows for both training and testing using the largest dataset available. This is especially important when the number of samples is initially small as is the case in microarray datasets.

1.4 Contributions

This thesis proposes a novel technique to construct biologically-interpretable classifiers using the genes in microarrays as features. The building block of our classifiers are *gene expression motifs* or *xMotifs* [MK03], small subsets of the genes which indicate the difference between distinct sample classifications. This helps to overcome the problem of irrelevant attributes in feature rich datasets such as microarrays. Unlike a traditional "black box" classifier, this also allows for groups of genes to be directly associated with a specific disease classification. We show how these groups of genes can be analyzed for common biological features, such as participation in a common cellular pathway or a mutation in a specific chromosomal region. The result is a description of specific diseases in terms of their underlying biological mechanisms. We hope that by interpreting the underlying biological distinctions between cancers, our contribution may lead to specific drug targets for additional investigation.

Chapter 2

Survey of Related Research

The chapter summarizes previously published classification techniques for biological problems which rely on microarrays as their dataset. Some of these classifiers are designed around a specific two-class classification problem, thus limiting their usefulness in a broader multi-class experiment. Others attempt to build a classifier that can recognize any number of classifications, but fail in interpretability.

2.1 Neighborhood Analysis

Golub *et al.* [GST+99] was the first published research to address the microarray classification problem by attempting to classify acute leukemias. Motivated by the success of the compound imatiniv mesylate on treating specific types of leukemias [DL00], the dataset in this work included microarrays collected from patients with either acute lymphoblastic leukemia (ALL) or acute myeloid leukemia (AML).

To construct a classifier, a list of differential genes was constructed based on how well each gene individually differentiated between the two classes using a method called "neighborhood analysis". An ideal gene vector would have all high expression in one cancer and low expression in another cancer or vice versa. Each gene's expression vector was compared with this ideal vector and the genes whose vectors were closest to the ideal vector were said to be neighbors of that ideal vector. Only the 50 closest genes were used in the classifier. To classify a new sample, genes were compared against the genes in the classifier and each gene submitted a weighted vote with the magnitude of each vote dependent on the expression level of the new sample and the degree of that gene's correlation with the

ideal gene vector during training. The resulting sum of votes gave a prediction and a prediction strength. By allowing for 7 "no-calls" with low prediction strength, leave-one-out cross-validation (LOOCV) tests showed this to work on the ALL/AML dataset with 100% accuracy.

This first technique for solving the microarray based disease classification problem mainly served to illustrate the potential of this idea. Further studies using the same data were able to achieve the same accuracy without "no calls". This technique owes it's success to the fact that there are individual genes which have a very good ability to independently distinguish between these classes. One advantage of this approach is that it produces a reduced set of discriminating genes that can be considered biologically. Golub *et al.* [GST+99] validate this analysis by looking at some of the individual genes in this set, finding genes with a known role in cancer.

2.2 Unsupervised Clustering

Another technique used for evaluating the potential for disease classification revolves around using one of many different techniques that organize the known samples into groups or clusters of high similarity based on the expression levels of their genes. These algorithms do not use the disease classification of the samples as part of the algorithm's input, and are thus prone to detecting distinctions in the samples studied that are unrelated to the disease in question. These techniques do improve upon the "neighborhood analysis" approach by simultaneously considering multiple genes.

Alon *et al.* [ABN+99] uses a hierarchical type of unsupervised clustering. Each sample s is represented as a vector V_s, whose components are the expression levels of the genes measured for that sample. Two clusters are formed using a deterministic-annealing algorithm that separates the samples into two similar groups based on their euclidean distance. Each of these resulting clusters is then repeatedly separated into two further subgroups until each subgroup contains only one sample, forming a binary tree. For classification, only the centroids of the first two clusters are used.

The dataset represents a collection of 62 colon biopsy samples of colon epithelial cells collected from colon cancer patients. The classes represent whether the sample was taken from healthy tissue or directly from tumor tissue, but all samples were taken from patients

with colon cancer. The number of genes in this dataset is over 6,500. While LOOCV was not performed, the tumor samples do indeed self-organize into two major clusters that show high specificity to a tumor/normal distinction. Only 7 of 62 or 11% of the samples were misclassified in this manner.

It is reported that many of the genes with the most statistically significant difference between tumor and normal are muscle genes. This observation indicates that perhaps the distinction observed was related to tissue composition instead of cancerous state. Most of the tumor cells are sampled from epithelial tissue, whereas normal cells are include a mixture of tissue types. In order to further validate their approach, Alon *et al.* [ABN$^+$99] remove the 1,500 genes with the most significant differences between tumor and normal tissues, in order to avoid the tissue-specific bias. They find that the clustering technique continues to perform well even using only genes that do not individually separate the two tissue types well. This indicates that there are patterns of expression which can be used for classification that are not related to individual genes, the basis for "neighborhood analysis".

Alizadeh *et al.* [AED$^+$00] attempts to classify B-cell malignancies and normal tissue, of which there are nine distinct classes. The dataset used in this study includes 96 specialized microarrays of these nine classes across 17,856 genes. The classification problem is then a multi-class classification. The clustering method used is hierarchical clustering [ESBB98] which repeatedly finds the pairs of genes with the smallest euclidean distance and merges them. After each pair is found, an average vector representing that pair is generated and used for further pairing. This approach, like that of Alon *et al.* [ABN$^+$99], generates a binary tree of the samples, consisting of clusters of samples that are selected without respect to the class labels.

The entire set of genes was used for the hierarchical clustering. Examining large trees in the resulting clustering dendogram shows automatic groupings of the samples into their respective classes, although the grouping is not perfect with 7 of the 96 samples (7%) clearly out of place. The results do however show that a hierarchical approach to classification is successful in multi-class data.

Tomida *et al.* [TKY$^+$04] extend the hierarchical clustering method to use only a subset of the genes available on the microarray. In this analysis they are attempting to generate a classifier that will predict patient survival for non-small-cell lung cancer (NSCLC) patients. They select the top 100 genes whose signal-to-noise ratio most clearly distin-

guished favorable from fatal cases of NSCLC. A hierarchical clustering is performed using only these 100 genes. Two major clusters were formed representing favorable and fatal outcome. Out of the 50 patients, 34 were predicted correctly using this method (68% accuracy).

Hierarchical clustering fails to take advantage of the fact that not all genes are equal in their ability to distinguish between classes, but it does allow for a computationally simple analysis of any number of class distinctions. Tomida *et al.* [TKY+04] shows how unsupervised clustering can be modified to only examine genes which individually distinguish cancer types well, but the results from Alon et al [ABN+99] indicate that there are subtle but predictive patterns in combinations of multiple genes that individually do not predict well.

2.3 Supervised Machine Learning

Before the advent of the classification problem for diseases, a number of supervised machine learning techniques had been developed for other problems. Supervised machine learning techniques are characterized by the fact that the input to the classifier learning phase includes both the sample class labels and the samples. This is in contrast to the unsupervised clustering methods in Section 2.2 which do not take advantage of the class labels during clustering. Many researchers naturally applied supervised machine learning techniques to disease classification.

One of the simplest machine learning classifiers is the nearest neighbor classifier [DH73]. The nearest neighbor classifier works as follows: For each test sample s, find the most similar example in the training set S and predict that s has the same label as that example. The definition of similarity is specific to the problem at hand, for microarray data, either the pearson correlation or euclidean distance is usually used. Often in microarray analysis a nearest neighbor classifier is trained only on the subset of genes whose signal-to-noise ratio most clearly distinguishes the classes of interest.

Other more direct methods denoted as *large margin classifiers* attempt to learn a decision surface that separates two different classes of samples. These decision surfaces are surfaces in higher dimensional R^d space corresponding to the number of attributes of the samples being classified, as well as additional dimensions for non-linear transformations of these attributes. Large margin classifiers suffer from a problem inherent in microarray

datasets. Namely, the number of attributes (genes) is orders of magnitude larger than the number of training samples. This means that there are many decision surfaces that can be used to separate the sample classes, not all of which are relevant. Generally, large margin classifiers work better when the number of samples is large in comparison to the number of attributes.

The most well studied large margin classifiers are support vector machines (SVM), developed by Cortes & Vapnik [CV95]. Support vector machines attempt to find a hyperplane in R^d space such that the hyperplane splits the input space into two spaces corresponding to two different sample classifications. Ambiguities are resolved by maximizing the distance from the closest example samples to the hyperplane. The hyperplane is derived by solving a quadratic programming optimization problem that can by solved efficiently by several existing algorithms. Support vector machines can be extended to non-linear classification by applying kernel transformation functions which can be thought of as adding extra dimensions of attributes to the samples that represent non-linear transformed space. A number of different kernel transformations exist. Commonly used ones are polynomial transformations of different degree and radial-basis transformations.

Boosting is a method for constructing a good classifier by repeated calls to multiple "weak learners". A weak learner is any classification algorithm that has an accuracy better than random chance. Boosting occurs in stages where a weak learner is repeatedly added to a learning function with some weight proportional to the accuracy of the weak learner. The data is reweighted based on the accuracy of the learning function for classifying specific sample points. The sample points that are incorrectly classified by the learning algorithm get *boosted* in importance repeatedly until they are correctly classified.

In Ben-Dor *et al.* [BDBF$^+$00], multiple classifiers are compared using three different data sets and across varying sizes of gene inputs. The first two data sets are the leukemia [GST$^+$99] and colon cancer [ABN$^+$99] datasets described in Section 2.2. The third is a data set of 32 Ovarian cancer samples, 15 of which are cancerous and the other 17 are normal tissue. The microarray contains approximately 100,000 clones from ovarian clone libraries. These three datasets are each classified by six different classifiers with varying numbers of genes.

The first classifier is based on a binary hierarchical approach such as those in Section 2.2. The other classifiers are all traditional supervised classifiers: nearest neighbor using a pearson correlation, support vector machines, and boosting. There are two dif-

ferent kernels for the support vector machines (a linear and quadratic kernel) and two different iteration counts for Boosting yielding a total of six classifiers. The boosting weak learners are simple gene expression thresholds for individual genes. That is, a weak learner would classify a sample as one class if the expression level for a specific gene is below some value, and another class if the expression level is above the same value.

Finally, a gene selection score is used to select genes which individually distinguish between the classes. Differing numbers of genes are then used for each of the classifier types. The results of this research show that in all three datasets, using only a subset of all the genes improves classification performance, but the size of the ideal subset depends on which classifier used and which dataset. In the colon dataset, the SVM with a quadratic kernel performs the best, with 84% accuracy. In the ovarian dataset, boosting and both SVM's achieve 100% accuracy. In the leukemia dataset, nearest neighbor, boosting, and both SVM's achieve 99% accuracy.

This work highlights the major problem with traditional classification for microarrays, namely that not all genes are useful for performing classification, and in fact some genes are detrimental to the results. By reducing the gene space, improved accuracy is shown to increase. The other issue with using SVMs or Boosting, is the natural two-class limitation imposed by these proceedures.

Yeang *et al.* [YRT+01] avoids the two-class limitation imposed by SVMs by testing various combination strategies. They first prepare a dataset of 109 samples from 14 tumor classes on a microarray with 16,063 known genes and expressed sequence tags. They compare a weighted voting algoritm [GST+99], a nearest-neighbors algorithm, and an SVM algorithm on this dataset using two different binary classifier combination strategies. The first strategy, one-vs-all (OVA), builds k (the number of classes) binary classifiers which distinguish one class from all the other classes grouped together. The classifier that has the highest confidence score on a new sample is used to make the classification. The second strategy, all-pairs (AP), builds $\frac{k(k-1)}{2}$ classifiers which distinguish each pair of classes. For each class, there are $k-1$ classifiers that distinguish it from other classes. For a new sample, the confidence of those $k-1$ classifiers are summed, and the class with the greatest overall confidence is the winning class. The OVA SVM classifier outperformed all other methods achieving 81% accuracy.

Ramaswamy *et al.* [RTR+01] extend the dataset in Yeang *et al.* [YRT+01] to include 308 samples from the same 14 tumor classes. They uses a support vector machine with

a one-vs-all heuristic which was found to be optimal in the prior study. This follow up study examines the effect of using a smaller number of genes in the classifier. The least predictive genes are iteratively removed by selecting those genes whose weight in a trained support vector machine is the smallest.

The results of LOOCV validates the use of all the genes in a support vector machine classifier (78% accuracy). The support vector machine only performs worse as the number of genes is decreased. It can also be noted from the results that the support vector machine's accuracy does not increase much beyond 100 genes (less than 1% of the total genes). Classifiers other than SVMs are shown to perform better on a gene set reduced by the SVM classifier. These results are predictable, as genes which have little weight in a support vector machine are already not contributing much to the result of the support vector machine. Removing them should not significantly change the results for an SVM, but for other classifiers, this pruning can improve the results.

2.4 Biclustering

While unsupervised hierarchical clustering can be applied to both genes and samples within the same dataset, the clusters are nevertheless calculated using the assumption that related genes behave similarly across all conditions and related conditions have similar gene expressions across all genes. Biclustering simultaneously clusters both genes and conditions avoiding this assumption. A bicluster is a subset of genes and a subset of samples with a high similarity defined according to the specific technique used. The literature contains various definitions of a bicluster that vary according to what definition of similarity is used [CC00, MK03, KSG04]. Different problems can be solved by selecting different definitions of similarity.

Gene expression biclustering offers a number of advantages over other classification techniques mentioned previously. Tomida *et al.* [TKY+04] and Ben-Dor *et al.* [BDBF+00] show that classification of disease samples can be improved by selecting a smaller subset of genes before performing the classification. Biclustering allows for multiple subsets of genes to be selected that may play independent roles in the classification of disease. The results of Alon *et al.* [ABN+99] indicate that genes which individually do not have much classification power can still contain patterns which differentiate disease. Biclusters can also lead to very simple biological interpretation as they identify the relationship of small

subsets of genes to small subsets of samples.

Cheng and Church [CC00] were the first to study biclustering of gene expression values. The measure of a bicluster in this work is "mean squared residue". The mean squared residue is calculated as the variance of the set of all expression values in the bicluster, plus the mean row and column variance. The goal is to find large biclusters that have low mean squared residue, which is shown to be an NP-Hard problem.

Cheng and Church [CC00] use a greedy algorithm to find large biclusters with a variance below a specific threshold. The algorithm begins with the initial state of the entire matrix of expression values. Rows and columns are removed iteratively such that the resulting matrix has lower mean squared residue. This approach is guaranteed to converge on a single bicluster, and this bicluster will have low mean squared residue. When a bicluster is found, it is reported, and all of the values in the original expression matrix corresponding to this bicluster are set to new random values. This process is repeated until some number of user defined biclusters have been extracted.

This research highlights one of the major difficulties with bicluster discovery in gene expression data. In the general case, bicluster discovery is an NP-Hard problem solvable only in time $O(2^n 2^m)$ for an expression data matrix with n rows and m columns. The randomization of elements in the original matrix creates two new problems. First, overlapping biclusters will be difficult to find as the first bicluster found will randomize elements in the other bicluster. The second bicluster may no longer meet the mean squared residue requirements. Second, new biclusters may possibly be formed when changing the matrix that are not valid in the original data. We expect that after many iterations of the algorithm, the matrix would tend to have more random values than real data, and the quality of the biclusters discovered would decline.

In the research by Califano *et al.* [CST00], a biclustering algorithm called SPLASH is used. Initially, the SPLASH algorithm normalizes each gene vector using a non-linear transformation that fits the expression values onto a uniform distribution over the interval [0,1]. A bicluster is defined as a set of genes and samples such that the range of expression values for every gene is less than some parameter δ.

To build a classifier from these biclusters, first only biclusters corresponding to a particular class are built. For each gene, SPLASH builds probability densities for the expression values of samples in the bicluster and those in the alternative class. SPLASH assigns a score to each bicluster based on the difference between the two probability densities.

SPLASH selects the subset of the biclusters with the best score such that every sample in each class is found within at least one bicluster. Finally a winner-takes-all method chooses the bicluster best matching a new sample using the probability densities of the bicluster.

Analysis is done on a dataset of 60 human cancer cell lines with a three different classifications. The first is a melanoma vs. healthy subset of 21 samples. Second, analysis is performed on a subset of 17 cells with/without a mutation in p53. The last classification is between 3 groups of cell lines, differentiated by the effect of the anti-growth drug Chlorambucil GI_{50}. The results show that the biclustering classifier performed on par with that of a support vector machine.

In Murali & Kasif [MK03], the idea of a conserved gene expression motif or *xMotif* is presented. This is a bicluster whereby the expression values for the genes in the bicluster are simultaneously conserved in the samples in the bicluster. A gene's expression level is conserved if the gene is expressed at approximately the same level in all the samples. A few other requirements are added to the algorithm, namely that the bicluster must contain at least some number of samples and that the genes not in the bicluster must not be conserved in more than some fraction of the samples in the bicluster. The state of a gene is a discrete value given to an expression value in a preprocessing step, which divides a gene's expression range into a number of smaller subranges.

Instead of a deterministic search for xMotifs, this work takes random sets of samples called a discriminant and attempts to add both genes and samples to that discriminant to form an xMotif of maximal size. This process is repeated until a satisfactory set of xMotifs have been discovered. This approach is proven to find the largest xMotif early with high probability.

This algorithm is applied to the leukemia dataset [GST$^+$99] in an unsupervised fashion. The algorithm finds four xMotifs that are almost exclusively ALL patients and one xMotif that is almost exclusively AML patients. This illustrates that there are indeed some strong xMotif patterns that can be used for classification within this dataset.

Tanay, Sharan, & Shamir [TSS02] use biclustering to discover groups of biologically interesting genes instead of solving a classification problem. They first simplify the gene-sample array by changing each expression value from a real-valued expression value to either up, down, or normally regulated. They consider only the up and down regulated expression values. This simplification is represented as a bipartite graph where the nodes in the graph are genes and samples, and the edges are the up and down expression values.

In this representation, biclusters are large bicliques in the bipartite graph, and the problem of finding biclusters is reduced to finding bicliques, an NP-Complete problem. To reduce the complexity of this problem, only genes with a bounded number of edges are used.

This approach was analyzed for it's ability to functionally annotate genes with an unknown function. By selecting biclusters which had a large proportion of genes annotated with a particular Gene Ontology annotation [Con01], the other genes in the same bicluster could be annotated with the same Gene Ontology Annotation. Using cross-validation, this approach successfully labeled 81.5% genes.

Chapter 3

Methods

We propose a novel framework for classification of gene expression microarray data that allows for natural multiclass classification and simple biological interpretability. Our classifier is based on the notion of condition-specific clusters of co-expressed genes within microarray data called xMotifs. As in earlier work [MK03], an xMotif is a subset D of genes and a subset C of samples with the property that each gene in D is expressed to a similar extent in each sample in C; we provide a precise definition of "similar extent" in Section 3.3. Our classifier consists of a set of xMotifs for each class in the data. Given a new sample, we match how well the sample matches the set of xMotifs for each class and predict the class of the new sample based on how well the xMotifs match each class. Section 3.7 describes the details of how an xMotif is scored against a sample.

Our approach has several desirable features and potential biological advantages:

1. By representing each class as a set of xMotifs, we build a single classifier for multiclass data, as opposed to other classification techniques that resort to constructing multiple one-versus-all or pairwise classifiers.

2. Since an xMotif represents condition-specific co-expression, finding multiple xMotifs within a class could indicate and potentially lead to the discovery of new sub-classes of conditions and diseases.

3. Since our classifier represents each class by a set of xMotifs, comparing and contrasting the xMotifs for different classes can identify genes that are present in xMotifs for multiple classes but are expressed differently in different classes. Such genes may highlight similarities and differences in the gene expression programmes of distinct

diseases.

4. The statistical test that we use to include a gene in a bicluster only depends on the ranks of the expression values of the gene. Thus, our method is less likely to be confounded by noise in the data than other techniques that explicitly use the gene expression values themselves.

5. We can compute the functional annotations enriched in the set of genes of a bicluster. Such functional enrichments provide additional biological interpretations of an xMotif.

3.1 Definitions and Notation

The problems we are referring to in this chapter all involve calculations on a single dataset. For the sake of brevity, all references to genes and samples can be assumed to be genes and samples in the same dataset $\mathcal{D}$. We define a dataset $\mathcal{D}$ as a set S of samples, a set G of genes, and a expression value $E_{s,g}$ for every pair (s, g) such that $s \in S$ and $g \in G$. The symbols s and g, where used, denote a specific sample and gene respectively. In addition, we denote the set of gene expression values corresponding to sample s as $E_{s,*}$ and the set of gene expression values for gene g as $E_{*,g}$.

For a specific sample $s \in S$, we denote the class label of s as C_s. We let C be the set of all distinct classes of samples in dataset $\mathcal{D}$. We denote the number of distinct classes as $|C|$. A specific class is denoted by c, and the number of samples labelled with c is denoted by $|c|$.

We denote an xMotif X as a set $X_S \subset S$ of samples, a set $X_G \subset G$ of genes, and the corresponding expression values $E_{s,g}$ for every pair (s, g) such that $s \in X_S$ and $g \in X_G$. Additional properties of an xMotif are described later using this terminology. The score of an xMotif X, defined later, is denoted by $\textsc{score}(X)$.

We use the standard statistical notation of α to denote a significance level which is distinct from the notation a used as a sample count in Section 3.3.

3.2 Data Normalization

For any individual gene g, we can consider the gene expression values that this gene can take on as a continuous random variable whose distribution depends on many different

factors, both biological and experimental. Some genes have a larger or smaller variance in expression than other genes, and thus what would be a large expression change for one gene would be a small expression change for another gene. Different DNA microarray platforms, different experimental protocols, and different sample populations also all have an effect on the distribution of gene expression measurements.

In order to take into account different distributions between genes, we normalize each gene expression measurement $E_{s,g}$ for a specific sample s and gene g to a value that is uniformly distributed in the range [0,1] corresponding to that expression value's rank in the complete set $E_{*,g}$ of expression values for that gene. For an expression value $E_{s,g}$ ranked i out of the $|S|$ expression values in $E_{*,g}$ sorted in non-decreasing order, the transformed expression value of $E_{s,g}$ is:

$$t(E_{s,g}) = \frac{i-1}{|S|-1}$$

This formula gives transformed expression values in the range [0,1] and will assure that there are values at both extremes of the range. As the number of expression values $|S|$ approaches infinity, $t(E_{s,g})$ better approximates the cumulative probability of finding an expression value less than or equal to $E_{s,g}$ in the observed distribution of gene g. This normalization has the additional property that a subset of expression values for gene g which follow a different distribution from the that of $E_{*,g}$ will not be uniformly distributed in the transformed space.

3.3 Conservation of Expression

In order to define an xMotif, we must first define what it means for a gene to be expressed to a "similar extent" in a set of samples. Let $n = |S|$. Given a set $S' \subseteq S$ of $k = |S'|$ samples and a gene g, let $I_{g,S'}$ be the interval spanned by the expression values of gene g in the samples in S'. Note that the expression values of gene gene in other samples in $S \setminus C$ may also lie inside the interval $I_{g,S'}$. Let a be the number of samples in S whose expression values lie inside $I_{g,S'}$; $a \geq k$ since the expression values for g in all of the samples in S' lie inside $I_{g,S'}$.

Let $P_g(n,k,a)$ be the probability that when we select k samples uniformly at random from n samples, the number of samples with expression values in $I_{g,S'}$ is at most a. We say that a gene g is *conserved with significance* α in the samples in C if $P_g(n,k,a) \leq \alpha$.

We derive $P_g(n,k,a)$ as follows: First observe that the number of ways of selecting k

samples from n samples is $\binom{n}{k}$. Next, restrict the problem to finding the number of ways of choosing k samples from n samples that span exactly b total samples in n. There are $n-b+1$ ways of fixing the smallest sample in k which also forces the choice for the largest sample in k if there are exactly b elements between the smallest and largest sample in k. Finally, there are $\binom{b-2}{k-2}$ ways of selecting the other $k-2$ samples. Extending this for all valid values of b where $k \leq b \leq a$, we get the following calculation of $P_g(n,k,a)$:

$$\frac{\sum_{b=k}^{a}(n-b+1)\binom{b-2}{k-2}}{\binom{n}{k}} = \frac{(\frac{nk}{a}-k+1)\binom{a}{k}}{\binom{n}{k}}$$

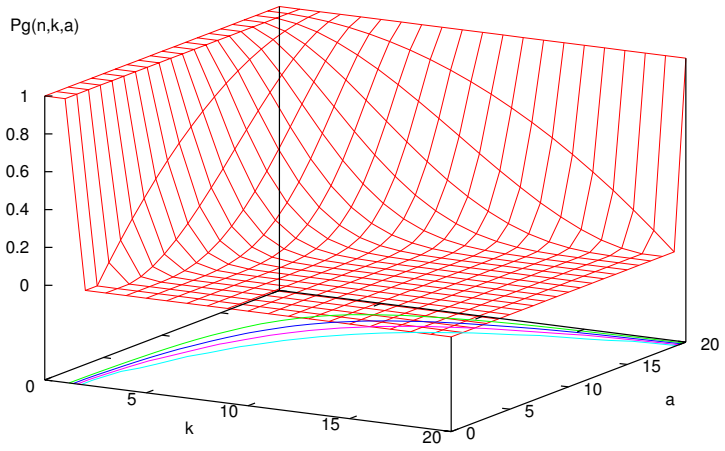


Figure 3.1: Plot of $P_g(n,k,a)$ for $n = 20$

Figure 3.1 is a plot of $P_g(n,k,a)$ for $n = 20$. The curves on the $k-a$ plane represent projections of the different $P_g(n,k,a)$ surface for constant values of $P_g(n,k,a)$.

3.4 Expression Motif

We now formally define an xMotif using the notion of conservation. Given a set of genes G whose expression levels are measured across a set S of samples and a threshold α, a *gene expression motif* or *xMotif* X is a pair (X_S, X_G) where $X_S \subset S$ and $X_G \subset G$, such that a gene g belongs to X_G if and only if g is conserved with significance α in the samples in X_S. Our definition of conservation ensures that we do not include a gene g' in an xMotif

if the interval spanned by the expression values of g' in the samples in X_S contains many samples not in X_S. Given X_S, we can calculate the set of genes X_G in the corresponding xMotif by calculating $P_g(n, k, a)$ for each gene $g \in G$ and including g iff $P_g(n, k, a) \leq \alpha$.

We define the score of an xmotif X, SCORE(X) to be $\prod_{g \in X_G} P_g(n, k, a)$, the product of the conservation scores of the genes in X_G. In this case, an xMotif with a lower score is preferred. This scoring function has the benefit that xMotifs with large numbers of genes or samples are scored lower, and also xMotifs with smaller expression ranges for their genes are scored lower.

We can now formally state the problem that we want to solve: Given a set S of samples, a set G of genes, a set of expression values for each gene-sample pair, and a parameter α, $0 < \alpha \leq 1$, find the xMotif X whose score is the smallest over all xMotifs in the dataset. Even in the simpler case when α is just a width, SCORE is the number of genes, and xMotifs must have at least some fraction of the samples, the problem is NP-Complete.

3.5 Algorithm for finding Low-Scoring xMotifs

Our algorithm to find an xMotif X relies on the notion of a *discriminating set* of samples δ for X with the property that the set of genes conserved with significance α in X_S are exactly X_G. Algorithm 1 describes the steps of our probabilistic method to compute a set of xMotifs in $\mathcal{D}$. Algorithm 1 proceeds by selecting n_d discriminating sets of samples uniformly at random from the set of all samples in a random class $c \in C$. For each discriminating set δ, the algorithm finds the set of genes δ_G such that for each gene $g \in \delta_G$, g is conserved in δ with signficance α. If $\delta_G \neq \emptyset$, the algorthim tries to improve the score of the xMotif (δ, δ_G) using a gradient descent. The algorithm then adds or removes a sample s such that $C_s = c$ according to how well this change will improve the xMotif. Out of all possible changes to the samples in the xMotif, the algorithm adds or deletes the sample that best improves the score of the the xMotif (δ, δ_G), SCORE(δ, δ_G). This process repeated until no simgle sample changes can be made that improve the score of the xMotif further.

We say that an xMotif X is *homogeneous* if for every sample $s \in S$ such that $C_s \neq c$, if there exists a gene $g \in X_G$ such that the expression level $E_{s,g}$ for gene g in sample s is not within the interval spanned by the expression levels of g in X_S, then we consider the xMotif X *homogeneous*; we accept X. If no such sample exists, we reject and discard

xMotif X.

Algorithm 1 ComputeXMotifs(S, G, C, n_d, α): the algorithm for discovering xMotifs for all classes in L.

```
1: for each class c in C do
2:    X_c = ∅
3: for n = 1 to n_d do
4:    choose a class c ∈ C uniformly at random.
5:    choose a discriminating subset δ of the samples in c uniformly at random.
6:    Compute D' ⊂ G such that for all g ∈ D', the samples in δ are conserved with
      significance α in g
7:    Set BestScore ← score(δ, D')
8:    while δ has not changed after one iteration do
9:       Set δ' ← δ
10:      for each sample s ∈ S such that λ(s) = L_w do
11:         if s ∈ δ then
12:            Set δ_CANDIDATE ← δ − {s}
13:         else
14:            Set δ_CANDIDATE ← δ ∪ {s}
15:         Compute D'_CANDIDATE ⊂ G such that for all g ∈ D'_CANDIDATE, the samples in
            δ_CANDIDATE are conserved with significance α in g
16:         if score(δ_CANDIDATE, D'_CANDIDATE) < BestScore then
17:            Set δ' ← δ_CANDIDATE
18:            Set BestScore ← score(δ, D'_CANDIDATE)
19:      Set δ ← δ'
20:   Recompute D' ⊂ G such that for all g ∈ D', the samples in δ are conserved with
      significance α in g
21:   if (δ, D') is a homogeneous xMotif then
22:      Add the xMotif (δ, D') to X_w
23:   else
24:      discard xMotif (δ, D')
25: Return X
```

3.6 Running time of the xMotif Algorithm

The running time of this algorithm is analyzed for a single discriminating set δ, as the number of discriminating sets denoted n_δ can be selected by the user. We begin by precomputing $P_g(n, k, a)$ for all $k \leq a \leq n$ where $n = |S|$. The time to calculate each $P_g(n, k, a)$ value is constant if we can precompute the factorials used in the combination function. These factorials can be precomputed once in $O(n = |S|)$ time. The precomputation of $P_g(n, k, a)$ for all $k \leq a \leq n$ is $O(n^2)$.

For each gene in G, we can then determine the minimum and maximum expression value of the samples in δ in $O(n)$ time. By using the precomputed values for $P_g(n, k, a)$, the time to find the set of genes D' that are conserved with significance α in δ is $O(nd)$ where $d = |G|$. Furthermore, we then build all xMotifs that are different from (δ, D') by

one sample of the same class as the samples in δ in order to perform a gradient descent. We can do this in time $O(dn^2)$. The test for homogeneity can be performed in $O(dn)$ time. Therefore, the total time in the worst case for a single discriminating set is $O(dn^2)$. We repeat this calculation using gradient descent steps until the score of the xMotif cannot be improved. The number of iterations of this gradient descent is difficult to bound.

If we output an xMotif as soon as we compute it, we do not need to maintain more than a constant number of xMotifs in memory at any given time during this process. Each xMotif takes $O(d+n)$ space, and the entire dataset takes $O(dn)$ space. The precomputed tables of $P_g(n,k,a)$ require only $O(n^2)$ space. The total space requirement for the xMotif discovery algorithm in the worst case is thus $O(dn+n^2)$.

3.7 Constructing an xMotif-Based Classifier

The algorithm in Section 3.5 outputs a set $\mathcal{X}$ of xMotifs for each class $c \in C$. Our goal now is to build a concise description of each class $c \in C$ using the set $\mathcal{X}$. Our classifier h is based on a subset $\mathcal{X}'$ of $\mathcal{X}$. For each sample s in S, we include into $\mathcal{X}'$ the xMotif $X \in \mathcal{X}$ with the lowest SCORE value such that $s \in X_S$. The number of xMotifs in $\mathcal{X}'$ is at most the number of samples in the training data.

We will classify a new point by comparing its distance to each xMotif in $\mathcal{X}'$. To perform this comparison, we represent an xMotif X by a vector M_X defined as follows: each element of M_X corresponds to a gene in X_G. The value of the element is the average normalized expression value of that gene in all the samples in X_S. Thus the ideal vector M_X represents an average sample in xMotif X. This step is illustrated in Figure 3.2 where a 5x5 xMotif, which has expression values represented by color, is shown averaged into a single ideal vector.

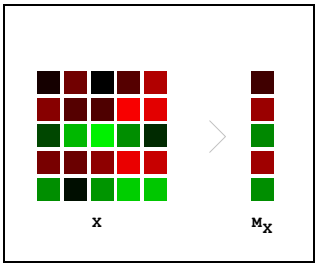


Figure 3.2: Converting a Bicluster into an Ideal Vector

In order to classify a new sample s', we first normalize its gene expression values

with respect to the samples in S. For each value $E_{s',g}$ corresponding to the expression value in our new sample for gene $g \in G$, we must find the sample with the $s^+ \in S$ smallest expression value and the sample $s^- \in S$ with the largest expression value such that $E_{s^+,g} < E_{s',g} < E_{s^-,v}$. The normalized expression value for s' is the mean of the normalized expression values for $E_{s^+,g}$ and $E_{s^-,g}$. If there exists an expression value $E_{s,g} = E_{s',g}$, we use the normalized expression value of $E_{s,g}$ for s' instead. Given an xMotif X, we denote by s'_X the vector of normalized expression values that correspond to the genes in X_G.

We can now describe how to measure the similarity of a new sample s' to the ideal vector M_X for an xMotif X using the sample vector s'_X. Both M_X and s'_X are vectors of dimension $|X_G|$. We define the similarity $\sigma(X, s')$ between the sample s' and the xMotif X as the Euclidean distance between M_X and s'_X. Note that both M_X and s'_X are points in the $|X_G|$-dimensional unit hypercube.

In order to meaningfully compare the distance of a sample s' to multiple xMotifs with different numbers of genes, we must take into account the fact that the distribution of the distance between a pair of random points in the unit hypercube varies with the dimension of the hypercube. For example, the mean distance between two random points increases with the dimension. We account for this phenomenon as follows: given $\sigma(X, s')$ for a sample s and an xMotif X, we measure the significance of obtaining a distance of $\sigma(X, s')$ or less when we pick two points uniformly at random in a unit hypercube of dimension equal to $|X_G|$. Unfortunately, we are not aware of a closed-form solution to this distribution. Therefore we estimate this distribution empirically by sampling 10,000 pairs of points from a $|X_G|$-dimensional unit hypercube.

In practice we observe that when $2 < d \leq 100$ for dimension $d = |X_G|$, the distribution of distances closely resembles the normal distribution. The maximum difference between the observed distribution's cumulative density function and the cumulative probability function of a normal distribution (the Kolmogorov-Smirnov statistic [Lil67]), with mean and standard deviation equal to the sample mean and standard deviation from the observed distribution of distances, is at most 0.018 for any dimension d such that $2 < d \leq 100$. Although the observed distributions of distances distributions are not normal[1], approximating them as a normal distribution appears reasonable.

[1]This can be shown by considering the range of possible values in a normal distribution(infinite) and in the euclidean distance metric(finite)

Given an xMotif X, we compute sample means and variances of Euclidean distances in a d-dimensional hypercube using 10,000 random pairs of test points in the hypercube and the computed distance between each pair. A normal distribution based on these values is used as an estimate for the significance of a similarity score of $\sigma(X, s')$ in d dimensional space. The significance can be calculated from the z-score of the observed distance. This significance is dimension independent and can thus be used for direct comparison of the similarity of s to multiple xMotifs. We then classify a sample s' using a winner-takes-all metric by finding the class of xMotif X in $\mathcal{X}$ with the highest significance score for $\sigma(X, s')$.

3.8 Functional Enrichment

In order to evaluate our classifier biologically, we perform a postprocessing analysis on a set of xMotifs that is commonly called functional enrichment. Functional enrichment tries to discover any biological significance implied by finding that a set X_G of genes participate in a single xMotif. For example, those genes may all encode for proteins that localize to a certain compartment of a cell or participate in a specific biological pathway. Given the set of genes in an xMotif, let G^+ be the set of genes in G that are annotated by function f. We ask the following questions: If we select $|X_G|$ genes from G uniformly at random without replacement, what is the probability that we would select $|G^+|$ or more genes annotated with the function f. We calculate this probability using the hypergeometric distribution. An experimenter-selected threshold then determines whether or not this probability is low enough to constitute a functional enrichment for the set of genes X_G.

The following definitions only consider the genes that have at least one functional annotation. $|G|$ denotes the total number of genes in a dataset. Let $|G^+|$ be the total number of genes in the dataset with functional annotation f. Within an xMotif X, let $|X_G|$ be the total number of genes in X and let $|X_G^+|$ be the total number of genes in xMotif X with functional annotation f. The probability of finding $|X_G^+|$ or more genes in X_G annotated by function f is given by the following summation:

$$P(|G|, |G^+|, |X_G|, |X_G^+|) = \sum_{i=|G^+|}^{\min(|X_G|,|X_G^+|)} \frac{\binom{|X_G^+|}{i}\binom{|G|-|X_G^+|}{|X_G|-i}}{\binom{|G|}{|X_G|}}$$

Given a sufficient number of different functions to search for, most any xMotif will be enriched with some function due to random chance. This problem is known as the *multiple hypothesis problem*. After the hyper-geometric probability is calculated, we correct for the *multiple hypothesis problem* by applying the False Discovery Rate(FDR) test [BH95] for

multiple hypotheses. This test simply rejects hypotheses that could occur by chance given the number of total hypothesis. We use a standard 0.05 α significance value for the FDR test.

Chapter 4

Results

4.1 Description of Experimental Procedure

The xMotif algorithm has two tunable parameters: the number of iterations n_δ and the significance threshold α for determining if a gene is conserved in a set of samples. Larger values of n_δ serve to find additional xMotifs, while simultaneously slowing down the training process. We arbitrarily select $n_\delta = 100$ iterations for all datasets in our study. The significance threshold α needs to be tuned on a per-dataset basis. Some datasets have many patterns of expression with very low significance scores; if we use a high value of α for such datasets an xMotif may contain many genes whose expression values are not correlated to the class distinctions. Other datasets have subtler patterns of expression and a small value of α will not find significant patterns at all. In order to select an appropriate value for α, we perform a self-validation procedure where we train the classifier using all the samples in S and classify each sample in S using the classifier. We adjust the value of α to achieve optimal accuracy on self-validation and then use this value of α for leave-one-out cross-validation (LOOCV) and for constructing the classifier.

We compare our classifier results to those of a support vector machine (SVM) [CV95]. SVMs attempt to find a hyperplane in $\mathbf{R}^d$ space such that the hyperplane splits the input space into two spaces corresponding to two different sample classifications. Ambiguities are resolved by maximizing the the distance from the closest example samples to the hyperplane. We use the software package libSVM [FCL05] with a radial basis function to train and test SVMs. libSVM comes with a number of tools used to estimate the best training parameters for a particular dataset. We use these tools and also report the time

required to run them.

After performing LOOCV, we retrain the classifier using all of the data points for further analysis. We report the training time on a 2.4GHz computer running Mandrake Linux 10.0. Finally, we analyze the xMotifs found by running enrichment analysis on the xMotifs as described in Section 3.8. Only a summary of the enrichment analyis is presented in this thesis, detailed information can be found online at the following URL: http://bioinformatics.cs.vt.edu/∼ggrothau/xMotif/

4.2 Overall Results

The comparison of accuracy and runtime for both an xMotif based classifier and an SVM classifier are reported in Table 4.1. The comparison looks at 5 different datasets which all involve distinctions between types of cancer.

Table 4.1: Overall results for 5 datasets

Dataset	xMotif Accuracy	SVM Accuracy	xMotif Train Time	SVM Train Time
Leukemia	93%	98.6%	7 minutes	2 seconds
Alizadeh Lymphoma	100%	97%	4 minutes	1 second
Global Cancer	47.4%	67.4%	11 minutes	82 seconds
Central Nervous System	75%	78%	21 seconds	1 second
Shipp Lymphoma	83%	97%	5 minutes	3 seconds

Table 4.2 shows some of the overall intermediate statistics during the xMotif algorithm. The number of xMotifs and Genes are for only those xMotifs that are used in the classifier.

Table 4.2: Statistics for 5 datasets

Dataset	Sample Count	xMotif Count	Mean # Genes per xMotif
Leukemia	72	11	118
Alizadeh Lymphoma	66	7	442
Global Cancer	190	39	2736
Central Nervous System	36	9	188
Shipp Lymphoma	77	16	810

4.3 Leukemia

Golub *et al.* [GST+99] classify a dataset of 72 acute leukemia patients. These patients are divided into two groups: 25 patients suffering from Acute Myeloid Leukemia (AML) and 47 patients suffering from Acute Lymphoblastic Leukemia (ALL). Their dataset includes 6,817 genes measured by an Affymetrix Microarray. They were able to perfectly classify the samples if allowed to leave 7 of the samples labelled as "uncertain". For comparitive purposes, the overall accuracy was thus 65 out of 72 samples (90.3%).

4.3.1 Classification Results

Our xMotif-based classifier correctly classifies 66 of 72 samples, achieving an accuracy of 93%. Training parameters were $p = 5 \times 10^{-5}$ with 100 training iterations. Training using the entire data set took 7 minutes. In comparison, LOOCV for libSVM correctly classifies 71 of 72, achieving 98.6% classification accuracy, and training using the entire dataset takes 2 seconds.

4.3.2 Final Enrichment

Reported below are selected corrected enrichment scores for interesting small functions as well as citations that indicate the value of these functions for diagnosis of the cancers. The functions listed as "Non-Classfier Functions" are functions found in xMotifs that were not used in the classifier.

Table 4.3: Enrichment scores of selected leukemia xMotifs

Function	ID	Enrichment Score
Lysozyme Activity	GO:0003796	9.4×10^{-6}
19p13	Cytogenetic Band	9.8×10^{-4}
Non-Classifier Functions		
Positive Chemotaxis	GO:0050918	4.1×10^{-6}
Ferritin Complex	GO:0008043	5.4×10^{-5}

- "All children with acute lymphatic leukemia (ALL) had significantly reduced levels of **lysozyme** at diagnosis, and none of the children fell within the normal range. ... Determination of serum **lysozyme** activity in children with acute leukemia is of value both for diagnosis and for evaluating the effect of therapy." [BM78]

- "Significant inhibition of **chemotaxis** was seen in patients with ALL (p less than 0.001) and CLL (p less than 0.01), whereas function in CML and AML patients was not significantly depressed." [NWT+80]
- "Extremely high serum **ferritin** levels were seen in acute myeloblastic leukemia before treatment ... We conclude that serum **ferritin** concentration must be valued as a clinically useful tumor marker in these types of leukemia" [AS85]
- "Cytogenetic translocations involving chromosome band 19p13, the site of the E2A gene, have previously been reported in pediatric acute lymphoblastic leukemias (ALL) in association with a precursor-B cell immunophenotype and poor prognosis." [KOSA99]

Additional enrichment results can be found online at

http://bioinformatics.cs.vt.edu/~ggrothau/xMotif/ALL-AML/

4.4 Multi-Class Lymphoid Malignancies

Alizadeh *et al.* [AED+00] present and analyze a dataset consisting of the three most prevalent lymphoid malignancies:

1. Diffuse Large B-Cell Lymphoma (DLBCL), 46 samples
2. Follicular Lymphoma (FL), 9 samples
3. Chronic Lymphocytic Leukemia (CLL), 11 samples

This dataset contains 66 samples measured on a specialized DNA Microarray containing 4,026 genes which are known or suspected to be related to lymphoid cells or cancer. Alizadeh *et al.* [AED+00] does not attempt to assess classification performance with this dataset; instead they attempt to discover novel distinctions amongst classes.

4.4.1 Classification Results

We performed LOOCV for this dataset achieving 100% classification accuracy. Training parameters were $p = 1 \times 10^{-4}$ with 100 training iterations. Training using the entire data set took 4 minutes. In comparison, SVM leave-one-out cross-validation correctly classifies 64 of 66 achieving 97% classification accuracy and trains in about 1 second.

4.4.2 Functional Enrichment

Reported below are corrected enrichment scores for selected functions as well as citations that indicate the value of these functions for diagnosis:

Table 4.4: Enrichment scores of selected lymphoma xMotifs

Cancer Class	Function	ID	Enrichment Score
DLBCL	nuclear division	GO:0000280	2.9×10^{-8}
DLBCL	cell cycle	GO:0007049	4.3×10^{-6}
FL	ferritin complex	GO:0008043	2.0×10^{-4}
DLBCL	nucleocytoplasmic transport	GO:0006913	3.9×10^{-4}

- "...follicular lymphoma (FL) ...and B-Cell chronic lymphocytic lymphoma (B-CLL), are distinguished by a relatively low proliferative index, small cell size, formation of large tumoral masses in lymph nodes, bone marrow or external locations, and a paradoxical combination of advanced clinical stages associated with low clinical aggresivity. This clinicopathic presentation seems to be the final consequence of singificant advantages to cell accumulation as a result of alterations in **apoptosis regulators** rather than in **cell cycle control** genes. ...[Large B-Cell Lymphoma is] associated with more frequently localized clinical stages but a higher clinical aggresivity, as a consequence of alterations in **cell cycle regulators** ..." [SBSAP03]
- "The serum **ferritin** levels correlated with the [Lymphoma] tumor mass." [AS90]
- "Nuclear pore complexes are large, elaborate macromolecular structures that mediate the bidirectional **nucleocytoplasmic traffic**. ...Diffuse large cell lymphomas and a lymphoblastic lymphoma stained strongly and extensively [to the nucleoporin complex Nup88]." [GMO+00]

Additional enrichment results can be found online at

http://bioinformatics.cs.vt.edu/~ggrothau/xMotif/Alizadeh/

4.5 Global Cancer Map

In 2001, Ramaswamy, et al. [RTR+01] published a dataset of 218 tumor samples from 14 different common cancer classes. Unfortunately, we were only able to obtain a subset of

this dataset containing 190 tumor samples. This publication used a one-vs-all support vector machine to evaluate cancer classification assessed using leave-one-out cross-validation. Overall prediction accuracy was 78%.

1. Breast Cancer (BR), 11 Samples
2. Prostate Cancer (PR), 10 Samples
3. Lung Cancer (LU), 11 Samples
4. Colorectal Cancer (CO), 11 Samples
5. Lymphoma (LY), 22 Samples
6. Bladder Cancer (BL), 11 Samples
7. Melanoma (ME), 10 Samples
8. Uterine Cancer (UT), 10 Samples
9. Leukemia (LE), 30 Samples
10. Renal Cancer (RE), 11 Samples
11. Pancreatic Cancer (PA), 11 Samples
12. Ovarian Cancer (OV), 11 Samples
13. Mesothelioma (MS), 11 Samples
14. Brain Cancer (BN), 20 Samples

4.5.1 Classification Results

We performed LOOCV for this dataset correctly classifying 90 of 190 achieving 47.4% accuracy with an xMotif-based classifier. Training parameters were $p = 5 \times 10^{-3}$ with 100 training iterations. Training using the entire data set took only 11 minutes. In comparison, LOOCV for libSVM correctly classifies 128 of 190 achieving 67.39% classification accuracy and trains in about 22 seconds. Determining the optimal training parameters for libSVM training takes 82 minutes, however.

Table 4.5 displays the errors made in LOOCV with the xMotif based classifier. Each row corresponds to a correct sample class and each column to a predicted sample class. The value in a row labelled u and column labelled v indicate the number of samples belonging to class u that were predicted as belonging to class v.

4.5.2 Functional Enrichment

Reported below are corrected enrichment scores for selected functions as well as citations that indicate the value of these functions for diagnosis:

- "**Heterogeneous nuclear ribonucleoprotein** A1 interferes with the binding of the human T-cell **leukemia** virus ..." [LZD+97]
- "Structural changes of the long arm of chromosome 4 or 15 and a break in **6p21** were also associated with **T-lymphoma.**" [MKS+87]

Table 4.5: xMotif Classification Detailed Results

	BR	PR	LU	CO	LY	BL	ME	UT	LE	RE	PA	OV	MS	BN
BR	9					1					1			
PR	1	4			3						1		1	
LU	5				2	1					1		2	
CO	6				2		1				1		1	
LY					20	1							1	
BL	6					4							1	
ME	3						4				1		2	
UT	3					1			1		3		2	
LE					1				29					
RE	2				1				2		2	3	1	
PA	6								1		2		2	
OV	2					4	1		3		1			
MS	1				2		1		1		4		2	
BN					1				1				2	18

Table 4.6: Selected Global Cancer Map Enrichment Scores

Cancer Class	Function	ID	Enrichment Score
Leukemia	heterogeneous nuclear ribonucleoprotein	GO:0030530	1.7×10^{-8}
Lymphoma	6p21	Cytogenetic Band	2.1×10^{-5}
Breast	hexose metabolism	GO:0019318	6.1×10^{-5}
Blastoma	Actin Cytoskeleton	GO:0030036	3.5×10^{-5}

- "GLUT12 may have a role in **hexose** supply to **breast cancer** cells." [RDS+03]
- "Galectin modulates human **Glioblastoma** cell migration into the brain through modifications to the **actin cytoskeleton** ..." [CBL+02]

Additional enrichment results can be found online at

http://bioinformatics.cs.vt.edu/~ggrothau/xMotif/GlobalCancer/

4.6 Embyronal Tumors Central Nervous System

Pomeroy *et al.* [PTG+02] present and analyze a dataset consisting of the distinct embryonal tumors of the central nervous system:

1. Medulloblastomas - 10 Samples
2. Primitive Neuroectodermal Tumors (PNET) - 6 Samples
3. Rhabdoid Tumors - 10 Samples
4. Malignant Gliomas - 10 Samples

There are 36 samples of these four cancers measured on oligonucleotide microarrays containing 6,817 genes. Pomeroy, et al. [PTG+02] use a k-nearest neighbor algorithm with a weighted voting algorithm for multi-class classification achieving 83% accuracy.

4.6.1 Classification Results

We performed leave-one-out cross-validation for this dataset achieving 75% classification accuracy with 29 of 36 correct classifications($P = 5.4 \times 10^{-9}$). Training parameters were $p = 5 \times 10^{-4}$ with 100 training iterations. Training using the entire data set took only 21 seconds. Interestingly, the xMotif algorithm finds only xMotifs with less than 10 genes during cross-validation for PNET tumors and all 6 PNET tumors are misclassified. Removing the PNET tumors improves the classification accuracy to 29 of 30 correct classifications achieving 97% accuracy.

In comparison, SVM leave-one-out cross-validation correctly classifies 28 of 36 achieving 78% classification accuracy and trains in about 1 second. Selecting only the non-PNET tumors for classification, SVM correctly classifies 29 of 30 tumors achieving 97% classification accuracy.

4.6.2 Functional Enrichment

Functional enrichment calculations of xMotifs discovered in this dataset reveal corrected enrichment scores as low as 1.3×10^{-14} for functions with large numbers of genes such as *GO: intracellular*. Scores this low indicate that these xMotifs are biologically significant, but given 3,049 genes in *GO: intracellular*, such a function is too generic to be of use to a researcher. Other enrichments were found, but a literature search turned up no corroborating evidence for these functions.

Additional enrichment results can be found online at

http://bioinformatics.cs.vt.edu/∼ggrothau/xMotif/Pomeroy/

4.7 Shipp Lymphomas

Shipp, et al. [SRT+02] present and analyze a dataset of two distinct types of lymphoma tumors:

1. Diffuse Large B-Cell Lymphoma (DLBCL) - 58 Samples
2. Follicular Lymphoma (FL) - 19 Samples

There are 77 samples of these two cancers measured on oligonucleotide microarrays containing 6,817 genes.

4.7.1 Classification Results

We performed leave-one-out cross-validation for this dataset achieving 83% classification accuracy with 64 of 77 correct classifications. Training parameters were $p = 5 \times 10^{-3}$ with 100 training iterations. Training using the entire data set took only 5 minutes. In comparison, SVM leave-one-out cross-validation correctly classifies 75 of 77 achieving 97% classification accuracy and trains in about 3 seconds.

4.7.2 Functional Enrichment

Reported below are corrected enrichment scores for selected functions as well as citations that indicate the value of these functions for diagnosis:

Table 4.7: Selected Lymphoma Enrichment Scores

Class	GO Function	GO ID	Enrichment Score
FL	Cell Cycle	GO:0007049	2.2×10^{-4}
DLBCL	Nucleocytoplasmic Transport	GO:0006913	3.5×10^{-4}
DLBCL	Nuclear Pore	GO:0005643	1.4×10^{-4}

- "...follicular lymphoma (FL) ...and B-Cell chronic lymphocytic lymphoma (B-CLL), are distinguished by a relatively low proliferative index, small cell size, formation of large tumoral masses in lymph nodes, bone marrow or external locations, and a paradoxical combination of advanced clinical stages associated with low clinical aggresivity. This clinicopathic presentation seems to be the final consequence of singificant advantages to cell accumulation as a result of alterations in **apoptosis regulators** rather than in **cell cycle control** genes. ...[Large B-Cell Lymphoma is] associated with more frequently localized clinical stages but a higher clinical aggresivity, as a consequence of alterations in **cell cycle regulators** ..." [SBSAP03]

- "Nuclear pore complexes are large, elaborate macromolecular structures that mediate the bidirectional **nucleocytoplasmic traffic**. ...Diffuse large cell lymphomas and a lymphoblastic lymphoma stained strongly and extensively [to the nucleoporin complex Nup88]." [GMO$^+$00]

  Additional enrichment results can be found online at

  http://bioinformatics.cs.vt.edu/∼ggrothau/xMotif/Shipp/

Chapter 5

Conclusions

5.1 Discussion of Results

We have developed a novel technique for classifying gene expression data sets based on the concept of a gene expression motif or xMotif. Our classifier allows for a degree of biological interpretability that is unavailable in previous work. In addition to examining genes individually, we can assess whether the set of genes in an xMotif are functionally enriched in a disease-specific context. By representing each class independently as a set of xMotifs, our technique allows for simple multi-class classification, avoiding the problem of building a multi-class classifier using multiple binary classifiers. The use of gene expression ranks avoids to some degree the problems of dataset differences that are present in other approaches such as a non-normalized support vector machine.

We demonstrated high accuracy for an xMotif-based classfier with LOOCV for a variety of data sets including multiclass datasets. Our accuracy is not as high as SVMs, but is statistically significant on multiple datasets. Our choice of methodology is limited by the need for biological interpretability through functional enrichment, a limitation not imposed on SVMs.

In addition to classification ability, our xMotif-based classfier using functional enrichment analysis is shown to provide an insight into the biological distinctions between different diseases. We present numerous examples of functions enriched in the xMotifs that are both statistically significant and confirmed by prior research.

5.2 Future Work

This thesis suggests many directions for future research. One idea we are considering is to use a standard-deviation based statistic instead of a range-based statistic in the gene selection step of the xMotif selection algorithm. Changing the statistic should improve the algorithm's robustness against noise. Range-based statistics are sensitive to outliers. However, constructing xMotifs based on standard deviation based statistics introduces new challenges. There is the issue of free-riding samples. If we have an xMotif with a small set of samples and deviation scores well below a specified threshold across the genes it is possible to add samples that are unrelated to the pattern represented by the xMotif simply because adding the sample does not deselect any gene. It would be interesting to modify the current xMotif algorithm to allow gene selection based on a standard-deviation based statistic while simultaneously ensuring that a sample in the xMotif is an outlier only for a small subset of the genes.

It became apparent that some disease classes have more subtle patterns of expression than others, even within a single dataset. Relaxing the α threshold enough to find significant xMotifs within one class may require including many erroneous genes in an xMotif for a different class. Therefore, using different class-specific α thresholds during the gene selection step of the xMotif algorithm has the potential to find more interesting patterns and to improve classification accuracy. It may even be possible for the xMotif algorithm itself to select optimal α values for a class.

A number of available improvements could be made to discover more meaningful biological interpretations of xMotifs. The functional annotation datasets used in this thesis are by no means exhaustive: simply by incorporating new annotations into the analysis, we may be able to find more biologically-significant enriched functions. We also observed that some xMotifs had both up and down regulated genes. In some cases, functions enriched in these xMotifs annotated only either up or down regulated genes. By separating the genes in an xMotif into up or down regulated gene sets we may be able to detect new enriched functions that were previously not enriched. By using regulation specific gene sets, we could directly link diseases to function. For example, we might hypothetically find that lung cancer related genes on chromosome 14 are up-regulated. In addition to analyzing gene sets that are specific to a single xMotif, potentially interesting results could occur by finding genes that are common to multiple diseases or unique to only one in a

dataset with many classes.

There are many additional uses for the xMotif system. For example we could find groups of coexpressed genes in normal tissue. We can take a dataset and find xMotifs in that dataset without consideration of any class labels. Enriched xMotifs in this gene set could be used for novel gene functional predictions. Extending this idea, by transposing the dataset , we can find samples that are conserved across sets of genes. All of the genes in such a transposed xMotif would be up and down regulated in unison across all of the samples in that xMotif. Since our technique does not require all samples to be contained in the transposed xMotif, we can find genes that may only participate in a common function under certain cell conditions, a subtle observation that may be experimentally difficult to discover, but simple to validate.

Appendix A

The BiVoC System

The xMotif algorithm produces a set of biclusters which are then analyzed in terms of their classification ability and functional enrichment. It is also useful to visualize these biclusters graphically. BiVoC, an acronym standing for Bidimensional Visualization of Clusters, was developed as a software tool designed to visualize a set of xMotifs in a dataset. In general, the BiVoC algorithm can lay out a set of biclusters.

The BiVoC algorithm constructs a single two-dimensional layout of the dataset and the biclusters within, potentially with repeated dataset rows and columns. The algorithm attempts to minimize the number of rows and columns in the layout while ensuring that the rows (columns) of each bicluster are contiguous in the layout. This property ensures that each bicluster is visually represented as a contiguous sub-matrix in the layout. This definition is general enough to visualize data other than biclusters in gene expression biclustering such as itemsets in binary retail data indicating relations between customers and items purchased. The algorithm's performance is evaluated on synthetically generated datasets and used for examining biclusters.

Jiang and Karp have shown that finding the layout of minimum size with the bicluster contiguity constraint is NP-Hard [Jia98] and restrict the problem by imposing a known ordering or interleaving of the biclusters. As far as we know, no approximation algorithm with a bounded performance guarantee exists for this general problem. Botzoglou and Istrail present a 2-approximation for this problem under the assumption that each bicluster has a column and row unique only to that bicluster [BI00]. We develop a polynomial time heuristic that attempts to construct a layout of minimal size and is guaranteed to compute the layout of minimal size exactly when there is a solution with no repeated

rows or columns. Our algorithm runs in $O(nm + n^2 \log n)$ time where n is the number of biclusters and m is the number of rows and columns in all of the biclusters.

A.1 Related Research

A binary matrix has the *Consecutive Ones Property* (COP) for rows if its columns can be arranged without repeats such that all of the ones in each row are consecutive [BL76]. See Figure A.1 for an example of a matrix with the COP. Determining whether a matrix has the COP has applications in a number of areas including testing for graph planarity [BL76], recognizing interval graphs [Hsu02, BL76], and DNA mapping using unique sequence tagged site Probes [AKNW93, LH03]. Booth and Leuker describe a data structure called the PQ-Tree which they use to represent all legal permutations of column orderings in a COP Matrix in linear time in the number of ones in the matrix.

$$\begin{pmatrix} \mathbf{1} & \mathbf{0} & 1 & 1 & 0 \\ \mathbf{0} & \mathbf{0} & 1 & 1 & 1 \\ \mathbf{1} & \mathbf{0} & 1 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 1 & 1 & 1 \\ \mathbf{0} & \mathbf{0} & 0 & 1 & 1 \end{pmatrix} \qquad \begin{pmatrix} \mathbf{0} & \mathbf{1} & 1 & 1 & 0 \\ \mathbf{0} & \mathbf{0} & 1 & 1 & 1 \\ \mathbf{0} & \mathbf{1} & 1 & 0 & 0 \\ \mathbf{1} & \mathbf{1} & 1 & 1 & 1 \\ \mathbf{0} & \mathbf{0} & 0 & 1 & 1 \end{pmatrix}$$

(a) A COP matrix before column reordering

(b) Equivalent COP matrix after reordering the first 2 columns

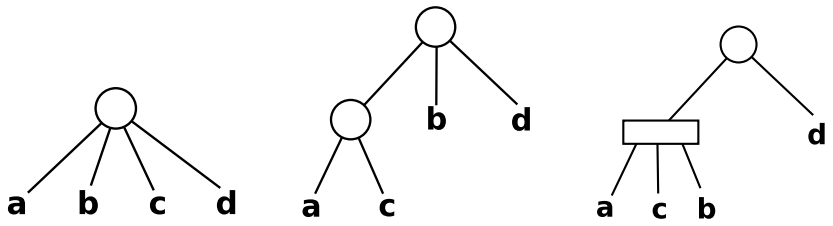
Figure A.1: An illustration of the COP

In practice, most matrices do not have the COP. Researchers have studied generalizations of the COP problem; however, most generalizations of the COP problem are NP-complete or NP-hard. For example, seeking the column ordering for a non-COP matrix that minimizes the number of gaps between the ones in each row can be reduced to the traveling salesman problem [AKWZ94]. In Section A.4, we will demonstrate that the visualization problem we are studying is equivalent to a generalization of the COP problem; in this generalization we are allowed to repeat as well as rearrange columns in order to ensure that the consecutive ones of every original row occur in at least one contiguous set of columns in that row. This problem is known to be NP-Hard [Jia98].

The most common application of this generalization of the COP is hybridization mapping with non-unique probes [Kar93]. A DNA probe is a short sequence of amino acids that can hybridize to a complementary segment of a chromosome. Biologists experimen-

tally can construct libraries of short overlapping sections (clones) of a chromosome. For each clone, they can determine which DNA probes hybridize with it. Thus, each clone can be thought of as the set of probes which hybridize to it. We can represent this data as a binary matrix where the rows represent clones, the columns represent probes, and the binary values represent the incidence of clones to probes. The goal is to determine an ordering of the probe DNA sequences that explains the clone data, while reducing the number of overall probes required in the explanation.

Most of the literature describing algorithms for the non-unique DNA probe layout problem takes advantage of the Lander-Waterman model [LW88] of clone/probe distributions along a chromosome [Kar93]. This Lander-Waterman model assumes that clones are distributed evenly across the chromosome and follow a Poisson distribution [LW88]. The additional assumption that the many algorithms exploiting the Lander-Waterman model require is that for the set S of probes found in any particular clone, there is no other clone whose probes are a strict subset of S [AKWZ94, BI00]. The only algorithm with a bounded approximation guarantee to this problem that we are aware of [BI00] requires a stronger version of this assumption that states that each clone must contain a probe not found in any other clone. Other proposed algorithms exploit additional properties that are domain specific such as knowing the ordering or interleaving of the clones [Jia98], allowing for probes to be added and removed in the presence of noise in the data [LH03], or using statistical fingerprints that can be commonly found along chromosome data [MS99]. None of these algorithms are applicable to our problem since the data we are dealing with may not have the required properties.



(a) Initial PQ-Tree T formed from set $U = a, b, c, d$

(b) REDUCE(T,a, c)

(c) REDUCE(T,b, c)

Figure A.2: The reduce operation on a PQ-Tree. P nodes are represented as circles, and Q nodes as rectangles.

A.2 PQ Trees

The problem of determining a column ordering for a matrix M that proves that M has the COP can be solved with a data structure called the PQ-Tree [BL76]. To define the PQ-Tree data structure, it is convenient to reformulate the problem as follows: Let U be the set of columns of M. We seek an ordering of the elements of U that satisfy an input of r restrictions. Each restriction is a subset S_i of U; we would like the elements of S_i to be consecutive in the derived ordering. The subset S_i corresponds to row i in M. The elements of S_i are exactly those columns that have a one in row i of M.

A PQ-Tree T is a data structure capable of representing all r legal permutations of U given the restrictions $S_i, i \leq i \leq n$. The PQ-Tree T represents these permutations using two types of internal nodes: P-nodes and Q-node. A P-node has the property that the children of the P node can be legally permuted in any way while still satisfying the restrictions. The children of a Q-node are linearly ordered in T. A valid permutation of a Q-node can contain the children either in this order or its reverse. A valid permutation of the leaves U in a PQ-Tree T is produced by traversing in-order the leaves of T for any valid permutation of the internal P and Q nodes. Figure A.2a illustrates a PQ-Tree where all possible permutations of it's leaves are valid. Figure A.2c illustrates a PQ-Tree where the only valid permutations of the leaves are the sequences $bcad, acbd, dbca$, and $dacb$.

A PQ-Tree T supports only one operation, the REDUCE operation. The REDUCE operation takes a reduction set S and a T as input and constructs a new PQ-Tree T' such that all reduction sets on T' that were applied to T are contiguous within all valid permutations of T', and S is also contiguous within all valid permutations of T'. The REDUCE operation fails if there are no remaining legal permutations [BL76]. The REDUCE operation takes time linear in $|S|$. An example of a REDUCE operation is illustrated in Figure A.2. After the REDUCE operation in Figure A.2 the valid permutations from this tree are the sequences $bcad, acbd, dbca$, and $dacb$.

The PQ-Tree data structure solves only the specific problem of finding the column permutations when there is a valid permutation with no repeats. If there is no such permutation, the PQ-Tree algorithm simply returns a failed state with no further information.

A.3 Definitions

Before describing the algorithm, it will be convenient to define some terms we will use in the rest of the paper. We denote the input matrix by D and use R and C to denote the set of rows and columns of D, respectively. Given subsets $R' \subseteq R$ and $C' \subseteq C$, we define a *bicluster* $B(R', C')$ to be the sub-matrix of D spanned by the rows in R' and the columns in C'. A *layout* $\mathcal{L}(\mathcal{R}, \mathcal{C})$ of the matrix D is a two-dimensional matrix specified as follows:

1. $\mathcal{R}$ is an ordered multi-set of rows of $\mathcal{L}$ with the property that each element of $\mathcal{R}$ is an element of R,

2. $\mathcal{C}$ is an ordered multi-set of columns of $\mathcal{L}$ with the property that each element of $\mathcal{C}$ is an element of C, and

3. $\mathcal{L}_{ij}$, the element in the ith row of $\mathcal{L}$ and the jth column of $\mathcal{R}$ is equal to $D_{i'j'}$, where i' is the row of D corresponding to the the ith row of $\mathcal{L}$ and j' is the column of D corresponding to the jth column of $\mathcal{R}$.

It is appropriate to consider $\mathcal{L}$ to be a layout of D since $\mathcal{L}$ specifies the order of its rows and columns. Allowing $\mathcal{R}$ and $\mathcal{C}$ to be multi-sets allows rows and columns of D to be repeated in the layout $\mathcal{L}$. The *size* of $\mathcal{L}$ is $|\mathcal{R}||\mathcal{C}|$.

A bicluster $B(R', C')$ is *contiguous* in a layout $\mathcal{L}(\mathcal{R}, \mathcal{C})$ if and only if the elements of R' (respectively, C') appear consecutively at least once in the ordered multi-set $\mathcal{R}$ (respectively, $\mathcal{C}$). We say that the layout $\mathcal{L}(\mathcal{R}, \mathcal{C})$ is *valid* with respect to a set of biclusters S if every bicluster $B \in S$ is contiguous in $\mathcal{L}(\mathcal{R}, \mathcal{C})$.

We can now formally define the problem we want to solve: Given a matrix D and a set S of biclusters in D, find a layout $\mathcal{L}$ of D such that $\mathcal{L}$ is valid with respect to S and $\mathcal{L}$ has the smallest size among all valid layouts of D. Note that we can construct a valid layout trivially by concatenating the rows of all the biclusters in S to form $\mathcal{R}$ and constructing $\mathcal{C}$ analogously.

A.4 The Layout Algorithm

Note that we can construct the layout $\mathcal{L}$ by determining $\mathcal{R}$ and $\mathcal{C}$ independently. In the rest of this section, we describe the algorithm to construct $\mathcal{C}$, the ordered multi-set of the columns in the layout $\mathcal{L}$. We can compute the ordered multi-set $\mathcal{R}$ analogously.

We describe the algorithm in two stages. We first transform the problem of constructing $\mathcal{C}$ to a generalization of the COP problem. We then present an algorithm to solve this transformed problem. This transformation is convenient since we can express our algorithm in terms of operations on PQ-trees.

We start by constructing a new binary matrix M that represents the columns of the biclusters in S. Each column on M corresponds to a column of the input matrix D. Let d be the number of biclusters in S. M contains d rows, one for each bicluster in S. The entry M_{ij} is 1 if bicluster $B_i \in S$ contains the column j; otherwise, M_{ij} is 0. We can now reformulate the problem of constructing $\mathcal{C}$ as follows: find a linear ordering $\mathcal{C}$ of the columns of M with the property that $\mathcal{C}$ can contain repeated columns of M and that the ones in each row of M appear contiguously.

Before describing the algorithm, we define some notation. Let c be the number of columns in M. Each PQ-Tree has as leaves some subset of the c columns in M. Let $set(T)$ denote the set of columns (leaves) in a PQ-tree T. Given two PQ-trees T and T', let $\sigma(T, T')$ denote the set similarity $\frac{set(T) \cap set(T')}{set(T) \cup set(T)}$ between the columns in T and T'. Our algorithm executes the following steps:

1. For each row i of M, construct a PQ-tree T_i consisting of a single P-node, whose children are exactly the columns in M that contain ones in row i of M. Let $\mathcal{T}$ be the set of all these PQ-trees.

2. For every pair $1 \leq i < j \leq c$, compute the set similarity $\sigma(T_i, T_j)$.

3. Sort the set similarity values in $\{\sigma(T_i, T_J), 1 \leq i < j \leq c\}$ in descending order. Let Σ denote this sorted order.

4. Repeat the following steps until Σ is empty:

   (a) Remove the smallest element from Σ. Let T and T' be the PQ-trees in $\mathcal{T}$ with this similarity value.

   (b) Let R be the set of REDUCE operations that have invoked to construct the tree T'. For each restriction $r \in R$, invoke the operation REDUCE(T, r). If any reduce operation fails, go to step (a). If all the reduce operations succeed, let T'' be the resulting PQ-tree.

   (c) Delete T and T' from $\mathcal{T}$.

   (d) For each $T \in \mathcal{T}$, insert $\sigma(T, T'')$ into Σ.

(e) Insert T'' into $\mathcal{T}$.

In its essence, the algorithm is a series of REDUCE operations. The failure of a REDUCE(T, r) operation means that the restrictions on the valid permutations of the columns in T are not compatible with the restrictions imposed by r.

A.5 Time Analysis

We will analyze the running time first for determining the column permutations only. Let the number of ones in the matrix M by denoted by m and the number of biclusters be denoted by n. Constructing the initial PQ-Trees in step 1 takes $O(m)$ time. To compare any particular pair of xMotifs takes $O(c)$ time. There are n^2 pairs to compare, and then the resulting n^2 similarity scores are sorted. To sort all of these values, corresponding to steps 2-3, takes $O(cn^2 + n^2 \log n)$ time. Step 4 is repeated at most n^2 times, and each iteration constructs a new PQ-Tree in $O(m)$ time [BL76] for a total cost of $O(mn^2)$ time. As m is strictly larger than c, the total time to determine the column permutations is $O(mn^2 + n^2 \log n)$. This time is independent of the column or row count given m, so repeating this operation for the row permutations does not change the time complexity.

A.6 Implementation

We implemented and tested these algorithms on a 2.8GHz Pentium computer running Fedora Core 2 Linux. The algorithms were implemented in C++ The software package is available under the GNU General Public License at http://bioinformatics.cs.vt.edu/∼ggrothau/BiVoC/. There are two executable programs distributed in this package. The first, `layout` implements the BiVoC algorithm as described in this paper to determine a layout $\mathcal{L}$ as a text file list of rows and columns. The second executable, `drawlayout`, uses this text file as input and allows the user a number of options for visualizing their data including:

- control over the itemset colors.
- whether rows/columns are shown if they are not present in any bicluster.
- class data for the columns as seen in Figure 4(b).
- different output image formats: postscript, png, and gif
- both binary and real-valued input formats, and options for comment fields

A.7 Testing

The Bivoc system was tested against synthetic datasets for an analysis of the performance. We then evaluated the BiVoC system using different types of real datasets to illustrate it's effectiveness.

A.7.1 Synthetic Data Sets

We created synthetic datasets by generating artificial matrices with r rows and r columns and u unique row and column identifiers. We then generated biclusters by sampling I sets of rows and columns from the matrix. For each dataset, we recorded the time required to run the BiVoC algorithm and the number of rows and columns in the output layout. The efficiency of the resulting layout is estimated by dividing the number of rows and columns in the layout the number of rows and columns in the dataset. Lower values of efficiency are better. We repeated this process using five random synthetic datasets for a number of different values of u and i, fixing $R = 100$ and averaged the results. Some of the efficiency values are less than one, which indicated that our algorithm was able to reduce the required size of the database to smaller than the original database while preserving the contiguity of the biclusters in the layout.

Table A.1: BiVoC Timing Values on Random Data in seconds.

	Unique Items				
itemsets	10	30	50	70	90
20	0.168	0.328	0.462	0.52	0.532
40	1.23	2.514	3.046	3.574	4.008
60	4.074	7.992	11.238	11.71	12.81
80	9.484	19.586	25.546	29.652	29.446
100	17.982	37.966	48.418	50.916	56.112

Table A.2: BiVoC Efficiency Values on Random Data.

	Unique Items				
itemsets	10	30	50	70	90
20	0.184	0.842	1.316	1.254	1.428
40	0.304	1.16	1.632	2.04	2.074
60	0.398	1.496	2.262	2.26	2.508
80	0.512	1.65	2.358	2.726	2.698
100	0.48	1.808	2.582	2.686	2.996

A.7.2 Transcriptional Regulation in Yeast

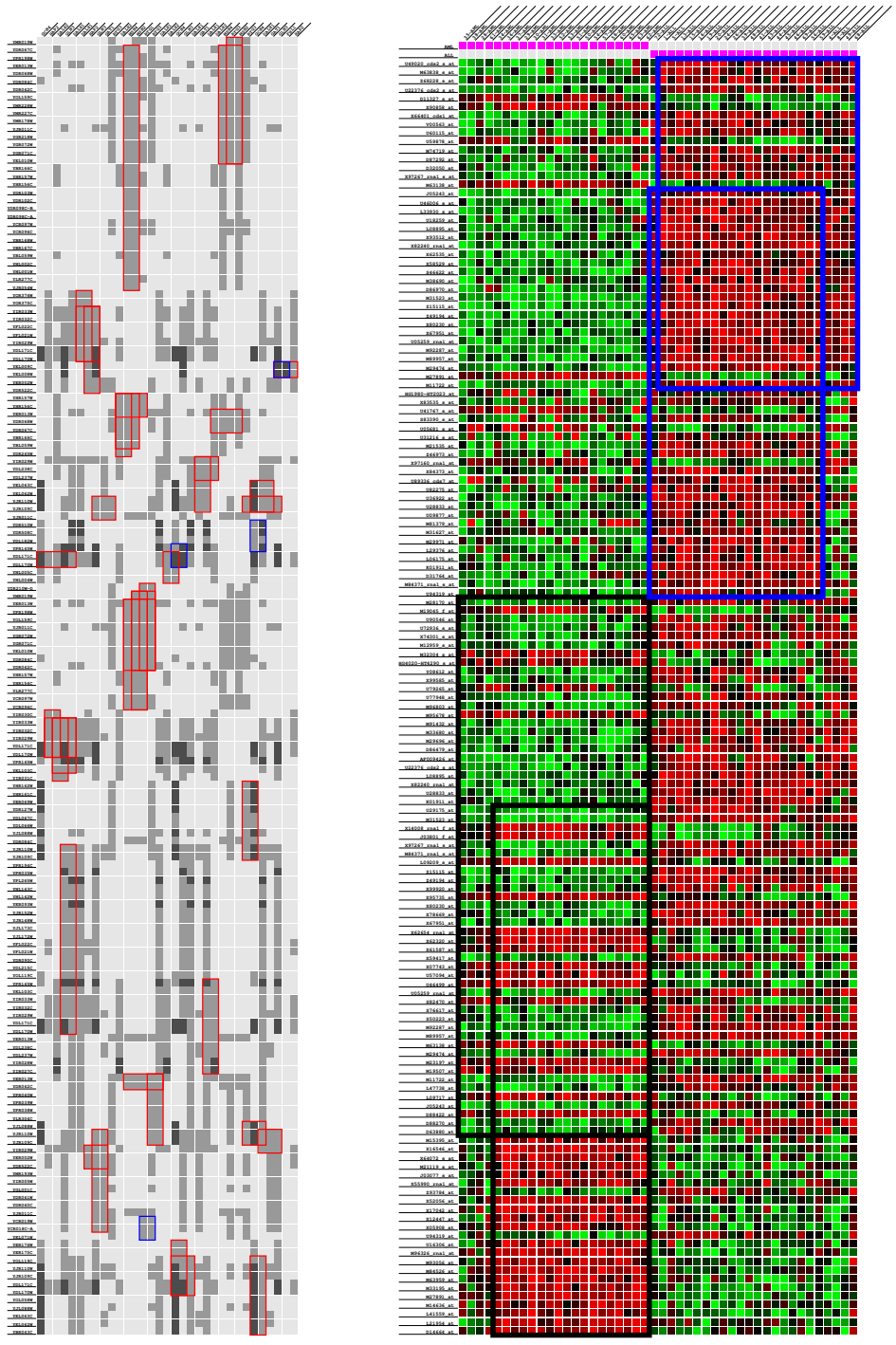
To demonstrate the ability of our visualization algorithm to analyze data other than real-valued biclusters, we analyzed datasets of transcriptional regulation in two experimental conditions in yeast. Each dataset is a binary matrix whose columns represent transcription factors and whose rows represent genes in budding yeast. The matrix entry contains a 1 if a biological experiment, ChIP-on-chip in this case, indicates that the transcription factor binds to the promoter of gene, thus potentially regulating the expression of the gene. An important problem that arises in the analysis of this data is determining if a set of genes are collectively regulated by a set of transcription factors and whether this combinatorial regulation changes when the cell is exposed to stress.

The two matrices correspond to normal growth conditions [LRR+02] and to growth conditions under exposure to a compound called rapamycin, which mimics nutrient starvation [BJGL+03]. We ran the *Apriori* [AIS93] algorithm on both of these datasets. The Apriori algorithm finds biclusters such that all of the elements in the bicluster are one in the original data matrix. We applied our visualization algorithm on biclusters with more than three genes and three transcription factors. Figure 4(a) displays the resulting layout. Biclusters obtained from the data under normal growth conditions are shown as blue boxes and rapamycin-induced biclusters are shown as red boxes. The underlying grey squares here indicate the number of databases in which that relationship was present, darker indicating more databases. The image strikingly demonstrates that under exposure to rapamycin, the transcriptional network activated in the cell is very different from the normal activation network. Very few genes are co-regulated by the same set of transcription factors in both conditions. A biologist can easily interpret these images and construct multiple hypotheses to validate in the laboratory.

A.7.3 ALL/AML Cancer Classification

The Acute Lymphoblastic Leukemia (ALL) and Acute Myeloid Leukemia (AML) dataset [GST+99] contains 72 microarray data sets of ALL and AML cancer patients. For each patient, there are 7129 real-valued measurements of individual gene expression. These are the output results from microarray experiments which measure mRNA levels in the cancer cells.

We ran the xMotif algorithm to compute biclusters in this dataset and selected some



(a) Rapamycin Response Network

(b) ALL/AML Cancer Classification

Figure A.3: BiVoC Experimental Visualizations

representative biclusters from the results to visualize. Figure 4(b) displays the layout. The individual array values are translated into a red/green spectrum spanning the range of expression values for each gene. The view of the biclusters within the context of the larger gene expression array highlights clearly the distinct gene expression patterns between these two similar cancer types. The two purple bars near the top of the image indicate the cancer type of each column.

Appendix B

libGO

libGO is a library developed in C++ that encapsulates the Gene Ontology(GO) [Con01], a directed acyclic graph whose nodes correspond to a well-defined set vocabulary of gene functions. In the Gene Ontology, a tag describes a particular function in some way. libGO currently parses and recognizes all tags currently available in the Gene Ontology and provides accessors to the following tags through library methods:

- ID
- Alternative IDs
- Subsets
- Synonyms
- Xref Analogs
- Name
- Namespace
- Definition
- Comments
- Obsolete

The Gene Ontology specifies functions as parent-child relationships is-a and part-of. If a function F annotates a gene g, and F is part-of (or is-a) function F', then g automatically has the function F', by definition. libGO calculates these functional relations, returning all the functions in GO that annotate g. Parents can be selected as is-a, part-of, or both relations in order to perform transitive closure. libGO's parent transitive closure is used by xMotif to compute all functional annotations for a gene given the functions that that gene is initially annotated with. The libGO software package is available under the GNU General Public License at http://bioinformatics.cs.vt.edu/~ggrothau/libGO/.

Appendix C

libEnrichment

libEnrichment is a library developed in C++ that presents a interface for computing functional enrichment using hyper-geometric statistics. In xMotif, libEnrichment is used to find functions that are enriched in a particular xMotif as decribed in Section 3.8.

libEnrichment is implemented as a C++ templated set of classes. Its methods support direct computation of the hyper-geometric statistics as described in Section 3.8. Its methods also support enriched annotation searches including bonferroni correction, holms correction, or false discovery rate correction [Dra03]. The results of these searches provide access to the enrichment score, the relation of type T with that score, and the values explaining the enrichment calculation. The libEnrichment software package is available under the GNU General Public License at http://bioinformatics.cs.vt.edu/∼ggrothau/libEnrichment/.

Bibliography

[ABN+99] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences of the United States of America*, 96(12):6745–6750, 1999.

[AED+00] Ash A. Alizadeh, Michael B. Eisen, R. E. Davis, Chi Ma, Izidore S. Lossos, Andreas Rosenwald, Jennifer C. Boldrick, Hajeer Sabet, Truc Tran, Xin Yu, John I. Powell, Liming Yang, Gerald E. Marti, Troy Moore, James Hudson, Lisheng Lu, David B. Lewis, Robert Tibshirani, Gavin Sherlock, Wing C. Chan, Timothy C. Greiner, Dennis D. Weisenburger, James O. Armitage, Roger Warnke, Ronald Levy, Wyndham Wilson, Michael R. Grever, John C. Byrd, David Botstein, Patrick O. Brown, and Louis M. Staudt. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000.

[AIS93] Rakesh Agrawal, Tomasz Imilienski, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, volume 22, pages 207–216, New York, NY, USA, 1993. ACM Press.

[AKNW93] Farid Alizadeh, Richard M. Karp, Lee A. Newberg, and Deborah K. Weisser. Physical mapping of chromosomes: a combinatorial problem in molecular biology. In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete algorithms*, pages 371–381, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.

[AKWZ94] Farid Alizadeh, Richard M. Karp, Deborah K. Weisser, and Geoffrey Zweig. Physical mapping of chromosomes using unique probes. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 489–500, Philadelphia, PA, USA, 1994. Society for Industrial and Applied Mathematics.

[AS85] E. Aulbert and C. G. Schmidt. Ferritin—a tumor marker in myeloid leukemia. *Cancer Detection and Prevention*, 8(1-2):297–302, 1985.

[AS90] E. Aulbert and O. Steffens. Serum ferritin–a tumor marker in malignant lymphomas. *Onkologie*, 13(2):102–108, 1990.

[BDBF+00] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. *Journal of Computational Biology*, 7(3-4):559–583, 2000.

[BH95] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society*, 57:289–300, 1995.

[BI00] Serafim Batzoglou and Sorin Istrail. Phyiscal mapping with repeated probes: The hypergraph superstring problem. *Journal of Discrete Algorithms*, 1:51–76, 2000.

[BJGL+03] Z. Bar-Joseph, G. K. Gerber, T. I. Lee, N. J. Rinaldi, J. Y. Yoo, F. Robert, D. B. Gordon, E. Fraenkel, T. S. Jaakkola, R. A. Young, and D. K. Gifford. Computational discovery of gene modules and regulatory networks. *Nature Biotechnology*, 21(11):1337–1342, 2003.

[BL76] K. S. Booth and G. S. Lueker. Testing for the consecutive ones property, interval graphs, and planarity using pq-tree algorithms. *Journal of Computational Systems Science*, 13:335–379, 1976.

[BM78] D. Bratlid and P. J. Moe. Serum lysozyme activity in children with acute leukemia. *European Journal of Pediatrics*, 127(4):263–268, 1978.

[CBL+02] I. Camby, N. Belot, F. Lefranc, N. Sadeghi, Y. d. Launoit, H. Kaltner, S. Musette, F. Darro, A. Danguy, I. Salmon, H. J. Gabius, and R. Kiss.

Galectin-1 modulates human glioblastoma cell migration into the brain through modifications to the actin cytoskeleton and levels of expression of small gtpases. *Journal of Neuropathology and Experimental Neurology*, 61(7):585–596, 2002.

[CC00] Yizong Cheng and George M. Church. Biclustering of expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 93–103. AAAI Press, 2000.

[Con01] Gene Ontology Consortium. Creating the gene ontology resource: design and implementation. *Genome Research*, 11:1425–1433, 2001.

[CST00] A. Califano, G. Stolovitzky, and Y. Tu. Analysis of gene expression microarrays for phenotype classification. *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, 8:75–85, 2000.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning (Historical Archive)*, 20(3):273–297, 1995.

[DH73] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Willey & Sons, New Yotk, 1973.

[DL00] B. J. Druker and N. B. Lydon. Lessons learned from the development of an abl tyrosine kinase inhibitor for chronic myelogenous leukemia. *The Journal of Clinical Investigation*, 105(1):3–7, 2000.

[Dra03] Sorin Draghici. *Data Analysis Tools for DNA Microarrays*. Chapman & Hall/CRC, 2003.

[ESBB98] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the United States of America*, 95(25):14863–14868, 1998.

[FCD+00] T. S. Furey, N. Cristianini, N. Duffy, D. W. Bednarski, M. Schummer, and D. Haussler. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics*, 16(10):906–914, 2000.

[FCL05] R. Fan, P. Chan, and C. Lin. Libsvm: A library for support vector machines. Technical report, Department of Computer Science, National Taiwan University, 2005.

[GMO+00] V. E. Gould, N. Martinez, A. Orucevic, J. Schneider, and A. Alonso. A novel, nuclear pore-associated, widely distributed molecule overexpressed in oncogenesis and development. *American Journal of Pathology*, 157(5):1605–1613, 2000.

[GST+99] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286(5439):531–537, 1999.

[Hsu02] Wen-Lian Hsu. A simple test for the consecutive ones property. *Journal of Algorithms*, 43(1):1–16, 2002.

[Jia98] Tao Jiang. Mapping clones with a given ordering or interleaving. *Algorithmica*, 21(3):262–284, 1998.

[Kar93] Richard M. Karp. Mapping the genome: some combinatorial problems arising in molecular biology. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Theory of Computing*, pages 278–285, New York, NY, USA, 1993. ACM Press.

[KOSA99] H. S. Khalidi, M. R. O'Donnell, M. L. Slovak, and D. A. Arber. Adult precursor-b acute lymphoblastic leukemia with translocations involving chromosome band 19p13 is associated with poor prognosis. *Cancer Genetics and Cytogenetics*, 109(1):58–65, 1999.

[KSG04] Mehmet Koyuturk, Wojciech Szpankowski, and Ananth Grama. Biclustering gene-feature matrices for statistically significant dense patterns. In *Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference*, pages 480–484, Washington, DC, USA, 2004. IEEE Computer Society.

[LDB+96] D. J. Lockhart, H. Dong, M. C. Byrne, M. T. Follettie, M. V. Gallo, M. S. Chee, M. Mittmann, C. Wang, M. Kobayashi, H. Horton, and E. L. Brown.

Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nature Biotechnology*, 14(13):1675–1680, 1996.

[LH03] W. F. Lu and W. L. Hsu. A test for the consecutive ones property on noisy data–application to physical mapping and sequence assembly. *Journal of Computational Biology*, 10(5):709–735, 2003.

[Lil67] HW Lilliefors. On the Kolmogorov-Smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 1967.

[LRR+02] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J. B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young. Transcriptional regulatory networks in saccharomyces cerevisiae. *Science*, 298(5594):799–804, 2002.

[LW88] E. S. Lander and M. S. Waterman. Genomic mapping by fingerprinting random clones: a mathematical analysis. *Genomics*, 2(3):231–239, 1988.

[LZD+97] H. P. Li, X. Zhang, R. Duncan, L. Comai, and M. M. Lai. Heterogeneous nuclear ribonucleoprotein a1 binds to the transcription-regulatory region of mouse hepatitis virus RNA. *Proceedings of the National Academy of Sciences of the United States of America*, 94(18):9544–9549, 1997.

[MK03] T. M. Murali and S. Kasif. Extracting conserved gene expression motifs from gene expression data. *Pacific Symposium on Biocomputing 2003*, pages 77–88, 2003.

[MKS+87] N. Maseki, Y. Kaneko, M. Sakurai, M. Kurihara, K. Sampi, K. Shimamura, and S. Takayama. Chromosome abnormalities in malignant lymphoma in patients from Saitama. *Cancer Research*, 47(24 Pt 1):6767–6775, 1987.

[MS99] Guy Mayraz and Ron Shamir. Construction of physical maps from oligonucleotide fingerprints data. In *Proceedings of the Third Annual International Conference on Computational Molecular Biology*, pages 268–277, New York, NY, USA, 1999. ACM Press.

[NWT+80] D. A. Norris, W. L. Weston, D. G. Tubergen, B. Rose, and L. F. Odom. Monocyte chemotaxis in leukemia patients. *The Journal of Laboratory and Clinical Medicine*, 95(4):609–615, 1980.

[PTG+02] Scott L. Pomeroy, Pablo Tamayo, Michelle Gaasenbeek, Lisa M. Sturla, Michael Angelo, Margaret E. McLaughlin, John Y. H. Kim, Liliana C. Goumnerova, Peter M. Black, Ching Lau, Jeffrey C. Allen, David Zagzag, James M. Olson, Tom Curran, Cynthia Wetmore, Jaclyn A. Biegel, Tomaso Poggio, Shayan Mukherjee, Ryan Rifkin, Andrea Califano, Gustavo Stolovitzky, David N. Louis, Jill P. Mesirov, Eric S. Lander, and Todd R. Golub. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870):436–442, 2002.

[RDS+03] S. Rogers, S. E. Docherty, J. L. Slavin, M. A. Henderson, and J. D. Best. Differential expression of glut12 in breast cancer and normal breast tissue. *Cancer Letters*, 193(2):225–233, 2003.

[RTR+01] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America*, 98(26):15149–15154, 2001.

[SBSAP03] M. Snchez-Beato, A. Snchez-Aguilera, and M. A. Piris. Cell cycle deregulation in b-cell lymphomas. *Blood*, 101(4):1220–1235, 2003.

[SRT+02] M. A. Shipp, K. N. Ross, P. Tamayo, A. P. Weng, J. L. Kutok, R. C. Aguiar, M. Gaasenbeek, M. Angelo, M. Reich, G. S. Pinkus, T. S. Ray, M. A. Koval, K. W. Last, A. Norton, T. A. Lister, J. Mesirov, D. S. Neuberg, E. S. Lander, J. C. Aster, and T. R. Golub. Diffuse large b-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, 8(1):68–74, 2002.

[SSDB95] Mark Schena, Dari Shalon, Ronald W. Davis, and Patrick O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270(5235):467–470, 1995.

[TKY+04] S. Tomida, K. Koshikawa, Y. Yatabe, T. Harano, N. Ogura, T. Mitsudomi, M. Some, K. Yanagisawa, T. Takahashi, H. Osada, and T. Takahashi. Gene expression-based, individualized outcome prediction for surgically treated lung cancer patients. *Oncogene*, 23(31):5360–5370, 2004.

[TSS02] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18 Suppl 1, 2002.

[YRT+01] C. H. Yeang, S. Ramaswamy, P. Tamayo, S. Mukherjee, R. M. Rifkin, M. Angelo, M. Reich, E. Lander, J. Mesirov, and T. Golub. Molecular classification of multiple tumor types. *Bioinformatics*, 17 Suppl 1, 2001.

[ZYS03] H. Zhang, C. Y. Yu, and B. Singer. Cell and tumor classification using gene expression data: construction of forests. *Proceedings of the National Academy of Sciences of the United States of America*, 100(7):4168–4172, 2003.

[ZYSX01] H. Zhang, C. Y. Yu, B. Singer, and M. Xiong. Recursive partitioning for tumor classification with gene expression microarray data. *Proceedings of the National Academy of Sciences of the United States of America*, 98(12):6730–6735, 2001.

Vita

Gregory Allen Grothaus was born in Metarie, Louisiana on October 22nd, 1978. He graduated with a Bachelor of Science in Computer Science and Applications from Virginia Polytechnic Institute and State University in May 2003. During his first year of Master's studies at Virginia Polytechnic Institute and State University, he worked as a Graduate Teaching Assistant for a Computer Graphics course. After graduation, he will be joining Google in Mountain View, California in July 2005 as a Software Engineer.