University of Alberta

MAKING GENE SETS MORE COHERENT

by

Fariba Mahdavifard

A thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computing Science

Fall 2009
Edmonton, Alberta

Examining Committee

Russ Greiner, Department of Computing Science

Paul Stothard, Department of Agricultural, Food and Nutritional Science (AFNS)

Guohui Lin , Department of Computing Science

Abstract

One important goal in microarray data analysis is to learn a predictor using a patient's microarray data to predict some important characteristics of that patient. The high dimensionality of data makes learning such classifiers very challenging. We tried to use prior biological knowledge to tackle the challenges. Our colleagues have produced clusters of genes with a common function, called "PBT"s, for mouse and human. We hoped we could use each cluster as a single feature. This is most effective if each PBT is "coherent". They expect all PBTs to be coherent; but while mouse PBTs are coherent, human PBTs are not.

In this thesis we propose a method, called MkCoh, to improve the coherency of each PBT by removing and flipping some genes. We expected the predictors based on the revised PBTs to be more accurate than the ones based on either the original PBTs, or on the original gene expression values. However, our experimental results did not demonstrate this; we explored some possible reasons.

Acknowledgements

First, I would like to thank my supervisor, Dr. Russell Greiner, for being an enthusiastic and genius supervisor. I am grateful to have the honor and pleasure of working under his supervision. This research would have been impossible without his help, encouragement and patience.

Many thanks to Paul Stothard and Guohui Lin for a very careful reading of the thesis, and the helpful comments that improved the thesis. Also, I would like to thank the staff and faculty of the Department of Computing Science at the University of Alberta for providing such a nice academic environment.

I would like to express gratitude to my dear friends and colleagues. Many thanks go in particular to Nasimeh Asgarian for being a very helpful colleague and friend. I am also thankful to Mohammad Reza Salavatipour, Amin Jorati and Saman Vaisipour. I would like to thank all our colleagues at Alberta Transplant Applied Genomic Center. Also, I thank the staff at the Alberta Ingenuity Centre for Machine Learning.

Finally, the people close to my heart deserve the uppermost appreciation for their support and patience. Many special thanks to my dear parents and my dear brother. I always feel the warmth of their love. Last but not least, I would like to thank my dear husband, Arash, whose patience, love and support made my studies both possible and enjoyable. This thesis is dedicated to him.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Overview

One application of microarray data is the prediction task. In this thesis our goal is to learn a predictor that can predict some important characteristics of a patient based on the data from his/her microarray. The large number of genes and the small number of samples make building such classifiers very challenging. But each microarray feature is based on a gene, which has many known properties that could be used as prior knowledge. Chapter 2 reviews some approaches have been applied to microarray data analysis. Some of them are only based on the data, while others use prior biological knowledge to get better results.

The known properties of genes have motivated biologists to use prior knowledge to form clusters of (dozens to hundreds) genes with a common function. Our colleagues, the members of the Alberta Transplant Applied Genomics Centre (ATAGC[1]), produced such clusters called "PBT"s (pathogenesis-based transcript sets) [37, 31]. They have provided 38 human PBTs and 21 mouse PBTs where each PBT is a cluster of genes with a common function, and some PBTs overlap.

Our colleagues initially characterized mouse PBTs as "herd movements" in mouse transplant data — that is, the genes in each mouse PBT have similar expression patterns, or in other words they are highly correlated. They then created human PBTs, many by homology with the mouse PBTs. They expected to see the same "herd movements" in those human PBTs as well. However, our empirical studies showed this assumption was not always true: many pairs of genes in human PBTs had negligible, or negative, correlations.

In this thesis, we propose a method, called *MkCoh*, for efficiently modifying each given PBT, by removing some genes and "flipping" some others, in order to improve the pairwise correlation of the genes in that PBT. Chapter 3 describes *MkCoh* algorithm and shows the results of applying this algorithm on PBTs.

[1] http://www.atagc.med.ualberta.ca/

Our plan, all along, was to use PBTs (and revised PBTs) to reduce the dimensionality of data. We hoped we could use each cluster (rather than each gene) as a single feature, because when a gene set is coherent, we can view all the genes in that set as a single entity by collapsing the expression values of all the genes into a simple summary statistic (*e.g.*, the arithmetic mean). We expected that the predictors based on these revised PBTs should be more accurate than the ones based on either the original PBTs, or on the original gene expression values. Unfortunately, our experimental results based on some real biological datasets did not show that. Chapter 4 presents these results and explores possible reasons for this negative finding. Finally Chapter 5 draws conclusions and describes some future works.

1.2 Introduction to Microarrays

Recent advances in microarray technology have made it one of the essential tools for biologists, as it allows them to monitor expression levels of genes in a given organism; it allows biologists to obtain a "global" view of the cell. Microarray technology allows them to example tens of thousands of genes at the same time. The *samples* may correspond to different time points, different experimental conditions, different organs, diseased or healthy tissues, or different individuals.

Microarray technology has great potential to provide accurate medical diagnosis, by helping to find better treatments and cure for many diseases[2]. Microarray data analysis is an important research area in bioinformatics. Medical researchers anticipate microarray data will provide medically relevant information. They analyze these data seeking meaningful patterns in the gene expression levels because these patterns can increase their understanding of normal and disease states, *e.g.*, it can help them to determine the genes

[2]See Appendix A as an example.

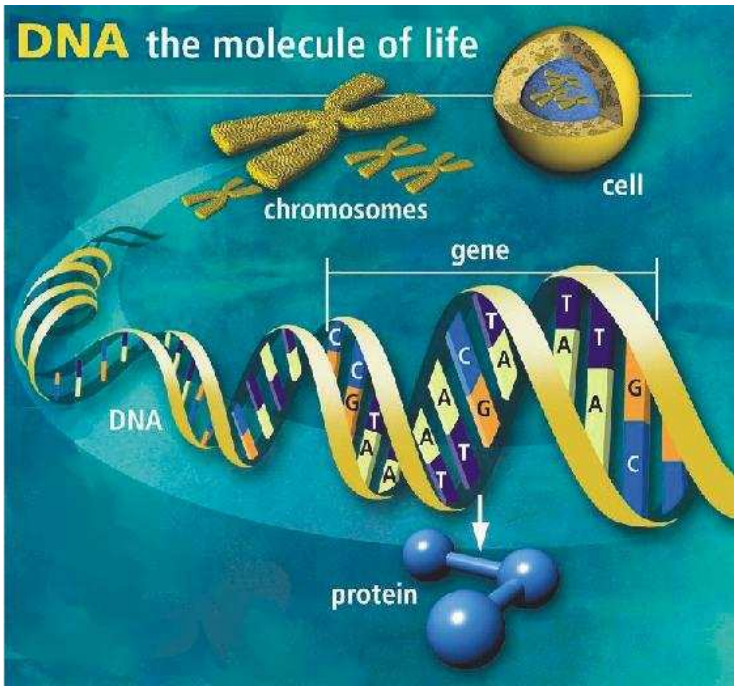


Figure 1.1: Cell, DNA, gene and protein [2].

involved in a disease. Machine learning and statistical techniques have been applied on microarray data to address some of the class discovery or class prediction biomedical problems such as predicting post-treatment outcome [34].

1.2.1 Biology Background

A *cell* is the fundamental unit of life as it contains all the structures and molecular constituents needed for life. Almost every cell of the body contains a full set of genes. *DNA (deoxyribonucleic acid)* carries the genetic information of a cell, in a set of one or more molecules, called chromosomes. As shown in Figure 1.1, each such chromosome, consists of thousands of genes that determine the hereditary traits of organisms. The *genes* do so typically by specifying the structure of *proteins*, which are large complex molecules that

do essential work [12, 34]. The sequence of amino acids in a protein specifies the shape and functions of that protein.

The term "Gene expression" describes the process by which information from a gene is used in the synthesis of a functional gene product; these products are often proteins. There is an intermediate between a gene and its corresponding protein that is called a messenger RNA (mRNA), which is the cell's template for creating that specific protein. To produce a protein from DNA, DNA is first transcribed to mRNA, which is then translated to protein.

Each gene codes for a protein, but in any cell only some of these genes are expressed, and it is this expressed subset that specifies the unique properties of each cell type [12]. Researchers evaluate the state of a cell based on what genes are expressed within it. A microarray measures the level of activity of genes by measuring the amount of mRNA for each gene.

A DNA-microarray is usually a small glass slide onto which the sequences from thousands of different genes are attached at fixed locations. As shown in Figure 1.2, DNA microarrays can be created by spotting a subsequence of every gene in a genome onto a glass microscope slide.

1.2.2 Microarray Technology

Microarray measures the amounts of each specific mRNA to find the active genes. To determine which genes are turned on and which are turned off in a given cell, a researcher must first collect the messenger RNA molecules present in that cell. The researcher then labels each mRNA molecule by attaching a fluorescent dye. Next, he places the labeled mRNA onto a DNA microarry slide. The messenger RNA that was present in the cell will then hybridize – or bind – to its complementary DNA on the microarray, leaving its fluorescent tag. The researcher then uses a scanner to measure the fluorescent areas on

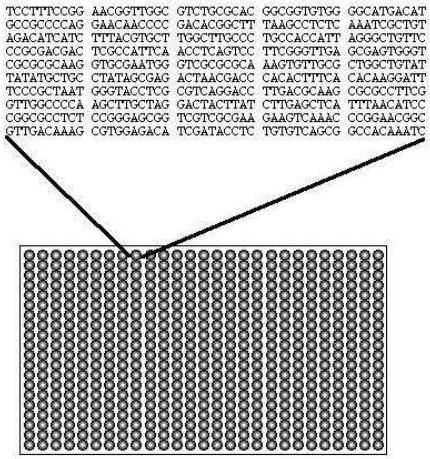


Figure 1.2: DNA microarrays can be created by spotting sequences from every gene in a genome onto a glass microscope slide [3].

the microarray. A very active gene will produce many molecules of messenger RNA, which hybridize to the DNA on the microarry and generate a very bright fluorescent area. Genes that are somewhat active produce fewer mRNAs, which results in dimmer fluorescent spots. Finding no fluorescence means that none of the messenger molecules have hybridized to the DNA, indicating that the gene is inactive [7].

Therefore, to obtain the raw microarray data, the microarray is then placed on a scanner that uses a specific frequency of light from a laser. This produces an image from the scanned array. To get information about gene expression levels, this image is analyzed; each spot on the array is identified, its intensity is measured and compared to the background. To obtain the final gene expression matrix from spot quantitations, all the quantities related to some gene have to be combined and the entire matrix has to be scaled to make different arrays comparable. This process is called normalization [5, 12]. After these steps, the microarray data is ready to be analyzed.

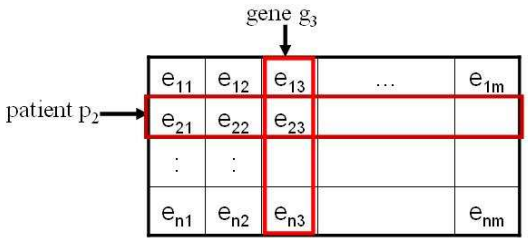


Figure 1.3: A gene expression matrix.

1.3 Microarray Data Analysis

In microarray data, the data from one sample corresponds to an $m \times 1$ vector of real numbers, where m is the number of genes, which is usually tens of thousands. The microarray data for all of the samples can be viewed as a matrix of expression levels. When we have n samples, that would be an $n \times m$ matrix of real numbers. Each row of the matrix contains the expression levels of the m genes, and each column contains the expression levels of a gene as it varies over the n samples. Figure 1.3 shows this matrix.

1.3.1 Challenges for Microarray Data Analysis

Medical researchers anticipate that microarray data will provide medically relevant information. Our colleagues have obtained microarray data from over a hundred samples hoping to find meaningful patterns. But microarray datasets have special characteristics that pose some challenges in their analysis.

Typical datasets have more samples than features $(n > m)$, but a microarray dataset has many more features than samples $(m \gg n)$. The number of features (genes), $|G| = m$, in a microarray is very large, in the order of tens of thousands; given the difficulty and financial cost of collecting microarray samples, the number of samples, $|P| = n$, is typically much smaller, usually less than a hundred. Such a dataset with the property of having a large

number of features is called a *high dimensional dataset*. The high dimensionality of data poses a challenge that given the small sample size, it is very difficult to find meaningful patterns [30]. In particular, the high dimensionality of the feature (gene) space, and the fact that many features are irrelevant or redundant, makes it challenging to analyze microarray data. Dealing with noise, outliers and missing values are the other challenges of working with microarray data. Also, there is still some limitations in microarray technology and measuring gene expressions; [41, 25] address some of these problems.

The following subsections describe the main tasks in microarray data analysis.

1.3.2 Classification and Regression

Finding the relationship within genes and samples, or between them, is an important task in microarray data analysis. In general, we can define *classification* as the process of predicting the class label of an instance. This model can be learned by applying a learning algorithm to a dataset, here the overall task includes two main steps: 1) learning a model from a dataset with known class labels, and then 2) classification, which is predicting the class of unlabeled data points. Figure 1.4 illustrates these two steps.

Biologists are interested in several classification tasks involving microarray data, which can be divided into these main groups [12]:

1. **Gene Classification:** using the expression values of a set of genes belonging to known classes, predict the class of a new gene, with known expression values.

2. **Gene Interactions:** it is known that the expression level of gene g is regulated by a set of other genes, G. Using the expression values of the genes in the set G, predict the expression value of g.

3. **Sample Classification:** using the expression values of a set of samples with known

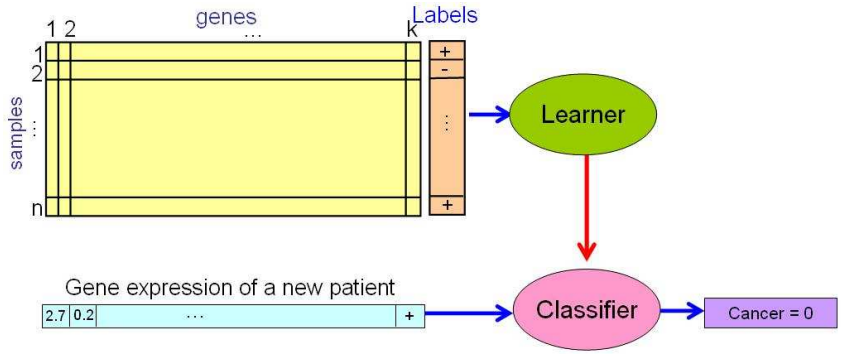


Figure 1.4: Learning and Classification.

classes, predict the class of a new sample based on its expression values.

4. **Time Series:** gene expression levels change over time, as proteins regulate gene transcription, due to a change in the state of a disease, or due to the effect of a certain treatment. Given expression levels of genes from different time points $\{t_1, ..., t_n\}$, predict the time point for a new gene given its expression values.

In this thesis, we focus on the third task, *i.e.*, sample (patient) classification. Sample classification requires learning a classifier from the data for different samples (with known classes) and their gene expression levels. Then, the learned model is used to predict the class of a new sample, see Figure 1.4.

In biology, there are many categories of classification including [12]:

Cancer versus normal (Diagnosis): We have the expression values of a set of patients, some of whom are known to have cancer and some of them do not. We use this data to build a classifier in order to predict if a new patient has cancer or not.

Type of cancer (Diagnosis): We have the expression values of a set of patients, all of whom have the same cancer but with different types. We build a classifier on this data in order to predict the type of cancer for a new patient.

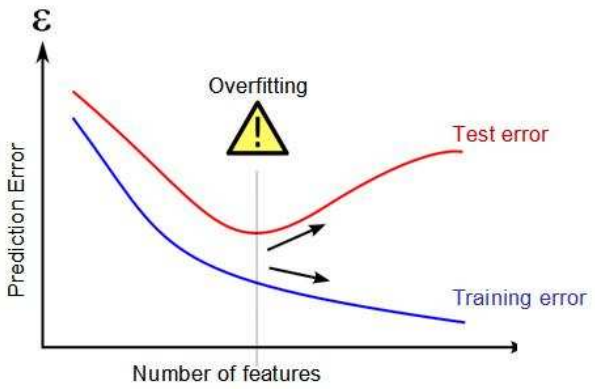


Figure 1.5: Overfitting [10].

Clinical outcome (Prognosis): We have the expression values of a set of patients, all of whom have the same cancer but with different types. The patients will all undergo the same treatment and after a certain amount of time they will be checked to determine if they have responded to the treatment or not. The class label here is whether their cancer recurred after a certain treatment or not. Therefore, the task here is to build a classifier in order to predict if a patient responds to a certain treatment or not.

For each of these classification tasks, a set of microarray experiments, each corresponding to mRNA from a different patient, is given. The mRNA is typically taken from the same cell type from each patient. Since this is *supervised learning*, the class labels for these patients are known as well. We use this data set (gene expression values of patients and the class label for each patient) to produce a classifier that can be used to predict the class of a new patient, given his or her gene expression values.

Overfitting

As we explained before, microarray data has high dimensions, with many more features than samples. *Overfitting* often occurs when we are dealing with such data. It means

that as the number of features (for a fixed number of training instances) increases, the prediction accuracy on training data increases, however the prediction accuracy on test data decreases, see Figure 1.5. Using all of the features (genes), especially when they are highly correlated with each other or when many of them are irrelevant, in order to build the classifier for predicting the patient's class, often leads to overfitting.

The best way to avoid over-fitting is to increase the number of training instances. But in some datasets, such as microarray data, it is often not possible to have more than just a few samples. Therefore we need to reduce the dimensionality of the data by finding the most relevant subset of genes and removing the others. This process, called gene selection, will be explained in Section 1.3.3.

Regression

For some biological experiments, we would like to learn a continuous valued function based on gene expression patterns and then predict the value of the function for a new sample. This process is called *regression*. Regression is similar to classification, in that they both learn a model from the labeled data of different samples and then use the learned model to predict the label of a new sample. But in regression tasks the labels (targets) are real valued numbers, while in classification tasks, class labels are categorical or nominal.

1.3.3 Feature Selection

Feature selection, also known as feature reduction, is the technique commonly used in machine learning to reduce the number of features in datasets. Most feature selection algorithms select a subset of relevant features by removing most irrelevant and redundant features from the data. We can find more meaningful patterns in data with a fewer number of redundant features. In high dimensional datasets, feature selection can help us to reduce

the high dimensionality of data.

When feature selection is applied to microarray data, the technique, here called *gene selection*, tries to detect informative genes in a DNA microarray experiment. By removing most irrelevant and redundant features from the microarray data, feature selection can improve the performance of learning models and therefore increase the accuracy of prediction.

1.3.4 Clustering

Clustering is the process of organizing objects into groups whose members are "similar" in some way. Clustering can be considered as the most important *unsupervised learning problem*; it is called unsupervised because it deals with finding a structure in a collection of unlabeled data. A *cluster* is a collection of objects that are "similar" to each other and are "dissimilar" to the objects belonging to other clusters.

It is often meaningful to cluster data with respect to features or samples. In microarray data, we can cluster both genes and samples. **Gene clustering** treats the genes as the objects and the samples as the features; it organizes genes in clusters based on their expression patterns. **Sample clustering** regards the samples as the objects and the genes as the features; it partitions the samples into the homogeneous groups. There is another type of microarray clustering that is called **biclustering**, which performs clustering on both genes and samples at the same time. It finds a *subset* of genes that have similar pattern under a specific *subset* of samples. Figure 1.6 shows these three types of microarray data clustering.

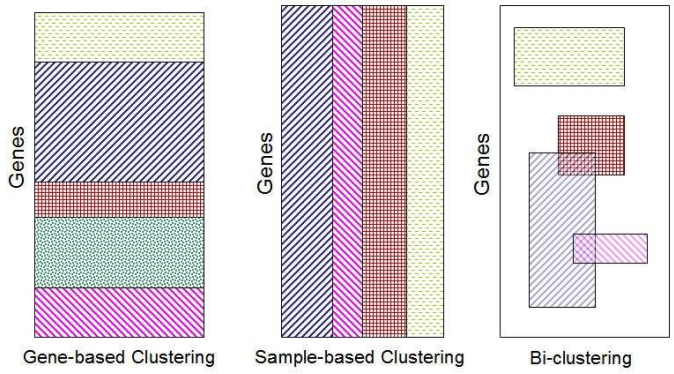


Figure 1.6: Gene-based clustering, Sample-based clustering and Biclustering.

Similarity Measure

Choosing a similarity measure is a critical step in clustering that depends on what we want to find or emphasize in the data. Two common similarity measures are:

1. **Euclidean distance** measures the "ordinary" distance between two points $x = \{x_1, x_2, ..., x_n\}$ and $y = \{y_1, y_2, ..., y_n\}$:

$$d_{euc}(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}, \tag{1.1}$$

   where n is the number of dimensions in the data vector.

2. **Pearson Linear Correlation** is much more common as the similarity measure in microarray data clustering, because it compares the overall shape of expression profiles rather than the actual magnitudes, *i.e.*, it considers genes similar when their values are "up" and "down" together.

   The *correlation* between two variables x and y is a number between -1 and +1 that measures the degree of their association. A positive value for the correlation implies

a positive association (large values of x tend to be associated with large values of y and small values of x tend to be associated with small values of y). A negative value for the correlation implies a negative or inverse association (large values of x tend to be associated with small values of y and vice versa). Figure 1.7 shows the scatter plot of positively correlated, uncorrelated and negatively correlated variables.

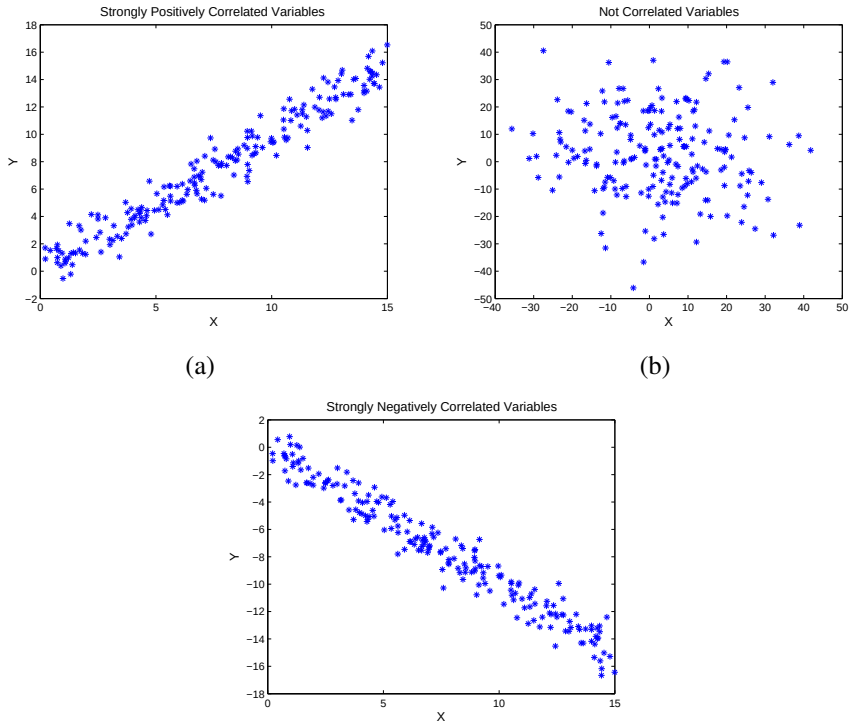


Figure 1.7: The scatter plot of (a) Strongly positively correlated variables, (b) Not correlated variables, (c) Strongly negatively correlated variables.

Chapter 2

Related Work

Microarray technology can simultaneously measure the expression levels of a large number of genes for a particular sample. The huge amount of biological information produced by these experiments has attracted many groups to analyze such data. Some approaches in microarray data analysis are only based on the data, *i.e.*, they usually do not use any prior biological information about the data. In fact these methods are applicable to any high dimensional data. Section 2.1 reviews some of these approaches. Other approaches use prior biological knowledge in microarray data analysis in order to improve the analysis results and also enhance the biological interpretability of the results. Section 2.2 reviews very briefly some of these methods.

2.1 "Large p, Small n" Problem

The main challenge of microarray data analysis is the high dimensionality of data. Any statistical analysis of gene expression data has to face the "large p, small n" problem, where p denotes the number of genes (features) and n denotes the number of instances (experiments); it has to avoid overfitting to construct a classifier with a good generalization ability. Most classical statistical approaches fail in such "large p, small n" problems, but with the advent of various types of high dimensional data there has been a dramatic growth in the development of statistical methodology in the analysis of such data.

Although there is no accurate grouping for these methods, we roughly divide them into two groups:

1. Dimensionality reduction methods: methods that involve feature selection or extraction ideas in order to reduce the feature space dimensionality. After this step, the selected (or extracted) feature set can be used for learning the classification or regression model.

2. Regularization-based methods: these methods take advantage of the data regularities in learning the classification or regression model. In these methods, selecting or extracting features is embedded in the learning process, *i.e.*, they do not have a separate feature selection (or extraction) step.

Kohavi *et al.* [28] find a similar grouping of feature selection algorithms, dividing these algorithms into filter methods versus wrapper methods. The essential difference is that a wrapper method makes use of the algorithm that will be used to build the final classifier, while a filter method does not. In our grouping, the first group roughly corresponds to filter methods, and the second group corresponds to wrapper methods.

2.1.1 Dimensionality Reduction Methods

One of the characteristics of high dimensional datasets is that, in many cases, not all the features are "important" for understanding the underlying phenomena of interest. Therefore, in many applications we need to use a dimensionality reduction technique to reduce the dimensionality of the original data prior to any modeling of the data [18].

Feature selection and feature extraction are two main methods for reducing dimensionality. In *feature selection*, we are interested in finding k of the p dimensions $(k < p)$ that give us the most information and we discard the remaining $(p - k)$ dimensions. In *feature extraction*, we are interested in finding a new set of k dimensions that are the combination of the original p dimensions [11].

Feature (gene) selection or extraction is also an important task in microarray data analysis, in order to reduce the dimensionality and to provide the most relevant set of features.

Feature Selection

The broadly used gene selection algorithms on microarray data share a common workflow [46]:

1. A single-gene based discriminative score is selected.
2. Genes are ranked based on such a discriminative score.
3. Top scored genes are then selected for further investigation.

Several ranking methods have been proposed to rank the genes in terms of their classification performance. Golub *et al.* [19] first introduced a ranking criterion for each gene in two-class classification. The criterion is defined as

$$\rho_j = \left| \frac{(\overline{x}_j^{(1)} - \overline{x}_j^{(2)})}{\sigma_j^{(1)} + \sigma_j^{(2)}} \right|, \tag{2.1}$$

where $\overline{x}_j^{(k)}$ and $\sigma_j^{(k)}$ denote the mean and standard deviation of the gene expression levels of gene j for all samples of class k.

Classical feature selection methods ranked each gene separately, but genes are well known to interact with each other. Hence, recent feature selection methods do not rank the genes independently; for example Xu *et al.* proposed a method that limits the correlation of the selected feature set in order to avoid selecting redundant genes [46].

Feature Extraction

The feature extraction methods expand the data in a new lower dimensional space in order to reduce the dimensionality of the data [39]. One of the best known and most widely used feature extraction methods is *Principal Components Analysis (PCA)*, which is a procedure that transforms a number of possibly correlated variables into a smaller number of

uncorrelated variables. PCA performs a linear mapping of the data to a lower dimensional space, selected to maximize the variance of the data in the low-dimensional representation.

Principal Components Analysis

Given a data set described by a set of numerical variables $\{X_1, X_2, ..., X_p\}$, the goal of PCA is to describe this data set with a smaller set of new variables. These variables will be linear combinations of the original variables, and are called the *Principal Components*. Reducing the number of variables used to describe data will lead to some loss of information; PCA minimizes this loss.

To clarify the main idea of PCA, let $\{X_1, ..., X_p\}$ be a set of real-valued random variables. We seek a derived variable $Z_1 = a_{11}X_1 + a_{21}X_2 + ... + a_{p1}X_p$ such that $var(Z_1)$[1] is maximized, where the a_{i1}'s are real-valued coefficients. We limit $a_1 = (a_{i1})$ to have unit Euclidean norm, $\|a_1\| = 1$. The derived variable Z_1, the largest principal component, attempts to capture the common variation in the variables X_i. If Z_1 is not enough to represent the original variables $\{X_1, ..., X_p\}$, we then look for a second derived variable, Z_2, uncorrelated with the first one, with the largest remaining variance, and so on.

In microarray data if we represent the expression array by $X_{ij} = [x_{ij}]$, where i indexes one of the p genes and j indexes one of the n instances, the largest *sample principle component* z_{1j} is defined to be the linear combination $z_{1j} = \sum_{i=1}^{p} a_{i1}x_{ij}, \|a_1\| = 1$, that has the largest instance variance.

Suppose that $C = [c_{ij}]$ is the $p \times p$ covariance matrix of the genes whose ijth entry is

$$c_{ij} = \frac{1}{n-1}\sum_{h=1}^{n}(x_{ih} - \overline{x}_i)(x_{jh} - \overline{x}_j), \tag{2.2}$$

where $\overline{x}_i$ and $\overline{x}_j$ are the sample means for genes i and j. The largest eigenvector of C

[1]For a random variable X with expected value $E(X) = \mu$, the variance of X is $var(X) = E[(X - \mu)^2]$.

defines the first principal component, the second largest eigenvector defines the second principal component, and so on.

The proposed *MkCoh* algorithm, which will be explained in Chapter 3, is similar to PCA, in that they both involve eigenvectors of the covariance matrix. However, *MkCoh* deals only with the first eigenvector (corresponding to the largest eigenvalue), and also discretizes this vector.

Gene Shaving

Cluster analysis is another important task in microarray data analysis. We described the main idea of clustering in Section 1.3.4. Gene shaving [20] is a clustering method based on PCA. It identifies subsets of genes with coherent[2] expression patterns that vary as much as possible across the samples. In other words, it extracts coherent clusters of genes where the average gene of each cluster has a large variance.

To clarify the main idea of gene shaving, let $X = [x_{ij}]$ be a $p \times n$ matrix of real-valued measurements, where the rows are genes and the columns are samples, and S_k is the indices of a cluster of k genes; the vector of n column averages of the expression values for this cluster is:

$$\overline{x}_{S_k} = \left(\frac{1}{k} \sum_{i \in S_k} x_{i1}, \frac{1}{k} \sum_{i \in S_k} x_{i2}, ..., \frac{1}{k} \sum_{i \in S_k} x_{in}, \right) \tag{2.3}$$

For each cluster of size k, gene shaving seeks a cluster S_k having the highest variance of the column averages:

$$S_k = \underset{S}{\operatorname{argmax}}\ var\{\overline{x}_S,\ |S| = k\} \tag{2.4}$$

This procedure generates a sequence of nested clusters S_k, in a top down manner, starting with $k = p$, the total number of genes, and decreasing down to $k = 1$. At each stage

[2] A coherent set is a set with a high value of average pair-wise correlation; Definition 1 and 2 in Chapter 3 define the pair-wise correlation and coherence.

it first computes the largest principal component of the current cluster of genes. Next, it computes the inner product (essentially the correlation) of each gene with the leading principal component, and discards ("shave off") a fraction (usually 10%) of the genes having the lowest (absolute) inner product. The process is repeated on the reduced cluster of genes. Algorithm 1 shows the main steps in the gene shaving algorithm.

Algorithm 1 Gene shaving algorithm (M, $X \in \Re^{p \times n}$)

1. Start with the entire expression matrix $X = [x_{ij}]$, each row centered to have zero mean.
$\mu_i = \frac{1}{n} \sum_{j=1}^{n} x_{i,j}$, $x_i := (x_i - \mu_i)$

2. Compute the leading principal component of the rows of X.

3. Discard ("shave off") the proportion α (typically $\alpha = 10\%$) of the genes having smallest absolute inner-product with the leading principal component.

4. Repeat steps 2 and 3 until only one gene remains.

5. This produces a nested sequence of gene clusters $S_p \supset S_k \supset S_{k_1} \supset S_{k_2} \supset ... \supset S_1$, where S_k is a cluster of k genes and $p > k > k_1 > k_2 > ... > 1$. Then estimate the optimal cluster size $\hat{k}$ using the gap statistic.

6. Orthognalize each row of X with respect to $\overline{x}_{S_{\hat{k}}}$, which is the average gene in S_k.

7. Repeat steps 1-6 with the orthognalized data, to find the second cluster. This process is continued until a maximum of M clusters are found, where M is chosen *a priori*.

The first 5 steps give the first cluster of genes. The gap test estimates the optimal cluster size k, by comparing the variances of each subset S_k in the shaving sequence to a similar sequence obtained from randomized data. Step 6 of this process orthogonalizes[3] the data to encourage discovery of a different (uncorrelated) second cluster.

The main drawback of the gene shaving process is that it requires repeated computations of the largest principal component of a large set of variables, which make it computa-

[3]In order to orthognalize gene x_i with respect to $\overline{x}_{S_{\hat{k}}}$, we must subtract the projection of x_i on to $\overline{x}_{S_{\hat{k}}}$ from x_i; $x_i^{new} = x_i^{old} - \frac{<x_i^{old}, \overline{x}_{S_{\hat{k}}}>}{<\overline{x}_{S_{\hat{k}}}, \overline{x}_{S_{\hat{k}}}>} \overline{x}_{S_{\hat{k}}}$ [42].

tionally expensive. Our *MkCoh* differs from gene shaving by considering highly-negative correlations between genes; in particular, *MkCoh* considers both negative and positive correlations between pairs of genes. By "flipping" the genes (Section 3.2), *MkCoh* can include genes that have highly negative correlations to the other included genes.

2.1.2 Regularization-based Methods

Regularization is a way to control the complexity of the model in order to avoid overfitting. There are different techniques to control the complexity. One famous method is to add a penalty term to the error function in order to discourage the coefficients from reaching large values. Therefore, in regularization-based methods, the total error function to be minimized is:

$$E(w) = E_D(w) + \lambda J(w) \tag{2.5}$$

where the coefficient λ controls the relative importance of the data-dependant error $E_D(w)$ and the regularization term $J(w)$. For example the regularization term can take the following form,

$$J(w) = \sum_{j=1}^{p} \|w_j\|^q \tag{2.6}$$

which is the L_q-norm of the parameter vector w (sum of the components, each to the power of q) [13].

Lasso is a well-known regularization-based method, mainly applicable in regression problems [21]. The error function used in Lasso is composed of squared error along with a regularization term that penalizes L_1-norm of coefficients of the linear model. For the given training set $X \in \mathbb{R}^{p \times n}$ and targets $Y \in \mathbb{R}^n$, the formulation of Lasso is

$$E(w) = \sum_{i=1}^{n} \left(y_i - w^T x_i\right)^2 + \lambda \|w\|_1 \tag{2.7}$$

where $\|w\|_1 = |w_1| + ... + |w_p|$.

A sufficiently large value of λ will make some of the coefficients w_j equal to zero, which leads to a *sparse* model. Because of this characteristic, Lasso is a commonly used approach in high dimensional data analysis [44].

The *SVM* (Support Vector Machine) [40] algorithm is another regularization-based method that penalizes the L_2 norm of w. Equation (2.8) is the error function used in classification SVM, which is composed of the hinge loss along with a regularization term that penalizes L_2-norm of coefficients of the linear model, $\|w\|_2^2 = w^T w = w_1^2 + ... + w_p^2$.

$$E(w) = \sum_{i=1}^{n} [1 - y_i f(x_i)]_+ + \lambda \|w\|_2^2, \tag{2.8}$$

where $[a]_+ = \max\{0, a\}$ and $f(x) = b + w^T x$ is the linear model fitted by SVM. The classification of a new instance x is made according to sign$(f(x))$.

SVM can be applied to both classification and regression problems. It has had many successful applications in microarray data analysis [29, 14].

As will be explained in Chapter 4, we applied several learning algorithms to the selected feature set to learn a predictor (model) that can predict some patient's characteristics. We usually achieved the best predictive performance by using Lasso or SVM.

2.2 Prior Biological Knowledge-based Methods

Some approaches use prior biological knowledge in microarray data analysis. Microarray experiments produce long lists of genes that are differentially expressed between two different situations. It is also known that the genes do not usually act alone in a biological system; they participate in a cascade of networks [15]. In order to better understand

the biology behind these data, it is relevant to include the available biological information of the genes [17]. We anticipate that incorporating biological knowledge into microarray data analysis will improve the results, as well as enhancing the interpretability of the results [33, 23].

Microarray data analysis methods can take advantage of pathways as the prior biological knowledge. Pathway information provides insights into the biological processes underlying microarray data. Combination of microarray data and pathway information may highlight the processes taking place in the cell and tissue and provide biological knowledge on the gene expression data. Pathway databases contain information mainly based on research performed with human and laboratory animals, moreover they are widely available in databases through the internet. There are three main sources of pathway and functional information, which can be either generic or species specific. The Gene Ontology project (GO) (http://www.geneontology.org) classifies genes into a hierarchy, placing gene products with similar functions together. The Kyoto Encyclopedia of Genes and Genomes (KEGG) (http://www.genome.jp/kegg/) provides searchable pathways and GenMAPP (http://www.genmapp.org) displays the gene expression on maps representing pathways and groupings of the genes [17].

Rapaport *et al.* [36] review some methods that use prior knowledge of the gene networks. They conclude that including prior knowledge of a gene network for the analysis of gene expression data leads to good classification performance and improved interpretability of the analysis results. The rest of this Chapter reviews some methods in microarray data analysis that use such prior biological knowledge.

One of the challenges in the medical research is to identify new prognostic markers that are more directly related to disease and that can more accurately predict the risk of disease in individual patients. Many studies have been done in this area to identify markers through analysis of gene expression profiles. Marker sets are usually selected by scoring

each individual gene for how well its expression pattern can discriminate between different classes of disease. But a more effective way for marker identification may be to combine gene expression measurements over groups of genes that fall within common pathways. Chuang *et al.* [16] applied prior protein-network-based knowledge to identify new markers in breast cancer. They identify markers not as individual genes but as subnetworks extracted from protein interaction databases. To integrate the expression and network data sets, they overlaid the expression values of each gene on its corresponding protein in the network and searched for subnetworks whose activities across the patients were highly discriminative of metastasis. This process involves several scoring and search steps. The resulting subnetworks provide new hypotheses for pathways involved in tumor progression. They have also achieved higher accuracy in the classification of metastatic versus non-metastatic tumors.

Gene clustering is another important task in microarray data analysis. Kennedy *et al.* proposed a method of gene clustering using information from the Gene Ontology [27]. The method uses this information to build clusters and also to extract meaningful cluster descriptions. The proposed methodology first reduces the number of genes coming from the microarray experiments to dozens of genes. The output of this stage is interesting from a statistical point of view, however it is difficult for biological interpretation. The Gene Ontology has been used to assist in the interpretation of the output. The list of genes is reclustered into groups of genes with similar biological functions. The broad goal of this work is to improve the understanding of genes related to a specific form of childhood cancer.

There are many articles concerned with clustering genes for gene function discovery using microarray gene expression data. Because co-expressed genes are likely to have the same biological function or be involved in the same biological process, clustering genes' expression profiles could provide a means for gene function discovery. Most existing

approaches ignore known functions of some genes, but some recent methods use a prior biological knowledge of known gene functions in the process of clustering. To discover new gene functions, they use biological pathways or gene functional annotation systems, such as Gene Ontology or KEGG [23, 33].

Estimating a gene network is another important topic in the field of bioinformatics. Several methodologies have been proposed for constructing a gene network based on gene expression data. But microarray data do not contain enough information for constructing gene networks accurately in many cases, therefore there are usually some drawbacks for the gene network construction using only microarray data. Recent methods have applied prior biological knowledge to get better results. Imoto *et al.* have proposed a statistical method for estimating a gene network based on Bayesian networks from microarray gene expression data together with biological knowledge [24].

In this thesis, we used clusters of genes with a common function, as prior biological knowledge. Our colleagues produced such clusters of genes, called PBTs, for mouse and human, which includes 38 human PBTs and 21 mouse PBTs. They expect the genes in each PBT to be highly correlated; but while mouse PBTs are coherent, human PBTs were not. Hence, we proposed a method, which will be explained in Chapter 3, to improve the coherency of the genes in each PBT. We hoped we could use each cluster (rather than each gene) as a single feature, because when a gene set is coherent, we can view all the genes in that set as a single entity. We expected that the predictors based on the PBTs should be more accurate than the ones based on the original gene expression values. Chapter 4 presents the results.

Over the last few years, many articles have been published dealing with high dimensional data [18, 22]. In microarray data analysis, as an instance of high dimensional data, some methods just use the data, but some others apply prior biological knowledge to improve the results and enhance the interpretability of the results. However, generally high

dimensional data and particularly microarray data analysis is still an active research area.

Chapter 3

Making Gene Sets More Coherent

3.1 The Coherence Task Problem

The prior knowledge of gene properties could help biologists to form clusters of genes with a common function. As a possible approach, we used these clusters to reduce the dimensionality of data. We hoped we could use each cluster (rather than each gene) as a single feature. Our colleagues have produced such clusters of genes, called PBTs, for mouse and human. They have produced 38 human PBTs and 21 mouse PBTs. Human PBTs range in size from 3 to 1798 genes, with an average size of 318.42 $\pm$ 447.82. The size of mouse PBTs range from 4 to 924 genes with an average size of 214.38 $\pm$ 243.78. Some PBTs overlap, *i.e.*, there are some genes included in more than one PBT. The medical researchers initially characterized mouse PBTs as "herd movements" in mouse transplant data, *i.e.*, mouse PBTs are "coherent" or highly correlated. Given that many human PBTs are constructed to resemble corresponding mouse PBTs, they expect to see the same coherence for human PBTs as well. In order to explain the problem more clearly, we need to define the *pair-wise correlation* and *coherence* more accurately.

In general, we can characterize each patient by a vector of m gene expression values (*e.g.*, of a biopsy taken of that patient's transplanted kidney), one for each gene in G; for ATI data $m = |G| = 54675$ (as we are using Affymetrix GeneChip® human Genome U133 Plus 2.0 Array). The ATI_{GFR} dataset includes $n = |P| = 137$ such patients. Here $P = \{p_i\}_{i=1,\ldots,n}$ is a vector set of patients where $p_i \in \Re^m$ and $G = \{g_i\}_{i=1,\ldots,m}$ is a vector set of genes where $g_i \in \Re^n$.

We let $e_{i,j} = \text{expression}(p_i, g_j) \in \Re$ refer to the expression of the j^{th} gene of the i^{th} patient, and for each gene g_j, let $\mu_j = \frac{1}{|P|} \sum_i e_{i,j}$ refer to the average expression value for that gene over all patients P.

Definition 1. Pair-wise Pearson Correlation

For any two genes, we define the pair-wise correlation matrix (w.r.t. $E = [e_{i,j}]$) as

$$\text{corr}(\, g_j,\, g_k \,) \quad = \quad \frac{\sum_i (e_{i,j} - \mu_j)(e_{i,k} - \mu_k)}{\sqrt{(\sum_i (e_{i,j} - \mu_j))(\sum_i (e_{i,k} - \mu_k))}} \tag{3.1}$$

where the summations are over the patients $p_i \in P$.

Definition 2. Coherence

We then define the "coherence" of any subset of genes $F \subset G$ as the "average pair-wise correlation"

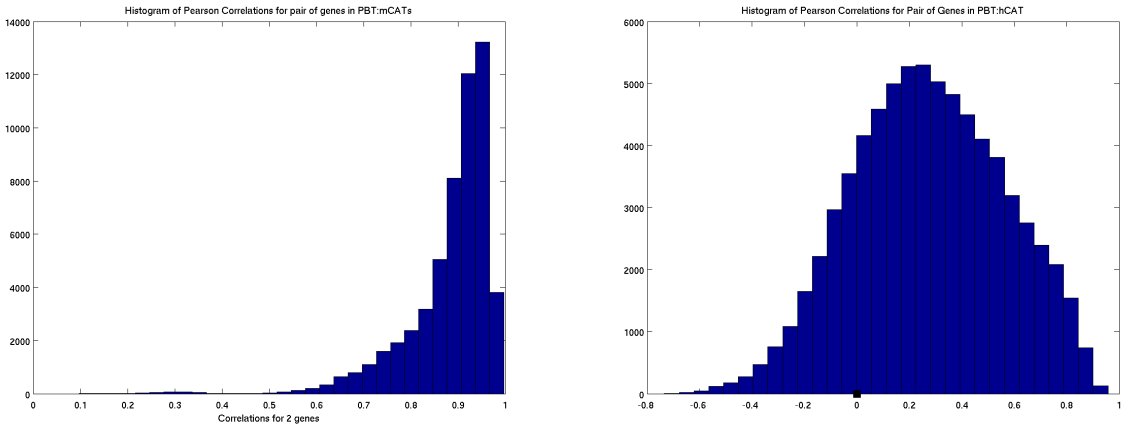
$$\text{coh}(\, F \,) \quad = \quad \frac{1}{|F|} \sum_{g_j, g_k \in F} \text{corr}(\, g_j,\, g_k \,) \tag{3.2}$$

Notice this average is per gene (and not per gene-pair).

To test the coherence assumption in mouse and human data, we computed all $\binom{n}{2}$ pair-wise Pearson correlation values over each n-gene PBT. We say a PBT is coherent if essentially all of the gene pair-wise correlation values will be large — near 1. We can also allow some genes to be strongly *negatively* correlated — *i.e.*, whenever gene g_j goes up, gene g_k goes down, and vice versa. This would mean correlation (g_j, g_k) is close to -1.

Figure 3.1(a) is a histogram of the $\binom{332}{2}$ pair-wise correlations of the 332 genes in the mouse PBT, "mCAT", and Figure 3.1(b) is a histogram of the pair-wise correlations of the 382 genes in the corresponding human "hCAT" PBT. We see that mCAT PBT is coherent, but this is not the case for hCAT PBT. Most of the mCAT genes' pair-wise correlations are close to 1, but most of the hCAT genes' pair-wise correlations are close to 0; *i.e.*, unlike mCAT genes, most pairs of genes in the human hCAT PBT are essentially uncorrelated, and very few are either strongly positively correlated, or strongly negatively correlated.

mCat PBT is not the only coherent mouse PBT— most of the mouse PBTs are coherent; the webpage [1, `ATI/Mouse_ATI`] presents histograms for all 21 mouse PBTs and the



(a) (b)

Figure 3.1: Histogram of the pair-wise correlations of the genes in (a) mouse mCAT PBT (332 genes), (b) corresponding human hCAT PBT (382 genes).

webpage [1, `ATI/Mouse_Human_PBTs`] presents histograms for all 38 human PBTs and the 17 mouse PBTs (including ones that corresponds to some human PBTs). Figure 3.2 compares the histogram of coherence over all mouse PBTs versus the histogram of coherence over all human PBTs. Figure 3.3 shows the histograms of the pair-wise correlations of the high variance subsets of the genes[1] in human and mouse data.

Considering all these issues, we can conclude that unlike the mouse PBTs, most human PBTs are not coherent. This has motivated us to find ways to modify human PBTs to be more coherent. To achieve this goal we proposed the *MkCoh* algorithm that will be explained in Section 3.2. Section 3.3 describes the results of using *MkCoh* algortihm on PBTs to make them more coherent.

[1]The high variance subset of the genes includes only those genes whose variance over the patients is more than a threshold.

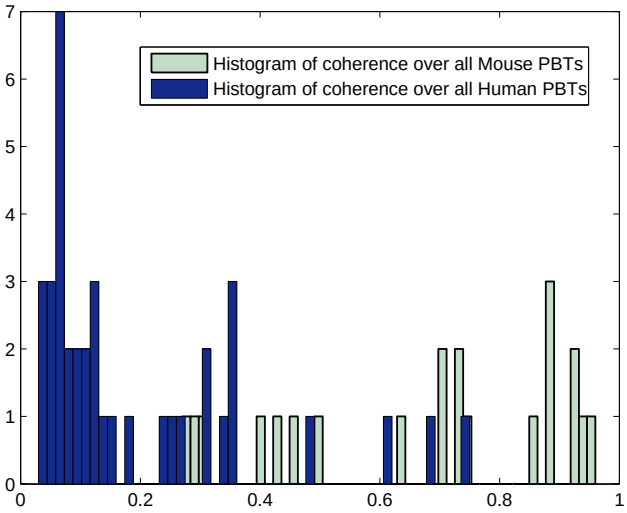


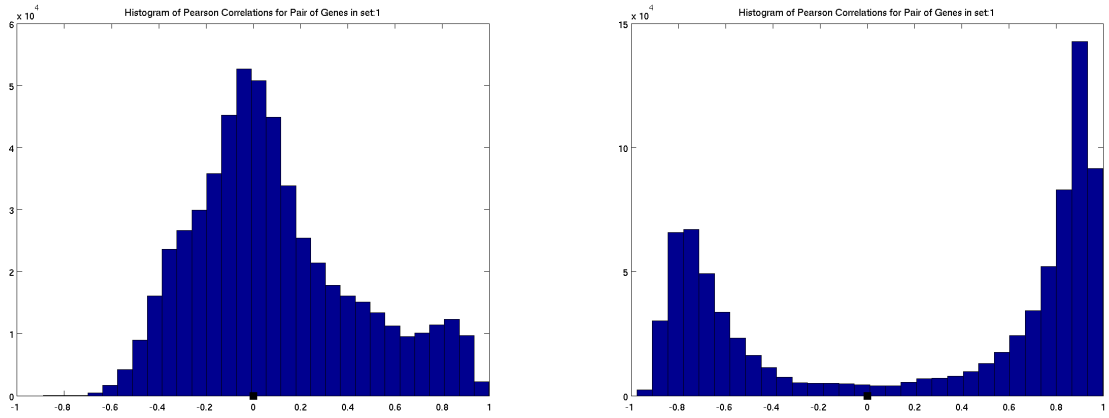
Figure 3.2: Histogram of coherence (average pair-wise correlation) over all PBTs; the lighter histogram is for mouse PBTs and the darker one is for human PBTs.

3.2 The *MkCoh* Algorithm

This section describes the *MkCoh* algorithm, which modifies a set of genes in a way to produce a set that is more coherent. Basically this process involves "removing" the genes that tend to be uncorrelated with many remaining genes, and "flipping" the genes that tend to be negatively correlated.

In general, given a gene set F, with associated correlation values $corr(\cdot, \cdot)$, we seek a modified gene set, $\mathrm{F}^{(mc)}$ with improved coherence; here $\mathrm{F}^{(mc)}$ can differ from F by (1) removing some genes and (2) "flipping" some other genes. We use "flipping" to change a gene that is negatively correlated to one that is positively correlated — in particular, flipping gene g_j means replacing g_j with g'_j, where

$$expression(p_i, g'_j) \quad := \quad -expression(p_i, g_j)$$



(a) (b)

Figure 3.3: Histogram of the pair-wise correlations of the genes in (a) human high variance subset (1292 genes), (b) mouse high variance subset (1050 genes).

over all patients p_i, which in turn means

$$corr(\, g'_j,\, g_k\,) \quad := \quad -corr(\, g_j,\, g_k\,)$$

Hence, if g_1 is negatively correlated with g_2, then g'_1 is positively correlated with g_2; if g_1 was anticorrelated with all the other genes in a PBT, then we would increase the PBT's coherence by flipping that g_1. Now imagine g_2 is *positively* correlated with g_7; here the new g'_2 would be negatively correlated; if g_2 is positively correlated with most of the PBT's genes, we probably should not flip it. If g_3 is on the cusp — with perhaps slightly more genes positively correlated than negatively correlated — we may not want to flip it. Of course, if we decide to flip g_1, this may tip the balance for g_3, encouraging us to flip g_3 as well. This in turn may have an effect of many other genes, suggesting that they be flipped as well. Given all of these flips, we may then decide not to flip g_1 after all. (Notice that flipping two genes leaves their correlation the same: $corr(\, g'_j,\, g'_k\,) = corr(\, g_j,\, g_k\,)$.)

It is therefore complicated to decide which genes to flip. Of course, this has been

ignoring the issues of removing some genes from the PBT; this too can affect the flip/no-flip decision.

We define the *coherence task* as the process of finding the most *coherent* subset of genes (based on flipping or removing). Using our definition of "coherence" in Equation (3.2), the most *coherent* subset is the subset of genes with the "maximum average pair-wise correlation". We can get this subset by removing some genes and flipping some other genes. This process can be formalized as the following definition:

Definition 3. Coherence Task
Given an $m \times m$ matrix $C = (c_{jk})$ where each $c_{jk} \in [-1, +1]$ is the correlation between two elements, compute the m-ary vector $x = \langle x_1, \ldots, x_m \rangle \in \{-1, 0, +1\}^m$ that maximizes coherence (Equation (3.2)), which here corresponds to

$$\left(\sum_{jk} c_{jk} x_j x_k \right) \; / \; \sum_{j} |x_j| \tag{3.3}$$

Of course, $x_j = 0$ means removing gene g_j, and $x_j = -1$ means flipping this gene. The denominator is just a way to compute the cardinality of the surviving set of genes (whether positive or negative). Recall this is "per gene", rather than "per gene pair".[2]

As $x_j \in \{-1, 0, +1\}$, notice $|x_j| = x_j^2$. This means we can rewrite Equation (3.3) as

$$\underset{x \in \{-1, 0, +1\}^n}{\operatorname{argmax}} \quad \frac{x^T C x}{x^T x} \; . \tag{3.4}$$

While this combinatorial problem is difficult to solve for *discrete* values of x_j, its continuous form is well known. That is, if we replace the constraint $x_j \in \{-1, 0, +1\}$ with

[2]This "average per-gene-pair" score $\frac{1}{|F|^2} \sum_{g_j, g_k \in F} corr(\, g_j, \, g_k \,)$ — which uses $\left(\sum_j |x_j| \right)^2$ in the denominator — is not appropriate as it removes too many genes. Indeed, there is a degenerate way to optimize it: just identify the largest single $(j^*, k^*) = \operatorname{argmax}_{j,k} |corr(\, g_j, \, g_k \,)|$; then set $b_{j^*} = 1$ and $b_{k^*} = \text{sign}(corr(\, g_j, \, g_k \,))$, and $b_i = 0$ for all other $i \neq j^*, k^*$.

$x_j \in [-1, +1]$, then the optimal x is simply the eigenvector with the largest eigenvalue.[3]

Given this vector of real valued x, the only remaining challenge is discretizing it: mapping each real value x_j to $\{-1, 0, +1\}$. Here, we find two break-points $\alpha, \beta \in [-1, +1]$ such that

$$x_j^{\delta} = \textit{discrete}_{\alpha,\beta}(\, x_j \,) \quad = \quad \begin{cases} -1 & \text{if} \quad x_j < \alpha \\ 0 & \text{if } \alpha < x_j < \beta \\ +1 & \text{if } \beta < x_j \end{cases}$$

We compute the α, β values that produce the discrete vector $x^{\delta^*} = \textit{discrete}_{\alpha,\beta}(\, x \,)$ that optimizes Equation (3.4) over all α and β values. This can be done by first sorting the elements of the initial x vector, and then considering all $\binom{m+2}{2}$ possible positions for $\alpha < \beta$, and evaluating Equation (3.4) for each. Figure 3.4 shows the histogram of sorted values of principle eigenvector for CECAT PBT; *MkCoh* algorithm has found α and β as thresholds to discretize F values.

We call the resulting algorithm *MkCoh*: Given a correlation matrix C_F over a set of genes F, *MkCoh*(C_F) returns the x^{δ^*} vector described above, specifying which genes to remove and which to flip, where $x_j^{\delta^*} = 0$ means remove gene g_j , and $x_j^{\delta^*} = -1$ means flip this gene. We will use the "$^{(mc)}$" superscript to indicate the modified PBT; hence "CECAT$^{(mc)}$ = *MkCoh*(CECAT)".

The only other issue deals with the observation that there is a "bit" of freedom for each eigenvector: that is, if x is an eigenvector, then so is $-x$. To simplify our explanations, we define the eigenvector as one whose majority of elements are positive.

Given a gene set F including N genes, the following steps are required to get $F^{(mc)}$:

1. Calculate the $N \times N$ pair-wise correlation matrix $C = [c_{jk}]$ using Equation (3.1). Here c_{jk} is the correlation between g_j and g_k.

[3]The standard formulation, requiring $x^T x = \sum_j x_j^2 = 1$, is sufficient to insure that $x_j \in [-1, +1]$. Also, as this C is a covariance matrix, we know it is positive semidefinite [40], which means its eigenvalues are non-negative.

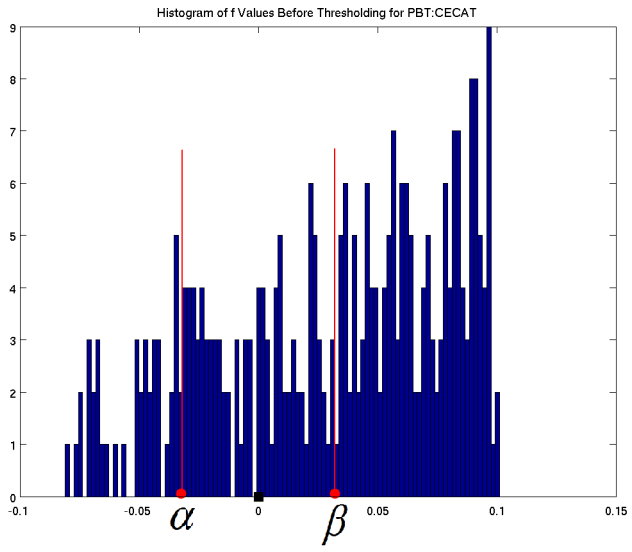


Figure 3.4: Histogram of sorted values of principle eigenvector for CECAT PBT, with optimal break-points α and β.

2. Find the first eigenvector of C, which is a real valued vector $x = (x_1, ..., x_N)$ with length N, $x_j \in [-1, +1]$. Sort the values of x in ascending order.

3. Compute two break-points $\alpha, \beta \in [-1, +1]$ to produce the discrete vector $x^{\delta^*} = \textit{discrete}_{\alpha,\beta}(\, x\,)$ that produces the smallest value of Equation (3.4) w.r.t. α and β.

4. To obtain $\mathrm{F}^{(mc)}$ for all the genes $g_{1 \le i \le m}$, remove g_i if $x_i^{\delta^*}$ is 0, flip g_i if $x_i^{\delta^*}$ is -1 and keep it if $x_i^{\delta^*}$ is 1.

Algorithm 2 describes *MkCoh* algorithm in more details. Although, there is a more efficient $O(n^2)$ algorithm, we show a simpler $O(n^3)$ version that is easier to understand. It takes a correlation matrix C_F over a set of genes F as the input and returns the x^{δ^*} vector that specifies which genes to remove and which to flip.

Algorithm 2 MkCoh Algorithm (C_F)

$x_{(1,\ldots,N)} \leftarrow \text{argmax}_{u:||u||=1}\, u^T C u$

$x \leftarrow$ sort x in ascending order

$x_0 \leftarrow x_1 - 1$
$x_{N+1} \leftarrow x_N + 1$

$V_{max} \leftarrow -\infty$

for $i = 0$ to N **do**
  $\alpha \leftarrow (x_i + x_{i+1})/2$
  for $j = i$ to N **do**
    $\beta \leftarrow (x_j + x_{j+1})/2$
    for $k = 1$ to N **do**

$$x_k^\delta = \begin{cases} -1 & \text{if} \quad x_k < \alpha \\ 0 & \text{if } \alpha < x_k < \beta \\ +1 & \text{if } \beta < x_k \end{cases}$$

    end for
    $V = \frac{{x^\delta}^T C x^\delta}{{x^\delta}^T x^\delta}$

    if $V > V_{max}$ **then**
      $V_{max} \leftarrow V$;
      $x^{\delta^*} \leftarrow x^\delta$;
    end if;
  end for
end for
return x^{δ^*}

3.3 Making Gene Sets More Coherent

The difference between the coherence property of mouse PBTs and human PBTs, explained in Section 3.1, motivated us to propose the *MkCoh* algorithm, which addresses the challenge of modifying each PBT to be more coherent. Section 3.2 defined this task more precisely. This section applies *MkCoh* algorithm on PBTs and compares PBT with PBT$^{(mc)}$ in terms of coherency.

We ran the *MkCoh* algorithm on all 38 PBTs, which produced more coherent gene sets PBT$^{(mc)}$s. Figure 3.5 presents the scatter plot of the average correlation of all 38 human PBTs and PBT$^{(mc)}$s; each (x,y) point shows one of the PBTs. It also includes the "x=y" line to show how much PBTs' average correlations are different from PBT$^{(mc)}$s' average correlations. We see that, except for the 3 points that are on the "x=y" line, all the others are above the line; therefore *MkCoh* algorithm has strictly increased the coherence for almost all the sets. Figure 3.6(a) is the histogram of CECAT PBT and Figure 3.6(b) is the histogram of CECAT$^{(mc)}$. Most of the histogram values of CECAT are around 0 (the mean value is 0.08). We can see that the average correlation of CECAT$^{(mc)}$ has improved notably (the mean value is 0.34). To get CECAT$^{(mc)}$, *MkCoh* algorithm removed 99 and flipped 41 genes of *CECAT* PBT. The webpage [1, `ATI`] presents histograms for all 38 human PBTs and PBT$^{(mc)}$s.

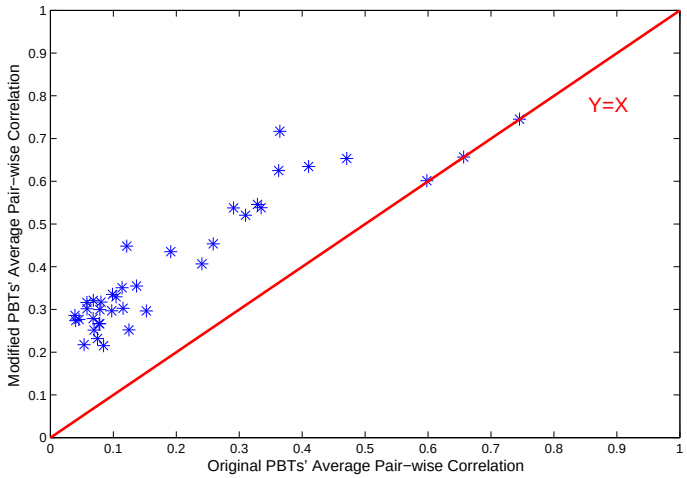
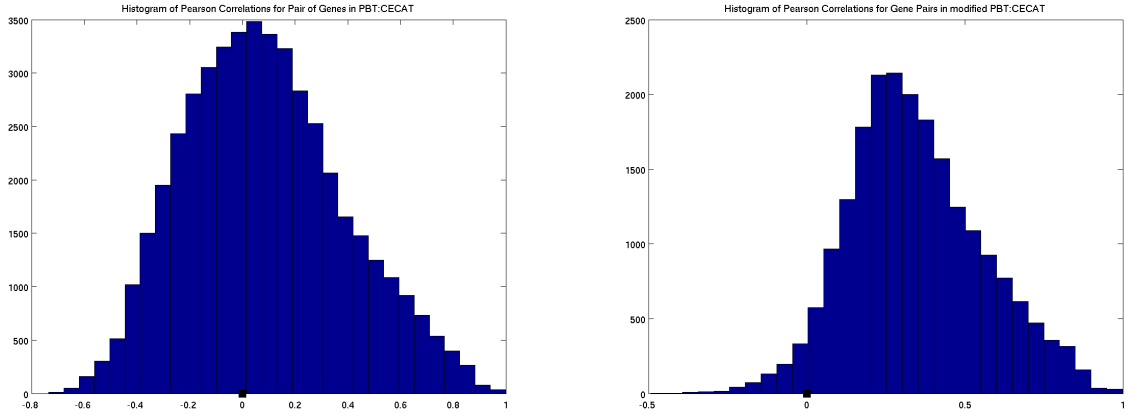


Figure 3.5: Scatter plot showing average correlations over the 38 original PBTs versus the PBT$^{(mc)}$s.



(a) (b)

Figure 3.6: Histogram of Pearson correlations for pair of genes in (a) CECAT, (b) CECAT$^{(mc)}$.

Chapter 4

Datasets and Results

We used PBTs and PBT$^{(mc)}$s as prior biological knowledge to reduce the dimensionality of microarray data. We hoped we could use each cluster (rather than each gene) as a single feature, because when a gene set is coherent, we can view all the genes in that set as a single entity, by collapsing the expression values of all the genes into a simple summary statistic (*e.g.*, the arithmetic mean). This process will reduce the dimensionality of microarray data from 50K to less than 50, while we hope we still keep the relevant information.

We saw in Chapter 3 that human PBTs are not coherent. However, applying *MkCoh* algorithm on human PBTs improved their coherency — almost all the resulting PBT$^{(mc)}$s were more coherent than PBTs. Using coherent PBT$^{(mc)}$s, we can collapse the expression values of all the genes into a simple summary statistic and get 38 values for each patient. We expected these 38 values to be a "concise" description of all the microarray data. But that would be true only if PBTs include all the relevant information of microarray data for the prediction task. Based on what medical researchers believed, we assumed that PBTs include all the significant genes for the prediction task.

4.1 Feature Sets

To validate the quality of PBTs and PBT$^{(mc)}$s, we used an objective measure: predictive accuracy, on some biologically important tasks. We expected PBTs to be more effective than all the microarray genes, and PBT$^{(mc)}$s to be more effective than the original PBTs as the features in the prediction task.

We explored ways to use PBTs and PBT$^{(mc)}$ to estimate several biological functions and compared the prediction results of using this prior knowledge with the results that used other possible sets of genes as the features. We sought ways to use the *set* of all 38 PBTs. One challenge here is finding a simple quantity to encode the expression values of all genes within each PBT, for each patient p. As the gene expression values varied

considerably for the different genes, we used the arithmetic mean:

$$a(p, F) = \frac{1}{|F|} \sum_{g_j \in F} e_{i,j} \tag{4.1}$$

where $e_{p_i,g_j} = \text{expression}(p_i, g_j)$ is expression of gene g_j for patient p_i[1]. We can then identify each patient p with the 38 values $\{a(p, F)\}$ over the PBTs $\{F_1, \ldots, F_{38}\}$, and attempt to build a regressor or classifier over these values.

Some medical researchers believed that we will remove the effects of significant genes in a PBT by using arithmetic mean of its genes, therefore we used the union of all PBTs' genes and also the union of all PBT$^{(mc)}$s' genes as two other possible feature sets.

To compare the results, we considered using all the microarray genes and also high variance subset of genes to estimate the objective function or predict the class labels. Also, we applied K-means clustering algorithm to find 100 clusters using correlation as the distance measure. We then averaged the value of genes in each cluster and used it as the feature associated with that cluster.

We also used 27 other "clinical features" of each patient — including age, gender, etc. Table 4.1 shows almost all the sets that we used as the features; we used these sets with and without clinical features in classification or regression tasks.

To verify the effectiveness of PBTs and PBT$^{(mc)}$s in a prediction task, we used several datasets. Table 4.2 describes these datasets, which includes both classification and regression tasks. We used Weka [9] software to apply SMO and Decision Tree Classifier (J48) to the classification problems, and SMOreg and Decision Tree Regressor (M5P) to the regression problems. We also applied Lasso to both of them. This project has been

[1]Note we also considered various other summary statistics, including log of geometric mean $\frac{1}{|F|} \sum_{g_j \in F} \log e_{i,j}$, average z-score $\frac{1}{|F|} \sum_{g_j \in F} \hat{e}_{i,j}$ where $\hat{e}_{i,j} = (e_{i,j} - \mu_g j)/\sigma_{gj}$ is the "z-score" and average absolute z-score $\hat{z}(p, F) = \frac{1}{|F|} \sum_{g_j \in F} |\hat{e}_{i,j}|$ where the mean is $\mu_{gj} = \frac{1}{|P|} \sum_p e_{i,j}$ and the variance is $\sigma^2_{gj} = \frac{1}{|P|} \sum_p (e_{i,j} - \mu_{gj})^2$; but the results were almost the same as the results of simple arithmetic mean.

Table 4.1: The following sets with/ without clinical features have been used as features for classification or regression tasks.

Feature Set	Description
"PBTs' AVG"	Summary statistics from all of the PBTs
"PBT$^{(mc)}$s' AVG"	Summary statistics from all of the PBT$^{(mc)}$s
"PBTs' genes"	Union of all the genes in all 38 PBTs
"PBT$^{(mc)}$s' genes"	Union of all the genes in all 38 PBT$^{(mc)}$s
"IQR filtered genes"	IQR filtered subset of genes
"IQR filtered PBTs"	IQR filtered subset of PBTs' genes
"IQR filtered PBT$^{(mc)}$s"	IQR filtered subset of PBT$^{(mc)}$s' genes
"All genes"	All gene expression values in the microarray
"HighVar(HV) genes"	High variance subset of the genes
"K-means clusters' AVG"	Summary statistics from 100 clusters found by k-means algorithm

Table 4.2: Datasets

Dataset	To Predict	Task	Size
ATI_{Lesion}	Lesion Values	Regression	$173 \times 54K$
ATI_{GFR}	GFR Scores Reject/No Reject	Regression Classification	$137 \times 54K$
BREAD	Relapse/ No Relapse	Classification	$132 \times 24K$
Breast cancer (Van't Veer et al., 2002)	Relapse/ No Relapse	Classification	$95 \times 14K$

done in collaboration with Nasimeh Asgarian[2], therefore some results are available on her webpage.

4.2 Results on ATI_{Lesion} Dataset

We had two main versions of ATI datasets, ATI_{Lesion} and ATI_{GFR}. ATI_{Lesion} includes 54K genes, and the values of the 12 lesions over 173 unique patients. Here, the goal is to

[2] `http://cs.ualberta.ca/~nasimeh/`

Table 4.3: The RMSE of Lasso in predicting Lesions using all PBTs' genes and all PBT$^{(mc)}$s' genes with and without clinical features, all the microarray genes and subset of high variance genes with clinical features

Lesion	Baseline	PBTs genes	PBT$^{(mc)}$s genes	PBTs genes+ Clinical	PBT$^{(mc)}$s genes+ Clinical	All genes	HV genes	K-means clusters' AVG
g	0.66	0.59	0.58	0.60	0.59	0.56	**0.55**	0.64
cg	0.87	0.81	0.78	0.77	0.76	0.77	**0.71**	0.83
i	0.92	0.64	**0.63**	0.64	**0.63**	0.68	0.68	0.66
ci	0.92	0.74	0.75	0.72	0.73	0.67	**0.61**	0.68
t	0.91	0.72	0.72	0.71	0.71	0.70	**0.69**	0.77
ct	0.86	0.70	0.71	0.67	0.69	0.62	**0.60**	0.67
v	0.41	**0.40**	**0.40**	**0.40**	**0.40**	0.41	**0.40**	0.41
cv	0.94	0.94	0.93	0.90	0.90	0.91	**0.84**	0.94
ah	1.06	1.01	0.97	0.88	**0.87**	0.99	0.99	1.03
mm	0.87	0.80	0.78	**0.77**	**0.77**	0.83	0.78	0.80
ptc	0.90	**0.79**	0.81	0.81	0.82	**0.79**	0.81	0.86
ptcml	3.19	3.04	3.02	3.00	**2.97**	3.32	3.20	3.49

predict twelve histologic lesions[3] called 'g', 'cg', 'i', 'ci', 't', 'ct', 'v', 'cv', 'ah', 'mm', 'ptc' and 'ptcml'. For each patient, each of the first eleven lesions has the value of {0, 1, 2, 3} and 'ptcml' has the value of {0,1,2}. Table 4.3 shows the first part of our results on the ATI_{Lesion} dataset. It compares the RMSE[4] of baseline[5] with the RMSE of Lasso using all PBTs' genes and all PBT$^{(mc)}$s' genes with and without clinical data as the features in predicting the lesions' values; here the RMSEs are based on leave-one-out cross validation. The bold numbers are the lowest RMSEs for each lesion.

[3]See Appendix A for more information about ATI dataset and also histologic lesions

[4]The RMSE (Root Mean Squared Error) measures the differences between predicted values by a model and the true values. $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(x_i) - t_i)^2}$ where $f(x_i)$ is the predicted value by the model f for sample case x_i; t_i is the true value for sample case x_i and n is the total number of samples.

[5]For regression problems the baseline is predicting the average value of targets; and for classification problems the baseline is predicting the majority of class labels.

Table 4.4 shows the next part of the results on the ATI_{Lesion} dataset. It compares the RMSE of Decision Tree (M5P) with the RMSE of SMOreg when they are using all PBTs' genes, all PBT$^{(mc)}$s' genes, the arithmetic average of PBTs and the arithmetic average of PBT$^{(mc)}$ as the features in predicting lesions' values; here the RMSEs are based on 10-fold cross validation. Table 4.5 shows our best results for each lesion by comparing the best results of Tables 4.3 and 4.4. The accuracy of prediction was good for each of the lesions. This table shows that Lasso regressor produced the best results.

For all the lesions, Figures 4.1 and 4.2 show the plot of true values versus the predicted values for each patient.

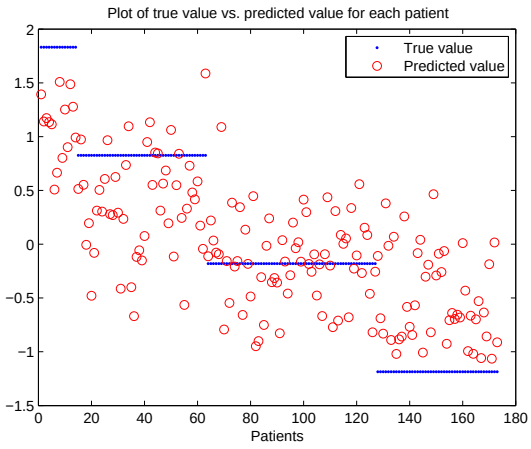
Our medical researchers asked us to each of the lesion values as binary: for lesions 'g', 'cg', 'v', 'cv', 'mm' and 'ptc', distinguish the value of $\{0\}$ versus the values of $\{1,2,3\}$ and for lesions 'i', 'ci', 't', 'ct', 'ah' and 'ptcml', distinguish the values of $\{0,1\}$ versus the values of $\{2,3\}$. We achieved good predictive performance for most lesions using Lasso on all the genes. The webpages [6, `ATI/Lesions/Name/table.html`] and [6, `ATI/Lesions/Names_Geo/TableGeo.html`] show the results.

Table 4.4: The RMSE values on lesion prediction using PBTs' genes, PBT$^{(mc)}$s' genes, PBTs' AVG and PBT$^{(mc)}$s' AVG as the feature sets. The number between parentheses shows the size of the feature set and the bold number is the lowest RMSE for that lesion.

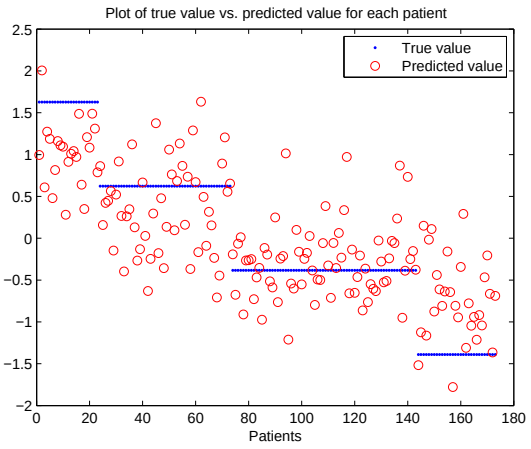
Lesion	Baseline	Regressor	PBTs' genes (7280)	PBT$^{(mc)}$s' genes (5353)	PBTs' AVG (38)	PBT$^{(mc)}$s' AVG (38)
g	0.66	M5P	0.71	0.73	0.62	**0.57**
		SMOreg	0.60	0.61	0.70	0.70
cg	0.87	M5P	0.89	0.94	0.87	0.86
		SMOreg	**0.79**	0.81	0.95	0.94
i	0.92	M5P	0.77	0.74	0.68	0.68
		SMOreg	0.68	0.69	0.68	**0.66**
ci	0.92	M5P	0.79	0.78	0.76	0.78
		SMOreg	**0.71**	**0.71**	0.78	0.80
t	0.91	M5P	0.82	0.82	0.80	0.77
		SMOreg	0.73	**0.72**	0.79	0.76
ct	0.86	M5P	0.75	0.73	0.73	0.73
		SMOreg	0.69	**0.68**	0.73	0.76
v	**0.41**	M5P	0.49	0.48	0.46	0.47
		SMOreg	0.46	0.45	0.45	0.45
cv	**0.94**	M5P	1.16	1.09	1.11	1.07
		SMOreg	1.08	1.09	1.10	1.12
ah	1.06	M5P	1.27	1.14	1.05	1.03
		SMOreg	**0.97**	0.98	1.05	1.02
mm	0.87	M5P	0.92	0.88	0.82	0.83
		SMOreg	0.80	0.81	0.81	**0.77**
ptc	0.9	M5P	0.85	0.84	0.92	**0.83**
		SMOreg	0.95	0.96	0.96	0.96
ptcml	3.19	M5P	3.77	3.59	3.63	3.62
		SMOreg	3.44	**3.30**	3.48	3.32

Table 4.5: The best results for each lesion

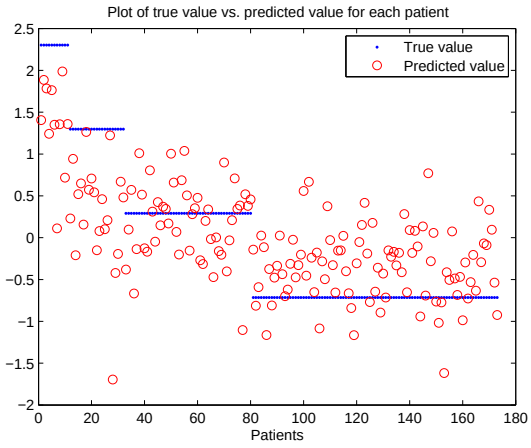
Lesion	Baseline	Lowest RMSE	Feature set	Regressor
g	0.66	0.55	HV genes	Lasso
cg	0.87	0.71	HV genes	Lasso
i	0.92	0.63	PBT$^{(mc)}$s	Lasso
ci	0.92	0.61	HV genes	Lasso
t	0.91	0.69	HV genes	Lasso
ct	0.86	0.60	HV genes	Lasso
v	0.41	0.40	PBTs	Lasso
			PBT$^{(mc)}$s	Lasso
			HV genes	Lasso
cv	0.94	0.84	HV genes	Lasso
ah	1.06	0.87	PBT$^{(mc)}$s+Clinical	Lasso
mm	0.87	0.77	PBTs+Clinical	Lasso
			PBT$^{(mc)}$s+Clinical	Lasso
			PBT$^{(mc)}$s AVG	SMOreg
ptc	0.9	0.79	Allgenes	Lasso
			PBTs+Clinical	Lasso
ptcml	3.19	2.97	PBT$^{(mc)}$s+Clinical	Lasso



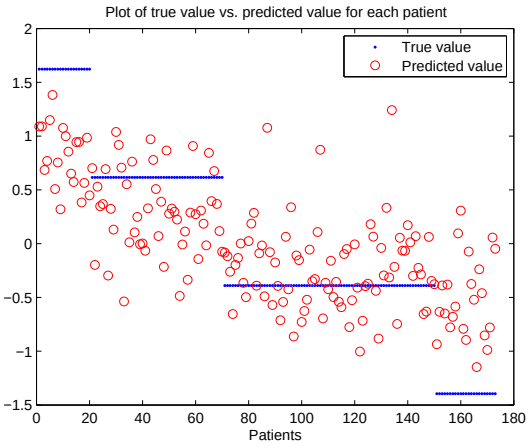
(a) i



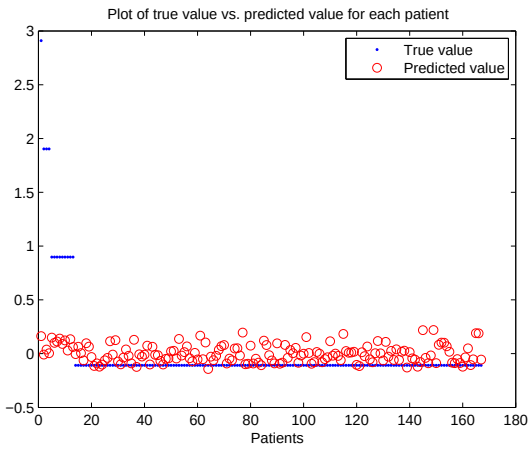
(b) ci



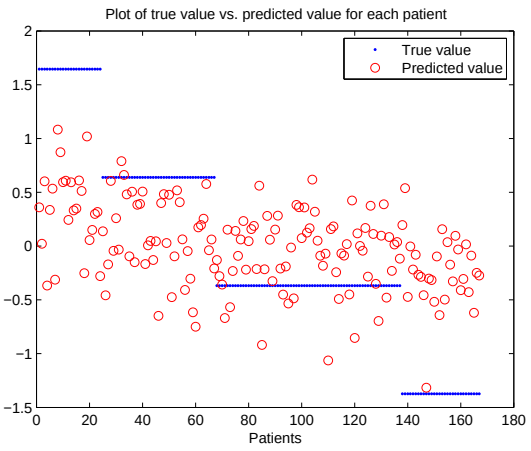
(c) t



(d) ct

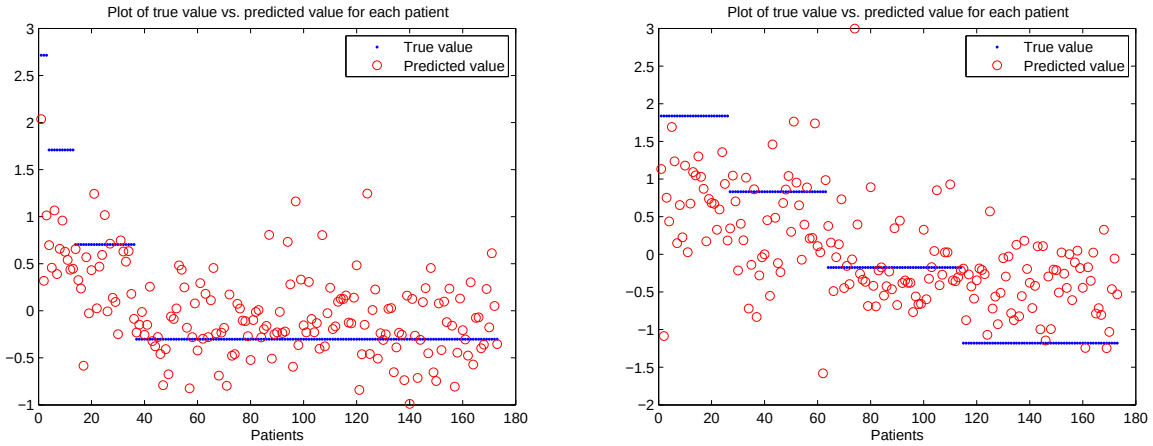


(e) v



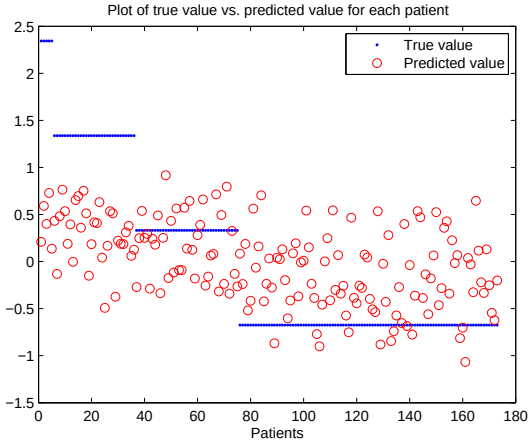
(f) cv

Figure 4.1: Plot of true values of some lesions versus the predicted values for each patient.

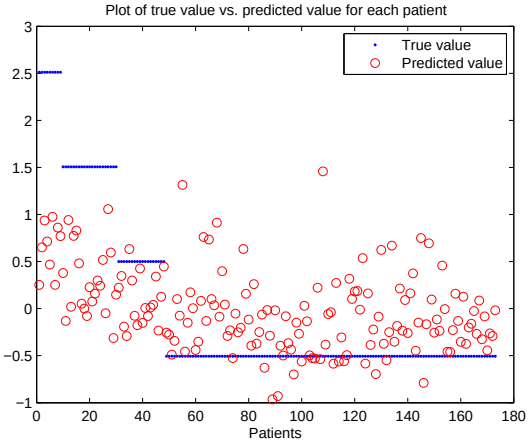


(a) g

(b) ah



(c) mm



(d) ptc

Figure 4.2: Plot of true values of some lesions versus the predicted values for each patient.

Statistical Analysis of Results

In many cases, the RMSE values in Table 4.3 are close to each other, which makes it difficult to compare the classifiers based on the different feature sets. Therefore we ran two statistical tests, paired t-test and Wilcoxon signed rank test, to determine if the differences are statistically significant. For each lesion, we compare the squared error values of leave-one-out cross validation of running Lasso on the seven feature sets : 1. All genes, 2. High variance genes, 3. K-means clusters' AVG, 4. PBT$^{(mc)}$s, 5. PBT$^{(mc)}$s+Clinical, 6. PBTs and 7. PBTs+Clinical.

For each lesion, we compare every pair of feature sets and for each pair we determine whether one of the sets performed significantly better than the other in terms of mean squared error. Therefore, for each lesion, we will have a 7×7 table whose $(i, j)^{th}$ element determines whether the feature set F_i is better (+1), worse (-1) or not significantly different (0) from feature set F_j. In this table, we are only dealing with the upper triangle – as the $(i, j)^{th}$ entry is just the negative of the $(j, i)^{th}$ entry. A feature set is better than another if the statistical test suggest that it is statistically better than the alternative set at the $p \leq 0.05$ value. The feature sets are then sorted based on their scores. The score of feature set F_i in lesion L is determined by

$$S(F_i, L) = \text{total number of wins in } L - \text{total number of losses in } L.$$

Table 4.6 shows the winners for all lesions. For each entry we show the score of the corresponding feature set. An entry of N/A means that there is no winner for the corresponding lesion, i.e. no feature set is better than any other set.

Using signed rank, in 5 out of 12 lesions there is no feature set that performs significantly better than the other feature sets, and using paired t-test, in 9 out of 12 lesions there is no significantly better feature set. Based on the results of the statistical tests, especially

Table 4.6: Best feature sets

Lesion	signed rank winner (S)	t-test winner (S)
g	All genes (1)	N/A
cg	N/A	N/A
i	PBTs (1)	N/A
ci	HV genes (2)	HV genes (4)
t	N/A	N/A
ct	HV genes (1)	HV genes (1)
v	N/A	N/A
cv	N/A	N/A
ah	PBT$^{(mc)}$s+Clinical (5) PBTs+Clinical (4)	PBT$^{(mc)}$s+Clinical (2) PBTs+Clinical (2)
mm	PBTs+Clinical (1)	N/A
ptc	N/A	N/A
ptcml	PBT$^{(mc)}$s (2) PBT$^{(mc)}$s+Clinical (2) PBTs (1) PBTs+Clinical (1)	N/A

signed rank test, on the rest of the lesions, we conclude that using the clinical data helps to improve the performance of the predictor. Feature sets which are based on using PBTs did slightly better than the rest of feature sets, although this result is not comprehensive over all lesions. Also, this results show that feature sets built by applying the *MkCoh* algorithm did not perform significantly better than those that just use PBTs.

4.3 Other Explorations

To improve the results, medical researchers suggested that we use the subset of IQR[6] filtered genes; they expected this subset to be more relevant than all the genes. In fact,

[6]IQR filter keeps only those genes that have fairly high variance.

IQR, like HV, filters out the genes that do not change over all samples (*e.g.*, house-keeping genes). We used IQR filtered genes and we repeated all the previous experiments on this subset of data, but we obtained similar results. The webpage [6, `ATI/Lesions/IQR/IQRTABLE.html`] provides all the results for the IQR filtered genes.

We also tried to extend each PBT$^{(mc)}$ by adding the high variance genes that are highly correlated with its genes, *i.e.*, adding g_i if

$$\frac{1}{|PBT^{(mc)}|} \sum_{g \in PBT^{(mc)}} corr(\, g_i,\, g\,) \geq T \tag{4.2}$$

where T is a user-specified threshold and $|PBT^{(mc)}|$ is the number of genes in PBT$^{(mc)}$; see [1, `Results/Extending/`]. We then used these extended PBT$^{(mc)}$'s as the features in learning the model. Unfortunately this did not improve the performance.

We next attempted to find collection of coherent gene sets from "scratch". Here, we first apply the *MkCoh* algorithm to 1050 high variance genes; this produced the first coherent cluster of 791 genes. We then examined the 1050-791=259 genes removed, from which we extracted a cluster of 82 coherent genes, and so on. [1, `Results/Clusters/`] shows the resulting seven clusters. However, using these clusters as the feature sets, did not improve the predictive performance noticeably.

We also used "KEGG Pathways" [26] as another source of prior biological knowledge. KEGG pathways describe the relationships among genes in pathways, which are each series of biochemical reactions controlling a specific cellular activities such as cell division or a programmed cell death. KEGG is a publicly available online database containing many known molecular pathways for different organisms. Each KEGG pathway is represented as a directed graph: each node corresponds to a set of genes or gene products having similar or related functions and the arcs are different interactions between them.

Our approach, like most of the current methods that use KEGG pathways, ignores the structure or relationships between the genes and simply uses the set of genes belonging to the pathway as selected features [36, 17].

We used 202 sets of KEGG pathways with the average size of 54.92 $\pm$ 58.00. We repeat all the previous process on these sets of genes. We applied *MkCoh* algorithm on KEGG pathways to get KEGG$^{(mc)}$ producing the results, shown in Table 4.7 — they are almost the same as PBTs' and PBT$^{(mc)}$s' results, *i.e.*, there is no difference in predictive performance.

4.3.1 Results on Other Datasets

We applied the idea of using PBTs and PBT$^{(mc)}$s as the features in prediction tasks on other datasets as well. The next data set is the another version of ATI dataset, ATI_{GFR}, which includes 54K genes and 137 patients. In this dataset, our goal was to estimate the following kidney functions (GFR scores [7]):

F_{now}**:** the GFR score of the kidney at the time of the biopsy

F_{6mo}**:** the GFR score of the kidney 6 months after the biopsy

$\Delta F = F_{6mo} - F_{now}$: the difference between F_{now} and F_{6mo}

Table 4.8 shows the results on this dataset. We did not get any promising results on this dataset. We even tried predicting Creatinine [8] values and using these values to compute GFR scores, but no signal has been found in this dataset to predict GFR scores.

The next dataset, called *BREAD ("Breast Cancer Relapse EArly Determinants")*, includes 132 patients and 24K genes. Here the main goal is to identify the markers that

[7] GFR (Glomerular Filtration Rate) is accepted as the best test to measure the level of kidney function and determine the stage of kidney disease [8].

[8] Creatinine is a chemical waste molecule that is generated from muscle metabolism [4].

Table 4.7: The RMSE values on lesion prediction using all the genes of KEGG and KEGG$^{(mc)}$ and also arithmetic average of KEGG and KEGG$^{(mc)}$

Lesion	Regressor	KEGG genes	KEGG$^{(mc)}$ genes	KEGG AVG	KEGG$^{(mc)}$ AVG
g	M5P	0.62	0.65	0.66	0.64
	SMOreg	**0.57**	**0.57**	0.68	0.71
cg	M5P	0.91	0.98	0.88	0.90
	SMOreg	**0.78**	0.81	0.85	0.96
i	M5P	0.83	0.80	0.73	0.70
	SMOreg	0.68	**0.66**	0.77	0.67
ci	M5P	0.77	0.80	0.78	0.83
	SMOreg	**0.69**	0.70	0.82	0.96
t	M5P	0.83	**0.73**	0.87	0.78
	SMOreg	0.75	0.75	0.86	0.87
ct	M5P	0.80	0.78	0.77	0.83
	SMOreg	**0.69**	**0.69**	0.80	0.94
v	M5P	0.49	0.49	0.49	0.50
	SMOreg	**0.44**	0.45	0.47	0.44
cv	M5P	1.14	1.10	1.12	1.13
	SMOreg	**1.09**	1.10	1.25	1.29
ah	M5P	1.16	1.13	1.13	1.05
	SMOreg	**0.98**	1.04	1.02	1.14
mm	M5P	0.94	0.90	0.89	0.90
	SMOreg	**0.79**	0.81	0.88	0.98
ptc	M5P	0.86	**0.85**	0.87	0.91
	SMOreg	0.86	**0.85**	0.97	1.00

distinguish between breast cancers that do not relapse from those that relapse early despite adjuvant therapy. Comparing the results of using PBTs, KEGG, PBT$^{(mc)}$s and KEGG$^{(mc)}$s, the best percentage of correctly classified instances was 64.39%, based on SMO on all the PBT$^{(mc)}$s genes. But using the PBTs and KEGG pathways, PBT$^{(mc)}$s and KEGG$^{(mc)}$s did not improve the results; other feature sets lead to a better prediction accuracy. You can find more results on BREAD dataset in the webpage [6, `Meetings/BREAD/`].

Table 4.8: The RMSE of Lasso in prediction of GFR scores $(F_{now}, F_{6mo}, \Delta F)$ using All-Genes, PBTs' AVG and PBT$^{(mc)}$s' AVG with/without Clinical data as the feature sets.

All Genes	PBTs	PBT$^{(mc)}$s	Clinical	F_{now}	F_{6mo}	ΔF
			+	16.80	21.59	**14.70**
+				**14.58**	21.53	15.4
+			+	14.96	21.67	15.43
	+			15.66	20.52	15.56
	+		+	14.66	**20.03**	14.79
		+		16.11	20.51	15.65
		+	+	14.59	20.12	14.86

We also used *Public Breast Cancer* [45] which includes 95 patients and 24K genes; here we only used the subset of 14K genes that was tagged by gene names. This is a classification problem, predicting relapse versus no relapse. Table 4.9 shows the percentage of correctly classified instances. The best known result for this dataset is about 73%, but the best result we achieved was 69.47% using SMO on all PBT$^{(mc)}$s genes.

Table 4.9: The percentage of correctly classified instances on $PublicBreastCancer$ dataset using PBT, PBT$^{(mc)}$s, KEGG and KEGG$^{(mc)}$

Regressor	PBTs' AVG	PBT$^{(mc)}$s' AVG	PBTs' genes	PBT$^{(mc)}$s' genes
SMO	63.15%	63.15%	67.36%	**69.47%**
J48	53.68%	58.94%	54.73%	64.21%

Regressor	KEGG AVG	KEGG$^{(mc)}$ AVG	KEGG genes	KEGG$^{(mc)}$ genes
SMO	58.04%	48.42%	66.31%	64.21%
J48	45.26%	56.84%	**67.36%**	64.21%

Chapter 5

Concluding Remarks

5.1 Conclusions

The prediction task is an important application of microarray data. In this thesis our goal was to learn a predictor that can predict some important characteristics of a patient based on the data from his/her microarray. We reviewed the challenges of microarray data analysis and summarized the approaches that have been applied to overcome these challenges. Many approaches are only based on the data, but some of them use prior biological knowledge as well.

Our colleagues used biological knowledge to produce clusters of genes, called PBTs, for Mouse and Human. They have provided 38 Human PBTs and 21 Mouse PBTs where each PBT is a cluster of genes with a common function, hence they expect the genes in each PBT to be highly correlated. Our studies in Chapter 3 showed that, unlike Mouse PBTs, Human PBTs are not coherent. We proposed a method, called *MkCoh*, to improve the coherency of the genes in each PBT by removing or flipping some genes. The revised PBTs, called PBT$^{(mc)}$, are more coherent than the original ones.

We applied several algorithms to learn a model that can predict some patient's characteristics, such as lesion values. We achieved good predictive performance using Lasso on all the gene expression values, but we hoped using each PBT$^{(mc)}$ (rather than each gene) as a single feature should improve our results, *i.e.*, we anticipated the predictors based on the PBT$^{(mc)}$ should be more accurate than ones based on either the original PBTs, or on the original gene expression values. Unfortunately, the experimental results based on some real biological datasets did not show that. To investigate possible reasons for this negative finding, we tried to extend PBTs$^{(mc)}$ by adding the high variance genes that are highly correlated with the genes inside each PBT. We also attempted to find coherent gene clusters within the complete set of high variance genes. In addition, we used KEGG Pathways as another source of prior biological knowledge. However, none of these approaches

improved the previous results.

5.2 Future Work

We tried many heuristics to use prior biological knowledge, in order to improve the results, but there are still some possibilities we would like to try in the future.

- Based on what medical researchers believed, we assumed that all the genes in a PBT should have a similar function (there should be just one "herd movement" in each PBT), therefore we found a subset of highly correlated genes in each PBT and we removed the other uncorrelated ones. However there might be more than one cluster of genes in each PBT, and those uncorrelated genes that we removed might be informative, *i.e.*, being uncorrelated with other genes is not a good reason for a gene to be removed. Hence, we should look for more than one cluster in each PBT.

- When we used KEGG pathways, we ignored the structure or relationships between the genes, *i.e.*, we only used the set of genes that belongs to the pathway as selected features. However, the structure of networks may contain many useful information for classification. Therefore, one possible approach is to take into account the structure of the networks in the feature set.

Bibliography

[1] Fariba Mahdavifard's Webpage. http://cs.ualberta.ca/~mahdavif.

[2] Genome Programs of the U.S. Department of Energy Office of Science. Retrieved September 10, 2009, from http://genomics.energy.gov/gallery/basic_genomics/gallery-01.html.

[3] Introduction to DNA Microarrays. Retrieved September 10, 2009, from http://www.bio.davidson.edu/people/macampbell/strategies/chipsintro.html.

[4] MedicineNet. Retrieved September 10, 2009, from http://www.medicinenet.com/creatinine_blood_test/article.htm.

[5] Microarrays: Chipping away at the mysteries of science and medicine. Retrieved September 10, 2009, from http://www.ncbi.nlm.nih.gov/About/primer/microarrays.html.

[6] Nasimeh Asgarian's Webpage. http://cs.ualberta.ca/~nasimeh.

[7] National Human Genome Research Institute. Retrieved September 10, 2009, from http://www.genome.gov/.

[8] National Kidney Foundation. Retrieved September 10, 2009, from http://www.kidney.org/kidneydisease/ckd/knowGFR.cfm.

[9] Weka Software. Retrieved September 10, 2009, from http://www.cs.waikato.ac.nz/ml/weka/.

[10] Wikipedia. Retrieved September 10, 2009, from http://en.wikipedia.org/wiki/Overfitting.

[11] E. Alpaydin. *Introduction to Machine Learning*. MIT Press, 2004.

[12] N. Asgarian. Rank-1 bicluster classifier. Master's thesis, University of Alberta, 2007.

[13] C. M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, October 2007.

[14] M. P. S. Brown, W. N. Grundy, D. Lin, N. Cristianini, C. W. Sugnet, T. S. Furey, M. Ares, Jr., and D. Haussler. Knowledge-based analysis of microarray gene expression data by using support vector machines. In *Proceedings of the National Academy of Sciences*, 2000.

[15] D. Cavalieri and C. Defilippo. Bioinformatic methods for integrating whole-genome expression results into cellular networks. *Drug Discovery Today*, 10(10):727–734, May 2005.

[16] H. Y. Chuang, E. Lee, Y. T. Liu, D. Lee, and T. Ideker. Network-based classification of breast cancer metastasis. *Molecular systems biology*, 3, October 2007.

[17] K. R. Curtis, M. Oresic, and A. Vidal-Puig. Pathways to the analysis of microarray data. *Trends in Biotechnology*, 23(8):429–435, August 2005.

[18] I. Fodor. A survey of dimension reduction techniques. Technical report, UCRL-ID-148494, LLNL, 2002.

[19] 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, October 1999.

[20] T. Hastie, R. Tibshirani, M. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. Chan, D. Botstein, and P. Brown. Gene shaving as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology*, 1(2), 2000.

[21] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning*. Springer, corrected edition, July 2003.

[22] D. C. Hoyle. Automatic PCA dimension selection for high dimensional data and small sample sizes. *Journal of Machine Learning Research*, 9:2733–2759, December 2008.

[23] D. Huang and W. Pan. Incorporating biological knowledge into distance-based clustering analysis of microarray gene expression data. *Bioinformatics*, 22(10):1259–1268, 2006.

[24] S. Imoto, T. Higuchi, T. Goto, K. Tashiro, S. Kuhara, and S. Miyano. Combining microarrays and biological knowledge for estimating gene networks via bayesian networks. In *CSB '03: Proceedings of the IEEE Computer Society Conference on Bioinformatics*, page 104, Washington, DC, USA, 2003. IEEE Computer Society.

[25] R. A. Irizarry, Z. Wu, and H. A. Jaffee. Comparison of affymetrix genechip expression measures. *Bioinformatics*, 22(7):789–794, April 2006.

[26] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucl. Acids Res.*, 28(1):27–30, January 2000.

[27] P. J. Kennedy, S. J. Simoff, D. B. Skillicorn, and D. R. Catchpoole. Extracting and explaining biological knowledge in microarray data. In *PAKDD*, volume 3056 of *Lecture Notes in Computer Science*, pages 699–703. Springer, 2004.

[28] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[29] Y. Lee and C.K. Lee. Classification of multiple cancer types by multicategory support vector machines using gene expression data. Technical report, Department of Statistics, University of Wisconsin, 2002.

[30] O. Maimon and L. Rokach. *Data Mining and Knowledge Discovery Handbook*. Springer, September 2005.

[31] T. F. Mueller, G. Einecke, J. Reeve, B. Sis, M. Mengel, G. S. Jhangri, S. Bunnag, J. Cruz, D. Wishart, C. Meng, G. Broderick, B. Kaplan, and P. F. Halloran. Microarray analysis of rejection in human kidney transplants using pathogenesis-based transcript sets. *American journal of transplantation*, 2007.

[32] T. F. Mueller, J. Reeve, G. S. Jhangri, M. Mengel, Z. Jacaj, L. Cairo, M. Obeidat, G. Todd, R. Moore, K. S. Famulski, J. Cruz, D. Wishart, C. Meng, B. Sis, K. Solez, B. Kaplan, and P. F. Halloran. The transcriptome of the implant biopsy identifies donor kidneys at increased risk of delayed graft function. *American journal of transplantation*, 2008.

[33] W. Pan. Incorporating gene functions as priors in model-based clustering of microarray gene expression data. *Bioinformatics*, 22(7):795–801, 2006.

[34] G. Piatetsky-Shapiro and P. Tamayo. Microarray data mining: facing the challenges. *SIGKDD Explor. Newsl.*, 5(2):1–5, 2003.

[35] L. C. Racusen, K. Solez, R. B. Colvin, S. M. Bonsib, M. C. Castro, T. Cavallo, B. P. Croker, A. J. Demetris, C. B. Drachenberg, A. B. Fogo, P. Furness, L. W. Gaber, I. W. Gibson, D. Glotz, J. C. Goldberg, J. Grande, P. F. Halloran, H. E. Hansen, B. Hartley, P. J. Hayry, C. M. Hill, E. O. Hoffman, L. G. Hunsicker, A. S. Lindblad, and Y. Yamaguchi. The Banff 97 working classification of renal allograft pathology. *Kidney Int*, 55(2):713–723, February 1999.

[36] F. Rapaport, A. Zinovyev, M. Dutreix, E. Barillot, and J. P. Vert. Classification of microarray data using gene networks. *BMC Bioinformatics*, 8(1):35+, February 2007.

[37] J. Reeve, G. Einecke, M. Mengel, B. Sis, N. Kayser, B. Kaplan, and P. F. Halloran. Diagnosing rejection in renal transplants: A comparison of molecular- and histopathology-based approaches. *American journal of transplantation*, June 2009.

[38] J. Reeve, G. Einecke, M. Mengel, B. Sis, N. Kayser, B. Kaplan, and P. F. Halloran. Diagnosing rejection in renal transplants: A comparison of molecular- and histopathology-based approaches. *American journal of transplantation*, June 2009.

[39] R. Schachtner, D. Lutter, A. M. Tomé, G. Schmitz, P. V. Gómez, and E. W. Lang. A matrix factorization classifier for knowledge-based microarray analysis. In Juan M. Corchado, Juan Francisco de Paz, Miguel Rocha, and Florentino Fernández Riverola, editors, *IWPACBB*, volume 49 of *Advances in Soft Computing*, pages 137–146. Springer, 2008.

[40] B. Schlkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond (Adaptive Computation and Machine Learning)*. The MIT Press, December 2001.

[41] A. Schulze and J. Downward. Navigating gene expression using microarrays - a technology review. *Nature Cell Biology*, 3(8):E190–E195, August 2001.

[42] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, New York, NY, USA, 2004.

[43] K. Solez, R. A. Axelsen, H. Benediktsson, J. F. Burdick, A. H. Cohen, R. B. Colvin, B. P. Croker, D. Droz, M. S. Dunnill, P. F. Halloran, P. Häyry, J. C. Jennette, P. A. Keown, N. Marcussen, M. J. Mihatsch, K. Morozumi, B. D. Myers, C. C. Nast, S. Olsen, L. C. Racusen, E. L. Ramos, S. Rosen, D. H. Sachs, D. R. Salomon, F. Sanfilippo, R. Verani, E. von Willebrand, and Y. Yamagushi. International standardization of criteria for the histologic diagnosis of renal allograft rejection: The Banff working classification of kidney transplant pathology. *Kidney Int*, 44:411–422, 1993.

[44] R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[45] L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530–536, January 2002.

[46] X. Xu and A. Zhang. Boost feature subset selection: A new gene selection algorithm for microarray dataset. In *International Conference on Computational Science*, pages 670–677, 2006.

Appendices

Appendix A

ATI Dataset

This appendix describes very briefly the challenges of kidney transplantation, how our colleagues at Alberta Transplant Applied Genomic Center collected and processed microarray data for ATI dataset, and finally how they scored the lesions.

A.1 Challenges of Kidney Transplantation

The quality of the donor organ is an important factor in predicting the performance of the transplanted organ. Unfortunately, it can be difficult to evaluate the quality of the organ at the time of transplantation [32].

Currently, histopathology[1] is the basis for assessing needle biopsies[2] in clinical medicine, but it is subjective. To determine whether a kidney transplant is being rejected, clinicians typically use a combination of histopathologic features defined by an international consensus, the Banff criteria [43]. Typically, a diagnosis of rejection is assigned based on empirically derived rules for lesion scoring. Although histopathology diagnoses correlate with treatment response and graft outcome, their accuracy has never been validated. Lesion grades are arbitrary and the agreement between two pathologists on lesion scoring is 10-50% and on diagnosis 45-70 % [38]. There are many other limitations [31] that affect treatment and produce inaccurate results for clinical trials. Finding relationship between histopathology and molecular phenotype, can improve the diagnoses using biopsies [38]. The assessment of gene expression in the donor organ can help researchers to determine organ quality and predict transplantation performance. The transcriptome[3] may provide a comprehensive measurement of the individual kidney's characteristics [32].

[1] Histopathology refers to the microscopic examination of a tissue in order to study the manifestations of a disease.

[2] A biopsy is a medical test involving the removal of certain cells or tissues for examination.

[3] The transcriptome is the set of all messenger RNA (mRNA) molecules produced in one or a population of cells.

A.2 ATI Dataset Preprocessing

Our colleagues at Alberta Transplant Applied Genomic Center take implant biopsies just before the end of the transplant surgery [32, 38]. The sample was immediately placed into RNA-later[4] for subsequent RNA extraction. Total RNA was isolated using the RNeasy Mini Kit (QIAGEN, Valencia, CA), and amplified according to AffymetrixR® protocol (Santa Clara, CA). RNA yields were measured by UV absorbance. RNA labeling and hybridization to the AffymetrixR® GeneChip microarrays (human Hu133 plus 2.0) was carried out according to the protocols included in the AffymetrixR® GeneChip Expression Analysis.

To prepare the data for more analysis, the raw data of all individual sample chips preprocessed using robust multi-chip averaging (RMA)[5]. After that other preprocessing steps, such as IQR filtering, are applied on data to filter out genes with low variability across the samples.

A.3 Lesion Scores

It is necessary to standardize the interpretation of allograft[6] biopsy, in order to guide therapy in transplant patients. A group of renal pathologists and transplant surgeons agreed to develop a schema for international standardization of nomenclature and criteria for the histologic diagnosis of renal allograft rejection, in 1991. In this schema, some rejection

[4]RNA-later is usually used for tissue storage. It stabilizes and protects cellular RNA. Therefore there is no need to immediately process samples or to freeze them for later processing.

[5]RMA (robust multi-chip averaging) is a program to compute gene expression summary values for Affymetrix Genechip. It consists of three steps: a background adjustment, quantile normalization and finally summarization.

[6]Allograft: The transplant of an organ or tissue from one individual to another of the same species with a different genotype.

indicators , such as Interstitial[7] inflammation, regarded as the principal factors to score lesions. Each lesion score has the value of $\{0, 1, 2, 3\}$; usually 0 means the kidney is healthy and 1, 2, 3 indicate different levels of disease severity. More details are available in [35, 43], Table A.3 provides a summary of scoring system and abbreviations of the histopathologic lesions [35, 43].

[7]Interstitial fluid is a solution that surrounds the cells of multicellular animals.

Table A.1: Scoring system and abbreviations of the histopathologic lesions [35, 43].

Lesion	ABBR	Score=0	Score=1	Score=2	Score=3
Interstitial inflammation	i	Interstitial mononuclear inflammatory cells in 0-9% of cortex	Interstitial mononuclear inflammatory cells in 10-25% of cortex	Interstitial mononuclear inflammatory cells in 26-50% of cortex	Interstitial mononuclear inflammatory cells in >50% of cortex
Tubulitis	t	No tubulitis	1-4 mononuclear inflammatory cells tubular cross-section	5-10 mononuclear inflammatory cells tubular cross-section	>10 mononuclear inflammatory cells tubular cross-section
Intimal arteritis	v	No arteritis	Subendothelial mononuclear inflammatory cells involving <25% of luminal area	Subendothelial mononuclear cells involving >25% of luminal area, no necrosis	Transmural inflammation and/or arterial fibrinoid necrosis with mononuclear cells
Glomerulitis	g	No glomerultiis	Mononuclear cells in <25% of glomeruli	Mononuclear cells in 26-75% of glomeruli	Mononuclear cells in >75% of glomeruli
Peritubular capillaritis	ptc	Absent or <10% of cortical peritubular capillaries with inflammatory cells	3-4 luminal inflammatory cells in =10% of cortical peritubular capillaries	5-10 luminal inflammatory cells in =10% of cortical peritubular capillaries	>10 luminal inflammatory cells in =10% of cortical peritubular capillaries
C4d peritubular capillary staining	C4d	0% of biopsy area that has a linear, circumferential staining in peritubular capillaries	1-9% of biopsy area that has a linear, circumferential staining in peritubular capillaries	10-50% of biopsy area that has a linear, circumferential staining in peritubular capillaries	>50% of biopsy area that has a linear, circumferential staining in peritubular capillaries
Transplant glomerulopathy	cg	Double contours in <10% of capillary loops in most severely affected glomerulus	Double contours in 10-25% of capillary loops in most severely affected glomerulus	Double contours in 26-50% of capillary loops in most severely affected glomerulus	Double contours in >50% of capillary loops in most severely affected glomerulus
Mesangial matrix increase	mm	No mesangial matrix increase	Present in 1-25% of nonsclerotic glomeruli	Present in 26-50% of nonsclerotic glomeruli	Present in >50% of nonsclerotic glomeruli
Interstitial fibrosis	ci	Interstitial fibrosis in 0-5% of cortex	Interstitial fibrosis in 6-25% of cortex	Interstitial fibrosis in 26-50% of cortex	Interstitial fibrosis in >50% of cortex
Tubular atrophy	ct	No tubular atrophy	Tubular atrophy in 1-25% of cortical tubules	Tubular atrophy in 26-50% of cortical tubules	Tubular atrophy in >50% of cortical tubules
Arterial fibrous intimal thickening	cv	No arterial fibrous intimal thickening	Arterial fibrous intimal thickening with 1-25% luminal narrowing	Arterial fibrous intimal thickening with 26-50% luminal narrowing	Arterial fibrous intimal thickening with >50% luminal narrowing
Arteriolar hyalinosis membrane multilayerin	ah	No arteriolar hyalinosis	Mild-moderate hyalinosis in at least one arteriole	Moderate-severe hyalinosis in more than one arteriole	Severe hyalinosis in many arterioles
Peritubular capillary basement	ptcml	1-2 basement membrane layers in peritubular capillaries assessed by electron microscopy	3-4 basement membrane layers in peritubular capillaries assessed by electron microscopy	5-6 basement membrane layers in peritubular capillaries assessed by electron microscopy	>6 basement membrane layers in peritubular capillaries assessed by electron microscopy